An Indonesian resource grammar (INDRA) : and its application to a treebank (JATI)

Moeljadi, David

2018

Moeljadi, D. (2018). An Indonesian resource grammar (INDRA) : and its application to a treebank (JATI). Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/82540>

<https://doi.org/10.32657/10220/46580>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**AN INDONESIAN RESOURCE GRAMMAR
(INDRA)
—AND ITS APPLICATION TO A TREEBANK (JATI)—**

**DAVID MOELJADI
SCHOOL OF HUMANITIES
2018**

**AN INDONESIAN RESOURCE GRAMMAR (INDRA)
—AND ITS APPLICATION TO A TREEBANK (JATI)—**

DAVID MOELJADI

School of Humanities

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2018

Acknowledgement

Many people have guided, helped, and supported me during my four year PhD candidature. It has been a period of intense learning for me in computational linguistics field, especially in grammar engineering. Without their guidance, support, and help, I might not survive. Thus, I would like to reflect on them and express my gratitude.

I would like to thank my supervisor, Dr. Francis Bond, who introduced me to computational linguistics field. Thank you very much for your guidance and help, as well as for all of the opportunities I was given to conduct my research, present it, and publish several papers. Thank you for your inspiration and advice to build INDRA and JATI. I also thank my co-supervisor, Dr. I Wayan Arka, for supporting me writing my dissertation and replying my e-mails. I would like to thank Dr. František Kratochvíl in my thesis advisory committee. Thank you for sharing your deep knowledge on Malay linguistics and your comments on my papers and dissertation drafts. Thank you for giving me an opportunity to join your expedition in Alor and for starting a Malay meeting group at NTU. The NTU Computational Linguistics Lab is a stimulating and friendly place to do research and work. I am grateful to Song Sanghoun for his technical help in the implementation, especially the control and raising constructions in the early stage; to Michael Goodman for his help in GitHub and tools such as gTest; to Lê Tuấn Anh who always has something to discuss and suggestions how to tackle problems; to Luís Morgado da Costa who organized a grammar engineering session and who always makes me laugh with his jokes; to Hannah Choi Yun Jung for her support and help (especially for Wordnet Bahasa); to Fan Zhenzhen who is always willing to help; to Giulia Bonansinga who lent a helping hand when I was in trouble; and to Wang Huizhen for sharing information about HPSG and MRS. I would also like to thank the Malay meeting members at NTU: Nomoto Hiroki, Atiqah, and Jojo. I thank the HSS (now SoH) staff: Nurazeen Binte Ismail, Amutha Shanmugam, and Christina Seet Mei Lin for their kind help on NTU administration/paper work. My PhD life was also supported by the NTU library staff. I would like to express my sincere gratitude to the HSS library support staff: Rashidah Ismail, Raihana Abdul Wahid, and Tan Chuan Ko for allowing me to borrow the fourth edition of KBBI paper dictionary for months and to Wong Oi May who helped me order the dictionary and other books such as reference grammars for Indonesian and Malay.

Beyond NTU, the wider circle of DELPH-IN consortium was very generous. I owe a

special gratitude to Dan Flickinger who always has time to discuss problems, especially during the Global Wordnet Conference at NTU in January 2018, when I was busy writing this dissertation. I would like to thank Emily Bender for the precious comments, especially on clitics and passive voice, and for organizing the VLAD sessions. I am grateful to all of the following people for many different kinds of assistance in (computational) linguistics field: thank you very much to people in Badan Bahasa, especially to Ibu Dora Amalia who gave me a precious experience to create the KBBI database and let me take part in a bigger KBBI project, I also thank her for the permission to use a part of the fifth edition of KBBI data for my treebank project; to Pak Adi Budiwiyanto, Ibu Menuk Suharjo, Pak Azhari Dasman Darnis, Mbak Vita Luthfia, Mbak Meryna Afrila, Mbak Winda Luthfita, Mbak Ambiya Ikrami Adji, and other staff who welcomed me warmly in the big KBBI project. Thank you very much for inviting me to various talks at different seminars and meetings in Jakarta. I thank Ivan Lanin for improving the KBBI database I created and making it more efficient. I also thank Lim Lian Tze who inspired me to write a paper about the KBBI Database and for many things: for her help in improving Wordnet Bahasa, replying my emails and questions about L^AT_EX and Overleaf, etc.

My PhD journey is also supported by people in Singapore outside NTU and outside (computational) linguistics field. Thank you to Pastor Stephen Tong (唐崇榮) whose spirit and teaching encouraged me to learn and live the Word. Thank you to the Reformed Evangelical Church Singapore (GRII Singapura) members: Ian Kamajaya who helped me learn programming during my first years at NTU and lent a helping hand to make the online KBBI page; Randy Sugianto (Yuku) who helped me with KBBI data, regular expressions, and other small things related to programming, also with the KBBI Android application; Juan Intan Kanggrawan who organized small group meetings and invited me to go there, through the meetings I learned how science and faith are related; Franky Chioh who gave me the book *In The Beginning was The Word* written by Vern Sheridan Poythress; my OSG small group members: Ko Carlos Wiyono Kurniawan who guides us patiently and lovingly, Steven Rusli, Ivan Edison, Harris, Frano Sumarauw, Willy Halim, thank you all for the prayers! I also thank my housemates in Singapore: I Made Riko, I Made Darmayuda (Kadek), Ratih Oktarini, and the little Bella, for supporting and helping me during my PhD life. I would express my special gratitude to my family: my mother Rut Widjiutami who loves me and prays for me every day, my uncle Subagio Budiharjo (CiKiang) who supported my big family (I still cannot believe that he passed away last December), and my cousins.

I am extremely grateful to the government of Singapore, through the Ministry of Education (MOE), and NTU, for full financial support in the form of an NTU scholarship which allowed me to work full-time on this dissertation for four years. My PhD project was supported in part by Singapore Ministry of Education (MOE) Tier 2 grant *That's what you meant: a Rich Representation for Manipulation of Meaning* (MOE ARC41/13) and MOE

Tier 2 grant *Grammar Matrix Reloaded: Syntax and Semantics of Affectedness* (MOE ARC21/13). I am grateful to the HSS (now SoH), NTU for a travel grant which enabled me to attend several conferences: the 19th International Symposium on Malay/Indonesian Linguistics (ISMIL 19) in Jambi, Grammar Engineering Across Frameworks 2015 (GEAF 2015) in Beijing, the Eighth Global WordNet Conference (GWC 2016) in Bucharest, the 12th DELPH-IN Summit at Stanford University, and the 20th International Symposium on Malay/Indonesian Linguistics (ISMIL 20) in Melbourne. I am also grateful to Fuji Xerox for the internship opportunity on Indonesian sentiment analysis in Yokohama in February-March 2016 and for supporting me to present my research results.

Portions of this work have previously been presented at the following conferences: Grammar Engineering Across Frameworks 2015 (GEAF 2015) in Beijing, China on 30 July 2015; Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar (HeadLex16) in Warsaw, Poland on 26 July 2016; Chulalongkorn International Student Symposium on Southeast Asian Linguistics (Chula-ISSSEAL) in Bangkok, Thailand on 9 June 2017; The 11th International Conference of the Asian Association for Lexicography (ASIALEX 2017) in Guangzhou, China on 10 June 2017; The 4th Atma Jaya Conference on Corpus Studies (ConCorps 4) in Jakarta, Indonesia on 21 July 2017; The 25th International Conference on Head-Driven Phrase Structure Grammar (HPSG 2018), in Tokyo, Japan, on 1 July 2018; and the International Symposium on Malay/Indonesian Linguistics (ISMIL), as well as DELPH-IN Summit in 2015, 2016, 2017, and 2018. I thank all the anonymous reviewers for the comments and participants for the discussion.

This dissertation is typeset with $\text{\LaTeX}$, using Overleaf (www.overleaf.com). The grammar INDRA is on GitHub (<https://github.com/davidmoeljadi/INDRA>). I am grateful to all the people who developed GitHub and Overleaf. My laptop crashed twice when I was writing this dissertation. I did not worry very much because my dissertation, as well as INDRA, can be accessed online. Again, I thank Lim Lian Tze and my supervisor, Francis, for replying to my e-mails and giving me useful suggestions for my dissertation, Overleaf, and $\text{\LaTeX}$.

δωρεὰν ἐλάβετε, δωρεὰν δότε
“you received without paying; give without pay”
Matthew 10:8b (ESV)

Singapore, August 2018

Contents

Acknowledgement	i
List of Abbreviations and Conventions	xi
List of Figures	xiii
List of Tables	xv
Abstract	xvii
I Background	xix
1 Introduction	1
1.1 Statement of research	2
1.2 Major contribution of the dissertation	2
1.3 Organization of the dissertation	4
2 Literature Review	7
2.1 Indonesian language	7
2.1.1 Historical and sociolinguistic background	7
2.1.2 Overview of Indonesian grammar	10
2.2 Theoretical framework	13
2.2.1 Head-driven Phrase Structure Grammar	13
2.2.2 Minimal Recursion Semantics	20
2.3 Computational background	29
2.3.1 Grammar engineering	29
2.3.2 Deep Linguistic Processing with HPSG Initiative	30
2.3.3 The development environment	31
2.3.4 Previous work on Indonesian computational grammar	34
2.3.5 Wordnet Bahasa	35
2.3.6 Indonesian part-of-speech tagger	40
2.4 Research method	42

2.4.1	Grammar development	42
2.4.2	Implementation in Type Description Language	45
2.4.3	Treebanking	49
2.4.4	Grammar evaluation	51
2.5	Summary	51

II INDRA 53

3 Basic structures 55

3.1	LinGO Grammar Matrix	55
3.2	Phrase structure rules and constituent order	57
3.2.1	Head-Initial Constructions	58
3.2.2	Head-Final Constructions	58
3.2.3	Head-Subject Phrase	58
3.2.4	Head-Complement Phrase	60
3.2.5	Head-Specifier Construction	60
3.2.6	Head-Adjunct Phrase	61
3.2.7	Coordination	62
3.2.8	Subordination	66
3.2.9	Fragments	68
3.3	Summary	70

4 Nouns 71

4.1	Noun subcategorization	72
4.1.1	Common nouns	75
4.1.2	Pronouns	78
4.1.3	Proper names	84
4.1.4	Locative and temporal nouns	84
4.2	Clitics	85
4.3	Determiners	92
4.4	Numerals and classifiers	95
4.5	Adjectives and prepositions	100
4.6	Reduplication	102
4.6.1	Reduplication without corresponding single bases	103
4.6.2	Reduplication of nouns and adjectives	104
4.7	Relative clause	108
4.7.1	Defining relative clause	108
4.7.2	Possessor topic-comment relative clause	112
4.8	Summary	115

5	Verbs	117
5.1	Verb subcategorization	117
5.1.1	Verb acquisition and subcategorization	120
5.1.2	Intransitive verbs	126
5.1.3	Verbs with clausal complementizer	127
5.1.4	Existential verb <i>ada</i>	130
5.1.5	Transitive verbs	132
5.2	Inflectional rules	137
5.2.1	Active voice	138
5.2.2	Passive voice	142
5.3	Auxiliaries	154
5.3.1	Temporal markers	155
5.3.2	Modals	157
5.4	Negation	160
5.5	Adverbs	162
5.6	Summary	164
6	Copula constructions	165
6.1	Introduction	165
6.1.1	Copula clauses with noun phrase predicates	166
6.1.2	Copula clauses with adjective phrase predicates	168
6.1.3	Copula clauses with prepositional phrase predicates	168
6.2	Analysis	169
6.2.1	Copula clauses with noun phrase predicate	169
6.2.2	Copula clauses with adjective phrase predicate	175
6.2.3	Copula clauses with prepositional phrase predicate	175
6.3	Negation	177
6.4	Generation	181
6.5	Summary	182
7	Compounds	183
7.1	Idiomatic compound	183
7.2	Verb-verb compound (serial verb construction)	185
7.2.1	Introduction	185
7.2.2	Previous works	186
7.2.3	Indonesian data	190
7.2.4	Analysis and computational implementation	194
7.2.5	Evaluation and conclusion	204
7.3	Noun-noun compound	204

7.4	Adjective-adjective compound	208
7.5	Summary	210

III JATI 211

8 KBBI database 213

8.1	Introduction	214
8.2	The KBBI dictionary format	216
8.3	Cleaning-up, conversion, and database creation	218
8.3.1	File and string manipulation	219
8.3.2	List of text filter and conversion	221
8.3.3	List of regular expression filter and conversion	221
8.3.4	Cleaning-up	222
8.3.5	Breaking down the components and creating a database	222
8.3.6	Dictionary data structure	224
8.4	The current state of the KBBI database and its applications	226
8.4.1	Targeted look-ups	226
8.4.2	Lexicography analysis	229
8.4.3	Linguistic analysis	230
8.4.4	Linking to other lexical resources	230
8.4.5	Online and offline applications	231
8.5	Summary and future work	231

9 JATI 233

9.1	Introduction	233
9.2	Related work	234
9.3	The corpus	235
9.4	Treebank development	236
9.5	Result and evaluation	238
9.6	Summary and future work	241

IV Summary 243

10 Conclusions 245

10.1	INDRA	246
10.2	JATI	247

V Bibliography and Appendices	249
Bibliography	251
A INDRA Meta-information	265
B Letter of permission	267
C List of publications and presentations	269

List of Abbreviations and Conventions

- *Italic font* is used for citing sentences, words, and other forms
- “Glosses” are given in double quotes immediately after the words or phrases
- (lit. “glosses”) is used to mark literal glosses or translations
e.g. *rumah sakit* “hospital” (lit. “sick house”)
- SMALL CAPITALS are used for names of features and grammatical glosses
- Sans serif is used for the names of predicates
- *Italic sans serif* is used for names of types
- *Slanted font* is used to introduce technical terms
- Underlining is used to highlight areas of interest in examples
- * (asterisk) indicates that the following example is ungrammatical
- ? (question mark) indicates that the following example is of questionable grammaticality

When describing feature structures, the following *paths*, i.e. sequences of features that can be followed from the root node, are often shortened.

Abbr.	Full path	Comment
HEAD	SYNSEM LOCAL CAT HEAD	Head
VAL	SYNSEM LOCAL CAT VAL SUBJ	Subject
VAL	SYNSEM LOCAL CAT VAL SPR	Specifier
VAL	SYNSEM LOCAL CAT VAL COMPS	Complements
HOOK	SYNSEM LOCAL CONT HOOK	Hook
LTOP	SYNSEM LOCAL CONT HOOK LTOP	Local top handle
INDEX	SYNSEM LOCAL CONT HOOK INDEX	Semantic index
XARG	SYNSEM LOCAL CONT HOOK XARG	External argument
RELS	SYNSEM LOCAL CONT RELS	Semantic relations
HCONS	SYNSEM LOCAL CONT HCONS	Handle constraints
ICONS	SYNSEM LOCAL CONT ICONS	Individual constraints
LKEYS	SYNSEM LKEYS	Pointer to main semantics

I follow the Leipzig glossing rules (<https://www.eva.mpg.de/lingua/resources/glossing-rules.php>). The following abbreviations are used for syntactic classes, functions, and other categories:

-	affix boundary
=	clitic boundary
1	first person
2	second person
3	third person
ACT	active
CLF	classifier
COP	copula
DEF	definite
EXCL	exclusive
FUT	future
INCL	inclusive
NEG	negation
PASS	passive
PL	plural
PRF	perfect
PROG	progressive
REDUP	reduplication
REL	relativizer
SG	singular

All the internet links are active at the time of submission on August 27, 2018.

List of Figures

2.1	Malay dialects (Adelaar, 2010, p. 203)	8
2.2	Diglossic situation in Indonesia (Paauw, 2009, p. 16)	10
2.3	Monophthongs and diphthongs in Indonesian (Soderberg & Olson, 2008)	11
2.4	Screenshot of the LinGO Grammar Matrix’s main page, taken on 22 April 2015	32
2.5	A small extract of a choices file (see also Table 3.1 on page 56)	32
2.6	The number of distinct orthographic forms in Wordnet Bahasa, KBBI, and KD (as of October 13, 2017)	39
2.7	A POS-tagged Indonesian sentence	41
2.8	A small extract of <code>lab3</code> test-suite	43
2.9	The process of grammar development (Bender et al., 2011, p. 10)	44
2.10	Screenshot of FFTB main page for MRS test-suite treebanking	49
2.11	Screenshots of treebanking process of <i>anjingnya sedang menggonggong</i> “the dog is barking”	50
3.1	Prepositions and subordinators in Indonesian	66
7.1	Analysis of Indonesian SVC	195
8.1	Screenshot of the online KBBI before 28 October 2016	215
8.2	Types of lexical resources, based on digital readiness (Lim et al., 2016) . .	215
8.3	Example entry <i>mereka</i> “they”	216
8.4	Example entry <i>AD</i>	216
8.5	Example entries <i>digitalin</i> and <i>digitalis</i>	217
8.6	Example entry <i>kaul</i> “vow”	217
8.7	Example entry <i>karam</i> “shipwrecked”	218
8.8	Example entries <i>keronsang</i> and <i>kerontang</i>	218
8.9	A part of KBBI Fourth Edition in Microsoft Excel and Word files	219
8.10	Screenshot of “KBBICleaner”	220
8.11	A part of the RTF file of KBBI	220
8.12	A part of the HTML file of KBBI	222

8.13	A part of the algorithm used to extract a number of fields in an entry . . .	224
8.14	The data structure for KBBI entry	225
8.15	Screenshot of the online KBBI homepage	231
8.16	Screenshots of the Android (left) and iOS (right) mobile applications . . .	232
9.1	Screenshot of FFTB main page for JATI	237
9.2	Screenshot of the annotation page for sentence number 3 in the KBBI test-suite	238
9.3	Screenshot of the annotation result with parse tree for sentence number 3 in the KBBI test-suite	238
9.4	Screenshot of the MRS semantic representation of the annotation result .	239
9.5	Evolution of coverage	240

List of Tables

2.1	Table of consonants in Indonesian (Soderberg & Olson, 2008)	12
3.1	Phenomena covered by filling in the LinGO Grammar Matrix questionnaire (M = Mintz (2002), L = Liaw (2004), S = Sneddon et al. (2010), A = Alwi et al. (2014), the number following each letter is the page number)	56
3.2	Constituent orders in Indonesian	57
4.1	Noun subcategorization in INDRA	76
4.2	Count nouns and mass nouns in Indonesian	77
4.3	Some combinations of cardinal units and group numbers	95
5.1	Verb subcategorization in INDRA (as of January 2018)	121
5.2	Three of six synsets of the verb <i>eat</i> and their verb frames in Wordnet . . .	123
5.3	The eleven most frequently used ERG verb types in the corpus	124
5.4	The eleven most frequently used ERG verb types in the corpus and their corresponding Wordnet verb frames	125
5.5	New verb types extracted from ERG and their corresponding names in INDRA	126
5.6	Sound changes in inflectional rule with <i>meN-</i>	139
6.1	Three types of basic copula clauses in Indonesian	166
7.1	Indonesian SVCs and the corresponding tags	193
7.2	Distribution of Indonesian SVCs in the corpus	193
7.3	The six relation classes for compound nouns (Ó Séaghdha, 2007)	205
8.1	Some examples of the conversion	222
8.2	Some inconsistencies in the KBBI format and the cleaning-up process . .	223
8.3	Some modifications in the KBBI definitions	223
8.4	Search results for all definitions of words with orthographic form <i>mereka</i> .	227
8.5	Search result for the proverb <i>sedia payung sebelum hujan</i>	227
8.6	Search results for all lemmas with labels <i>ark</i> (archaic) and <i>Jw</i> (Javanese) .	228
8.7	Search results for lemmas having definition sentences starting with <i>makanan</i>	228

8.8	Twenty most frequent words in KBBI definitions	229
8.9	Twenty most frequent genus words in KBBI definitions	229
8.10	Some derived words in KBBI entries, grouped by affixes	230
8.11	Some examples of aligned KBBI entries and the Wordnet synsets	230
9.1	Some KBBI definition sentences, before and after editing	236

Abstract

This dissertation describes the creation and the development of an open-source, broad-coverage Indonesian computational grammar, called Indonesian Resource Grammar (INDRA), within the framework of Head-Driven Phrase Structure Grammar (HPSG) (Pollard & Sag, 1994; Sag et al., 2003) and Minimal Recursion Semantics (MRS) (Copestake et al., 2005), using computational tools and resources developed by the DEep Linguistic Processing with HPSG-INitiative (DELPH-IN) research consortium. As a resource grammar, INDRA was employed to build an open-source treebank, called JATI. The research I have conducted on INDRA and its application to JATI was done in four years, from January 2014 to January 2018, during my PhD candidature. Previous work on the computational grammar of Indonesian are mainly done in the framework of Lexical-Functional Grammar (LFG) (Kaplan & Bresnan, 1982; Dalrymple, 2001) such as Arka (2010a) and Mistica (2013). A computational grammar of Indonesian called IndoGram (Arka, 2012) was developed within the LFG-based Parallel Grammar (ParGram) framework, using the Xerox Linguistic Environment (XLE) parser. To the best of my knowledge, no work on Indonesian HPSG has been done. Thus, the development of INDRA can also function as an investigation of the cross-linguistic potency of HPSG and MRS.

The approach taken is a corpus-driven approach. The scope is on the analysis and computational implementation of some basic Indonesian constructions and some phenomena in the Indonesian text: from the Nanyang Technological University Multilingual Corpus (NTU-MC) (Tan & Bond, 2012) and from definition sentences in the fifth edition of Kamus Besar Bahasa Indonesia (KBBI) (Amalia, 2016); the later contains 2,003 sentences and was treebanked, named JATI. The lexicon was semi-automatically acquired from various sources: the English Resource Grammar (ERG) (Copestake & Flickinger, 2000) via Wordnet Bahasa (Nurri Hirfana Mohamed Noor et al., 2011; Bond et al., 2014), the NTU-MC, and the KBBI definition sentence corpus. The coverage, i.e. the quality and the quantity of parsed sentences in the corpus by the grammar, is evaluated using test-suites.

INDRA can parse and generate complex noun phrases with clitics, determiners, numerals, classifiers, and defining relative clause; verb phrases with auxiliaries and voice markers; major copula constructions; compounds; coordination of words and phrases with the same part-of-speech; and subordination. However, at the time of submission, INDRA

still cannot handle phenomena such as equative, comparative, and superlative adjective phrases; coordination of words and phrases of different parts-of-speech; possessor topic-comment relative clause with more than one comment; imperatives; and constructions with Wh-question words. These are for future work. Despite its limitations, compared with IndoGram, INDRA has more precision in the analyses for some phenomena and has fifteen times more sentences in the open-source treebank. In addition, INDRA has the potential to be used in various applications such as multilingual machine translation and computer-assisted language learning. Since INDRA is developed in the DELPH-IN community along with other grammars such as the English Resource Grammar (ERG) (Flickinger et al., 2010) using the same semantics (MRS), a semantic-transfer-based machine translation system can be easily built. In summary, INDRA serves as the first, open-source computational grammar for Indonesian which covers most of the important constructions. INDRA has reached to a stage that it has the potential to be applied to various applications such as treebanking, machine translation, and computer-assisted language learning.

PART I:

BACKGROUND

Chapter 1

Introduction

“Languages are objects of considerable complexity, which can be studied scientifically.”

Sag et al. (2003, p. 2)

The present dissertation describes my four-year work done on the creation and the development of a *grammar*¹ of Indonesian, named Indonesian Resource Grammar or INDRA,² and a *treebank* for Indonesian, named JATI, up to the present time and the future development. INDRA is the first Indonesian Head-Driven Phrase Structure Grammar (HPSG) (Pollard & Sag, 1994; Sag et al., 2003) implemented as a computational grammar which can *parse*³ and *generate*⁴ Indonesian text. JATI is a *treebank*, or an annotated corpus of Indonesian text extracted from a subset of definition sentences in the fifth edition of the official Indonesian dictionary or KBBI (Amalia, 2016). It was parsed using INDRA. The present state of INDRA and JATI is available to be examined and can be downloaded from GitHub, a web-based hosting service for version control using a version control system for tracking changes in computer files.⁵ It has demo pages in *delphin-viz*⁶ and *demophin*.⁷ The computational tools employed to create INDRA and JATI, as

¹The term *grammar* in this dissertation follows the definition of grammar in Wasow (2004) which focuses on syntax and morphosyntax and includes semantics.

²INDRA stands for INDonesian Resource grAMmar. INDRA is a computational grammar and serves to be a resource for many NLP applications. The word *indra* in Indonesian is polysemous: ¹**in-dra** *noun* organ to taste, smell, hear, see, touch and feel something instinctively (intuitively); ²**in-dra** *noun* **1** king; **2** *Hinduism* (written with a capital letter) name of a god who rules the sky (Amalia, 2016). It is also a common Indonesian male name.

³*Parsing* is the act of determining the syntactic structure of a sentence. The goal is typically to represent “who did what to whom” in the sentence (Sproat et al., 2004, p. 608). It is assumed to involve starting with an input string and constructing all valid structures which can be associated with that string (Copestake, 2002, p. 85).

⁴*Generating* is the act of defining in a formally precise way a set of sequences (strings over some vocabulary of words) that represent the well-formed sentences of a given language (Sag et al., 2003, p. 525). It involves starting with a structure and deriving all valid strings (Copestake, 2002, p. 85).

⁵<https://github.com/davidmoeljadi/INDRA>

⁶<http://delph-in.github.io/delphin-viz/demo/>

⁷<http://chimpanzee.ling.washington.edu/demophin/indra/>

well as INDRA and JATI are open-source.

Hand-built grammars are used in many Natural Language Processing (NLP) applications, either to analyze spoken or written input or to generate text or speech, as *resource grammars*. Developing a computational grammar is often called *grammar engineering* (Copestake, 2002, p. 3). Grammar engineers both analyze natural languages in formal frameworks and implement those analyses in software. I have focused on a single natural language and developed a broad-coverage grammar of it, like the English Resource Grammar (ERG). I created a *starter grammar* of INDRA in January 2014 using the Matrix customization system in the LinGO Grammar Matrix (Bender et al., 2002), a web-based typological questionnaire which provides me a foundation for building INDRA. After that, the starter grammar was developed by analyzing more phenomena, modeling the analysis in HPSG, and implementing it so that it can be interpreted by a computer. The present dissertation documents the analysis, HPSG-model, and implementation of the phenomena covered in INDRA.

1.1 Statement of research

The aim of my research is to analyze and develop a detailed model of standard, formal Indonesian syntax and semantics explicit enough to be interpreted by a computer. My research is *breadth-first*, rather than *depth-first*, i.e. I firstly try to cover phenomena in Indonesian as broadly as possible, rather than focus on some particular phenomena and analyze as deeply as possible. I have done my research by building and developing an Indonesian resource grammar (INDRA) implemented within the framework of Head-Driven Phrase Structure Grammar (HPSG) (Pollard & Sag, 1994; Sag et al., 2003) and Minimal Recursion Semantics (MRS) (Copestake et al., 2005), as well as a treebank for Indonesian (JATI), using tools developed by The Deep Linguistic Processing with HPSG Initiative (DELPH-IN) research consortium.⁸ JATI is used to show that the model, implemented in INDRA, can predict and give all candidate parses for Indonesian sentences.

1.2 Major contribution of the dissertation

The main contributions of this dissertation are as follows.

1. This work is, to the best of my knowledge, the first to analyze, model, and implement linguistic phenomena in standard, formal Indonesian within the HPSG framework. The detailed model of Indonesian syntax and semantics contributes to the Indonesian linguistics, such as the analysis and modeling of copula constructions

⁸<http://www.delph-in.net>

and bound pronouns. The ultimate goal is a robust, linguistically-motivated computational grammar that can parse and generate well-formed Indonesian sentences. For example, in Indonesian, plurality can be expressed by noun reduplication or adjective reduplication; and possession can be expressed by full pronouns or their bound forms (only for singular pronouns), as illustrated in (1).

- (1) *batu kecil-kecilku*
 stone small-REDUP=1SG
 “my small stones”

Given the noun phrase in (1) as an input, the computational grammar can produce a syntactic tree, as shown in (2). The reduplicated form is represented as superscript two (2) in the tree. The left one is a syntactic tree with parts-of-speech information and the right one is with lexical and grammar rules which contribute to the noun phrase construction.

The computational grammar can also produce a well-formed semantics representation, as shown in (3). The semantics shows that the noun *batu* “stone” is third person plural, modified by *kecil* “small”, has a possessive relation (represented by *poss*) to first person singular pronoun, and is definite.

(3)

In addition to all possible syntactic trees and semantics, the computational grammar can generate well-formed sentences based on the semantics, as shown in (4). Depending on the context, any of the four noun phrases shown are possible. The reduplication can be applied to the noun or to the adjective and the possessor can be expressed either in the full form or in the bound form.

- (4) a. *batu kecil-kecil aku*
stone small-REDUP 1SG
- b. *batu kecil-kecilku*
stone small-REDUP=1SG
- c. *batu-batu kecilku*
stone-REDUP small=1SG
- d. *batu-batu kecil aku*
stone-REDUP small 1SG

2. The computational grammar can parse and treebank standard, formal Indonesian text, such as a subset of the definition sentences in the official Indonesian dictionary (KBBI).
3. The computational grammar and the treebank can be used as a resource for multiple applications, including natural language interfaces of various sorts and machine translation.

1.3 Organization of the dissertation

This dissertation consists of four parts, as follows.

1. **Part I** provides background information on the Indonesian language, theoretical framework, computational background, and research method.

- **Chapter 2** contains brief information about the linguistic part which includes the Indonesian language and the theoretical framework, also a brief explanation of the computational part, i.e. the development environment and the tools employed to build and develop INDRA, as well as the research method.
2. **Part II** is the main part of this dissertation. It starts with a description of basic phrase structure rules and constituent orders. Afterwards, it describes phenomena on nouns, verbs, as well as copula constructions and compounds. I describe some previous works at the beginning of each chapter.
- **Chapter 3** describes the first stage of INDRA, some basic constituent orders and phrase structures in Indonesian, as well as the implementation.
 - **Chapter 4** provides an overview of the Indonesian noun phrase, starting with noun subcategorization. It explains the main building components of an Indonesian noun phrase, i.e. bound pronouns, determiners, numerals and classifiers, adjectives and prepositions, reduplication, and relative clause.
 - **Chapter 5** outlines the Indonesian verb phrase. It starts with verb acquisition from English Resource Grammar (ERG) via Wordnet Bahasa and then explains the subcategorization, inflectional rules for active and passive voice, auxiliaries, negation, and adverbs.
 - **Chapter 6** describes Indonesian copula clauses with noun phrase, adjective phrase, and prepositional phrase predicates. It provides analyses and the implementation for copulas, as well as the negation.
 - **Chapter 7** contains four types of compound construction: idiomatic compound, verb-verb compound, noun-noun compound, and adjective-adjective compound. The characteristics of each type, the analysis, and the implementation are described.
3. **Part III** presents a description of JATI, an annotated *Kamus Besar Bahasa Indonesia* (KBBI) Indonesian monolingual dictionary definition sentence corpus, parsed using INDRA.
- **Chapter 8** is about the creation of KBBI database and its applications.
 - **Chapter 9** outlines the JATI treebank creation and development.
4. **Part IV** is the summary of this dissertation.
- **Chapter 10** presents overall conclusions regarding INDRA and JATI and suggests how the research can be further refined.

Chapter 2

Literature Review

This chapter presents some basic knowledge about the Indonesian language, the theoretical framework, computational background, and the research method. It is structured as follows: Section 2.1 outlines the historical, sociolinguistic, typological, morphological, and syntactic aspects of the Indonesian language, as well as an overview of the Indonesian grammar and some previous work done on the documentation of the Indonesian grammar or reference grammars. Section 2.2 explains briefly the HPSG and MRS theoretical framework and gives an example of how to analyze a simple Indonesian sentence. Section 2.3 describes some technical stuff such as the development environment of INDRA, previous work on computational grammar, as well as sources related to INDRA such as Wordnet Bahasa and Indonesian part-of-speech tagger. Section 2.4 outlines the research method of developing INDRA: from lexical acquisition and implementation to treebanking and evaluation. Section 2.5 summarizes.

2.1 Indonesian language

This section provides a brief description of the Indonesian language. Firstly, the historical and sociolinguistic background of Indonesian are introduced. Afterwards, the typological, morphological and syntactic aspects are briefly explained.

2.1.1 Historical and sociolinguistic background

Indonesian (ISO 639-3: ind), called *bahasa Indonesia* (lit. “the language (*bahasa*) of Indonesia”) by its speakers, is a Western Malayo-Polynesian language of the Austronesian language family. Within this subgroup, it belongs to the Malayic branch with Standard Malay spoken in Malaysia, Brunei Malay in Brunei, local Malay in Singapore and other Malay varieties spoken at various places in Indonesia such as Minangkabau and Makassar Malay (Lewis, 2009) (see Figure 2.1). The Indonesian language is spoken mainly in the Republic of Indonesia as the sole official and national language and as the common

Figure 2.1: Malay dialects (Adelaar, 2010, p. 203)

language for hundreds of ethnic groups living there (Alwi et al., 2014, pp. 1-2). In Indonesia it is spoken by around 43 million people as their first language and by more than 156 million people as their second language (2010 census data). It is over 80% cognate with Standard Malay (Lewis, 2009).

The history of the Indonesian language cannot be separated from its diglossic¹ nature which exists from the very beginning of the historical record when it is called Old Malay around the seventh century A.D. to the present day (Paauw, 2009, p. 3). With the coming of Islam and during the era of the Malay kingdoms from the twelfth to the nineteenth century A.D., a literary variety of Malay, known as Classical Malay, was codified and spread throughout the Malay world as a court language (Paauw, 2009, p. 14). During this period, the literary Malay in the Riau and Lingga group of islands, under Dutch influence, became the basis of the present-day standard Indonesian, while the one in the Malay peninsula, under British influence, is nowadays known as Standard Malay. The Dutch colonial used the literary language as a language of colonial administration and (to a limited extent) education and tried to develop and standardize the language. Classical Malay which was generally written in Arabic script, began to be uniformly written in Latin script after Charles Adrian van Ophuijsen introduced his system of spelling in 1901 (Abas, 1987, pp. 81-86).

One significant milestone that marks the adoption of Malay as the national language

¹Ferguson (1959, p. 336) who first described the concept of diglossia, defined it as a relatively stable language situation in which there are two types of language variety: "High" language variety which is a highly codified superposed variety, learned by formal education and is used for most written and formal spoken purposes but is not used by any sector of the community for ordinary conversation and "Low" language variety which is used for ordinary conversation.

in the present-day Indonesia is the Pledge of Youth by members of youth organizations in the first All Indonesian Youth Congress on 28 October 1928. The language was named Indonesian in the pledge. On the following day after the declaration of independence on 17 August 1945, the 1945 Constitution of the Republic of Indonesia was promulgated, in which it is stated in Section XV, Article 36 that the language of the state is Indonesian (*Bahasa Negara ialah Bahasa Indonesia*). Since then, many efforts have been undertaken to standardize the Indonesian language.

Two years after the independence, in 1947 the Faculty of Letters and Philosophy of the University of Indonesia set up an institute for language and cultural studies, which became the Division of Language and Culture (*Lembaga Bahasa dan Budaya*) in 1952 (Montolalu & Suryadinata, 2007, p. 44). After changing its name several times, in 2010 it became the Institution for Language Development and Cultivation (*Badan Pengembangan dan Pembinaan Bahasa*), under the Ministry of Education and Culture. Montolalu & Suryadinata (2007, p. 44) note that the main responsibility of this institution is to ensure that Indonesian becomes a national language in its true sense. One of the domains that has received much attention is the spelling reform. The Soewandi spelling system, released in 1947, replaced the 1901 Van Ophuysen spelling system (Abas, 1987, pp. 83-88). It was then replaced by the Perfected Spelling System (*Ejaan Bahasa Indonesia yang Disempurnakan*) or EYD which was released in 1972 (Abas, 1987, pp. 99-103). The Perfected Spelling System was then revised in 1987 and 2009. The Indonesian Spelling (*Ejaan Bahasa Indonesia*) or EBI replaced the Perfected Spelling System in 2015.

The language institute also publishes dictionaries, creates new terms and provides support for the standardization and propagation of the language. Pusat Bahasa published in 1988 a standard reference dictionary called “The Great Dictionary of the Indonesian Language of the Language Center” (*Kamus Besar Bahasa Indonesia Pusat Bahasa* or KBBI). This dictionary contains approximately 62,000 lemmas. Since then it has been revised and expanded three times: the second edition in 1991 contains approximately 72,000 lemmas, the third edition in 2001 contains approximately 78,000 lemmas, the fourth edition in 2008 contains approximately 90,000 lemmas, and the current fifth edition in 2016 contains approximately 108,000 lemmas. The third edition was the first one made online to public and had an official site (Alwi et al., 2008) which is currently inactive. The current online version of the fifth edition was released in 2016, together with the paper dictionary (Amalia, 2016) and the Android and iOS mobile application version. Another important decision made is to publish a good standard grammar entitled “A Standard Grammar of the Indonesian Language” (*Tata Bahasa Baku Bahasa Indonesia* or TBBI) in 1988. This reference grammar has been revised and expanded two times in 1993 and in 1998 (Alwi et al., 2014).

While much attention has been paid to the development and cultivation of the standard “High” variety of Indonesian, little attention has been particularly paid to describing and

Figure 2.2: Diglossic situation in Indonesia (Paauw, 2009, p. 16)

standardizing the “Low” variety of Indonesian. Sneddon (2006, pp. 4-6) calls this variety “Colloquial Jakartan Indonesian” and states that it is the prestige variety of colloquial Indonesian in Jakarta, the capital city of Indonesia, and is becoming the standard informal style. Paauw (2009, p. 40) mentions that Colloquial Jakartan Indonesian is a variety which has only been recognized as a separate variety recently. Historically, it developed from the Low Malay varieties spoken in Java by Chinese immigrant communities, which have been termed “Java Malay”. It has also been influenced by the Betawi language of Jakarta, a Low Malay variety which is thought to have been spoken in the Jakarta region for over one thousand years.

In addition to this “Low” variety, the regional languages spoken in various places in Indonesia which count more than 500 add the complexity of sociolinguistic situation in Indonesia. The “High” variety of Indonesian is used in the context of education, religion, mass media and government activities. The “Low” variety of Indonesian is used for everyday communication between Indonesians. The regional vernaculars (*bahasa daerah*) are used for communication at home with family and friends in the community. In some areas, Indonesian coexists with yet another regional lingua franca, which is often a Malay variety. For example, in the city of Larantuka of Flores Island, locals speak Indonesian, Larantuka Malay, and their local language Lamaholot in different contexts. This complex situation is well described in Paauw (2009) and shown in Figure 2.2.

2.1.2 Overview of Indonesian grammar

Because of its historical and sociolinguistic aspects, the “High” variety of Indonesian is perhaps the most well-described and studied Austronesian language of Indonesia with several comprehensive grammar available such as Macdonald (1976), Mintz (2002), Liaw (2004), Sneddon et al. (2010), and Alwi et al. (2014). Macdonald (1976) covers many im-

Figure 2.3: Monophthongs and diphthongs in Indonesian (Soderberg & Olson, 2008)

portant phenomena such as affixes, reduplication, conjunction, passive voice, existential sentence, topic-comment sentence, and various sentence patterns. Mintz (2002) compares Indonesian and standard Malay grammar and gives examples in both varieties. It emphasizes the order of words and phrases in a sentence. Liaw (2004) provides compact information on various grammatical phenomena with many examples. Compared with other reference grammars, he discusses more on prepositions, adverb-forming affixes, subordinative compounds and some minor sentences such as those used in block language and formulaic language. Sneddon et al. (2010) is perhaps the best reference grammar for Indonesian. It contains many descriptions of demonstratives, classifiers, partitives, articles, proper nouns and common nouns, count nouns and mass nouns, indefinite pronouns, aspect auxiliaries, particles, conjunction, derivational affixes, types of reduplication, relative clause, and various sentence patterns. Alwi et al. (2014) has more analyses and examples of verbs based on its morphological and syntactic types. It distinguishes verb based on the morphology, whether it should be used without affixes, with optional affixes, or obligatory affixes, as well as the number of arguments in a sentence, i.e. intransitive, semitransitive, monotransitive, and ditransitive. In addition to the aforementioned reference grammar, Hammarström et al. (2013) provide a comprehensive collection of 127 references of descriptive work in Indonesian such as reference grammars written in English, Indonesian, Dutch, German, French and Russian, dictionaries, as well as texts.² It is this “High” variety of Indonesian that I have modeled and implemented in INDRA.

Indonesian has 22 consonants, 6 vowels (monophthongs), and 3 diphthongs (Soderberg & Olson, 2008; Alwi et al., 2014, pp. 55-80). Table 2.1 shows all 22 consonants in Indonesian and the orthography. The phonemic values are given in the International Phonetic Alphabet (IPA). Figure 2.3 shows all 6 vowels and 3 diphthongs in Indonesian. The orthography for /e/ and /ə/ is <e>. Bracketed phonemes only appear in loan words or

²In this dissertation, I rely heavily on the reference grammar, especially Sneddon et al. (2010). There has been work within different frameworks on relevant issues, and a comparative discussion of alternative analyses is beyond the scope of this dissertation. As mentioned in Section 1.1, my research is *breadth-first* and I have been focusing on modeling and implementing as many phenomena as possible. Thus, comprehensive reference grammar serve as my primary source.

Table 2.1: Table of consonants in Indonesian (Soderberg & Olson, 2008)

	Bi-labial	Labio-dental	Dental	Alveo-lar	Post-alveolar	Palatal	Velar	Glottal
Plosive & affricate (Orthography)	p b p b		t t	d d	tʃ dʒ c j		k g k/q g	(ʔ) k/Ø
Nasal (Orthography)	m m			n n		ɲ ny	ŋ ng	
Flap/trill (Orthography)				r r				
Fricative (Orthography)		(f) f/v		s (z) s/x z	(ʃ) sy			h h
Approximant (Orthography)	w w					j y		
Lateral approximant (Orthography)				l l				

morphological boundaries.

Morphologically, Indonesian is a mildly agglutinative language, compared to Finnish or Turkish where the morpheme-per-word ratio is higher (Larasati et al., 2011). It has a rich affixation system, including a variety of prefixes, suffixes, circumfixes, and reduplication. Most of the affixes in Indonesian are derivational (Sneddon et al., 2010, p. 29). When affixes combine with roots or stems, a number of phonetic or phonological alternations through morphophonemic processes occur. Section 5.2 on page 137 describes a number of sound changes which occur when active voice prefix *meN-* combines with stems.

The order of constituents in an Indonesian sentence is highly constrained, similar to English. As is often the case with Austronesian languages of Indonesia, Indonesian has a basic word order of SVO with a nominative-accusative alignment pattern. In Indonesian, the heads normally precede their dependents (head-initial) (see Section 3.2 on page 57 on constituent order). The relation between a noun and the verb or a noun and the possessor is shown by the constituent order in the sentence or in the noun phrase, or by the use of a preposition. Argument changing operations are triggered by passive voice, as shown in Section 5.2.2 on page 142, and applicative constructions.

In a noun phrase with an adjective or a demonstrative or a relative clause, the head noun precedes the adjective or the demonstrative or the relative clause. Chapter 4 on page 71 explains the constituent order in a noun phrase. There is no case system in Indonesian, the same form of a noun is used for the subject, direct object, indirect object, possessor, etc.

Indonesian noun phrases do not require the presence of a determiner. A bare noun phrase in Indonesian can have singular, plural, or generic meaning. However, reduplication which is typically for countable nouns can show plurality and countability (see Section 4.6.2 on page 104 on noun reduplication). There is no agreement between verbs and pronouns or the number of the subject NP like in English, e.g. subject-verb agreement. However, there is an agreement for types of classifier and the head noun (see Section 4.4 on page 95 on classifiers).

2.2 Theoretical framework

This section provides a brief introduction to the Head-driven Phrase Structure Grammar (HPSG) (Pollard & Sag, 1994; Sag et al., 2003) and the Minimal Recursion Semantics (MRS) (Copestake et al., 2005), in particular to the version of HPSG and MRS used in the Grammar Matrix (Bender et al., 2002; Bender et al., 2010) (see Section 2.3.3 on page 31 on the development environment),³ which was employed to build a *starter grammar* for INDRA. This section also gives detailed examples how HPSG and MRS can be employed to analyze a simple Indonesian sentence. Some explanations in this section were cited from Drellishak (2009).

2.2.1 Head-driven Phrase Structure Grammar

Many formal models of the syntax of natural language are Context-Free Grammars (CFG), also called Phrase-Structure Grammars. These grammars are based on constituent structure, which was formalized by Chomsky (1956). A CFG consists of a set of rules and a lexicon of symbols (parts-of-speech/POS) and words, as illustrated in (5). The set of rules groups and orders the symbols. The lexicon connects the symbols with the words.

- (5) a. A set of rules:
- $S \rightarrow NP VP$
 - $NP \rightarrow N$
 - $VP \rightarrow V$
- b. A lexicon of POS and words:
- N: *anjing* “dog”
 - V: *menggonggong* “bark”

From the set of rules and the lexicon in (5), we can create a sentence, as shown in (6).

³<http://matrix.ling.washington.edu/customize/matrix.cgi>

- (6) *Anjing menggonggong.*
 dog bark
 “Dogs bark.” (own data)

We can also visualize it using a *parse tree*,⁴ as illustrated in (7).

Head-Driven Phrase Structure Grammar (HPSG), which was developed by by Ivan Sag and Carl Pollard in the mid 1980s, adds the idea that the lexical head of each constituent or the grammatically most important word in a phrase is passed up the parse tree (Jurafsky & Martin, 2009). HPSG is *surface oriented*, positing no additional abstract structures and thus providing a reasonably simple structure that is directly associated with the string of words that form each sentence. HPSG is *mono-stratal*, i.e. orthography, syntax, semantics, pragmatics are all handled in a single structure or the *sign*. The sign, i.e. a pairing of form and meaning, is the primary or elementary unit in HPSG, modeled through *Typed Feature Structures* (TFSs). Signs in HPSG include words, phrases, sentences and utterances. HPSG is *lexicalist*, i.e. the bulk of syntactic and semantic properties are defined in the lexicon. Thus, the information contained in the *feature structure* of a sign includes both the syntax and the semantics.⁵ HPSG embodies the claim that determining which strings of words constitute well-formed sentences and specifying the linguistic meaning of sentences depend mostly on the nature of words.

Feature structures are data structures which take the form of sets of *feature* (also called *attribute*) and *value* pairs. They can be used to represent objects. Features or attributes are unanalyzable atomic symbols from some finite set. Values are either atomic symbols or feature structures themselves. Every feature structure is of a certain *type*. HPSG models the grammatical properties of signs using types. Each type is associated with a particular *feature structure*. Types have features and the values of features are themselves types, leading to the possibility of nested structures. Feature structures are usually illustrated with an *Attribute-Value Matrix* (AVM) or a *Directed Acyclic Graph* (DAG) (Jurafsky & Martin, 2009, pp. 524-526; Fokkens, 2014, pp. 27-28).

⁴*Parse tree* is an abbreviation for the complete representation of a structure corresponding to the valid parse (Copestake, 2002, p. 89).

⁵This subsection on HPSG mainly describes the syntax and the next subsection on MRS mainly describes the semantics.

AVMs are formatted as a series of feature names on the left, with the values of the features on the right, and with both columns surrounded by square brackets ([]). The type of the feature structure is shown at the top of the AVM. Simplified AVMs for *anjing* “dog” and *menggonggong* “bark” in INDRA are shown in (8).

- (8) a.
$$\begin{array}{l} \text{anjing:} \left[\begin{array}{l} \text{nonhuman-noun-lex} \\ \text{HEAD } \text{commonnoun} \\ \text{VAL } \left[\begin{array}{l} \text{valence} \\ \text{SPR } \langle \left[\text{HEAD } \text{det} \right] \rangle \rangle \end{array} \right] \end{array} \right] \end{array}$$
- b.
$$\begin{array}{l} \text{menggonggong:} \left[\begin{array}{l} \text{intr-verb-lex} \\ \text{HEAD } \text{intransitive} \\ \text{VAL } \left[\begin{array}{l} \text{valence} \\ \text{SUBJ } \langle \left[\begin{array}{l} \text{HEAD } \text{subj-noun} \\ \text{VAL } \left[\text{SPR } \langle \rangle \rangle \end{array} \right] \rangle \\ \text{COMPS } \langle \rangle \end{array} \right] \end{array} \right] \end{array} \right]$$

AVM (8a) has a type of *nonhuman-noun-lex* (non-human noun).⁶ It has two features: HEAD and VAL (VALENCE). The HEAD encodes the part-of-speech of a sign and takes values like *noun*, *verb*, *det*, etc and their subtypes or supertypes. The VAL controls which words

⁶AVM (8a) is a compact version of the following AVM (see **List of Abbreviations and Conventions** on page xi).

- (9)
$$\begin{array}{l} \text{anjing:} \left[\begin{array}{l} \text{nonhuman-noun-lex} \\ \text{SYNSEM } \left[\begin{array}{l} \text{lex-synsem} \\ \text{LOCAL } \left[\begin{array}{l} \text{local-min} \\ \text{CAT } \left[\begin{array}{l} \text{cat} \\ \text{HEAD } \text{commonnoun} \\ \text{VAL } \left[\begin{array}{l} \text{valence} \\ \text{SPR } \langle \left[\begin{array}{l} \text{synsem-min} \\ \text{LOCAL } \left[\begin{array}{l} \text{local-min} \\ \text{CAT } \left[\begin{array}{l} \text{cat} \\ \text{HEAD } \text{det} \end{array} \right] \end{array} \right] \rangle \rangle \end{array} \right] \rangle \rangle \end{array} \right] \end{array} \right] \end{array} \right] \end{array} \right] \end{array}$$

The features HEAD and VAL are actually of type *cat* (*category*). They are in a feature structure, which is the value of the feature CAT (CATEGORY), the one which deals with the syntax in an AVM. When it is desirable to address a feature several levels deep in the structure without constraining the intermediate levels, the AVM may be abbreviated by leaving out intermediate names entirely.

and phrases are allowed to combine to form larger phrases and in what order. In AVM (8a), the HEAD's value is *commonnoun*, a subtype of *noun*, while the VAL's value is another feature structure of type *valence* containing SPR (SPECIFIER). In addition to types and feature structures, the values of features may also be lists, represented by a pair of angle brackets (< >). The feature SPR's value is a list of a feature structure. Copestake (2002, p. 121) notes that SPR takes a list value, which is either a single element list or an empty list. It is used for the determiner of a noun. The noun *anjing* “dog” can take a determiner *ini* “this” as its specifier and thus the value of SPR is a list of length one, i.e. a list of one feature structure in which the value of its HEAD is *det* (*determiner*).

The AVM (8b) shows that *menggonggong* “bark” is of type *intr-verb-lex* (intransitive verb). Its HEAD is *intransitive*, a subtype of *verb* and its VAL is a feature structure which has two features: SUBJ (SUBJECT) and COMPS (COMPLEMENTS). The SUBJ's value is a list of one feature structure. Its HEAD is *subj-noun*, a subtype of *noun*, with its SPR list is empty. This empty list means that the demand that has to be satisfied has been fulfilled and a complete constituent is formed, i.e. it is *saturated*. In other words, *menggonggong* “bark” needs an NP as its subject. Because the intransitive verb *menggonggong* “bark” does not take any complements, the value of COMPS is a list of length zero (illustrated in the AVM as < >).

The feature structures in HPSG are typed and hierarchical. Types are arranged into a hierarchy via the mechanism of *inheritance*. Each *subtype* has one or more *supertypes* which inherits all properties, including the constraints, from its supertypes: the features and the constraints on their values, but it can add further constraints and/or further appropriate features, too. Subtypes are therefore more specific than (or at least as specific as) their supertypes. Subtypes can inherit from more than one supertype. This is called *multiple inheritance*. Groups of types and their inheritance relationships are often shown as trees. A simplified hierarchy for the type *noun* is shown in Tree 10 (see Section 4.1 on page 72 for the full type hierarchy). The value of HEAD in the AVM for *anjing* “dog” in (8a) is a type *commonnoun*, which inherits from *noun*.

(10)

In HPSG, lexical entries, *lexical rules*,⁷ and *phrase structure rules*⁸ are all feature structures. To see how the AVMs in (8) and phrase structure rules combine together, consider the syntactic analysis produced by INDRA for the sentence *anjing menggonggong* “dogs bark” in (6). First, the parse tree, showing only parts-of-speech:

In HPSG, the part-of-speech labels are nothing more than abbreviations for feature structures. For example, an N is a sign with [HEAD *noun*] and a non-empty SPR list, i.e. it still needs a specifier; an NP, in contrast, has an empty SPR list: it needs no specifier. A V is a sign with [HEAD *verb*] and a non-empty SUBJ and COMPS list, i.e. it still needs a subject and a complement; and a VP, in contrast, has an empty COMPS list. Since *menggonggong* “bark” is an intransitive verb and does not have any complements, it is put under VP. Thus, the HPSG parse tree in (11) is different from the CFG parse tree in (7).

Expanding the feature structures of each node shows how the phrase structure rules and valence lists interact to license this sentence. The syntactic composition using feature structures is illustrated in (12).⁹ In each AVM in this tree, I added a feature STEM which has a list as its value, containing one or more strings surrounded by double quotation marks (“ ”), representing each constituent’s orthography or spelling. I also added a feature ARG-ST (ARGUMENT STRUCTURE), which contains the argument structure list. It appears only on lexical items (words or lexical rules), not on phrases, for modeling their argument structure. In addition, the AVM may be abbreviated by leaving out square brackets and replaced by a vertical bar (|). Thus, VAL [SPR < >] can be written as VAL | SPR < >

The NP node in (12) shows that the *bare-np-phrase-rule*¹⁰ changes the N *anjing* “dog” to an NP. Copestake (2002, p. 19) notes that a Typed Feature Structure (TFS) that encodes a rule consists of the daughters and the mother. A TFS that encodes a rule can be thought of as consisting of a number of ‘slots’, into which the phrases for the daughters and the mother fit. The mother is the TFS as a whole, while the daughters are the elements in the list which is the value of the ARGS (ARGUMENTS) feature. The *bare-np-phrase-rule* is a

⁷*Lexical rules* operate on single lexical items, optionally changing their spelling and building up new signs over their inputs. Grammars usually have a number of lexical rules proportional to the number of inflectional morphemes in the language.

⁸*Phrase structure rules* license the combination of one or more words into a single, larger phrase.

⁹Notice in particular that this tree with AVMs is slightly different from the usual ones. I put the rules inside the tree. The name of the rules are in magenta color.

¹⁰*bare-np-phrase-rule* is typically used in implemented grammars and not (necessarily) in theoretical HPSG (Fokkens, 2014, p. 33).

unary phrase rule and a *headed rule*. All rules in HPSG are either *headed* or *non-headed*. In headed rules, one of the daughters is identified as the *head daughter*. The HEAD of the phrase is identified with the HEAD of that daughter. The *bare-np-phrase-rule* takes only one daughter, which is its HEAD-DTR (HEAD DAUGHTER). The HEAD's value is *noun* and the SPR's HEAD's value is *det*. It retains the HEAD-DTR's HEAD value but changes the value of the SPR to an empty list. It states that the mother is *saturated* and is a phrase (NP).

(12)

S

HPSG is unification- and constraint-based. In *unification*, feature structures representing smaller constituents, such as words and phrases, are combined into larger constituents according to *constraints* of the lexical entries based on the *type hierarchy*. Two feature structures can unify if their values are compatible at every level of their structure, i.e. they have the same type or if one of them is a subtype of the other. The *nonhuman-noun-lex* can unify with the HEAD-DTR in the *bare-np-phrase-rule* because its HEAD's value is *commonnoun* which is a subtype of *noun* (see Tree (10)).

The *subj-head-phrase-rule* unifies the NP and the VP. The result is an S. The information on the constituent mother (S) is calculated on the basis of the constraints on the rule and the information provided by the daughters (NP and VP). In *subj-head-phrase-rule*, the HEAD-DTR's HEAD is of type *verb* and it demands an NP as its SUBJ (it has empty SPR list). The other daughter is the NON-HEAD-DTR (NON HEAD DAUGHTER), an NP. The mother has the same HEAD's value as the HEAD-DTR but the value of the SUBJ is an empty list, i.e. it is saturated.

One important part of the HPSG formalism is structure sharing, which is used to express that information in feature structures is identical. It is indicated by boxed numbers in feature structure descriptions. An identical number at several places in an AVM expresses the fact that the respective feature structures or types are identical. The *bare-np-phrase-rule* in (12) shows that the value of the feature HEAD-DTR is identified with the feature structure of a noun *anjing* “dog”, indicated by a boxed number [4]. The rule demands the HEAD-DTR's HEAD to be *noun*. As shown in (10), the supertype *noun* subsumes the subtype *commonnoun* and thus, the noun *anjing* “dog” can be the HEAD-DTR of the rule. The *subj-head-phrase-rule*'s HEAD-DTR is identified with the VP, indicated by a boxed number [3] and its NON-HEAD-DTR is identified with the NP, indicated by a boxed number [1]. The NP also serves as the value of the feature SUBJ in the VP. The HEAD value of the verb *mengggonggong* “bark” is passed to the HEAD-DTR of the *subj-head-phrase-rule*, identified with the mother of the rule, and up the tree to the HEAD of the S, indicated by a boxed number [2]. The HEAD of the noun *anjing* “dog” is also passed up to the NP, indicated by a boxed number [5].

2.2.2 Minimal Recursion Semantics

INDRA uses MRS (Copestake et al., 2005) as its semantic framework. MRS is a *flat, non-recursive* semantic formalism, adaptable for HPSG typed-feature structure cross-linguistically and suitable for parsing and generation. MRS is not a theory of semantics but rather a system of semantic representations. MRS representation was designed to deal with some of the problems in *semantic transfer* approaches to machine translation, especially to model the ambiguity often found in sentences with quantification, e.g. *every dog chases some*

white cat, using underspecification of *scope* relations. However, a discussion of how MRS represents scope is beyond the scope of this dissertation. MRS can be converted into more familiar languages, such as *predicate calculus* (Copestake, 2002, p. 26).

The semantic structures in MRS are underspecified for scope and thus suitable for representing ambiguous scoping. The primary interest in MRS is in finding the correct lexemes and the relationships between them that are licensed by the syntax. Thus, a representation language where scope can be ignored when it is irrelevant but retrieved when it is needed is required (Copestake et al., 2005, p. 286). The heart of an MRS representation is a *bag* of *elementary predications* (EPs). Formally, it is a bag, rather than a *set*, because there might be repeated elements. EPs indicate relations with the associated arguments. For example, the meaning of *anjing* “dog” can be represented in predicate logic as a one-place relationship, as shown in (13a). Similarly, *menggonggong* “bark”, and *anjing menggonggong* “dogs bark” can be represented as follows.

- (13) a. $anjing(x)$
 b. $menggonggong(x)$
 c. $menggonggong(x), anjing(x)$

Note that the structure in (13c) is a list of EPs being conjoined. The order of the members in the list is arbitrary. In HPSG, EP is represented as a *typed feature structure* (TFS), usually illustrated with an AVM.

In TFSs, the idea of encoding semantics is to build it up in parallel with the syntax. AVM (14) illustrates both the syntax and the semantics encoding of *anjing* “dog” in one big structure (see also a compact version of AVM illustrating the syntax encoding of *anjing* “dog” in AVM (8a) and its full version in AVM (9) in Footnote 6 on page 15).

(14)

The overall semantic structure is represented by the feature *CONT* (CONTENT). Its value is a feature structure, of type *mrs*. The type *mrs* has features *HOOK*, *RELS*, *HCONS* and *ICONS*. The value of *HOOK* is of type *hook*. *HOOK* is used to group together the features that specify the parts of an MRS which are visible to semantic functors. It is the information in *HOOK* that is available for coindexation with some external elements. *RELS* (RELATIONS) is the feature that introduces the bag of EPs. It is implemented as a *difference list*, represented with the brackets *< ! ! >*. In AVM 14, the *RELS* has one EP of type *noun-relation*. *HCONS* (HANDLE CONSTRAINT) introduces the *hande constraints* which are also implemented as a difference list. *ICONS* (INDIVIDUAL CONSTRAINT) deals with anaphors and information structure. It is also implemented as a difference list. AVM (15) shows in detail the content of the type *mrs* in AVM (14). To distinguish it from other types of AVM, I purposely changed the name of the type to *nonhuman-noun-lex*.

As mentioned before, the heart of an MRS representation is a bag or a list of *elementary predicates* (EPs), i.e. a combination of a single predicate with its arguments. In AVM (15), RELS (RELATIONS) is a feature that takes a difference list of EPs as a value. The structure of EPs is flat (no embedding) and the conjunction between EPs is implicit. Each EPs consists of a *relation* which has features PRED (PREDICATE), LBL (LABEL), and ARG0 (ARGUMENT ZERO). In AVM (15), the EP is of type *noun-relation*.

PRED is a predicate representing the meaning of the lexical item. The value of PRED for all types denoting relations is a string value corresponding to the predicate symbol (predicate name) and ends in *_rel*. In AVM 15, we see that the value of PRED is *_anjing_n_rel* (n represents the POS noun and rel stands for relation). The value of LBL is a *handle* (h), which is used to express scope relations. The only possible value of ARG0 is a variable standing for an individual (x) or an event (e). For common nouns such as *anjing* “dog”, the value of ARG0 is of type *ref-ind* (*referential index*) which serves as a pointer to the entity referred to by the noun. The type *ref-ind* has a feature PNG (PERSON-NUMBER-GENDER). It has feature structures as its value: the feature PERNUM (PERSON-NUMBER) has a value *3rd* (third person) and the feature ANIMCOUNT (ANIMACY-COUNTABILITY) has *non-human* as its value. The singular or plural number is not specified because a bare noun phrase in Indonesian such as *anjing* “dog” is underspecified for numbers. The value of ANIMCOUNT is a type *non-human* because Indonesian countable nouns use different classifiers for inanimate, human, and non-human. The information of animacy (and countability) is important for establishing correct classifiers (see Section 4.4 on page 95).

The feature HOOK has feature structures as its value: the feature LTOP (LOCAL TOP) has

its value identified with the value of LBL in RELS; the feature INDEX is coindexed with the INDEX of the semantic head daughter or ARG0. Copestake (2002, p. 141) notes that access to the semantics of a phrase is always via its INDEX. There is never a case where the RELS list is accessed directly. The HCONS (HANDLE CONSTRAINT) is a set of *handle* constraints which reflect syntactic limitations on possible scope relations among the atomic predications. The ICONS (INDIVIDUAL CONSTRAINT) deals with anaphors and information structure. It constraints two referential entities and says that the the two variables refer to the same thing (e.g. anaphor). Since AVM (15) has only one atomic predication or EP, the values of both HCONS and ICONS are empty.

AVM (16) shows the MRS of an intransitive verb *menggonggong* “bark”. In this AVM, the EP in the RELS has two arguments: ARG0 and ARG1. The ARG0 is of type *event*. It has a feature E (EVENT) which has feature structures as its value: the feature TENSE has a value *tense*, the feature ASPECT has a value *aspect*, and the feature MOOD has a value *mood*. The verb *menggonggong* “bark” is underspecified for tense, aspect, and mood. The ARG1 is of type *ref-ind* which refers to an entity or a noun. The PERNUM and ANIMCOUNT are underspecified for person, number, animacy, and countability. This means that the argument of this intransitive verb (the subject) can be any noun, irrespective of its person, number, and animacy, and countability.

Since *menggonggong* “bark” is an intransitive verb, it has only one argument ARG1. Predicates which require more arguments, such as monotransitive and ditransitive verbs, introduce them with ARG1, ARG2, and ARG3, as required. Note that the argument features are simply numbered rather than being given descriptive, semantic (thematic) role labels, such as AGENT and PATIENT, or very specific features for relations, such as EATER for the subject of *makan* “eat”, which is traditional in the HPSG literature. It is assumed that it can be reconstructed from the lexicon. The type *hook* has a feature XARG (EXTERNAL ARGUMENT) which picks out (the INDEX of) the SUBJ. This corresponds to the unexpressed subject in *control* environments (see Section 7.2 on page 185). In AVM (16), it is identified with the value of ARG1.

AVM (17) shows in detail the semantics composition of (6) using feature structures. Similar to AVM (12), it shows unification and also equivalence of arguments, which is implemented by coindexation. In this AVM, I omitted empty sets of HCONS and ICONS, as well as *_rel* in each predicate, and I wrote *mgg* rather than *menggonggong* in the PRED of *menggonggong* “bark” to save space. The AVM for *subj-head-phrase-rule* is also omitted but will be mentioned in the explanations.

(16)

ARG0 2ARG1 3

In AVM (17), *bare-np-phrase-rule* is a rule which is used for bare noun phrases. Copestake et al. (2005) states that the meaning of these phrases is a very complex issue and this is beyond the scope of this dissertation. It adds one EP that become part of the EP bag of the constructed phrase (NP) and sentence (S). The feature C-CONT (CONSTRUCTIONAL CONTENT) specifies a new main semantic contribution for this rule. Here it is assumed that the structure of a bare noun phrase is syntactically the same as an NP structure with a lexical type for quantifiers such as “some”, “every”, and “exist”, which introduces a *quant-relation* (*quantifier relation*) type. The *bare-np-phrase-rule* provides an underspecified quantifier predicate *exist_q_rel*. In other HPSG computational grammars such as the English Resource Grammar (ERG) and JACY Japanese grammar (Siegel et al., 2016), it is named *udef_rel*. Its sole argument ARG0 takes the identifier of a noun (*x2*), i.e. the referential index that the quantifier binds.

(17)

S

It introduces two additional features: *RSTR* (RESTRICTOR) and *BODY*. As scopal features, they have handles as their values.¹¹ The *RSTR* is related to the top handle of the quantifier's restriction (*h1*) via *HCONS* (HANDLE CONSTRAINT).

The *HCONS* is a set of handle constraints which reflect syntactic limitations on possible scope relations among the atomic predications. It is used instead of direct specification. A scope-resolved MRS is a maximally linked structure, whereas *linking* here means adding equalities between handles. Handles which appear as the value of scopal arguments are called *holes*. The *bare-np-phrase-rule* adds a *qeq* constraint via *HCONS*. The *qeq* (*equality modulo quantifiers*) relation (or $=_q$) always relates a hole *h* to a label *l* and indicates that either the value *l* of *LARG* (LABEL ARGUMENT) fills the hole *h* in *HARG* (HOLE ARGUMENT), i.e. $h=l$, or *l* is indirectly linked to *h* via one or more *floating quantifiers*. The *BODY* is left unbound: this is what allows quantifier to have varied scoping possibilities.

In this composition of the N daughter and the new relation, the *LTOP* and *INDEX* serve as pointers into the *RELS* list. The *LTOP* and *INDEX* of the NP mother come from those of the *C-CONT*. Copestake (2002, p. 142) notes that the *INDEX* on the phrase as a whole is coindexed with the *INDEX* of the semantic head daughter. The *ARG0* of the NP is identified with the *ARG1* of the VP by the following chain of identities: the N identifies its *ARG0* with the *INDEX*; the *bare-np-phrase-rule* identifies the *HOOK* of the NP with the *HOOK* of the *quant-relation*; the *subject-head-phrase-rule* identifies the *SYNSEM* (which includes the *HOOK*) of the NP with the *SUBJ* of the VP (*SUBJ* < [*HOOK* [*INDEX* [2] *x*]] >); and finally the verb lexical item identifies the *INDEX* on its *SUBJ* with its *ARG1*. Thus, the *ARG1* in the semantic head daughter *menggonggong* “bark” is coindexed with the *ARG0* of *anjing* “dog” which has the value *x2*.

MRS provides types that implement the principle of *semantic compositionality* via *RELS* and *HCONS*. The lists of *RELS* and *HCONS* of the NP and the S are built up of the lists of their daughters and those of their additional semantic contributions in the *C-CONT*. Both syntactic and semantic information on the mother is calculated on the basis of the constraints on the rule and the information provided by the daughters. The *HOOK* of the S is identified with the *HOOK* of the head daughter.

¹¹Scope-taking predicates include quantifiers as well as clause-embedding verbs, scopal modifiers, and scopal operators, such as negation. With scope-taking predicates, one or more argument positions point to the label of the predicate they scope over. However, this is done indirectly through handle constraints (the value of the feature *HCONS*). This indirection allows for the underspecification of quantifier scope.

In addition to the feature structure style representation using AVM, following Copestake et al. (2005), an MRS can also be displayed using an alternative linear notation. The MRS composition for *anjing menggonggong* “dogs bark” using the linear notation is shown in (18). The linear notation is represented as a tuple $\langle \text{GT}, \text{LT}, \text{R}, \text{C} \rangle$, where GT is the G_{TOP} (GLOBAL TOP handle), LT is the L_{TOP}, R is the RELS, and C is the HCONS. The root condition stipulates that the global top is qeq to the local top of the root phrase. In this tree, the global top handle is omitted except on the root. Empty sets of handle constraints are also omitted. For convenience, *h0* will be consistently used for the global top in this dissertation.

Moreover, the MRS of *anjing menggonggong* “dogs bark” can also be illustrated using a dependency graph, called Dependency Minimal Recursion Semantics (DMRS), as shown in (19). The structure of DMRS is minimal, the predicates are shown with simple inventory of links and without variables.

In DMRS (19) above, TOP is the G_{TOP} or the global top handle. It points to the (LBL of the) predicate *_menggonggong_v*. The ARG1 of *_menggonggong_v* points to the (INDEX or ARG0 of the) predicate *_anjing_n*. NEQ means that the LBL of *_menggonggong_v* is not equal with the LBL of *_anjing_n*. The RSTR of *exist_q* links to the (LBL of the) predicate *_anjing_n*. H indicates that the value of the RSTR is a handle. For the reasons of space and for the sake of simplicity, Dependency Minimal Recursion Semantics (DMRS) will be mainly used for MRS representations in this dissertation.

2.3 Computational background

This section provides the computational background of developing INDRA. Section 2.3.1 gives a brief explanation of grammar engineering. Section 2.3.2 contains an overview of the Deep Linguistic Processing with HPSG Initiative (DELPH-IN) consortium. Section 2.3.3 provides a brief explanation of the tools employed to build and develop INDRA. Section 2.3.4 mentions some previous work done on computational grammar related to Indonesian and INDRA. Section 2.3.5 describes Wordnet Bahasa, a lexical resource used in INDRA's lexical acquisition. Section 2.3.6 describes Indonesian part-of-speech tagger used in INDRA's unknown word handling.

2.3.1 Grammar engineering

The common practice of language documentation or descriptive grammar may include the following areas: phonology (the study of sound systems), morphology (the study of word structure), syntax (the study of sentence structure), semantics (the study of language meaning) and pragmatics (the study of language use). Grammar engineering is similar to grammar documentation because it tries to describe the language as used by native speakers but it is particularly focused on syntax and semantics. In addition, grammar engineering allows us to have the computer do the work of checking the models for consistency and to test against a much broader range of examples (Bender & Fokkens, 2010).

Sag et al. (2003, pp. 13, 21) note that syntax plays a crucial role in human language processing because it imposes constraints on how sentences can or cannot be construed and develops a set of rules that will predict the acceptability of (a large subset of) sentences in a language. Some of the goals of grammar engineering are as follows.

1. to tell for any arbitrary string of words whether it is a well-formed sentence or not and to give a possible range of syntactic and semantic representations (interpretations)
2. to consider how the grammar of one language differs from the grammar of other languages
3. to consider what our findings might tell us about human linguistic abilities in general

The well-formedness of sentences can be tested by asking native speakers for their judgments of acceptability. However, considering variation across speakers, linguistic and nonlinguistic context, the use of multiple sources such as data from actual usage, i.e. written and spoken corpora, is always a good idea (Sag et al., 2003, pp. 2-3).

Flickinger et al. (2010) mention that the necessary components in grammar engineering are as follows.

1. Linguistic theory. A solid linguistic theory which has rigid mathematical foundation, tractable computational model, and universal to different languages (see Section 2.2 on page 13 on theoretical framework).
2. Grammar engineering platform, which is used for implementation of the formalism (description language). It should have grammar editor, processor which includes parser and generator, graphical user interface, profiling system, and treebanking tools (see Section 2.3.3 on the next page about the development environment).
3. Linguistic resources, such as corpora, test-suites, treebanks, and reference grammars which include existing grammars for other languages on the same platform and existing grammars for one language on other platforms (see Section 2.3.3 on the facing page on the development environment, Chapter 8 on page 213 on KBBI dictionary, and Chapter 9 on page 233 on JATI treebank).
4. Research method (see Section 2.4 on page 42)

INDRA aims to be a computational grammar which has the goals of telling the well-formedness of sentences in Indonesian, giving a possible range of interpretations, as well as considering the similarities and differences between Indonesian and other languages. INDRA also aims to have all the necessary components in grammar engineering field mentioned above.

2.3.2 Deep Linguistic Processing with HPSG Initiative

The Deep Linguistic Processing with HPSG Initiative (DELPH-IN)¹² consortium is a research collaboration between linguists and computer scientists which builds and develops open source grammar, tools for grammar development and Natural Language Processing (NLP) applications using HPSG and MRS.

More than fifteen grammars of various languages with different typological characteristics have been created and developed within DELPH-IN,¹³ some of them are as follows:

1. English Resource Grammar (ERG) (Flickinger, 2000). The first implemented HPSG grammar and a robust English grammar, used for several applications such as machine translation and language education. It has 38,294 lexical items, 81 lexical rules, 212 grammar rules, 215 features, 9,086 types, and a large treebank called “Redwoods” (Oepen et al., 2002).¹⁴
2. Jacy (Siegel et al., 2016). A medium sized Japanese grammar, used for Japanese-English machine translation. As of 10 January 2018, it has 56,880 lexemes, 69

¹²<http://www.delph-in.net>

¹³see <http://moin.delph-in.net/GrammarCatalogue> for the DELPH-IN grammar catalogue

¹⁴<http://lingo.stanford.edu/redwoods/>

lexical rules, 120 rules, 193 features, 2,569 types, and a treebank called “Hinoki” (Bond et al., 2004).

3. Zhong (Fan et al., 2015).¹⁵ A grammar for Chinese languages, including simplified and traditional Mandarin Chinese as well as Cantonese and Min Nan. As of 10 January 2018, it has 38,263 lexemes, 2 lexical rules, 59 rules, 201 features and 2,842 types.

Together with ERG, Jacy, and Zhong, INDRA is being developed in the DELPH-IN community.

DELPH-IN grammars define typed feature structures using Type Description Language (TDL) (Copestake, 2000). An example of TDL type definition from INDRA is shown in Example (21). There are some differences between the standard HPSG theory and its implementation in DELPH-IN grammars due to the computational efficiency (Fokkens, 2014). One of the differences is most rules in DELPH-IN grammars are either unary or binary.

2.3.3 The development environment

The development of grammar is a combination of linguistic analysis and software development. A variety of tools to build and develop a grammar is needed. The DELPH-IN community provides several open-source tools for grammar development as well as an on-line wiki¹⁶ for the documentation.

The Linguistic Knowledge Builder (LKB) (Copestake, 2002) is mainly used in the grammar development environment. The LKB system was initially developed at the University of Cambridge Computer Library. Its first version was implemented in 1991 and has been updated at the Center for the Study of Language and Information (CSLI), Stanford University (Copestake, 2002, p. ix). It is now part of the Linguistic Grammars Online (LinGO) project, partially supported by the National Science Foundation of the United States of America. The LKB system is a grammar and lexicon development environment for typed feature structure grammars which enables grammar developers, even linguists which have very little knowledge in computer science, to write grammars and lexicons for natural languages to be parsed and generated (Copestake, 2002, pp. 6-7). It can be used to both parse and generate with the grammar,¹⁷ and is particularly good for debugging process. It has tools for interactive unification, examination of the type hierarchy, exploration of parse chart, and so forth.

The LinGO Grammar Matrix (Bender et al., 2002; Bender et al., 2010)¹⁸ is a web-

¹⁵<http://wiki.delph-in.net/moin/ZhongTop>

¹⁶<http://moin.delph-in.net/>

¹⁷Copestake (2002, p. 142) notes that the Linguistic Knowledge Builder (LKB) includes a generator which can produce strings on the basis of semantic representations

¹⁸<http://matrix.ling.washington.edu/customize/matrix.cgi>

Figure 2.4: Screenshot of the LinGO Grammar Matrix’s main page, taken on 22 April 2015

section=word-order	section=number
word-order=svo	number1_name=sg
has-dets=yes	number1_supertype1_name=number
noun-det-order=noun-det	number2_name=pl
has-aux=yes	number2_supertype1_name=number
aux-comp-order=before	
aux-comp=vp	section=person
multiple-aux=yes	person=1-2-3
	first-person=incl-excl

Figure 2.5: A small extract of a choices file (see also Table 3.1 on page 56)

based questionnaire for writing new DELPH-IN grammars, developed at the University of Washington (see Figure 2.4). It consists of a set of *libraries*, each of which supports a particular linguistic phenomenon and includes a section of the questionnaire and a syntactic analysis of the target phenomenon that can be customized and included in output grammars. The documentation for answering the questionnaire is provided in Fokkens et al. (2012). The matrix is originally based on ERG, with reference to the German HPSG grammar and especially to Jacy (Fokkens, 2014). As a starter kit, it was created to be useful cross-linguistically, providing a wide range of phenomena and basic files to make the grammars compatible with DELPH-IN parsers and generators. After a list of questions in the questionnaire is filled out, the matrix generates a *choices* file (see Figure 2.5) and builds a new grammar. The new grammar can be manually edited and extended afterwards.

The fundamental element of the matrix is a file called `matrix.tdl`. It defines the top of the hierarchy of typed feature structures and general rules for combining subjects, complements or modifiers with a head as well as coordination structures. It also provides the basics to construct semantics based on MRS (Fokkens, 2014).

In addition to LKB and LinGO Grammar Matrix, several open-source tools provided by DELPH-IN for grammar development are as follows:

1. Parsers and generators:

- (a) Platform for Experimentation with efficient HPSG processing Techniques (PET) (Callmeier, 2000), a very efficient HPSG parser, for processing. PET reads the same source files as LKB and produces identical results but allows a high-efficiency batch processing.
- (b) Answer Constraint Engine (ACE) (Packard, 2013).¹⁹ An efficient processor for DELPH-IN HPSG grammars. Its parsing and generation performance are both around 15 times faster than LKB. In certain common configurations, ACE is significantly faster than PET.
- (c) another grammar engineering environment (agree) (Slayden, 2012). agree can work with DELPH-IN style TDL grammars configured for LKB or PET within Mono and Microsoft's .NET framework.

2. Treebanking:²⁰

- (a) ITSDB or [incr tsdb()] (pronounced *tee ess dee bee plus plus*) (Oepen & Flickinger, 1998). A tool for testing, profiling the performance of the grammar (analyzing the coverage and performance), tracking changes, and annotating treebanks.
- (b) Full Forest Treebanker (FFTB) (Packard, 2014). A treebanking tool for DELPH-IN grammars, allowing the selection of an arbitrary tree from the “full forest” without enumerating/unpacking all analyses in the parsing stage. It is partly integrated with [incr tsdb()] and the LOGON tree.

3. Machine translation engine:

- (a) LOGON (Oepen et al., 2007). The LOGON infrastructure is a collection of software, grammars and other linguistic resources for transfer-based machine translation (MT).
- (b) Answer Constraint Engine (ACE) (Packard, 2013)²¹

¹⁹<http://sweaglesw.org/linguistics/ace/>

²⁰Treebanking is a process of making a syntactically annotated corpus by annotating each sentence with a parse tree (Jurafsky & Martin, 2009, p. 438).

²¹<http://sweaglesw.org/linguistics/ace/>

The various tools described above have documentation pages in the DELPH-IN wiki.²² I make extensive use of ACE, ITSDB, FFTB and LOGON for the development of INDRA. I also employ other tools such as typediff²³ to investigate and compare phenomena in INDRA with the ones in other DELPH-IN grammars, such as ERG and Jacy; delphin-viz²⁴ for DELPH-IN data structure visualizations and demo interface; Demophin,²⁵ a DELPH-IN web demo; and PyDelphin,²⁶ a set of Python libraries for the processing of DELPH-IN data.

As a part of the grammar development process, in order to make tracking of modifications possible, the latest version of INDRA is regularly saved or backed up in GitHub²⁷ and contains the following items:

1. The grammar (including the lexicon)
2. Some *test-suites* (see Section 2.4) and treebanks
3. Python scripts for integrating the Indonesian POS Tagger (see Section 2.3.6) to INDRA

It is licensed under the MIT license, a free software license originating at the Massachusetts Institute of Technology (MIT), which grants any person to download, use, copy, modify, and distribute the grammar, as long as they include the copyright and permission notice. Information on how to download is on the GitHub page. The development of INDRA is partially documented in the DELPH-IN wiki.²⁸

2.3.4 Previous work on Indonesian computational grammar

To the best of my knowledge, there is no previous work done on Indonesian HPSG but much has been done on Indonesian Lexical Functional Grammar (LFG). In addition, there is a computational grammar for Indonesian LFG called IndoGram²⁹ (and other computational grammars for other languages) written in Xerox Linguistic Environment (XLE) (Arka, 2012) and two open-source treebanks³⁰ having 127 sentences in total, under CC-BY license. Arka & Meladel (2011) discuss the syntactic and the functional-semantic/pragmatic constraints associated with all types of negation in Indonesian. Arka (2011a) and Arka (2013b) examine the relation between the expressions of tense, aspect,

²²<http://moin.delph-in.net>

²³<https://github.com/ned2/typediff>

²⁴<https://github.com/delph-in/delphin-viz>

²⁵<https://github.com/goodmami/demophin>

²⁶<https://github.com/delph-in/pydelphin>

²⁷<https://github.com/davidmoeljadi/INDRA>

²⁸<http://moin.delph-in.net/IndraTop>

²⁹<http://clarino.uib.no/iness/xle-web>

³⁰<http://clarino.uib.no/iness/treebanks>

and mood (TAM) and possessive/definite nominalization =*nya* in Indonesian. Both papers demonstrate that =*nya* encodes past time reference. Arka & Manning (2008) deal with active and passive voice in Indonesian. Arka (2010a) deals with categorial multifunctionality, i.e. the phenomenon of the same form of word or phrase which can appear in different functions in a sentence: as a predicate, a modifier, and an argument. Arka et al. (2009) present an analysis of the suffix *-i* in Indonesian, focusing on the issues of applicative-causative polysemy of the suffix and its alternation with suffix *-kan*. Meladel et al. (2009) investigate reduplication in Indonesian, focusing on verb reduplication that has the agentive voice affix *meN-*. Arka (2013a) discusses the nonverbal predication, showing that it is related to the future/non-future tense and the nominal predication is syntactically different from adjectival and prepositional predication. He argues that the absence of copula is associated with non-future tense, while an inchoative predication must be used for the nominal predication in the future tense. His finding that Indonesian does show a grammatical tense in relation to syntactic restriction uncovered a phenomenon which many grammarians fail to see. Arka (2014) works on syntactic, semantic and pragmatic properties of Indonesian control constructions with an interplay among the voice system, morpho-semantic-lexical properties of the control verbs, animacy of the argument and information structure of argument focusing. Arka (2010b) reports his analysis on two passives in Indonesian with *di-* and *ter-*. Musgrave (2001) in his Phd thesis concludes that the contrasting properties of non-subject arguments of prefixed and bare verbs suggest a divide between two types of clause structure in Indonesian.

Some phenomena have been implemented in both IndoGram and INDRA such as copula constructions which have similar analyses but INDRA covers more copula verbs with a refined type hierarchy (see Chapter 6 on page 165), some have been implemented only in IndoGram (and vice versa). Compared with IndoGram, INDRA has more precision and less ambiguity in the parsing for some phenomena. In terms of openness, INDRA as well as the tools used to develop INDRA are more open (they can be downloaded easily since most of them are in GitHub) than IndoGram and the tools (one must send emails to the developers and ask for permission to use). Since INDRA is developed along with other grammars in DELPH-IN which employ MRS as the semantics, it has the potential to be used in applications such as semantic-transfer-based machine translation. IndoGram, to the best of my knowledge, does not have such potential. In addition, INDRA has been used to parse fifteen times more sentences in the open-source treebank.

2.3.5 Wordnet Bahasa

Lexicon or a repository of words is vital in building and developing a grammar. Thus, a dictionary or a lexical database with comprehensive data of words is very important as a lexical source for a computational grammar. INDRA uses Wordnet Bahasa (Nurri

Hirfana Mohamed Noor et al., 2011; Bond et al., 2014) as an important part of its lexical source. Wordnet Bahasa was created based on Princeton Wordnet (PWN) (Fellbaum, 2005), a large English lexical database.

PWN was built at the Cognitive Science Laboratory of Princeton University in 1985. Different from other dictionaries and lexical databases, PWN groups nouns, verbs, adjectives, and adverbs into sets of cognitive synonyms or *synsets*, each expressing a distinct concept. They are interlinked through a number of semantic relations (Fellbaum, 2005). Since its creation, many other wordnets in different languages have been built based on PWN (Bond & Paik, 2012; Bond & Foster, 2013). They are available online at the Extended Open Multilingual Wordnet (1.2),³¹ hosted at Nanyang Technological University in Singapore. One of them, Wordnet Bahasa, is built as a lexical database in the Malay language (at present it contains data from Indonesian and Standard Malay).³² It combines data from several lexical resources: the French-English-Malay dictionary (FEM), the Kamus Melayu-Inggeris (KAMI), and wordnets for English, French and Chinese (Nurri Hirfana Mohamed Noor et al., 2011, p. 258). This subsection on Wordnet Bahasa will appear in Moeljadi et al. (forthcoming).

History of Wordnet Bahasa

Wordnet Bahasa is created by merging three wordnets: the Malaysian Wordnet (Lim & Hussein, 2006), the Indonesian Wordnet (Riza et al., 2010a), and the Wordnet Bahasa (Nurri Hirfana Mohamed Noor et al., 2011). All three wordnets opted for the *expand* approach, i.e. Malay lemmas were added to the Princeton Wordnet structure, taking the basic semantic structure of English and adding Malay words to the synsets. The main difference is in their coverage.

The Malaysian Wordnet (Lim & Hussein, 2006) was the first wordnet built for Malaysian. It was based on sense alignments produced by hand-aligning definitions from the Kamus Inggeris Melayu Dewan (KIMD), an English-Malaysian dictionary (Johns, 2000) and the Princeton Wordnet (PWN) (version 1.6). Based on the aligned definitions, Lim & Hussein (2006) built 12,429 noun synsets and 5,805 verb synsets. There are several issues with the resulting wordnet, as pointed out by Lim & Hussein (2006), such as the incomplete coverage, the nature of the dictionary used (KIMD), in which some Malaysian equivalents are definitions or not lexicalized and not valid lemmas. In addition, the wordnet is missing useful information that the PWN has, including Malaysian definitions for the glosses, verb frames, and sense frequencies.

The Indonesian Wordnet (Riza et al., 2010a) used the *expand* approach using a version of the lexicon from the Asian Wordnet Project (Riza et al., 2010a) which contained

³¹<http://compling.hss.ntu.edu.sg/omw/>

³²Wordnet Bahasa will be extended to accommodate other varieties of Malay language such as the local Malay variety spoken in Singapore and Brunei Malay.

some 25,755 synsets. Indonesian lemmas were added by human translators using online dictionaries and machine translation system as aids. However, the suitable Indonesian lemmas do not exist for all synsets. In addition, the Indonesian Wordnet has noise and less coverage of common words.

The Wordnet Bahasa (Nurril Hirfana Mohamed Noor et al., 2011) also used the *expand* approach, but used a multiple-pivot following the approach of Bond & Ogura (2007), aligning Malaysian to English using French and Chinese and semantic codes as extra information to constrain the translations. Wordnet Bahasa used two dictionaries: the French-English-Malaysian Lexicon (FEM) (Lafourcade et al., 2003), which contains entries in French, English, and Malaysian as well as hypernyms in French; and the KAmus Melayu-Inggeris (KAMI), which contains entries in Malaysian, English, and Chinese (Quah et al., 2001). It also used wordnets for English (PWN) (Fellbaum, 1998a), Chinese (Xu et al., 2008), and combined French wordnet from the French part of Euro WordNet (Vossen, 1998) and the Wordnet Liberé du Français (Sagot & Fišer, 2008), as well as semantic classes from the Goi-Taikei ontology (Ikehara et al., 1997). To map between the Goi-Taikei ontology and wordnet, the mappings produced by CoreNet (Kang et al., 2010) were used.

Each Malaysian word in the lexicon is linked through pivots to every synset that has the same part-of-speech. There are three pivots for this: the English term, the French or Chinese term, and the hypernym. After linking through the terms, semantic classes are used to see if the hypernym is compatible with the synset's hypernyms. After matching all the candidates, thresholds were set based on the number of languages matched, the hypernym matching, and the amount of ambiguity. Finally the 5,000 most common synsets used in the British National Corpus³³ (Fellbaum & Vossen, 2007) were hand checked. During this process, candidates that were only used in either Malaysian or Indonesian were marked as such. The resulting Wordnet Bahasa had 19,207 synsets, 48,111 senses, and 19,460 unique words (counting hand-checked and high-quality automatic candidates). This is still quite small, in terms of types, but as the high frequency synsets are all in, it should have high token coverage when used to tag texts.

In order to make a more useful resource with better coverage for both Malaysian and Indonesian, the wordnets were *merged* and cleaned up. The Malaysian Wordnet was mapped to PWN version 3.0 using the mappings provided by Daudé et al. (2003); the Indonesian and Bahasa wordnets were both based on version 3.0. Each entry was decided if it was used in Indonesian, Malaysian or both.

Some extra entries, such as languages, territories, and dates were also added from the Unicode Common Locale Data Repository (CLDR),³⁴ on person names from the English wordnet, some translations from Wikipedia, and a few entries found missing in corpora,

³³<http://wordnet.cs.princeton.edu/downloads.html>

³⁴<http://cldr.unicode.org/>

as described in Bond et al. (2014). The extended combined wordnet covers 90.6% of the open class tokens in the corpus, an improvement of 4.7% over the simple merge. Roughly half of the remaining unknown words are proper nouns. The combined Wordnet Bahasa had 49,668 synsets, 145,696 senses, and 64,431 unique words.

Afterwards, Wordnet Bahasa was cleaned up using part-of-speech and language information in KBBI and Kamus Dewan (KD) (Moeljadi, 2014a).³⁵ If an Indonesian or Malay word in Wordnet Bahasa was found not in KBBI, that word was marked as a bad lemma. If the part-of-speech of a particular word in Wordnet Bahasa is not the same as the one in KBBI, it was also marked as a bad lemma. After the cleaning up, Wordnet Bahasa had 40,493 synsets, 118,903 senses, and 43,113 unique words. The cleaning up also involved adding the correct lemmas to concepts that are otherwise undifferentiated in English. For example in Malay and Indonesian there are three lemmas for what we understand as the English rice, namely *nasi* for cooked rice, *beras* for uncooked rice or grains and *padi* for the rice plant. In addition, more data were contributed: Kateglo³⁶ from Ruli Manurung and Wikipedia data from Lian Tze Lim. I added 12,668 definitions from the Asian Wordnet Project (Riza et al., 2010b) and identified four semantic relations (hypernym, instance hypernym, synonym, and hyponym) in 57.10% of noun and verb definitions (Moeljadi & Bond, 2016). I added 34 numeral classifiers for Indonesian from KBBI dictionary. I also added new synsets, lemmas, and definitions from KBBI, KD, and PWN using scientific names as pivot, i.e. 494 new Indonesian lemmas and 489 new Malaysian lemmas, a total of 223 new synsets, as well as 431 new definitions. The latest raw data is released under the MIT license.³⁷ Lemmas and synsets from KBBI and KD not yet included in Wordnet Bahasa will be further added (see Figure 2.6 on the next page).

Applications of Wordnet Bahasa

Wordnet Bahasa has been used to support research in a variety of topics. It has been used to help analyze Malay tweets (Saloot et al., 2014) and Indonesian tweets (Le et al., 2016), as well as for general Malay semantic processing (Chu et al., 2014).

In more linguistic research, it has been used to annotate text from the Nanyang Technological University — Multilingual Corpus (NTU-MC), which contains 2,975 Indonesian sentences from three sources: Singapore Tourism Board website,³⁸ a Sherlock Holmes short story “the Speckled Band”, and a short story written by Akutagawa Ryunosuke “The Spider’s Thread” (Petrolito & Bond, 2014; Wang & Bond, 2014). Tagging is done with a web interface, and the text is gone through sequentially, so that the annotators can see the

³⁵The Institute Dictionary (*Kamus Dewan*) is a dictionary for Standard Malay published by The Institute of Language and Literature (*Dewan Bahasa dan Pustaka*). The fourth edition, which is the newest edition published in 2005, contains 75,387 lemmas including proverbs (Lian Tze Lim p.c.).

³⁶<http://kateglo.com/>

³⁷<https://sourceforge.net/p/wn-msa/tab/HEAD/tree/trunk/>

³⁸www.yoursingapore.com

Figure 2.6: The number of distinct orthographic forms in Wordnet Bahasa, KBBI, and KD (as of October 13, 2017)

context for each sentence. The corpus will be used to calculate sense frequencies which will be useful for lexicon interface and word sense disambiguation.

Having annotated the text in NTU-MC, Wordnet Bahasa has been used to model the decompositional semantics of pronouns, along with analyses for Chinese, English and Japanese (Seah & Bond, 2014a). Finally, it has been linked to the Semantic Domains from SIL International (Muhammad Zulhelmy bin Mohd Rosman et al., 2014). The semantic domains are designed to aid in the rapid construction and subsequent organization of lexicons for languages which may have no dictionaries at all. They are being employed to create a lexicon and wordnet for Abui.³⁹

Wordnet Bahasa can be also used to build and complement multilingual term lists for certain domains, which may lack terms in Malay. The Catalogue of Life (Roskov et al., 2015) is an online database of the world's known species of animals, plants, fungi and micro-organisms. The data contains some common names in a number of languages (most notably English), though not always available for all species. As no Malay common name was available in CoL2015,⁴⁰ Lim et al. (2016) matched CoL2015 records with Wordnet synsets using accepted scientific names as the pivot, thus creating a multilingual biodiversity checklist, providing translations in languages available in either CoL2016 and the Wordnet family (including Malay).

³⁹ Abui (ISO 639-3 abz) is a language spoken by approximately 16,000 speakers in the central part of the Alor Island in Eastern Indonesia.

⁴⁰ CoL2017 now contains more data in Malay.

Discussion and Future Plans

Wordnet Bahasa is actively maintained, currently in the Nanyang Technological University Computational Linguistics Lab in Singapore.⁴¹ There are a lot of things can be done to improve Wordnet Bahasa.

Incorrect senses are being cleaned up and the missing senses are being added to Wordnet Bahasa. Currently a small corpus containing the most frequent words used in the definition sentences of KBBI is being handchecked for incorrect lemma forms as well as missing senses. The work involves removing lemmas from concepts that were incorrectly added during the automated processes as well as creating new concepts that are relevant to both Indonesian and Malaysian.

In addition, many parties are encouraged to take part in improving Wordnet Bahasa, especially Badan Pengembangan dan Pembinaan Bahasa (The Language Development and Cultivation Agency) or Badan Bahasa in Indonesia, Dewan Bahasa dan Pustaka (Institute of Language and Literature) in Malaysia, and Dewan Bahasa dan Pustaka (Institute of Language and Literature) in Brunei Darussalam. Badan Bahasa gave a permission to use the information in KBBI V for Wordnet Bahasa. It was decided in the Majlis Bahasa Brunei Darussalam-Indonesia-Malaysia (MABBIM) executive meeting in April 2017 that each country (Indonesia, Malaysia, and Brunei Darussalam) should build its own team to improve Wordnet Bahasa and contribute to a single rich lexical resource.

Concretely, Indonesian definition sentences and examples from KBBI V will be added and with DPB Malaysia and Brunei's cooperation, definition sentences and examples in Standard Malay will also be added to make the wordnet more accessible to Malay speakers. Wordnet Bahasa will be linked to KBBI and vice versa. Simultaneously, more corpora will be annotated with Wordnet Bahasa in order to get more frequency information and further check for gaps in coverage. In addition, derivational morphology will be added. Indonesian and Malaysian words will be linked to their stems and variant forms. Other dialects of Malay will be covered and a graph containing only the Malay structures will be built so that its structure and connectivity can be investigated.

2.3.6 Indonesian part-of-speech tagger

Part-of-speech (POS) tagging is a process of assigning parts-of-speech to words. It takes words and tagsets as the input and produces a POS tag for each word as the output. It is useful for resolving ambiguity, i.e. to specify the correct part-of-speech of a word having more than one part-of-speech in a given context. In the development process of INDRA, POS tags are important, particularly for dealing with unknown words. INDRA uses Indonesian Part-Of-Speech (POS) Tagger (Rashel et al., 2014),⁴² a tool developed

⁴¹<http://compling.hss.ntu.edu.sg>

⁴²<http://bahasa.cs.ui.ac.id/postag/tagger>

Adi	VB		
bisa	MD	MD,NN	rule-11
membawa	VB		
es	NN		
krim-es	NN		
krim	NN		
itu	PR		
ke	IN		
rumah sakit		NN	
.	Z		

Figure 2.7: A POS-tagged Indonesian sentence

by a team from the Faculty of Computer Science of the University of Indonesia for POS tagging in Indonesian. It receives Indonesian text as an input and gives lines of words with the respective POS tags as an output.

Indonesian POS Tagger is based on Indonesian Morphological Analyzer (MorphInd) (Larasati et al., 2011),⁴³ a tool which handles both morphological analysis and lemmatization for a given surface word form. Indonesian POS Tagger applies a rule-based approach by employing hand-written disambiguation rules. It has six main modules: multi-word expression tokenizer, name entity recognizer, closed-class word tagging, open-class word tagging, rule-based disambiguation, and resolver for unknown words. It applies tags to every token, starting from closed-class words to open-class words and disambiguates the tags based on a set of manually defined rules. It has 23 tagsets of parts-of-speech, such as NNP for proper nouns, NN for common nouns, VB for verbs, MD for modals and auxiliary verbs, and JJ for adjectives. Figure 2.7 shows a result of POS tagging an Indonesian sentence in Example 20 using the Indonesian POS Tagger.

(20) *Adi bisa membawa es krim-es krim itu ke rumah sakit.*

Adi can bring ice cream-REDUP that to hospital

“Adi can bring those ice creams to hospital.” (own data)

Figure 2.7 shows the Indonesian POS Tagger correctly disambiguates the word *bisa* with hand-written rule-11. The word *bisa* in Indonesian is ambiguous, as a noun it means “poison” and as a modal it means “can”. A compound noun *rumah sakit* “hospital” (lit. “sick house”, *rumah* means “house” and *sakit* means “sick”) is correctly tagged as a common noun (NN). However, it incorrectly tags Andi, a proper noun, as a verb although there is no such verb in Indonesian and segments a reduplicated compound noun *es krim-es krim*

⁴³<http://septinalarasati.com/morphind/>

2.4 Research method

“Grammar engineering is spiraling upward.” Francis Bond (p.c.)

This section illustrates the method employed for grammar engineering or building and developing a computational grammar. In general, the research method is a mixture of linguistic analysis and computational implementation. By doing grammar engineering, we must consider every detail of language phenomena which might not come to our mind when we document or describe a language on paper. Using the LinGO Grammar Matrix (see Section 2.3.3 on page 31) as a starter kit for a new grammar is convenient because we do not have to create and write the TDL files from scratch one by one although it is possible that the analyses we made in the questionnaire have to be manually edited or deleted at a later stage.

This section consists of four subsections. Subsection 2.4.1 describes the overall grammar development mechanism, including a short description of how the grammar acquires the lexicon and the process of building a syntactically annotated corpus or a treebank. Subsection 2.4.2 on page 45 shows how to implement an Indonesian simple sentence in TDL files. Subsection 2.4.3 on page 49 gives a brief description on treebanking. Subsection 2.4.4 on page 51 summarizes the grammar evaluation procedure.

2.4.1 Grammar development

The aim of grammar development is to parse and generate text. A sample of the text illustrating a particular language phenomenon or construction in the form of grammatical and ungrammatical sentences is selected and formatted in interlinearized glossed text according to Leipzig glossing rules⁴⁴ into one or several sample files called *test-suites*.⁴⁵ Test-suite can be divided into two types: phenomena based test-suite which contains particular phenomena and natural test-suite which is taken from a parallel corpus such as NTU-MC or naturally occurring text. Figure 2.8 shows a small extract of a phenomena based test-suite.

Copestake (2002, p. 121) notes that the objective of grammar engineering is to increase coverage while avoiding overgeneration and thus a set of test sentences is essential to make sure this objective is being met as the grammar is changed. The test-suite file in Figure 2.8 consists of a header which contains the information of the language name, language code, author(s) of test-suite, the date and the source if the sentences are taken from reference

⁴⁴<https://www.eva.mpg.de/lingua/resources/glossing-rules.php>

⁴⁵The general guidelines and formatting of test-suites can be checked at <http://compling.hss.ntu.edu.sg/courses/hg7021/testsuites.html>

2.7. RESEARCH METHOD 45

Language: Indonesian
 Language code: ind
 Author: David Moeljadi
 Date: 2014-12-05

#1	#2
#word order	#word order
Source: author	Source: author
Vetted: t	Vetted: t
Judgment: g	Judgment: u
Phenomena: {wo}	Phenomena: {wo}
Saya makan kue	Saya kue makan
saya makan kue	Saya kue makan
1sg eat cake	1sg cake eat
'I eat cake(s).'	'I eat cake(s).'
#grammatical	#ungrammatical bec. the word order is SOV

Figure 2.8: A small extract of lab3 test-suite

grammars or web pages. Each example includes a number, optional comment(s) begin with #, source, vetted (t means checked with a native speaker, otherwise f), judgment (g for grammatical and u for ungrammatical), phenomenon, sentence in three layers: in standard orthography, with morpheme boundaries and morpheme-by-morpheme glosses, and translation.

After analyzing the phenomenon based on reference grammars and other linguistics literature, the analysis is modeled in HPSG and implemented by manually adding or editing some TDL files. The implementation of similar phenomenon in ERG, Jacy, and Zhong may also be referred to. Afterwards, the grammar is compiled and tested by parsing sample sentences or test-suites (both the previous and the recent ones). Typically, the grammar will not be able to parse some sample sentences and get the correct MRS or it will not have perfect coverage of the test-suites. Because of this, the grammar writer will investigate the problematic sentences which cannot be parsed well with correct MRSs or those which have high numbers of readings or parse results (higher than expected ambiguity). If the problems are found, the grammar will be debugged until both the new and the previous phenomena are covered correctly. Sometimes the debugging process takes a long time because a new analysis has to be made and modeled in HPSG and implemented subsequently. Afterwards, a solution is tested using the interactive environment of the LKB. If the grammar writer thinks it is correct, the sample sentences or the test-suites will be parsed again and treebanked using the new grammar, and a new profile will be created. The new profile can be compared against the previous ones in various dimensions, in-

Figure 2.9: The process of grammar development (Bender et al., 2011, p. 10)

cluding coverage and efficiency. This process goes repetitively, as shown in Figure 2.9. If problems are not found or the debugging process has finished with a good result, the grammar will be updated in GitHub.

In order to build a robust and broad-coverage grammar, the lexicon plays an important role. Increasing the amount of words in the lexicon will develop the grammar extensively. Lexical acquisition can be manually done, i.e. inputting the lexical items one by one, or semi-automatically done via linguistic resources such as Wordnet and annotated corpus, i.e. extracting and mapping the lexicon to the computational grammar automatically using a computer program and checking the correctness with native speaker(s) because it is anticipated that there will be many cases in which exact matching is not possible. Preliminary work on lexical acquisition is described in Section 5.1.1 on page 120.

In addition, because the number of open class words (e.g. nouns) grows, it is almost impossible to know how many there are and to acquire all of them to the computational grammar. Thus, the task of automatically identifying the lexical class of unknown words or *unknown word handling* is important to make the grammar more robust. DELPH-IN grammars such as Jacy and Zhong use ACE’s YY-mode in order to deal with this issue

(Song, 2015a; Song, 2015b).

2.4.2 Implementation in Type Description Language

This subsection describes how the HPSG and MRS analysis for the sentence *anjing menggonggong* “dogs bark” in Section 2.2 can be implemented using a *description language* in the LKB (see Section 2.3.3) in INDRA. The purpose of a description language is to allow the grammar designer to write definitions of types and of TFSs in a way which is readable by a machine. Copestake (2002, p. 17) states that the language in which the type and its constraint are defined is referred to as *description language*. The description language used in the grammar files in INDRA is similar to the Attribute-Value Matrix (AVM) notation and it is syntactically based on the Type Description Language (TDL) originally used in the DISCO/PAGE system (Krieger & Schäfer, 1994). TDL looks like an ASCIIised version of the AVM syntax (Copestake, 2002, pp. 99-100).

Copestake (2002) mentions that the source file organization in the LKB is based on the assumption that a single file contains only a single class of object. Thus, there is a single file for lexical entries and a single file for grammar rules. All these (source) files are read into the LKB under the control of a *script file*, which is written in *Lisp* programming language. I will explain firstly how to write the lexical items in `lexicon.tdl`, after that how to write the definitions of the lexical types in `indonesian.tdl`, and lastly the rules to make and combine the lexical items to a phrase or sentence in `indonesian.tdl` and `rules.tdl`. There is another, very important file which is the fundamental element of the LinGO Grammar Matrix (see Section 2.3.3) and provides the basics to construct semantics based on MRS, called `matrix.tdl`. It defines the top of the hierarchy of typed feature structures and general rules for combining subjects, complements, or modifiers with a head, as well as coordination structures. The description of this file is beyond the scope of this dissertation.

Lexical items such as *anjing* “dog” and *menggonggong* “bark” are simply listed in a file called `lexicon.tdl`. Example 21 shows the TDL descriptions of those lexical items.

(21)

```

anjing := nonhuman-noun-lex &
[ STEM < "anjing" >,
  SYNSEM.LKEYS.KEYREL.PRED "_anjing_n_rel" ].

menggonggong := intr-verb-lex &
[ STEM < "menggonggong" >,
  SYNSEM.LKEYS.KEYREL.PRED "_menggonggong_v_rel" ].

```

In TDL, a lexical entry is a triple consisting of lexical type (e.g. *nonhuman-noun-lex*), orthography (e.g. "anjing"), and semantic predicate symbol (e.g. "_anjing_n_rel"). The *type identifiers* *anjing* and *menggonggong* stand left of the symbol `:=` which is followed by at least one supertype (*nonhuman-noun-lex* for *anjing* and *intr-verb-lex* for *menggonggong*). Spaces may not occur in identifiers, such as type names, names of lexical entries, and features. A symbol `&` connects different supertypes or supertypes and constraints. It also represents unification (Copestake, 2002).

Square brackets `[ ]` define further constraints on the type. Inside the square brackets, features and their values are separated from each other by white spaces. There is a convention that features are written in uppercase and types in lowercase. Dots (.) can be used to define paths of features. Dots correspond to the vertical bars used in Section 2.2. Lists begin with a `<` and end with a `>`. Double quotation marks (" ") represent string values. A comma (,) separates list elements, e.g. `< a, b, c >`. If the lexical item contains more than one word, e.g. *anjing hutan* “dhole” (lit. “forest dog”), the value of the STEM would be `< "anjing", "hutan" >`.

A comma (,) separates the constraints inside the square brackets. Each lexical item specifies the value of its main relation (EP) through the feature `SYNSEM | LKEYS | KEYREL`, whose value is the relation of the item exposes for semantic composition. For each lexical item which has a predicate, we need to write explicitly the value of `PRED`, i.e. `_anjing_n_rel` and `_menggonggong_v_rel`, as a string inside double quotation marks. Lastly, all TDL descriptions are terminated by a full stop (.)

The lexical types *nonhuman-noun-lex* and *intr-verb-lex* encode both syntax and a skeleton semantic structure. They are described in a file called `indonesian.tdl`. The lines of code in 22 shows the descriptions of the lexical type *nonhuman-noun-lex* and its supertypes.

(22)

```

nonhuman-noun-lex := animate-noun-lex &
[ SYNSEM.LOCAL.CONT.HOOK.INDEX.PNG.ANIMACY non-human ].

```

```

animate-noun-lex := countable-noun-lex &
  [ SYNSEM.LOCAL.CONT.HOOK.INDEX.PNG.ANIMCOUNT animate ].

countable-noun-lex := common-noun-lex &
  [ SYNSEM.LOCAL.CONT.HOOK.INDEX.PNG.ANIMCOUNT countable ].

common-noun-lex := noun-lex &
  [ SYNSEM.LOCAL [ CAT.HEAD commonnoun,
    CONT.HOOK.INDEX.PNG.PERNUM 3rd ] ].

noun-lex := basic-noun-lex & basic-one-arg & no-hcons-lex-item &
  non-mod-lex-item &
  [ SYNSEM.LOCAL.CAT.VAL [ SPR < #spr &
    [ LOCAL.CAT.HEAD det ] >,
    COMPS < >,
    SUBJ < >,
    SPEC < > ],
    ARG-ST < #spr > ].

```

The lexical type *nonhuman-noun-lex* has a supertype *animate-noun-lex* with a constraint: the ANIMCOUNT's value is *non-human*. The type *animate-noun-lex* has a supertype *countable-noun-lex* with a constraint: the ANIMCOUNT's value is *countable* (see the noun type hierarchy in 10 on page 16). The type *countable-noun-lex* inherits from *common-noun-lex* which has *3rd* as its PERNUM's value. The type *noun-lex* has four supertypes and constraints: the value of its SPR is a list of one feature structure having *det* as its HEAD. This feature structure is coindexed with the item inside the list of ARG-ST. The coindexation is represented by a symbol #. A symbol & connects the coindexation with the coindexed type or feature structure. The COMPS, SUBJ, and SPEC's values are all empty lists, represented by the notation < >, which represents a list with no elements. The four supertypes of *noun-lex* are described in `matrix.tdl`.

The lines of code in 23 shows the descriptions of the lexical type *intr-verb-lex* and its supertypes.

(23)

```

intr-verb-lex := intransitive-verb-lex.

intransitive-verb-lex := main-verb-lex & intransitive-lex-item &
  [ SYNSEM.LOCAL.CAT [ HEAD intransitive,
    VAL.COMPS < > ] ].

main-verb-lex := indonesian-verb-lex & basic-verb-lex &
  [ ARG-ST.FIRST.LOCAL.CAT.HEAD subj-noun ].

```

```

indonesian-verb-lex := verb-lex &
  [ SYNSEM.LOCAL [ CAT [ HEAD.AUX -,
                        VAL [ SPR < >,
                            SPEC < >,
                            SUBJ < #subj > ] ],
    CONT.HOOK.XARG #xarg ],
  ARG-ST.FIRST #subj &
    [ LOCAL [ CAT.VAL [ SPR < >,
                        COMPS < > ],
    CONT.HOOK.INDEX #xarg ] ] ].

```

```

verb-lex := non-mod-lex-item &
  [ SYNSEM.LOCAL.CAT [ HEAD verb,
                      HC-LIGHT - ] ].

```

intr-verb-lex has two supertypes: *main-verb-lex* and *intransitive-lex-item*, with constraints: the value of *COMPS* is an empty list and the head of the first argument in *ARG-ST* is *noun*. The type *main-verb-lex* inherits from *indonesian-verb-lex* and *basic-verb-lex*. *indonesian-verb-lex* inherits from *verb-lex* which has the following constraints: its *HEAD.AUX*'s value is minus which means that it can take auxiliaries, the value of its *SUBJ* is coindexed with the value of the first argument in *ARG-ST*, the *INDEX* of its *SUBJ* is coindexed with the *XARG*, and the values of *SPR* and *COMPS* in the *SUBJ* are empty lists which means the *SUBJ* must be a phrase. *verb-lex* inherits from *non-mod-lex-item* with *HEAD*'s value is *verb* and *HC-LIGHT*'s value is minus. The descriptions of *intransitive-lex-item*, *basic-verb-lex*, and *non-mod-lex-item* are in *matrix.tdl*.

Lastly, the rules *bare-np-phrase* and *subj-head-phrase* are also described in *indonesian.tdl*, shown in 24.

(24)

```

bare-np-phrase := basic-bare-np-phrase &
  [ SYNSEM.LOCAL.COORD -,
    C-CONT.RELS <! [ PRED "exist_q_rel" ] !> ].

subj-head-phrase := decl-head-subj-phrase & head-final &
  [ HEAD-DTR.SYNSEM.LOCAL.CAT.VAL.COMPS < > ].

```

bare-np-phrase inherits from *basic-bare-np-phrase* with the following constraints: the value of *COORD* (COORDINATION) is minus⁴⁶ and it adds a new relation (EP) having a *PRED* of type *exist_q_rel* in the *C-CONT*. *subject-head-phrase* inherits from *decl-head-subj-phrase* and *head-final* with a constraint: the *HEAD-DTR*'s *COMPS* should have an empty list,

⁴⁶*COORD* is a boolean which is used to enforce the syntax of coordination. *COORD* must be minus (-) for all phrases that are not participating in coordination.

Figure 2.10: Screenshot of FFTB main page for MRS test-suite treebanking

in other words, the *HEAD-DTR* must be a *VP*. The descriptions of *basic-bare-np-phrase*, *decl-head-subj-phrase*, and *head-final* are in `matrix.tdl`.

The rules must be described in a special file called `rules.tdl`. The descriptions are shown in 25, which state that *bare-np* inherits from *bare-np-phrase* and *subj-head* inherits from *subj-head-phrase*.

(25)

```
bare-np := bare-np-phrase.
subj-head := subj-head-phrase.
```

2.4.3 Treebanking

A *treebank* is a syntactically annotated corpus of sentences with parse trees. Treebanking is included into the grammar development process. Treebanks are important in natural language processing, for example for grammatical knowledge acquisition or statistical language model induction. In the DELPH-IN community, they are semi-automatically built using Full Forest Treebanker (FFTB). Figure 2.10 shows an FFTB page of the Indonesian MRS test-suite. MRS test-suite contains a representative set of sentences designed to show some of the semantic phenomena (see also Section 2.4.4).⁴⁷

FFTB is a tool for treebanking with DELPH-IN grammars that allows the users to select manually a tree from the “full forest” of possible trees without listing or specifying all analyses in the parsing stage and store it into database for statistical ranking of candidate parses, transfers, and translations. The basic approach to the syntactic annotation

⁴⁷<http://moin.delph-in.net/MatrixMrsTestSuite>

Figure 2.11: Screenshots of treebanking process of *anjingnya sedang menggonggong* “the dog is barking”

is grammar-based corpus annotation. First, the corpus or the test-suite is parsed using a grammar, and then the treebanker or the annotator selects the correct analysis for each item in the corpus, or occasionally rejects all analyses. The selection process is done through a choice of *discriminants*. The system computes features that distinguish between parsers and the annotator selects or rejects features until only one parse is left. The left figure in Figure 2.11 shows that the grammar has two candidate trees of the following sentence.

(26) *Anjingnya sedang menggonggong.*

dog=DEF PROG bark

“The dog is barking.” (MRS Test-suite no. 65)

The enclitic *=nya* in the grammar has two representations (it is displayed as *n* in the screenshots in Figure 2.11): as a definite marker (*determiner-nya-lex*) like “the” and as a 3SG possessive marker (*enclitic-pronoun-3rd-sg-nohon-noun-lex*) like “his” or “her” in English. Because the intended meaning is “the dog” not “his/her dog”, *determiner-nya-lex* was chosen and the intended tree was parsed and treebanked in the right figure.

Using FFTB, the grammar writers or treebankers can note some interesting findings or linguistic analyses item by item and refine the grammar so that it can have wider coverage. If the grammar writers made some changes to the grammar, the treebankers only need to look at items in the corpus or test-suite which were affected, for example in coverage (more items can be parsed), precision (more items have correct semantics) or in the range of interpretation (become more ambiguous so that new decisions need to be made). The data stored in the treebank is the derivation trees which consist of rules and lexical items used to construct each parse.

2.4.4 Grammar evaluation

During the grammar development process, regression testing is important when the grammar writers make some changes to the grammar. INDRA is evaluated using a regression testing tool called gTest.⁴⁸ gTest is a DELPH-IN grammar testing tool which helps automate testing of DELPH-IN style HPSG grammars. The tool supports regression tests against gold [incr tsdb()] profiles, as well as coverage tests and semantic tests against [incr tsdb()] skeletons.

2.5 Summary

My PhD research focuses on building and developing a robust Indonesian resource grammar (INDRA) for parsing and generating the “High” variety of Indonesian sentences within the framework of HPSG and MRS using tools developed by DELPH-IN community. INDRA is aimed to be a resource grammar, for various purposes and applications. One of many applications is to build a syntactically and semantically annotated corpus or a treebank.

This chapter has described the historical and sociolinguistic background of the Indonesian language here, as well as an overview of the grammar. In spite of the rich literature of the Indonesian grammar, syntactic and semantic analysis in the framework of HPSG and MRS have not yet received much attention. As an open-source computational grammar, INDRA is an implementation of Indonesian HPSG and MRS. This chapter also has provided a brief description of the HPSG and MRS theoretical framework, as well as the tools developed in the DELPH-IN community which are employed to build and develop INDRA. Wordnet Bahasa is employed as a lexical source and Indonesian POS Tagger is used for unknown word handling in INDRA. Finally, this chapter has presented a brief overview of the research method, including implementation in TDL files, treebanking, and grammar evaluation. The latest version of INDRA is stored in GitHub and can be downloaded under the MIT license.

⁴⁸<https://github.com/goodmami/gtest>

PART II:

INDRA

Chapter 3

Basic structures

This chapter describes the basic structures of INDRA. Firstly, the initial stage of creation and development of INDRA using an open-source starter kit for the development of broad-coverage precision HPSG grammars, called the LinGO grammar matrix, is described in Section 3.1. Since its creation, grammar components such as the lexicon, grammar rules and constraints have been gradually added. Section 3.2 gives the core phrase structure rules and outlines some constituent word orders in INDRA. Other non-core rules such as fragments are also explained. All rules are from the grammar matrix, unless otherwise stated. Finally, Section 3.3 gives a summary.

3.1 LinGO Grammar Matrix

INDRA was created using the Linguistic Grammars Online (LinGO) grammar matrix in January 2014, firstly by filling in the required sections of the online page of the LinGO Grammar Matrix customization questionnaire,¹ hosted and developed at the University of Washington (Bender et al., 2002; Bender et al., 2010). The LinGO Grammar Matrix covers basic grammar phenomena such as word order, tense-aspect-mode (TAM), sentential negation, coordination, and morphology. Table 3.1 on the following page summarizes the phenomena covered, the options chosen for Indonesian, and the evaluation source from some reference grammars.

The lexical items for each noun and verb subcategory were added and the morphological inflection affixes to support the active-passive voice were also included in the Grammar Matrix. However, the Matrix does not handle sound changes as in the nasal assimilation process of *meN-* “ACT” and thus has to be manually added. Section 5.2.1 on page 138 discusses this manual extension. Other phenomena which cannot be handled using the Grammar Matrix were covered by modifying the TDL files manually. These include clitics, decomposed words, modals, adverbs, relative clauses, compounds, reduplications,

¹<http://matrix.ling.washington.edu/customize/matrix.cgi>

Table 3.1: Phenomena covered by filling in the LinGO Grammar Matrix questionnaire (M = Mintz (2002), L = Liaw (2004), S = Sneddon et al. (2010), A = Alwi et al. (2014), the number following each letter is the page number)

Phenomena	Chosen options for Indonesian	Source
Basic word order	SVO	L186, S265, A329
Word order of DET and nouns	Noun-DET	M104, L32, S133-134, A267
Word order of AUX and verbs	AUX-VP	M45-46, L62-68, S204-211, A163-167
Number	SG and PL	M281-285, L4, S20-22, A290-292
Person	1, 2, 3 (INCL and EXCL in 1PL)	M86-94, L29, S165, A256
Tense	FUT <i>akan</i> and underspecified	M72-73, L67, Arka (2013a)
Aspect	PRF <i>sudah</i> and PROG <i>sedang</i>	M(75, 82), L63, S204-205, A165
Sentential negation	<i>tidak</i> as an ADV modifying VP	M299, S202, A391
Coordination	monosyndeton <i>dan</i> (last coordinand is marked, e.g. “A B and C”, also allows “A and B and C”)	M63-65, L165-167, S346-347, A303
Yes/no question	sentence initial question word <i>apakah</i>	M262, L244, S320, A366
Lexical types	Noun subcategorization and verb subcategorization	L111-114, S(65, 72, 139), A(95-98, 288-290)
Lexicon	nouns, verbs, adjectives, auxiliaries and determiners	own data
Morphology	active <i>meN-</i> and passive <i>di-</i> voice inflection	S29

Table 3.2: Constituent orders in Indonesian

Parameter	Constituent order	Phrase structure rule	Head
Subject and predicate	NP-VP/PP/AP	<i>subj-head</i>	final
Predicate and object	V-NP	<i>head-comp</i>	initial
Agent and head verb in passive II	NP-V	<i>comp-head</i>	final
Predicate and optional object	V(-NP)	<i>basic-head-opt-comp</i>	initial
Predicate and modifier	VP-ADV/PP	<i>head-adj-int</i>	initial
Auxiliaries and modals	Aux-V	<i>head-comp</i>	initial
Question words	Sentence initial	<i>head-comp</i>	initial
Head VP/AP/PP and negation	Neg-VP/AP/PP	<i>adj-head-scop</i>	final
Scopal adverb	ADV-VP	<i>adj-head-scop</i>	final
Adpositions	P-NP	<i>head-comp</i>	initial
Head noun and demonstratives	N-D	<i>head-spec</i>	initial
Head noun and adjective	N-ADJ	<i>head-adj-int</i>	initial
Head noun and relative clauses	N-RelCl	<i>head-adj-int</i>	initial
Numbers and head noun	Num-N	<i>adj-head-int</i>	final
Head noun and numbers+classifiers	NumCL-N	<i>adj-head-int</i>	final
Head adjective and modifying adverb	ADV-ADJ	<i>adj-head-int</i>	final
Coordination of two items	X-Conj-Y	<i>coord</i>	headless
Coordination of more than two items	X-Y-Conj-Z	<i>coord</i>	headless

and copula constructions. Some phenomena, such as pronouns, proper names, and adjectives, were added via the Grammar Matrix and were subsequently constrained so that they cannot parse phrases such as **saya kaya* (Intended: “rich I”). Indonesian derivational affixes were not dealt with in INDRA because they are computationally expensive, i.e. processing them needs a lot of memory. Rather than combining roots with derivational affixes using rules, all derived words or lexical items are listed up in the lexicon. After creating INDRA via the LinGO Grammar Matrix, a small test-suite called *lab3*, containing 66 positive (grammatical) and 40 negative (ungrammatical) examples was created to evaluate the grammar (as Lab 3 of HG7021 at NTU).

3.2 Phrase structure rules and constituent order

This section outlines some core phrase structure rules, syntactic headedness, and constituent orders in Indonesian, which are modeled in INDRA. The syntactic head of a construction is the constituent which determines the syntactic distribution of the whole, i.e. the one which encodes the part of speech information which determines the syntactic distribution of an element and contributes the head information. In Indonesian, the syntactic head tends to precede other constituents (head-initial). However, Indonesian also has head-final phrases, as shown in Table 3.2.

3.2.1 Head-Initial Constructions

Indonesian has a strong tendency to be head-initial (Macdonald, 1976, pp. 24-25; Sneddon et al., 2010, pp. 26-28). In a noun phrase with an adjective, a demonstrative or a relative clause, the head noun precedes the adjective, the demonstrative or the relative clause. The type *head-initial* is defined in the matrix. It inherits from *binary-headed-phrase* with some constraints, as shown in (27). It constraints the arguments to have two items, the first of which is the head daughter preceding the second one (non-head daughter).

$$(27) \quad \text{head-initial :} \left[\begin{array}{ll} \text{binary-headed-phrase} & \\ \text{HEAD-DTR} & \boxed{1} \\ \text{NON-HEAD-DTR} & \boxed{2} \\ \text{ARGS} & \langle \boxed{1}, \boxed{2} \rangle \end{array} \right]$$

As shown in Table 3.2, *head-initial* is used in many phrase structure rules, such as *head-comp* and *head-spec* rules, which will be described in the next subsections.

3.2.2 Head-Final Constructions

Indonesian also has head-final phrases, e.g. in a noun phrase construction where numerals and classifiers precede the head noun (Alwi et al., 2014, pp.251-255). Similar to *head-initial*, the type *head-final* is also defined in *matrix.tdl*. It inherits from *basic-binary-headed-phrase* with some constraints, as shown in (28). Different from *head-initial*, the first item in the *ARGS* list of *head-final* is the non-head daughter. It is followed by the head-daughter.

$$(28) \quad \text{head-final :} \left[\begin{array}{ll} \text{basic-binary-headed-phrase} & \\ \text{HEAD-DTR} & \boxed{1} \\ \text{NON-HEAD-DTR} & \boxed{2} \\ \text{ARGS} & \langle \boxed{2}, \boxed{1} \rangle \end{array} \right]$$

head-final is used in phrase structure rules such as *subj-head*, which is described in the following subsection.

3.2.3 Head-Subject Phrase

The types used to define the phrase structure rules are arranged into a hierarchy. The part of the simplified type hierarchy relevant for the rules discussed in this section is shown in (29).

The type *head-nexus-phrase*, which serves as the root of this small hierarchy, inherits from *headed-phrase*, which is the locus of the Head Feature Principle: the HEAD value of the mother is identified with the HEAD value of the HEAD-DTR. This constraint is shown in (30).

(30)

$$\left[\begin{array}{l} \textit{headed-phrase} \\ \text{HEAD} \quad \boxed{1} \\ \text{HEAD-DTR} \mid \text{HEAD} \quad \boxed{1} \end{array} \right]$$

The type *head-nexus-phrase* passes up the NON-LOCAL.REL and NON-LOCAL.QUE values of the head daughter to the mother (see Sag, 1997). The type *head-valence-phrase* passes up the NON-LOCAL.SLASH value of the head daughter to the mother.

The phrase structure rule *subj-head* inherits from *head-final* and *decl-head-subj-phrase* which is from *basic-head-subj-phrase* with some constraints, as shown in (31).

(31)

$$\textit{decl-head-subj-phrase} : \left[\begin{array}{l} \textit{basic-head-subj-phrase} \\ \text{SUBJ} \langle \rangle \\ \text{HEAD-DTR} \mid \text{SUBJ} \langle \boxed{1} \rangle \\ \text{NON-HEAD-DTR} \mid \text{SYNSEM} \boxed{1} \end{array} \right]$$

The HEAD-DTR's SUBJ is not empty and needs to be filled in by the NON-HEAD-DTR. The *subj-head* rule makes the value of the mother's SUBJ empty. In INDRA, one more constraint is added: the value of the head daughter's COMPS should be empty, i.e. this rule can be applied if the COMPS is saturated. It is shown in (32).

(32)

$$\textit{subj-head-phrase} : \left[\begin{array}{l} \textit{decl-head-subj-phrase} \\ \text{HEAD-DTR} \mid \text{COMPS} \langle \rangle \end{array} \right]$$

This rule is used to connect a subject with its predicate, for example *anjing* “dog” with *menggonggong* “bark” in the clause *anjing menggonggong* “dogs bark”, as shown previously in (12).

3.2.4 Head-Complement Phrase

The phrase structure rule *head-comp* inherits from *head-initial* and *basic-head-1st-comp-phrase* which is from *basic-head-comp-phrase* with some constraints, as shown in (33). This rule makes the value of the mother’s COMPS empty by filling in the value of the head daughter’s COMPS with the non-head daughter.

$$(33) \quad \text{basic-head-1st-comp-phrase} : \left[\begin{array}{l} \text{basic-head-comp-phrase} \\ \text{COMPS } \langle \rangle \\ \text{HEAD-DTR} \mid \text{COMPS } \langle \boxed{1} \rangle \\ \text{NON-HEAD-DTR} \mid \text{SYNSEM } \boxed{1} \end{array} \right]$$

head-comp connects a predicate with its object, for example *mengejar* “ACT-chase” with Budi in the phrase *mengejar Budi* “ACT-chase Budi”, as shown in (171).

In addition to *head-comp*, I made a rule *comp-head*. The type *comp-head*, on the other hand, inherits from *head-final* and *basic-head-1st-comp-phrase* with an additional constraint: the value of the head daughter’s HEAD is of type *passive-two* since this rule is only applied in passive type two construction, as shown in (34).

$$(34) \quad \text{comp-head-phrase} : \left[\begin{array}{l} \text{basic-head-1st-comp-phrase} \\ \text{HEAD-DTR} \mid \text{HEAD } \text{passive-two} \end{array} \right]$$

comp-head connects the NP agent of passive type two with its bare passive verb, for example *dia* “3SG” with *jemput* “pick up” in the phrase *dia jemput* “picked up by him/her”, as shown in (191).

3.2.5 Head-Specifier Construction

The phrase structure rule *head-spec* combines a noun with its demonstrative, as shown in (96) and (97). It inherits from *head-initial* and *basic-head-spec-phrase* with some constraints, as shown in (35).

$$(35) \quad \text{head-spec-phrase : } \left[\begin{array}{l} \text{basic-head-spec-phrase} \\ \text{SPR} \langle \rangle \\ \text{HEAD-DTR} \mid \text{SPR} \langle \boxed{1} \rangle \\ \text{NON-HEAD-DTR} \mid \text{SYNSEM} \boxed{1} \end{array} \right]$$

This rule makes the value of the mother's SPR empty by filling in the value of the head daughter's SPR with the non-head daughter.

3.2.6 Head-Adjunct Phrase

Phrase rules related with adjuncts come from a supertype *head-mod-phrase* and its subtype *basic-head-mod-phrase-simple*, as shown in (36). They are all defined in `matrix.tdl`.

The phrase rule *head-adj-int* (*head-adjunct-intersective*) combines a head with its modifiers, for example a verb with its adverbs, as shown in (204), or a noun with its adjectives and relative clauses, as shown in (109) and (127). It inherits from *head-adj-phrase* and *isect-mod-phrase*. The type *head-adj-phrase* inherits from *basic-head-mod-phrase-simple* and *head-initial*. The type *isect-mod-phrase* inherits from *basic-head-mod-phrase-simple*. This rule identifies the LTOP of the non-head daughter (adjunct) with the LTOP of the head daughter. It contributes the value of the LOCAL feature of the modifiee, of type *intersective-mod*. Its AVM is shown in (37).

$$(37) \quad \left[\begin{array}{l} \text{basic-head-mod-phrase-simple} \\ \text{HEAD-DTR} \mid \text{LTOP} \boxed{1} \\ \text{NON-HEAD-DTR} \left[\begin{array}{l} \text{HEAD} \mid \text{MOD} \langle \text{LOCAL } \textit{intersective-mod} \rangle \\ \text{LTOP} \boxed{1} \end{array} \right] \end{array} \right]$$

The phrase rule *adj-head-int* combines a numeral with a noun, as shown in (101) and (106a), and adverb of degree with an adjective, as shown in (109). It inherits from *adj-head-phrase* and *isect-mod-phrase*. *adj-head-phrase* inherits from *basic-head-mod-phrase-simple* and *head-final*.

The phrase rule *adj-head-scop* (*adjunct-head-scopal*) combines a negation word (e.g. *tidak* “NEG”) with its negated constituent and a scopal adverb (e.g. *mungkin* “maybe”) with its scoped constituent, as shown in (212) and (238). It inherits from *adj-head-phrase* and *scopal-mod-phrase*. The *scopal-mod-phrase* rule identifies the HOOK of the non-head daughter (adjunct) with the HOOK of the constructed phrase in the C-CONT. It contributes the value of the LOCAL feature of the modifiee, of type *scopal-mod*. Its AVM is shown in (38).

$$(38) \left[\begin{array}{l} \text{basic-head-mod-phrase-simple} \\ \text{NON-HEAD-DTR} \left[\begin{array}{l} \text{HEAD} \mid \text{MOD} \left\langle \left[\text{LOCAL } \textit{scopal-mod} \right] \right\rangle \\ \text{HOOK } \boxed{1} \end{array} \right] \\ \text{C-CONT} \mid \text{HOOK } \boxed{1} \end{array} \right]$$

3.2.7 Coordination

Coordination is morphosyntactic means of linking two clauses of equal grammatical status (Payne, 2008, p. 336). In coordination, two or more elements of the same or similar grammatical category are combined into a single larger element of the same category. Indonesian uses lexical marking in form of coordinators or conjunctions *dan* “and”, *atau* “or”, *tetapi* “but”, *serta* “as well as” to link words, phrases, and clauses. Indonesian conjunction/coordinator usually occurs once, preceding the final coordinand, similar to English *A, B, and C*. They can combine all phrase types including NP, VP, and AP.

INDRA uses the analysis and implementation of coordination in Drellishak & Bender (2005), the same as the one in the Matrix. Since there are differences in the semantic effects of coordination for individuals and events, e.g. NP coordination rules must introduce a quantifier, coordinated NPs have plural number etc, separate rules for the coordination of different phrase types are posited, as illustrated in (40) and (43). For each phrase type, three coordination rules are posited to produce an unlimited number of coordinands: *top-coord-rule*, *mid-coord-rule*, and *bottom-coord-phrase*. Parse tree (40) and (43) show these rules. A boolean feature COORD is defined on the value of LOCAL. The value of COORD of lexical items, ordinary phrase structure, and lexical rules is minus. The rules *mid-coord-rule* and *bottom-coord-phrase* make it plus. *top-coord-rule* and *mid-coord-rule* derive from *binary-phrase*, rather than *binary-headed-phrase*, and they are thus headless, in order to prevent some problems with agreement.

Example (39) shows coordination in noun phrase. It consists of three NP constituents with a conjunction *dan* “and” preceding the last one. Its parse tree is shown in (40) and its DMRS in (41). The conjunction *dan* “and” is of type *conj-lex* and is defined in the

Matrix. Its *HEAD* is of type *conj* and it modifies nothing. Its *SPR*, *COMPS*, and *SUBJ* are saturated. It introduces *coordination-relation* in the semantics. The value of its *PRED* is *_and_coord_rel*, its *LBL* and *C-ARG* are coindexed with the *LTOP* and *INDEX*.

(39) *Aceh, Medan, dan Palembang*

Aceh Medan and Palembang

“Aceh, Medan, and Palembang” (based on Sneddon et al., 2010, p. 329)

Parse tree (40) shows how the three coordination rules interact for NPs. In INDRA, the *np-coord-phrase* rule needs to be constrained as follows: the *HEAD* value of the left and right daughter, as well as the entire phrase should be of type *subj-noun*. The coordination rule *np1-bottom-coord* takes the conjunction as *CONJ-DTR* and combines it with the last NP constituent (*NONCONJ-DTR*), makes them an *NP-B* phrase, and adds a relation *_and_coord_rel*. It changes the value of *COORD* of the *NONCONJ-DTR* from minus to plus. It links the *ARG0* of the *NONCONJ-DTR* to the *R-INDEX* of the *_and_coord_rel*. The *np1-mid-coord* rule takes one left NP constituent as *LCOORD-DTR* (the *COORD* value is minus) and *NP-B* as *RCOORD-DTR* (the *COORD* value is plus), makes them an *NP-M* phrase with a plus *COORD* value, and adds an implicit coordination relation *coord_c_implicit_rel*. The implicit coordination relation has features *L-INDEX* and *R-INDEX* whose values are identified with the *INDEX* of the *LCOORD-DTR* and *RCOORD-DTR* respectively. Its *LBL* and *C-ARG* are identified with the *LTOP* and *INDEX* of the *NP-M* phrase. The *np1-top-coord* rule takes the first NP constituent as the left daughter (the *COORD* value is minus) and *NP-M* as the right daughter (the *COORD* value is plus), makes them an *NP-T* phrase with a minus *COORD* value, and adds a quantifying relation *exist_q_rel*.

Example (42) shows coordination in verb phrase. It consists of three active transitive verbs with a conjunction *dan* “and” preceding the last one. All verbs have the same subject and object. In Indonesian, when two or more transitive verbs having the same subject and object are coordinated, the subject is usually occur before the first verb and the object after the last verb (Sneddon et al., 2010, p. 348). The parse tree of Example (42) is shown in (43) and its DMRS in (44).

(42) *Adi mengejar, menangkap, dan memukul Budi.*

Adi ACT-chase ACT-catch and ACT-hit Budi

Adi chased, caught, and hit Budi. (own data)

(43) a.

subj-head

(44)

(45) a. *Budi petani, bekerja keras, dan sibuk setiap hari.*
 Budi farmer work hard and busy every day
 “Budi is a farmer, works hard, and is busy everyday.” (own data)

- b. *Mereka sudah pergi ke Jakarta atau ke Bandung.*

3PL PRF go to Jakarta or to Bandung

“They have gone to Jakarta or to Bandung.” (Sneddon et al., 2010, p. 347)

3.2.8 Subordination

An Indonesian sentence can consist of a main clause, which can stand alone as a sentence, and a subordinate clause, which is preceded by a subordinating conjunction (or a subordinator) and cannot stand alone, occurring either before or after the main clause to add information to it (Sneddon et al., 2010, pp. 349-350). Subordinating conjunctions link a subordinate clause to the main clause in a sentence. However, some of them, such as *karena* “because”, *sebelum* “before”, *sesudah* “after”, also function as prepositions which link a noun phrase to the predicate (Sneddon et al., 2010, pp. 199-200; Liaw, 2004, pp. 175-176; Alwi et al., 2014, p. 302), as shown in Figure 3.1.

Figure 3.1: Prepositions and subordinators in Indonesian

Example (46) shows the word *karena* “because” functions both as a subordinator (46a) and as a preposition (46b). The parse tree of Example (46a) is shown in (48).

- (46) a. *Dia tidak datang karena ayahnya meninggal.*

3SG NEG come because father=3SG die

“He didn’t come because his father died.” (Sneddon et al., 2010, p. 199)

- b. *Dia tidak datang karena kematian ayahnya.*

3SG NEG come because death father=3SG

“He didn’t come because of the death of his father.” (Sneddon et al., 2010, p. 199)

In INDRA, subordinators belong to a lexical type *subord-conj-lex*, which inherits from *basic-subord-conjunction-lex* and *basic-icons-lex-item*. They are treated as scopal adverbs modifying a main clause and take an obligatory subordinate clause as its COMPS. The AVM

(48) a.

b.

(49)

3.2.9 Fragments

In addition to the core phrase structure rules mentioned in the previous subsections, IN-DRA has a rule for fragments, i.e. non-sentential utterances like *Budi*, as an answer to the question *Siapa yang pergi ke sana?* “Who went there?”. I follow Schlangen & Lascarides (2003) for the analysis and implementation of fragments. Although they are non-sentential, their intended meaning is typically associated with full sentences.

The fragment rule for noun phrases is a unary phrase rule, shown in (50). It takes a saturated noun phrase as its daughter. The noun phrase daughter is lifted to the level of sentences. The root-sign has all the syntactic features of a sentence. The value of its HEAD is of type *fragment_head*. All valence requirements are satisfied. It is semantically like a sentence. The rule adds the basic element of the semantics of fragments *unknown-v-relation* which inherits from *event-relation* via C-CONT. It introduces a predicate *unknown_v_rel*, an anaphoric element expressing the underspecification in the content of fragments. Its only argument slot ARG is co-indexed with the INDEX of the noun phrase

daughter. This rule produces a super-saturated fragment phrase whose `HEAD` value is of type *fragment_head*, which inherits from *head*. The meaning of the fragment independent from its context is that it will resolve to a proposition, of which the main predicate is unknown, but one participant in the main event of the proposition is specified, even though its exact role is not.

The sign for the NP fragment *Budi* is shown in (51) and its DMRS in (52).

(51) FRG
 |
 NP
Budi

3.3 Summary

This chapter starts with the creation and initial development stage of INDRA using the LinGO Grammar Matrix which covers a handful of basic, core phenomena and structures. This chapter has also presented the basic phrase structure rules in INDRA and outlined the basic constituent orders. The main constituent orders and phenomena can be covered by using a small number of phrase structure rules. While most of the phrase types needed for Indonesian are head-initial (Sneddon et al., 2010), there are a handful of instances of head-final phrases. The coordination rules in the matrix at present cannot connect constituents having different parts-of-speech. One of the possible solutions is to make new rules for each possible combination. This is for future work. Other phenomena not covered in the matrix, such as fragments, were analyzed and implemented by manually editing the TDL files. Next chapters in this part describe these phenomena and their implementation in INDRA.

Chapter 4

Nouns

This chapter describes my analysis of Indonesian noun phrases. The class of nouns includes words such as *batu* “rock”, *pohon* “tree”, and *rumah* “house”. They express the most time-stable concepts, i.e. they characteristically do not vary considerably over time. Prototypically, they express highly and obviously time-stable concepts (Payne, 2008, p. 33). Semantically, they refer to entities. In MRS, their predicates have ARG0 and the value is a referential index, represented by x. Syntactically, they serve as heads of noun phrases,¹ which usually have the grammatical functions as subjects and objects of clauses. Payne (2008, p. 86) states that noun phrase elements include determiners, numerals, genitives (possessors), modifiers (attributive adjectives), relative clauses, noun classifiers, and the head noun. The order of the elements in an Indonesian noun phrase is described in Alwi et al. (2014, pp. 251-255). A complex Indonesian noun phrase is shown in Example (53). The underlined word *anjing* “dog” in the example is the head noun, i.e. the noun that is modified by all the other elements.

- (53) *tiga ekor anjing besar adik saya yang menggonggong itu*
three CLF dog big younger.sibling 1SG REL bark that
“those three big dogs of my younger brother that are barking” (based on Payne, 2008, p. 86)

The parse tree of Example (53) is shown in (54) and its DMRS in (55). As can be seen from Example (53), there is no morphological distinction between singular and plural in an Indonesian noun, unlike in English *dogs* with a plural bound morpheme *-s*. However, there are some ways, such as numerals and reduplication, that mark the concept of plurality in Indonesian. These markers or noun modifiers will be explained in this chapter.

¹The head of an NP is the one word within the phrase that refers to the same entity that the whole phrase refers to (Payne, 2008, p. 33).

```

graph TD
    NP1[NP] --- N1[N]
    NP1 --- D[D]
    D --- itu[itu]
    N1 --- CL1[CL]
    CL1 --- NUM[NUM]
    CL1 --- CL2[CL]
    NUM --- tiga[tiga]
    CL2 --- ekor[ekor]
    N1 --- N2[N]
    N2 --- N3[N]
    N2 --- yangP[N yangP/N]
    N3 --- N4[N]
    N3 --- NP1[NP]
    N4 --- N5[N]
    N4 --- ADJ1[ADJ]
    N5 --- anjing[anjing]
    ADJ1 --- ADJ2[ADJ]
    ADJ2 --- besar[besar]
    NP1 --- N6[N]
    N6 --- N7[N]
    N6 --- NP2[NP]
    N7 --- adik[adik]
    NP2 --- saya[saya]
    yangP --- yang1[yang]
    yangP --- S/N[S/N]
    S/N --- VP[VP]
    VP --- menggonggong[menggonggong]
  
```

The diagram illustrates the semantic network for the sentence "Kardus bekas anjing besar adik saya yang suka menggonggong itu mengganggu saya." The nodes represent the words in the sentence, and the edges represent the semantic relations between them. The relations are labeled with codes such as ARG1/EQ, ARG2/NEQ, and RSTR/H.

The main clause structure is as follows:

- card(3)** (Kardus) is related to **_ekor_x** (bekas) via **ARG1/EQ**.
- _ekor_x** is related to **_anjing_n** (anjing) via **ARG1/EQ**.
- _anjing_n** is related to **_besar_a** (besar) via **ARG1/EQ**.
- _besar_a** is related to **_adik_n** (adik) via **ARG1/EQ**.
- _adik_n** is related to **pronoun_n** (saya) via **ARG1/EQ**.
- pronoun_n** is related to **pronoun_q** (yang) via **ARG2/NEQ**.
- pronoun_q** is related to **poss** (suka) via **ARG2/NEQ**.
- poss** is related to **def_q** (itu) via **ARG1/EQ**.
- def_q** is related to **compound_p** (mengganggu) via **ARG1/EQ**.
- compound_p** is related to **_menggonggong_v** (mengganggu) via **ARG1/EQ**.
- _menggonggong_v** is related to **distal_q** (saya) via **ARG1/EQ**.

The relative clause structure is as follows:

- pronoun_q** (yang) is related to **poss** (suka) via **RSTR/H**.
- poss** (suka) is related to **def_q** (itu) via **RSTR/H**.
- def_q** (itu) is related to **compound_p** (mengganggu) via **RSTR/H**.
- compound_p** (mengganggu) is related to **_menggonggong_v** (mengganggu) via **RSTR/H**.

The diagram also shows a **TOP** node at the end, indicating the end of the sentence.

4.1 Noun subcategorization

Nouns in INDRA are subcategorized according to their syntactic functions, i.e. as a subject or a complement in a clause, as a possessor or a non-possessor dependent noun (non-head daughter) in a noun phrase, and as a complement (which has a semantic role as agent) in a passive type one construction or in a passive type two construction.

(56)

Type hierarchy in (56) shows the upper part of the noun hierarchy in INDRA. The type *noun* has four subtypes: *comp-noun* (its subtypes are shown in 58), *possessor* (its subtypes are shown in 57), *pass1agent* (its subtypes are shown in 60), and *pass2agent* (its subtypes are shown in 57).

(57)

The type hierarchy in (57) shows the type *pronoun-super* can be both the agent of a passive construction type two and the possessor dependent noun in a noun phrase. The type *pronoun-super* is the only possible type to be the agent of a passive construction type two in the noun hierarchy. It has two subtypes: *pronoun-substitute* and *pronoun*. The type *pronoun* refers to the *personal pronouns*, i.e. pronouns which refer to people, as mentioned in Sneddon et al. (2010, pp. 164-166). It is further divided into *pronoun-3person* and *pronoun-non3person*. It is described in Section 4.1.2. The type *pronoun-substitute* refers to the *pronoun substitutes* in Sneddon et al. (2010, pp. 166-168), i.e. personal names and kinship terms used instead of pronouns. Personal names, proper nouns, or proper names are the specific names of persons, spelled with a capital letter. They are commonly used as substitutes for “I” and “you”. They are described in Section 4.1.3. In addition to *pronoun-super*, the type *possessor* has another subtype *enclitic*. Sneddon et al. (2010, pp. 170-172) call this as the suffixed form of pronouns. This type has two subtypes: *enclitic-3person* and *enclitic-non3person*, as shown in (58) and will be explained in Section 4.2.

(58)

The type hierarchy 58 shows that in addition to *enclitic*, the type *comp-noun* has another type called *subj-noun*. The type *comp-noun* is the supertype for all nouns that can occur as complements of active verbs. The enclitics or the suffixed forms of pronouns can occur as objects of active verbs, as mentioned in Sneddon et al. (2010, p. 170). The type *subj-noun* has three subtypes: *dependent-noun* (its subtypes are shown in 59), *demnoun*, and *pronoun-super*. The type *dependent-noun* serves as the type for the dependent noun in a noun phrase. The type *demnoun* is the type for nouns which can occur as *demonstrative pronoun*, such as *ini* “this” and *itu* “that” (Sneddon et al., 2010, p. 164). It is described in Section 4.1.2.

(59)

The type *dependent-noun* has two subtypes: *pronoun-substitute* and *commonnoun*, as shown in (59). The type *commonnoun* is the type for *common noun* which refers to a person or thing which is a member of a set, not the specific name of that person or thing (Sneddon et al., 2010, p. 132). This type is further divided into three subtypes: *nounnyadef*, *nounbare*, and *nounpos*, which will be described in the next sections.

(60)

The type *pass1agent* serves as the semantic agent (or the syntactic complement) in passive type one. Sneddon et al. (2010, p. 256) note that in passive type one the agent or the actor is third person, that is pronoun or a noun. This type has three subtypes, i.e. *dependent-noun*, *pronoun-3person*, and *enclitic-3person*.

In the semantics, nouns have two special features: *PERNUM* for *person* and *number*; and *ANIMCOUNT* for *animacy* and *countability*, which will be explained in Section 4.1.1. In addition, for compatibility with DELPH-IN-standard MRS well-formedness, nouns must be associated with a quantifier. The type hierarchy for quantifier relations is shown in (61).

(61)

The type *quant_rel* has six subtypes. The subtype *exist_q_rel* has been mentioned in Section 2.2.2, it is an underspecified quantifier predicate which is named *udef_rel* in other HPSG grammars such as ERG and JACY. The subtype *demon_q_rel* is further defined in (68) in Section 4.1.2 on demonstrative pronouns. The subtype *indef_q_rel* and *pronoun_q_rel* are mentioned in Section 4.1.2 on indefinite pronouns and personal pronouns, respectively. The subtype *proper_q_rel* is mentioned in Section 4.1.3 on proper names. Finally, the subtype *def_q_rel* is discussed in Section 4.2 on clitics and in Section 4.3 on determiners.

INDRA has 23 lexical types for nouns. Table 4.1 on the following page shows all the lexical types, together with their *HEAD* types, *PERNUM* and *ANIMCOUNT* values, and examples.

4.1.1 Common nouns

Common noun refers to a person or thing which is a member of a set. It is not the specific name of that person or thing (Sneddon et al., 2010, p. 132). Semantically, it is normally a one-place predicate² with an *ARG0* whose value is of the type *ref-ind* (*referential-index*), represented by *x*, and the value of the feature *PERNUM* is *3rd*. AVM (62) shows a simplified MRS representation of *anjing* “dog”.

²Some nouns (relational nouns) are two-place predicates.

Table 4.1: Noun subcategorization in INDRA

Noun type	HEAD	PERNUM	ANIMCOUNT	Example	Freq.
<i>inanimate-noun-lex</i>	<i>commonnoun</i>	<i>3rd</i>	<i>inanimate</i>	<i>buku</i> “book”	2766
<i>nonhuman-noun-lex</i>	<i>commonnoun</i>	<i>3rd</i>	<i>non-human</i>	<i>anjing</i> “dog”	110
<i>human-noun-lex</i>	<i>commonnoun</i>	<i>3rd</i>	<i>human</i>	<i>manusia</i> “human”	271
<i>uncountable-noun-lex</i>	<i>commonnoun</i>	<i>3rd</i>	<i>uncountable</i>	<i>dedaunan</i> “foliage”	35
<i>proper-name-lex</i>	<i>propername</i>	<i>sg</i>	<i>animcount</i>	<i>Budi</i>	349
<i>pronoun-1st-sg-nohon-noun-lex</i>	<i>pronoun-non3person</i>	<i>1sg</i>	<i>animcount</i>	<i>aku</i> “1SG”	1
<i>pronoun-1st-sg-hon-noun-lex</i>	<i>pronoun-non3person</i>	<i>1sg</i>	<i>animcount</i>	<i>saya</i> “1SG”	1
<i>pronoun-1st-pl-incl-noun-lex</i>	<i>pronoun-non3person</i>	<i>1pl_incl</i>	<i>animcount</i>	<i>kita</i> “1PL.INCL”	1
<i>pronoun-1st-pl-excl-noun-lex</i>	<i>pronoun-non3person</i>	<i>1pl_excl</i>	<i>animcount</i>	<i>kami</i> “1PL.EXCL”	1
<i>enclitic-pronoun-1st-sg-nohon-noun-lex</i>	<i>enclitic-non3person</i>	<i>1sg</i>	<i>animcount</i>	<i>-ku</i> “1SG”	1
<i>pronoun-2nd-sg-nohon-noun-lex</i>	<i>pronoun-non3person</i>	<i>2sg</i>	<i>animcount</i>	<i>kamu</i> “2SG”	1
<i>pronoun-2nd-sg-hon-noun-lex</i>	<i>pronoun-non3person</i>	<i>2sg</i>	<i>animcount</i>	<i>Anda</i> “2SG”	1
<i>pronoun-2nd-pl-noun-lex</i>	<i>pronoun-non3person</i>	<i>2pl</i>	<i>animcount</i>	<i>kalian</i> “2PL”	1
<i>enclitic-pronoun-2nd-sg-nohon-noun-lex</i>	<i>enclitic-non3person</i>	<i>2sg</i>	<i>animcount</i>	<i>-mu</i> “2SG”	1
<i>pronoun-3rd-sg-nohon-noun-lex</i>	<i>pronoun-3person</i>	<i>3sg</i>	<i>animcount</i>	<i>dia</i> “3SG”	2
<i>pronoun-3rd-sg-hon-noun-lex</i>	<i>pronoun-3person</i>	<i>3sg</i>	<i>animcount</i>	<i>beliau</i> “3SG”	1
<i>pronoun-3rd-pl-noun-lex</i>	<i>pronoun-3person</i>	<i>3pl</i>	<i>animcount</i>	<i>mereka</i> “3PL”	1
<i>enclitic-pronoun-3rd-sg-nohon-noun-lex</i>	<i>enclitic-3person</i>	<i>3sg</i>	<i>animcount</i>	<i>-nya</i> “3SG”	1
<i>n+dem-lex</i>	<i>demnoun</i>	<i>3rd</i>	<i>animcount</i>	<i>ini</i> “this”	2
<i>n+det-lex</i>	<i>pronoun</i>	<i>3rd</i>	<i>animcount</i>	<i>sini</i> “here”	5
<i>n_-_month-lex</i>	<i>propername</i>	<i>3sg</i>	<i>inanimate</i>	<i>Januari</i> “January”	13
<i>n_-_day-lex</i>	<i>propername</i>	<i>3sg</i>	<i>inanimate</i>	<i>Minggu</i> “Sunday”	7
<i>n_-_temp-lex</i>	<i>commonnoun</i>	<i>3rd</i>	<i>inanimate</i>	<i>pagi</i> “morning”	15
<i>n_-_measure-lex</i>	<i>commonnoun</i>	<i>3rd</i>	<i>uncountable</i>	<i>hektar</i> “hectare”	1
<i>relational-n-lex</i>	<i>commonnoun</i>	<i>3rd</i>	<i>uncountable</i>	<i>depan</i> “front”	25

Table 4.2: Count nouns and mass nouns in Indonesian

Parameter	Count noun	Mass noun
Bare NP	<i>daun</i> “leaf”	<i>dedaunan</i> “foliage”
Quantifier	<i>semua daun</i> “all leaves”	<i>semua dedaunan</i> “all foliages”
Numeral	<i>dua daun</i> “two leaves”	? <i>dua dedaunan</i> “two foliages”
Reduplication	<i>daun-daun</i> “leaves”	?? <i>dedaunan-dedaunan</i>
Numeral and classifier	<i>dua buah daun</i> “two leaves”	?? <i>dua buah dedaunan</i>
Numeral and taxonomic cl.	<i>dua jenis daun</i> “two kinds of leaf”	<i>dua jenis dedaunan</i> “two kinds of foliage”

Sneddon et al. (2010, p. 132) and Macdonald (1976, p. 78) note that Indonesian common nouns can be further divided into count(able) nouns and (non-count or uncountable) mass nouns. Payne (2008, p. 41) states that count nouns are nouns that refer to things that can be counted and mass nouns are those that refer to substances, such as water, sand, and air.

In Indonesian, specifically, count nouns refer to things which can be counted or modified by numerals. They can be directly preceded by a numeral or a classifier. They can also be reduplicated to indicate plurality. On the other hand, uncountable nouns or mass nouns typically cannot be modified by numerals and classifiers or be reduplicated to indicate plurality. However, they can be preceded by certain mensural quantifiers to enumerate or show the quantity. Sneddon et al. (2010, p. 132) note that some mass nouns can be used as count nouns when the context makes it clear that a certain measure or amount is being referred to, e.g. *dua kopi* “two cups/glasses of coffee” (lit. “two coffees”). The syntactic differences between count nouns and mass nouns are summarized in Table 4.2.

The feature ANIMCOUNT specifies the countability and the animacy of a noun. The count(able) nouns can be modified by numerals and classifiers, while the uncountable nouns cannot. The type hierarchy in (63) shows the type *countable* are subcategorized into *inanimate*, *non-human*, and *human* based on the three main classifiers in Indonesian: the classifier *buah* (lit. “fruit”) is for inanimate nouns, *ekor* (lit. “tail”) for non-human animate nouns, and *orang* (lit. “person”) for human nouns (Sneddon et al., 2010, p. 139; Alwi et al., 2014, p. 288). The type hierarchy in (64) shows the lexical type hierarchy for common nouns together with the PERNUM and ANIMCOUNT features, as well as the examples.

4.1.2 Pronouns

Pronouns are free forms that function alone to fill the position of a noun phrase in a clause (Payne, 2008, p. 43). A pronoun stands for a noun when it is clear who or what is being

spoken about (Sneddon et al., 2010, p. 164). Among the seven types of pronoun mentioned in Sneddon et al. (2010, p. 164), i.e. demonstrative pronouns, personal pronouns, indefinite pronouns, number pronouns, locative pronouns, reflexive pronouns, and interrogative pronouns, only four have been implemented in INDRA. The next subsections describe the demonstrative pronouns, personal pronouns, indefinite pronouns, and locative pronouns which have been analyzed and implemented. The semantic analyses and implementations of demonstrative, locative, and indefinite pronouns employ *decomposition*. Decomposing words is important to get more refined semantics.

Demonstrative pronouns

Indonesian has two demonstrative pronouns: *ini* “this” and *itu* “that” (Sneddon et al., 2010, p. 164). They occur as subject or object of a clause, as illustrated in the following examples.

- (65) a. *Ini besar.*
this big

“This is big.” (Sneddon et al., 2010, p. 164)

- b. *Budi makan itu.*
Budi eat that

“Budi eats that.” (own data)

I decompose the demonstratives pronouns *ini* “this” and *itu* “that” into the equivalent of *barang ini* “this thing” and *barang itu* “that thing”, respectively. DMRs (66a) and (66b) show the semantic representations of the examples above. The demonstrative pronoun *ini* “this” is decomposed into *entity_n_rel* and *proximal_q_rel* and *itu* “that” is decomposed into *entity_n_rel* and *distal_q_rel*. The semantic type hierarchy for demonstratives *proximal_q_rel* and *distal_q_rel* is shown in (68) and described in the next subsection on locative pronouns.

b.

Locative pronouns

As mentioned in the previous subsection, Indonesian has two demonstrative pronouns: *ini* “this” and *itu* “that”, similar to the ones in English. However, unlike English which only has two locative pronouns *here* and *there*, Indonesian has three locative pronouns similar to Japanese locative pronouns *ここ* *koko*, *そこ* *soko*, and *あそこ* *asoko*.

These Indonesian locative pronouns are *sini* “here (near speaker)”, *situ* “there (not far off)”, and *sana* “there (far off)” (Sneddon et al., 2010, pp. 133, 195). Similar to Japanese, Indonesian locative pronouns cannot function as adverbs, they must have a preposition, e.g. *di sini* “at here” and *ke situ* “to there”.

There are many cases that the locative pronouns, such as *sini* “here”, are decomposable or can be mapped to multiple elementary predicates (EPs). For example, *sini* “here” can be thought of as *tempat ini* “this place” (Seah & Bond, 2014b). The way to model this is by decomposing locative pronouns into a reference to a place and a demonstrative pronoun. Thus, I defined type hierarchies for the head which refers to a place and the demonstrative. Type hierarchies for heads and demonstratives are shown in (67) and (68), respectively.

(67)

The demonstrative *itu* “that” has the predicate `distal_q_rel`; the locative pronouns *situ* “here” and *sana* “there (far off)” has the predicate `medial_q_rel` and `remote_q_rel`, respectively. They are the daughters of the predicate `distal_q_rel`. The implementation of *situ* “there (not far off)” in TDL is shown in (69). The `LKEYS.KEYREL` is used for the heads and `LKEYS.ALTKEYREL` for the demonstratives.

(69)

```

situ := n+det-lex &
  [ STEM < "situ" >,
    SYNSEM.LKEYS [ KEYREL.PRED "place_n_rel",
                  ALTKEYREL.PRED "medial_q_rel" ],
    TRAITS native_token_list].
  
```

MRS (70a) and DMRS (70b) show the semantic representation of the decomposed word *situ* “there (not far off)” which is preceded by a preposition *di* “at”. The `ARG0` of the head daughter *di* “at” is equated with the `INDEX` of the whole PP which has the value *e2*. The value of the `ARG2` of the head daughter (*x4*) is coindexed with the `ARG0` of `place_n_rel` and `medial_q_rel`. The `medial_q_rel` introduces `RSTR` which is related to the top handle of the quantifier’s restriction (*h7*) and linked to the `LBL` of `place_n_rel` (*h7*=_q*h5*).

Indefinite pronouns

Indefinite pronouns indicate that a person or thing is being spoken about without a particular person or thing being identified (Sneddon et al., 2010, pp. 175-176). For example, *seseorang* “someone” and *sesuatu* “something”. I decompose *seseorang* “someone” into *orang_n_rel* and *indef_q_rel*; and *sesuatu* “something” into *benda+hal_n_rel* and *indef_q_rel*. The predicate *indef_q_rel* inherits from *quant_rel*, as shown in Type hierarchy (68). The predicate *orang_n_rel* and *benda+hal_n_rel* are subtypes of *entity_n_rel*, as shown in Type hierarchy (67).

Personal pronouns

Personal pronouns are pronouns which refer to people (Sneddon et al., 2010, pp. 164-166). Personal pronouns in Indonesian encode person (1st, 2nd, 3rd), number (singular and plural), and inclusive/exclusive for first person plural because first person plural has inclusive and exclusive forms. The type hierarchy for person, number, and inclusive/exclusive is shown in (71). The supertype is *pernum*, having seven subtypes: *1st*, *2nd*, *3rd* for person, *sg*, *pl* for number; and *incl* and *excl* for inclusive/exclusive. The types *1sg* for first person singular, *2pl* for second person plural, *1pl_incl* for first person plural inclusive etc. inherit from the subtypes of *pernum*.

(71)

In the semantic representation, pronouns get a quantifier *pron_q_rel*. This is implemented by adding *pronoun_q_rel* in *personal-pronoun-lex*. The lexical type hierarchy for pronouns in (72) shows the *basic-pronoun-lex* introduces the type *noun-relation*, i.e. *pronoun_n_rel* as the value of its PRED. Its subtype *pronoun-lex* introduces a new predicate *pronoun_q_rel*. It has two subtypes: *personal-pronoun-lex* whose HEAD is of type *pronoun* and *enclitic-pronoun-lex* whose HEAD is of type *enclitic*. The subtypes of the type *personal-pronoun-lex* are mentioned in Table 4.1 on page 76 together with their HEAD and PERNUM values, as well as the lexemes. The type *enclitic-pronoun-lex* is discussed in Section 4.2 on clitics.

(72)

According to Sneddon et al. (2010, pp. 165-166), pronouns *aku* “1SG”, *engkau* “2SG”, *kau* “2SG”, *kamu* “2SG”, and *kalian* “2PL” are intimate forms, used to children and between equals who have close relationship with each other. The pronoun *saya* “1SG” usually indicates that an intimate relationship does not exist. The pronoun *Anda* “2SG”, written in capital letter, conveys social information, used in impersonal situations, such as addressing strangers of the same age as or younger than the speaker, it is most frequently used in advertisements and public announcements. The pronouns *kita* “1PL.INCL”, *kami* “1PL.EXCL”, *ia* “3SG”, *dia* “3SG”, and *mereka* “3PL” are socially neutral. The pronoun *beliau* “3SG” carries social connotation, referring to people who are held in high respect. The analysis and implementation of Indonesian pronouns, in relation with social relationship, intimacy, politeness, and honorifics, are for future work.

4.1.3 Proper names

Proper names, also called *personal names* or *proper nouns* are nouns that are used to address and identify particular persons, things, places, organizations, or events which do not usually appear with articles, modifiers, possessors, relative clauses, or other devices that render nouns more identifiable (Sneddon et al., 2010, p. 132; Payne, 2008, p. 39). In Indonesian, they are usually written in capitals.

In the MRS, all proper names have the same PRED called `named_rel`. Similar to common nouns, they have ARG0 whose value is of the type *ref-ind*, represented by *x*. The value of the feature PERNUM is *sg* and the value of ANIMCOUNT is *animcount*, i.e. underspecified for animacy and countability. However, different from common nouns, a special feature CARG (CONSTANT ARGUMENT) is introduced in proper names. The value is the proper name itself, as a string. In addition, they do not usually appear with modifiers (including determiners). Proper names restrict their quantifying relations to be of type `proper_q_rel`. This follows the standard practice in DELPH-IN. The MRS representation of a proper name *Budi* is shown in AVM (73).

$$(73) \left[\begin{array}{l} mrs \\ \text{TOP} \quad [0] h \\ \text{INDEX} \quad [2] x \\ \text{RELS} \quad \left\langle \begin{array}{l} \text{named_rel} \\ \text{LBL} \quad [4] h \\ \text{CARG} \quad "budi" \\ \text{ARG0} \quad [2] \left[\begin{array}{l} x \\ \text{PNG.ANIMCOUNT} \quad animcount \\ \text{PNG.PERNUM} \quad sg \end{array} \right] \end{array} \right\rangle, \left[\begin{array}{l} \text{proper_q_rel} \\ \text{LBL} \quad [6] h \\ \text{ARG0} \quad [2] x \\ \text{RSTR} \quad [7] h \\ \text{BODY} \quad [8] h \end{array} \right] \right\rangle \\ \text{HCONS} \quad \left\langle \left[\begin{array}{l} qeq \\ \text{HARG} \quad [0] h \\ \text{LARG} \quad [1] h \end{array} \right], \left[\begin{array}{l} qeq \\ \text{HARG} \quad [7] h \\ \text{LARG} \quad [4] h \end{array} \right] \right\rangle \\ \text{ICONS} \quad \langle \rangle \end{array} \right]$$

4.1.4 Locative and temporal nouns

In addition to the three locative pronouns mentioned in Section 4.1.2, Sneddon et al. (2010, pp. 195-196) note that there is a set of locative nouns which indicate location in relation to the following noun. The locative nouns include *atas* “top, above”, *depan* “front”, *samping*

“side”, also *kanan* “right”, *utara* “north” etc. In INDRA, these locative nouns are of type *relational-n-lex* and inherit from *inanimate-noun-lex*.

Beside locative nouns, there is a set of temporal nouns which indicate time, such as *pagi* “morning”, *besok* “tomorrow”, *Senin* “Monday”, and *Januari* “January”. At the present stage, I give a specialized lexical type *n__temp-lex*, which is a subtype of *inanimate-noun-lex* for temporal nouns such as *pagi* “morning” and *besok* “tomorrow”, without any constraints. The days of the week and the names of months, such as *Senin* “Monday” and *Januari* “,” are regarded as proper names (Sneddon et al., 2010, p. 222). They belong to lexical types *n__day-lex* and *n__month-lex*, respectively, which are also subtypes of *proper-name-lex*. They have *3sg* as the value of *PERNUM* and *inanimate* as the value of *ANIMCOUNT* because they can take a classifier *buah* for inanimate nouns, as shown in Example (74).

- (74) *Senin di minggu ini adalah sebuah Senin yang baru buat saya.*
 Monday at week this COP one-CLF Monday REL new for 1sg

“Monday in this week is a new Monday for me.” (<https://godeliva.wordpress.com/2008/09/22/the-black-monday/>, accessed on 28 December 2017)

4.2 Clitics

Indonesian has bound forms of pronouns *ku-* “1sg”, *kau-* “2sg”, *-ku* “1sg”, *-mu* “2sg”, and *-nya* “3sg”. Some grammarians, such as Liaw (2004, pp. 231-232), Alwi et al. (2014, pp. 258-264), and Macdonald (1976, pp. 70-73) note that they are called *clitics*. Macdonald (1976, p. 70) notes that clitics can occur before or after a particular item of the structure. The former is called *proclitics* (i.e. *ku-* “1sg” and *kau-* “2sg”) and the later is called *enclitics* (i.e. *-ku* “1sg”, *-mu* “2sg”, and *-nya* “3sg”). Other grammarians, such as Sneddon et al. (2010, pp. 170-172), simply call these as (prefixed or suffixed) bound forms. In Indonesian, both proclitics and enclitics are usually written as one word with the following or preceding item, same as affixes (prefixes, suffixes, circumfixes, and infixes) which are attached to the stem and written as one word. In this dissertation, I use the term *clitic*, *proclitic*, and *enclitic* and I will discuss that “proclitics” (i.e. what other people analyze them as “proclitics”) *ku-* “1sg” and *kau-* “2sg” are not really clitics, they should be treated as prefixes instead.³ Based on this analysis, I treat and implement proclitics and enclitics differently: proclitics are regarded as inflectional affixes and enclitics are regarded as words.

³I thank Emily M. Bender who pointed this out at the Grammar Engineering mini-meeting in Seattle, 4 January 2017.

To the best of my knowledge, there are at least two differences between Indonesian proclitics and enclitics syntactically. Firstly, the enclitics are not selective to the hosts they are attached to, while the proclitics can attach only to passive type two verbs.⁴ The enclitics may occur as objects of active verbs, as shown in Example (75).

(75) *Narti menungguku/mu/nya.*

Narti ACT-wait=1SG/=2SG/=3SG

“Narti is waiting for me/you/him/her.” (Sneddon et al., 2010, p. 170)

They are used with noun phrases to denote possession and attached to the head noun, as shown in Example (76a), or attached to the modifier (adjective) if the head noun is modified, as shown in Example (76b).

(76) a. *rumahku/mu/nya*

house=1SG/=2SG/=3SG

“my/your/his/her house” (Sneddon et al., 2010, p. 171)

b. *rumah besarku/mu/nya*

house big=1SG/=2SG/=3SG

“my/your/his/her big house” (based on Sneddon et al., 2010, p. 171)

They also follow some prepositions, as shown in Example 77.

(77) a. *Ini untukku/mu/nya.*

this for=1SG/=2SG/=3SG

“This is for me/you/him/her.” (own data)

b. *Saya dijemput olehnya.*

1SG PASS-pick.up by=3SG

“I was met by him/her.” (Sneddon et al., 2010, p. 257)

The enclitic =*nya* “3SG” has another function. It can occur as agent in passive type one construction (see Section 5.2.2 on page 142), as shown in Example (77b) and Example (78).

(78) *Narti ditunggunya.*

Narti PASS-wait=3SG

“S/he is waiting for Narti.” (lit. “Narti is being waited for by him/her.”) (Sneddon et al., 2010, p. 171)

⁴In all constructions in which the proclitics and enclitics can occur, corresponding free forms are also possible (Sneddon et al., 2010, p. 171).

There is another *=nya* which functions as a determiner. Sneddon et al. (2010, pp. 155-156) note that *=nya* is attached to the head noun, translated as “the”. It is discussed in Section 4.3.

On the other hand, the proclitics are used only with passive verbs type two (see Section 5.2.2 on page 144) to denote the agent, as shown in Example (79).

(79) *Buku ini sudah ku/kaubaca.*

book this PRF 1SG=/2SG=read

“I/you have read this book.” (lit. “This book has been read by me/you.”) (Sneddon et al., 2010, p. 171)

The second difference is that in coordination constructions, the enclitics can occur only once if the elements in the coordination construction have the same objects, as shown in Example (80a), or the same possessor, as shown in Example (80b), or the same ARG2 of prepositions, as shown in Example (80c), or the same agent in passive type one construction, as shown in Example (80d).

(80) a. *Narti menunggu dan menjemputku/mu/nya.*

Narti ACT-wait and ACT-pick.up=1SG/=2SG/=3SG

“Narti waits for and picks me/you/him/her up.” (based on Example 75)

b. *rumah dan mobil besarku/mu/nya*

house and car big=1SG/=2SG/=3SG

“my/your/his/her big house and car” (based on Example 76a)

c. *Ini dari dan untukku/mu/nya.*

this from and for=1SG/=2SG/=3SG

“This is from and for me/you/him/her.” (based on Example 77)

d. *Narti ditunggu dan dijemputnya.*

Narti PASS-wait and PASS-pick.up=3SG

“Narti is waited for and picked up by him/her.” (based on Example 78)

On the other hand, the proclitics must occur in all elements in the coordination construction, as shown in Example (81a). Some people consider Example (81b) is unacceptable, while some think it is not totally bad. Example (81c) is unacceptable. I searched *kubeli dan baca* “1SG=buy and read” via Google and got only 8 results but I got 687 results when I searched *kubeli dan kubaca* “1SG=buy and 1SG=read” (as of 26 January 2018).

- (81) a. *Buku itu sudah ku/kaubeli dan ku/kaubaca.*
 book that PRF 1SG=/2SG=buy and 1SG=/2SG=read
 “I/you have bought and read that book.” (lit. “That book has been bought and read by me/you.”) (based on Example 79)
- b. ??*Buku itu sudah ku/kaubeli dan baca.*
- c. **Buku itu sudah beli dan ku/kaubaca.*

Zwicky & Pullum (1983) distinguish clitics from inflectional affixes and they propose six lines of evidence. Two of them are relevant to our discussions on Indonesian proclitics and enclitics in this section. Firstly, they state that the clitics can attach to words of any category (or part-of-speech), while the inflectional affixes are quite specific in their selections of stems. This point corresponds to Example (75)–(79) above. Secondly, no syntactic operations exist which treat a word combined with clitics as a unit, while words with inflectional affixes are treated as units. This point corresponds to Example (80) and (81) above. Thus, it is concluded that *ku*- “1SG” and *kau*- “2SG” are inflectional affixes, not clitics. Similar to the English contracted form *n’t*, *ku*- “1SG” and *kau*- “2SG” may have had their origin as simple clitics but have been reanalyzed as inflectional affixes. In addition, tokenization in the preprocessing stage also contributes to the decision that *ku*- “1SG” and *kau*- “2SG” should be treated as affixes. The following paragraphs describe the tokenization.

In INDRA the enclitics are tokenized and treated as separate words. At the present stage, the tokenization is done in a special file dealing with regular expression preprocessing, called *vanilla.rpp*. The lines of tokenization for words with clitic =*mu* “2GS” is shown in (82).

(82)

!(.+)mu	\1 -mu
!(berte il ja ka pene sa se ta terja) -mu	\1mu

The first line in (82) says that if there is a word ending in *mu*, the tokenizer will always separate *mu* from the preceding strings and change *mu* to *-mu* (preceded by a hyphen). The first line will separate every words ending in *mu*, even if they are in the lexicon file, such as *bertemu* “meet”, *ilmu* “science”, *kamu* “2SG”, and *penemu* “inventor”. In order to join these “words”, e.g. *berte* and *-mu*, after they are separated, we need a second line which says that if there are two “words” separated by a space, the first one belongs to a group of strings in the brackets such as *berte*, *il*, and *pene*; and the second one is *-mu*, join the first one with the second one. Thus, the system will recognize *bertemu* as one word. The drawbacks of this tokenization are: firstly, we need to list all words in Indonesian ending

in *mu* so that they can be regarded as one word again after they are separated; secondly, there are words with enclitics that are ambiguous, e.g. *keramu* can mean either “keramu tree” or a combination of *ker* “monkey” and *=mu* “2sg” meaning “your monkey” and *penanya* can mean either “questioner” (from a prefix *peN-* denoting an actor and a stem *tanya* “ask a question”) or a combination of *pena* “pen” and *=nya* “3sg” meaning “his/her pen”. The system should be able to recognize all possibilities and give all interpretations. The problem of building a good tokenizer for Indonesian is left for future work.

I made a lexical type hierarchy for enclitics, as shown in (83). The lexical type *enclitic-pronoun-lex* is a subtype of *pronoun-lex*, shown in (72). The *HEAD* value of *enclitic-pronoun-lex* is of type *enclitic*. The type hierarchy for *enclitic* is shown in (58), as a part of a bigger type hierarchy for *comp-noun*. The type *enclitic* is also a subtype of *possessor*, as shown in (56). The type *enclitic-3person* is special because it is also a subtype of *pass1agent*. By defining these type hierarchies, we can make sure that the enclitics can occur as objects/complements (of active verbs and prepositions, as well as of passive type one verbs for *=nya* “3sg”) and possessors.

Noun phrase constructions with enclitics denoting possessors have special analysis and implementation because they introduce a relation for possession *poss_rel* and the entire noun phrase has a definite quantifying relation *def_q_rel*. I made a phrase rule *n-pron-compound*, similar to a noun-noun compound rule (see Section 7.3 on page 204), the *HEAD* of the head daughter is of type *nounpos* and the *HEAD* of the non-head daughter is of type *possessor*. The non-head daughter’s *SPR* is empty but the head daughter’s *SPR* is identified with the *SPR* of the noun phrase mother. The mother’s *HEAD* is of type *nounnyapos*. This rule adds a relation *poss_rel* which has two arguments: its *ARG1* is identified with the *INDEX* of the head daughter and its *ARG2* is identified with the *INDEX* of the non-head daughter or the possessor.

The parse tree of Example (76a) *rumahku* “my house” is shown in (84a). The tree on the right side shows the rule *n-pron-compound* combines *rumah* “house” with the enclitic =*ku* “1sg”. The unary rule *poss-np* is similar to the rule for bare noun phrases, but instead of introducing *exist_q_rel*, it adds a different quantifying relation *def_q_rel*. This rule takes a head daughter of type *nounnyapos* having an optional determiner. The DMRS representation (84b) shows the predicate *_rumah_n_rel* has a quantifying relation *def_q_rel*. The predicate *poss_rel*’s ARG1 is identified with the ARG0 of *_rumah_n_rel* and its ARG2 is identified with the ARG0 of *_pronoun_n_rel* which has a quantifying relation *pronoun_q_rel* and its PERNUM is *1sg*. The semantic representation shown in (84b) is exactly the same as the one of *rumah aku* “my house” with the possessor in full, free form, not in bound, enclitic form. Thus, INDRA can generate both *rumahku* “my house” and *rumah aku* “my house” from *rumahku* “my house”.

The head daughter has an optional slot in its SPR which can be filled in by a determiner of type *det-non-nya* (see Section 4.3 on determiners). Example (85) illustrates a noun phrase with an enclitic denoting a possessor and a determiner. The parse tree of that example is shown in (86a) and its DMRS in (86b). Instead of the unary rule *poss-np*, the rule *head-spec* combines the noun phrase *rumahku* “my house” with determiner *ini* “this”. The quantifying relation of the entire phrase is introduced by the determiner. Thus, instead of *def_q_rel*, we have *proximal_q_rel*.

- (85) *rumahku ini*
house=1sg this
“this house of mine” (based on Example 76a)

On the other hand, the proclitics are not tokenized and treated as inflectional affixes in `i rules.tdl`, together with active prefix *meN-* and passive prefix *di-* (see also Section 5.2 on inflectional rules).⁵ The type description in (87) shows the implementation of *ku-* as an inflectional prefix which undergoes a rule *pas-two-prefix-ku-lex-rule* having an obligatory complement. This rule takes a daughter of type *tr-verb-lex* and builds a verb phrase mother of type *passive-ku*. The mother then undergoes a unary rule which makes the COMPS saturated and adds two relations *pronoun_n_rel* and *pronoun_q_rel*, of which the *PERNUM* is *1sg*.

(87)

```

pas-two-prefix-ku :=
%prefix (* ku)
pas-two-prefix-ku-lex-rule &
[ SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.OPT - ].

```

The parse tree of Example (79) is shown in (88) and its DMRS is shown in (89).

⁵In addition to the analysis of proclitics as affixes, it is more efficient not to tokenize *ku-* and *kau-* because in the fifth edition of KBBI dictionary there are 590 words starting with *ku-* and 27 words starting with *kau-*.

4.3 Determiners

In Indonesian, determiners like *ini* “this” and *itu* “that” follow the head noun, as shown in Example (90). They do not have plural forms like in English “these” and “those”, as shown in Example (90b).

- (90) a. *buku ini*
 book this
 “this book” (Sneddon et al., 2010, p. 133)
- b. *negeri-negeri itu*
 country-REDUP that
 “those countries” (Sneddon et al., 2010, p. 133)

They follow not only the head noun, but also all other constituents of the phrase, as shown in Example (53). Thus, they mark the end of a noun phrase.

In addition to *ini* “this” and *itu* “that”, the enclitic =*nya* “DEF” may carry the meaning of “the”. It generally refers to something understood from the context of the utterance or discussion (Mintz, 2002, pp. 108-109; Sneddon et al., 2010, pp. 155-156), as shown in Example (91).

- (91) *Kalau mau makan, nasinya di lemari.*
 if want eat rice=DEF at cupboard
 “If you want to eat, the rice is in the pantry.” (Sneddon et al., 2010, p. 155)

In determiners, I included indefinite numbers, such as *semua* “all” which precede the noun, and also the plural marker *para* (Sneddon et al., 2010, p. 138), as shown in Example (92).

- (92) *Semua orang bekerja di kota.*
 all person work at town
 “All people work in town.” (based on Sneddon et al., 2010, p. 138)

In INDRA, the common noun looks for a specifier, which can be filled in by a determiner. The determiner contributes a quantifying relation to the MRS of the noun phrase. The type hierarchy for this quantifying relation (*quant_rel*) is shown in (61). In DELPH-IN grammars, the HEAD value of the determiner is of type *det*. In INDRA, the type *det* is the supertype and it has two subtypes: *det-nya* for the determiner =*nya* “DEF” and *det-non-nya* for others, which has two subtypes: *det-nonquant* for *ini* “this” and *itu* “that”, and *det-quant* for indefinite numbers such as *semua* “all”, as shown in (93).

The type *determiner-lex* inherits from *basic-determiner-lex*. The type *basic-determiner-lex* contributes a quantifier relation to the semantics whose ARG0 is identified with the INDEX in the HOOK of the first item on its SPEC list, i.e. with the variable of the noun it quantifies. The subtypes of *det* become the HEAD of the lexical subtypes of *determiner-lex*, as shown in (94). The lexical type *det-det-lex* inherits from *determiner-lex* and *head-initial* and its head is of type *det-nonquant*. Determiners *ini* “this” and *itu* “that” belong to this lexical type. The lexical type *det-nya-lex* also inherits from *determiner-lex* and *head-initial*, but its head is of type *det-nya*. Enclitic =*nya* “DEF” belongs to this lexical type. The lexical type *det-pl-quant-lex* inherits from *determiner-lex* and *head-final* and its head is of type *det-quant*. It contributes the semantics of the noun head daughter (the PERNUM value is *pl*). Indefinite numbers such as *semua* “all” belong to this lexical type. The lexical type *det-sg-quant-lex* also inherits from *determiner-lex* and *head-final*, and its head is also of type *det-quant*. It contributes the semantics of the noun head daughter (the PERNUM value is *sg*). Indefinite numbers such as *satu-satunya* “the only one” belong to this lexical type.

Sneddon et al. (2010, p. 161) note that syntactic ambiguity may occur if there is more than one noun preceding a demonstrative. In other words, the demonstrative modifies a compound noun (see Section 7.3 on page 204 on compound nouns), as shown in Example (95).

(95) *rumah nelayan ini*
house fisherman this

“this fisherman’s house” or “the house of this fisherman” (Sneddon et al., 2010, p. 161)

Example (95) shows that *nelayan* “fisherman” is either a modifying noun describing what sort of house is being referred to, in which case *ini* “this” refers to the head word *rumah* “house”, as shown in (96), or it is a possessor, specifying the owner of the house, in which case *ini* “this” refers to *nelayan* “fisherman”, as shown in (97).

(96) a.

b.

(97) a.

b.

4.4 Numerals and classifiers

Numerals are words that are used to denote number. They can be divided into two groups: cardinal numbers which denote quantity, such as *satu* “one”, and ordinal numbers, such as *pertama* “first”, which denote relative position in a sequence (Liaw, 2004, p. 39). At the present stage, ordinal numbers have not been analyzed and implemented in INDRA. This section will discuss cardinal numbers only. Cardinal numbers consist of cardinal units, from one to nine, and group numbers such as *puluh* “tens”, *belas* “teens”, *ratus* “hundreds”, and *ribu* “thousands” (Sneddon et al., 2010, pp. 189-190). Group numbers combine with one of the cardinal units, as shown in Table 4.3. The cardinal unit *satu* “one” becomes *se-* “one” when it is combined with group numbers.

Table 4.3: Some combinations of cardinal units and group numbers

1	<i>satu</i>	10	<i>sepuluh</i>	100	<i>seratus</i>	1000	<i>seribu</i>
2	<i>dua</i>	20	<i>dua puluh</i>	200	<i>dua ratus</i>	2000	<i>dua ribu</i>
3	<i>tiga</i>	30	<i>tiga puluh</i>	300	<i>tiga ratus</i>	3000	<i>tiga ribu</i>
4	<i>empat</i>	40	<i>empat puluh</i>	400	<i>empat ratus</i>	4000	<i>empat ribu</i>
5	<i>lima</i>	50	<i>lima puluh</i>	500	<i>lima ratus</i>	5000	<i>lima ribu</i>
6	<i>enam</i>	60	<i>enam puluh</i>	600	<i>enam ratus</i>	6000	<i>enam ribu</i>
7	<i>tujuh</i>	70	<i>tujuh puluh</i>	700	<i>tujuh ratus</i>	7000	<i>tujuh ribu</i>
8	<i>delapan</i>	80	<i>delapan puluh</i>	800	<i>delapan ratus</i>	8000	<i>delapan ribu</i>
9	<i>sembilan</i>	90	<i>sembilan puluh</i>	900	<i>sembilan ratus</i>	9000	<i>sembilan ribu</i>

Smaller numbers follow larger numbers, as shown in Example (98).

(98) *dua ribu enam ratus tiga puluh sembilan*

two thousands six hundreds three tens nine

“two thousand six hundred and thirty-nine” (Sneddon et al., 2010, p. 190)

A cardinal number precedes a count noun to indicate the number of things being referred to (Sneddon et al., 2010, p. 136), as shown in Example (99a). It cannot follow the head noun, as shown in Example (99b). It cannot precede a reduplicated noun, as shown in Example (99c) (see Section 4.6.2 on reduplication of nouns).

(99) a. *sepuluh kota*

ten town

“ten towns” (Sneddon et al., 2010, p. 136)

b. **kota sepuluh*

town ten

(intended: “ten towns”) (based on Sneddon et al., 2010, p. 136)

c. **sepuluh kota-kota*

ten town-REDUP

(intended: “ten towns”) (based on Sneddon et al., 2010, p. 136)

Syntactically and semantically, Indonesian cardinal numbers can be analyzed based on their elements, i.e. cardinal units, group numbers, and some arithmetic operations “times” and “plus”. For example, *sepuluh* “ten” can be analyzed as *se-* “one” and *puluh* “tens”, having three predicates: one for “one”, one for “tens”, and one for an arithmetic operation “times”. Thus, the semantics of “ten” is similar to “one times tens”. Similarly, the semantics of *dua ribu enam ratus tiga puluh sembilan* “two thousand six hundred and thirty-nine” is similar to “two times thousands plus six times hundreds plus three times tens plus nine”.

However, at the present stage, cardinal numbers are treated as separate lexemes in IN-DRA’s lexicon. I wrote a script that generated all numbers from 1 to 9999 and put them into the lexicon. I made two lexical types: *a_-_num-sg_le* for a singular number *satu* “one” and *a_-_num-pl_le* for plural numbers from *dua* “two” to *sembilan ribu sembilan ratus sembilan puluh sembilan* “nine thousand nine hundred and ninety-nine”. Type descriptions for singular number 1 and plural number 9999 in the lexicon are shown in (100).

(100)

```
satu := a_-_num-sg_le &
[ STEM < "satu" >,
  SYNSEM.LKEYS.KEYREL.CARG "1",
  TRAITS native_token_list ].

sembilan_ribu_sembilan_ratus_sembilan_puluh_sembilan :=
a_-_num-pl_le &
[ STEM < "sembilan", "ribu", "sembilan", "ratus",
  "sembilan", "puluh", "sembilan" >,
  SYNSEM.LKEYS.KEYREL.CARG "9999",
  TRAITS native_token_list ].
```

The types *a_-_num-sg_le* and *a_-_num-pl_le* are both inherit from *a_-_num_le*, whose HEAD is of type *adj-number*. Numbers in INDRA are treated as adjectives modifying nouns, following the English grammar ERG and the Chinese grammar Zhong. The nouns that can be modified are constrained to those having the HEAD value of type *commonnoun* and the ANIMCOUNT is of type *countable*. Semantically, the lexical item for numbers *a_-_num_le* introduces a relation called *card-relation* which inherits from *named-relation* (used in proper names), having a feature PRED whose value is *card_rel* and a feature CARG

(101) a.

(102) a.

“a teacher” (Sneddon et al., 2010, p. 139)

- b. *dua ekor ular/ikan/kucing*
two CLF snake/fish/cat
“two snakes/fish/cats” (Sneddon et al., 2010, p. 139)
- c. *tiga buah meja/pena/rumah*
three CLF table/pen/house
“three tables/pens/houses” (Sneddon et al., 2010, p. 139)
- d. *sebuah rencana/perdebatan*
one-CLF plan/debate
“a plan/debate” (Sneddon et al., 2010, p. 141)

The number and classifier can immediately follow the head noun. The classifier is required when the number is placed after the noun, as shown in Example (103).

- (103) a. *ikan tiga ekor*
fish three CLF
“three fish” (Mintz, 2002, p. 322)
- b. **ikan tiga*
fish three
(Intended: “three fish”) (based on Mintz, 2002, p. 322) (see also Example 99b)

The type description for a classifier *ekor* is shown in (104). It inherits from a type *cl-nonhum-le* and it has a PRED value *_ekor_x_rel*.

(104)

```
ekor_cl := cl-nonhum-le &
[ STEM < "ekor" >,
  SYNSEM.LKEYS.KEYREL.PRED "_ekor_x_rel",
  TRAITS native_token_list ].
```

Type hierarchy (105)⁶ shows that the type *cl-nonhum-le*, together with *cl-hum-le* and *cl-inanim-le* are subtypes of *classifier-lex* which is a subtype of *basic-classifier-lex*. The HEAD is of type *classifier* which inherits from *adj* (adjective) and *det* (determiner). It modifies nouns whose HEAD is of type *commonnoun*. It takes an SPR whose HEAD is of type *adj-number*, i.e. the cardinal numbers. The type *classifier-lex* states that the SPR is obligatory. The subtypes *cl-inanim-le*, *cl-nonhum-le*, and *cl-hum-le* modify nouns whose *animcount* is of type *inanimate*, *non-human*, and *human*, respectively. The type hierarchy for *animcount* is shown in (63).

⁶In this type hierarchy, the feature ANIMCOUNT is abbreviated as ANIMC. to save some space.

(105)

Semantically, the LBL of a classifier is identified with the LBL of the modified noun (and also the LBL of the cardinal number). Its ARG1's value is identified with the ARG0 of the modified noun (and is also identified with the ARG1 of the cardinal number). The parse tree of Example (103), together with its variation (the number and classifier precedes the head noun), is shown in (106a). They have the same semantics representation, as shown in (106b).

Partitives, which indicate a particular amount of something, such as *botal* “bottle” and *genggam* “handful”, as well as measurement nouns, which refer to size, distance, volume, speed, weight, or temperature, such as *kilometer* “kilometre” and *derajat* “degree” (Sneddon et al., 2010, pp. 142-144), are grouped in one lexical type *n__measure-lex*, which

inherits from *inanimate-noun-lex*, without any constraints, in INDRA at the present stage. Adding constraints to this lexical type is left for future work.

4.5 Adjectives and prepositions

Adjectives can be used in a noun phrase to describe or specify some properties or characteristics of the head noun of the phrase, such as its shape or size, color, condition, or other characteristics (Payne, 2008, p. 63; Sneddon et al., 2010, pp. 180-181). The adjective can be preceded or followed by one of a number of modifying adverbs which show the amount or intensity of the quality indicated by the adjective (Sneddon et al., 2010, pp. 182-183). The adverbs which precede the adjective are, for example, *sangat* “very”, *terlalu* “too”, and *agak* “rather”. Those which follow the adjective are, for example, *sekali* “very” and *benar* “really”.⁷ Adjectives in Indonesian can be reduplicated (see Section 4.6.2 on page 104 on reduplication of adjectives). Two adjectives may form a compound or a kind of coordination without *dan* “and” occurring between them (see Section 7.4 on page 208 on adjective-adjective compound).

In addition to adjectives, Sneddon et al. (2010, pp. 154-155) and Alwi et al. (2014, pp. 254-255) note that a variety of prepositional phrases can occur in the noun phrase. Example (107) shows a head noun *petani* “farmer” which is modified by an adjective, a prepositional phrase, and a determiner. The modifying adjective is *kaya* “rich”, which is preceded by an adverb *sangat* “very”. The head of the prepositional phrase is *dari* “from” which takes a complement *Aceh*, a place name. The determiner is *itu* “that”.

- (107) *petani sangat kaya dari Aceh itu*
 farmer very rich from Aceh that
 “that very rich farmer from Aceh” (based on Sneddon et al., 2010, p. 154)

In HPSG, the syntactic feature `MOD` is used to allow a modifier to specify the kind of sign that it can modify. An adverb takes an adjective as its `MOD` value, as shown in (108). I made two lexical types for adverbs: *adverb-pre-adj-lex* which precedes an adjective (the `POSTHEAD` value is minus) and *adverb-post-adj-lex* which follows an adjective (the `POSTHEAD` value is plus).

- (108)
<i>adverb-adj-lex</i>	
[MOD < [HEAD <i>adj</i>] >]	
^	
<i>adverb-pre-adj-lex</i>
[POSTHEAD -]	<i>adverb-post-adj-lex</i>
[POSTHEAD +]

⁷Some verbs can be gradable, e.g. *sangat mencintai* “love very much” (lit. “very love”). That is, these adverbs are not a unique diagnostic test for adjectives in Indonesian.

An adjective takes a noun as its MOD value. The noun's HEAD value is of type *common-noun*. Adjectives in Indonesian follow the head noun and thus, they have a feature POST-HEAD whose value is plus. I made predicative adjectives as default in INDRA (see Section 6.1.2 on page 168 on copula clauses with adjective phrase predicates). Thus, in order to make INDRA be able to parse a noun phrase with modifying adjectives, I made a unary rule *adj-to-attr-rule* which takes a predicative adjective as its daughter and changes it to an attributive adjective.

A noun-modifying preposition takes a noun (its HEAD value is of type *subj-noun*) as its MOD value and an obligatory COMPS (its value is a noun of type *comp-noun*). Same as adjectives, noun-modifying prepositions in Indonesian also follow the head noun and thus, its POSTHEAD value is plus. In addition, predicative prepositions were made as default in INDRA (see Section 6.1.3 on page 168 on copula clauses with prepositional phrase predicates). A unary rule *adp-to-attr-n-rule* which takes a predicative preposition and changes it to an attributive preposition was made, too.

The parse tree of Example (107) is shown in (109). It shows both the adjective *kaya* “rich” and the preposition *dari* “from” undergo unary rules which change them from the default, predicative characteristic to attributive.

(109)

(111)

The MRS of Example (107) is shown in (110). The LBL of *_petani_n_rel* is identified with the LBLs of the modifying adjective *_kaya_a_rel* and the modifying preposition *_dari_p_rel*. Its ARG0 is identified with the ARG1s of both the modifying adjective *_kaya_a_rel* and the modifying preposition *_dari_p_rel*. The preposition's ARG2 is identified with the ARG0 of *named_rel*. The adjective-modifying adverb *_sangat_a_rel*'s ARG1 is identified with the ARG0 of the modified adjective *_kaya_a_rel*.

4.6 Reduplication

Indonesian has three kinds of reduplication: full reduplication, partial reduplication, and imitative reduplication. Partial reduplication and imitative reduplication are not produc-

tive. Sneddon et al. (2010, pp. 25-26) give some examples as follows:

- Partial reduplication
 - *tangga* “ladder” > +PART.REDUP > *tetangga* “neighbour”
 - *laki* “husband” > +PART.REDUP > *laki* “man”
- Imitative reduplication
 - *sayur* “vegetable” > +IMIT.REDUP > *sayur-mayur* “vegetables” (*mayur* cannot occur alone)
 - *ramah* “friendly” > +IMIT.REDUP > *ramah-tamah* “hospitable and friendly” (*tamah* cannot occur alone)
 - *warna* “color” > +IMIT.REDUP > *warna-warni* “all kinds of colors” (*warni* cannot occur alone)

I regard them as single words, do not analyze them based on their components and thus list all of them in `lexicon.tdl`.

Full reduplication is productive and this section mainly deals with it. In full reduplication, the entire word is repeated and the resulting two words or bases are separated by a hyphen in writing, with the second word or base is treated as the reduplicated part (Sneddon et al., 2010), for example *buku-buku* “books”. Macdonald (1976, p. 32) also notes that in writing, full reduplication is frequently indicated by placing the number two (2) after the form or base to be reduplicated, either on the line or raised above the line, for example *buku2* or *buku²*.

In the following subsections I describe two kinds of full reduplication in Indonesian: reduplication without corresponding single bases (Section 4.6.1), and reduplication of nouns and adjectives (Section 4.6.2).⁸

4.6.1 Reduplication without corresponding single bases

Reduplicated forms in Indonesian sometimes do not have unreduplicated counterparts and are often treated as single bases in dictionaries, such as in KBBI. Same as other types of full reduplication, they are usually written with a hyphen (Sneddon et al., 2010, p. 19). Such forms include *laba-laba* “spider”, *oleh-oleh* “gift”, *tiba-tiba* “suddenly”, *sia-sia* “futile”, *pura-pura* “pretend”, *megap-megap* “pant”, and *masing-masing* “each”.

There are other similar forms such as *gula-gula* “sweets”, *mata-mata* “spy”, and *langit-langit* “ceiling” which are inconsistently listed under the single base or as separate base.

⁸I developed parts of this analysis in the DELPH-IN Linguistic Analysis Design (LAD) meeting. The discussions are documented on the LAD Indonesian Morphology page <http://moin.delph-in.net/LADIndonesianMorphology#Redup>.

The meaning of the reduplicated form is different but nevertheless related to the meaning of the single base (Sneddon et al., 2010, p. 20). For example, *gula-gula* “sweets” is made from *gula* “sugar”, *langit-langit* “ceiling” is located up above, similar to *langit* “sky”, and *mata-mata* “spy” works as if s/he were someone’s *mata* “eye”. Same as partial reduplication and imitative reduplication mentioned above, full reduplication without corresponding single bases are also regarded as single words. Example (112) shows how *mata-mata* “spy” is written in `lexicon.tdl`.

```
(112) mata-mata := inanimate-noun-noun-lex &
      [ STEM < "mata²" >,
        SYNSEM.LKEYS.KEYREL.PRED "_mata-mata_n_rel",
        TRAITS native_token_list ].
```

However, full reduplication can also indicate plurality (Sneddon et al., 2010, p. 20) and thus *mata-mata* is ambiguous, it can mean ‘spy’ or ‘eyes’, which will be explained in the next subsection.

4.6.2 Reduplication of nouns and adjectives

Bare nouns in Indonesian are underspecified for number, they may be singular or plural. The major function of noun reduplication is to make them specific (plural) for number (Sneddon et al. (2010, p. 20), Macdonald (1976, pp. 33-34)). For example,

- *batu* “stone(s)” > +REDUP > *batu-batu* “stones”
- *es krim* “ice cream(s)” > +REDUP > *es krim-es krim* “ice creams”
- *mata* “eye(s)” > +REDUP > *mata-mata* “eyes”

Reduplicated forms without corresponding single bases, as described in Section 4.6.1, although they are treated as single bases, cannot be further reduplicated. The reduplicated form *mata-mata* “spy, spies” cannot be further reduplicated, thus **mata-mata-mata-mata* (intended meaning: “spies”) is ungrammatical.

Plurality of a noun can also be indicated by the reduplicated adjective modifying or describing it (Sneddon et al., 2010, pp. 20, 22; Macdonald, 1976, pp. 34-35). For example,

- (113) a. *Budi sedang melempar batu kecil.*
 Budi PROG meN-throw stone small
 “Budi is throwing a small stone/small stones.”
- b. *Budi sedang melempar batu kecil-kecil.*
 Budi PROG meN-throw stone small-REDUP
 “Budi is throwing small stones.”

- c. *Budi sedang melempar batu-batu kecil.*

Budi PROG meN-throw stone-REDUP small

“Budi is throwing small stones.”

- d. *?Budi sedang melempar batu-batu kecil-kecil.*

Intended meaning: “Budi is throwing small stones.”

Example (113a) illustrates that the non-reduplicated form *batu kecil* is underspecified for number. Example (113b) and (113c) show that both the reduplicated adjective *batu kecil-kecil* and the reduplicated noun *batu-batu kecil* denote plurality. However, the reduplicated form *batu-batu kecil-kecil*, as shown in Example (113d), is considered non-standard. Even so, I decided to make INDRA able to parse a reduplicated noun followed by a reduplicated adjective.

Reduplicated adjective can fill in the predicate position in a copula clause. It entails that the subject noun is plural, as shown in Example (114).

- (114) *Gambarmu bagus-bagus.*

drawing=2SG good-REDUP

“Your drawings are beautiful.” (Sneddon et al., 2010, p. 22)

In INDRA, regular expressions are used in handling reduplication of nouns and adjectives. I made a rule for full reduplication in `repp/vanilla.rpp` (`repp` stands for ‘regular expression preprocessing’). This rule states that any strings reduplicated with a hyphen in between is changed to the string itself with an iteration mark (superscript two ²). For example, if the input is *batu-batu*, it will be changed to *batu²*.

- (115) `!(.+)-\1` `\12`

The reduplication rules for noun and adjective are written in `lrules.tdl`. The rules state that for any strings with the iteration mark (²), apply the lexical rule *redup-noun-lex-rule* or *redup-adj-lex-rule*.

- (116) `redup-noun-suffix :=`

`%suffix (*2)`

`redup-noun-lex-rule.`

`redup-adj-suffix :=`

`%suffix (*2)`

`redup-adj-lex-rule.`

The *redup-noun-lex-rule* makes the number plural (PNG.PERNUM *pl*). The *n-enclitic-lex-rule-super* has *common-noun-lex* as its DTR. This rule pumps up the daughter’s syntactic HEAD, VAL, and POSTHEAD. The HEAD has a feature CARDIN with the value plus (+),

so that the reduplicated noun cannot co-occur with cardinal numbers and INDRA cannot parse **dua apel-apel* (intended: “two apples”). This rule adds the following semantic information: the attribute `PERNUM` has a value *pl* (plural).

$$(117) \left[\begin{array}{l} \text{redup-noun-lex-rule} \\ \\ \text{SYNSEM} \mid \text{LOCAL} \left[\begin{array}{l} \text{CAT} \left[\begin{array}{l} \text{HEAD} \quad \boxed{1} \left[\text{CARDIN} + \right] \\ \text{VAL} \quad \boxed{2} \\ \text{POSTHEAD} \quad \boxed{3} \end{array} \right] \\ \text{CONT} \mid \text{HOOK} \mid \text{INDEX} \mid \text{PNG} \mid \text{PERNUM } pl \end{array} \right] \\ \\ \text{DTR} \left[\begin{array}{l} \text{countable-noun-lex} \\ \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{CAT} \left[\begin{array}{l} \text{HEAD} \quad \boxed{1} \\ \text{VAL} \quad \boxed{2} \\ \text{POSTHEAD} \quad \boxed{3} \end{array} \right] \end{array} \right] \\ \\ \text{C-CONT} \left[\begin{array}{l} \text{HCONS} \quad \langle ! \quad ! \rangle \\ \text{ICONS} \quad \langle ! \quad ! \rangle \\ \text{RELS} \quad \langle ! \quad ! \rangle \end{array} \right] \end{array} \right]$$

The *redup-adj-lex-rule* takes *adjective-lex* as its daughter and states that the reduplicated adjective contributes to the semantics of the noun it modifies, i.e. its `PERNUM` has a value *pl*.

$$(118) \left[\begin{array}{l} \text{redup-adj-lex-rule} \\ \\ \text{VAL} \mid \text{SUBJ} \left\langle \left[\text{HOOK} \mid \text{INDEX} \mid \text{PNG} \mid \text{PERNUM } pl \right] \right\rangle \\ \\ \text{DTR } \text{adjective-lex} \end{array} \right]$$

Using this rule, INDRA can parse Example (114), as shown in (119) with a correct MRS, i.e. the noun’s `PERNUM` is *pl*, as shown in (120).

$$(119) \begin{array}{cc} & \text{S} \\ & \swarrow \quad \searrow \\ \text{NP} & \text{ADJ} \\ | & | \\ \text{N} & \text{ADJ} \\ \text{gambarmu} & \text{bagus}^2 \end{array}$$

$$(120) \left[\begin{array}{l} mrs \\ TOP \quad [0] h \\ INDEX \quad [2] e \\ \\ \left[\begin{array}{l} _gambar_n_rel \\ LBL \quad [4] h \\ \\ ARG0 \quad [3] \left[\begin{array}{l} x \\ PNG.ANIMACY \quad inanimate \\ PNG.PERNUM \quad 3pl \end{array} \right] \end{array} \right], \left[\begin{array}{l} def_q_rel \\ LBL \quad [5] h \\ \\ ARG0 \quad [3] x \\ RSTR \quad [6] h \\ BODY \quad [7] h \end{array} \right], \left[\begin{array}{l} poss_rel \\ LBL \quad [4] h \\ \\ ARG0 \quad [8] e \\ ARG1 \quad [3] x \\ ARG2 \quad [9] x \end{array} \right] \\ \\ RELS \quad \left\langle \left[\begin{array}{l} pronoun_q_rel \\ LBL \quad [10] h \\ \\ ARG0 \quad [9] \left[\begin{array}{l} x \\ PNG.ANIMACY \quad animacy \\ PNG.PERNUM \quad 2sg \end{array} \right] \end{array} \right], \left[\begin{array}{l} pronoun_n_rel \\ LBL \quad [13] h \\ \\ ARG0 \quad [9] x \end{array} \right], \left[\begin{array}{l} _bagus_a_rel \\ LBL \quad [1] h \\ \\ ARG0 \quad [2] e \\ ARG1 \quad [3] x \end{array} \right] \right\rangle \\ \\ HCONS \quad \left\langle \left[\begin{array}{l} qeq \\ HARG \quad [0] h \\ LARG \quad [1] h \end{array} \right], \left[\begin{array}{l} qeq \\ HARG \quad [6] h \\ LARG \quad [4] h \end{array} \right], \left[\begin{array}{l} qeq \\ HARG \quad [11] h \\ LARG \quad [13] h \end{array} \right] \right\rangle \end{array} \right]$$

INDRA can generate the following three sentences from the input sentence in Example (114): *Gambarmu bagus*² with a reduplicated adjective, same as the input sentence, *Gambar²mu bagus* with a reduplicated head noun, and *Gambar²mu bagus²* with both the head noun and the predicate adjective reduplicated. These three sentences have the same MRS.

When a noun is followed by a modifying noun, usually only the head noun is reduplicated, e.g. *buku-buku sejarah* “history books”, *toko-toko buku* “bookshops” (Sneddon et al., 2010, p. 21; Alwi et al., 2014, p. 246). However, as Sneddon et al. (2010, pp. 21-22) note, there is disagreement among Indonesians as to whether or not a noun and modifying noun should both be repeated, e.g. *surat-surat kabar* or *surat kabar-surat kabar* for “newspapers”. In addition, when a noun and a following noun both refer to more than one, it is often possible to repeat either noun (or both), e.g. *nama-nama universitas*, *nama universitas-universitas*, and *nama-nama universitas-universitas* for “names of universities”. INDRA currently treats these three forms differently, they have different MRSs corresponding to: “names of university”, “name of universities”, “names of universities” where the singular form is in fact underspecified for number.

4.7 Relative clause

A relative clause which functions as a nominal modifier has three elements: a noun phrase head, a clause modifying the head noun phrase, and a relativizer, i.e. a morpheme or particle that sets off (precedes) the clause (Payne, 2008, pp. 325-326). In Indonesian, the relativizer is *yang* “REL”. Sneddon et al. (2010, p. 294) mention that there are four types of relative clause in Indonesian: the defining relative clause (the most common type of relative clause), topic-comment relative clauses (including possessor topic-comment relative clause and object topic-comment relative clause), prepositional relative clause, and locative relative clause (the only relative clause which is not preceded by relativizer *yang*). This section will discuss the defining relative clause and the possessor topic-comment relative clause, which have been implemented in INDRA. Other types of relative clause are left for future work.

4.7.1 Defining relative clause

Sneddon et al. (2010, pp. 294-295) state that a defining relative clause is one whose subject corresponds to the head noun of the embedding phrase. Example 53 on page 71 at the beginning of this chapter shows a relative clause (marked by *yang*) following a head noun, a modifying adjective, and a possessor. Another example is shown in (121). In this example, the sentence (121a) can be embedded within the subject noun phrase of the sentence (121b), giving the sentence (121c).

- (121) a. *Orang itu duduk dekat jendela.*
 person that sit near window
 “That person is sitting near the window.” (Sneddon et al., 2010, p. 295)
- b. *Orang itu bekerja dengan saya.*
 person that work with 1SG
 “That person works with me.” (Sneddon et al., 2010, p. 295)
- c. *Orang yang duduk dekat jendela itu bekerja dengan saya.*
 person REL sit near window that work with 1SG
 “That person who is sitting near the window works with me.” (Sneddon et al., 2010, p. 295)

In this type of relative clause, only the subject can be relativized and be identical to the head of the embedding noun phrase. Thus, if the head noun has semantic role as agent, the verb must be in active voice; if the head noun stands as patient (or recipient or beneficiary in a ditransitive construction), the verb must be in passive voice (Sneddon et al., 2010, p. 295-296), as shown in Example (122).

(122) *Rumah yang dibangun Pak Dani tidak begitu besar.*

house REL PASS-build Mr Dani NEG so big

“The house Mr. Dani built isn’t very big.” (lit. “The house which was built by Mr. Dani is not so big.”) (Sneddon et al., 2010, p. 296)

Relative clauses can also derive from clauses with adjective phrase predicates and clauses with prepositional phrase predicates (see Chapter 6 on page 165 on copula clauses with adjective phrase predicates and copula clauses with prepositional phrase predicates), as shown in Example (123a) and Example (123b). Negation words and aspect or temporal markers should follow *yang*, as shown in Example (123). Noun relative clauses rarely occur. The relativizer *yang* which normally cannot precede a noun, can be used when the speaker wants to distinguish a person from other people with similar characteristics (Sneddon et al., 2010, p. 296), as shown in Example (123c).

(123) a. *mobil yang (tidak) biru*

car REL NEG blue

“the (not) blue car” (lit. “car which is (not) blue”) (based on Sneddon et al., 2010, p. 296)

b. *orang yang (sudah) di sini*

person REL PRF at here

“the person who was (has been) here” (based on Sneddon et al., 2010, p. 296)

c. *Soeharto yang (bukan) presiden*

Soeharto REL NEG president

“Soeharto who was (not) president” (based on Sneddon et al., 2010, p. 297)

Sneddon et al. (2010, pp. 150-151) note that *yang* cannot occur in a compound of a noun plus adjective sequence. The relativizer *yang* may be used to disambiguate a noun-adjective compound. For example, *kamar kecil* “toilet; small room” is ambiguous, the first meaning is idiomatic and the second meaning is constructive; however, when *yang* precedes the adjective, i.e. *kamar yang kecil* “small room”, only the second meaning is possible (see Section 7.1 on page 183 on idiomatic compounds).

A number of relative clauses can occur within the one noun phrase, as shown in Example (124).

(124) *mobil yang biru yang di sini*

car REL blue REL at here

“the blue car here” (lit. “car which is blue which is here”) (own data)

In INDRA, the lexical type *yang-super-lex*, which inherits from *lex-item*, has a HEAD feature whose value is of type *yang*. It takes a COMPS, which is a clause, having a feature

MC (MAIN CLAUSE) whose value is plus. It modifies a noun whose HEAD is of type *subj-noun*. It has a feature POSTHEAD whose value is plus, meaning that as a modifier, *yang* always occurs after the head noun. It is the supertype for both defining relative clause and possessor topic-comment relative clause.

For defining relative clause, I made a type *yang-comp-lex*, which inherits from *yang-super-lex*. Its HEAD is of type *yang*. It also inherits from *basic-one-arg* because it takes a COMPS. The COMPS' value has LOCAL and NON-LOCAL features. In the LOCAL feature, its HEAD is of type *+vjp* (*verb, adjective, and preposition*), with the value of its SUBJ is empty. In the NON-LOCAL.SLASH, it has a value of type *subj-noun*, its INDEX and LTOP are identified with the INDEX and LTOP of the MOD. The modified noun acts as filler of gap (or slash) in the relative clause.

Parse tree of Example (121c) is shown in (125). Its DMRS is shown in (126).

(125)

(126)

Parse tree of Example (124) is shown in (127). Its MRS is shown in (128). Its DMRS is shown in (129). The MRS representation of the noun phrase with *yang* “REL” is the same as the one without.

(129)

4.7.2 Possessor topic-comment relative clause

A copula clause with a noun phrase predicate in Example (130a) may derive a possessor topic-comment clause, as shown in Example (130b) (see Sneddon et al., 2010, pp. 287-290 for details of possessor topic-comment clauses), which in turn, may derive a possessor topic-comment relative clause, as shown in Example (130c).

- (130) a. *Nama sopir itu Ali.*
 name driver that Ali
 “That driver’s name is Ali.” (Sneddon et al., 2010, p. 287)
- b. *Sopir itu namanya Ali.*
 driver that name=DEF Ali
 “That driver’s name is Ali.” (lit. “Concerning that driver, his name is Ali.”)
 (Sneddon et al., 2010, p. 297)
- c. *sopir yang namanya Ali itu*
 driver REL name=DEF Ali that
 “that driver whose name is Ali” (based on Sneddon et al., 2010, p. 297)

Possessor topic-comment relative clause corresponds to relative clause with “whose” in English. The relativizer *yang* “REL” also occurs in this type of relative clause. The subject of the relative clause (*namanya* “name=DEF” in Example (130c)), having an enclitic =*nya* “DEF”, is the thing possessed by the head noun. The enclitic =*nya* “DEF” is identified with the head of the entire noun phrase (*sopir* “driver”). The head noun is modified by *yang* “REL”. It is the possessor of the subject noun in the relative clause (Sneddon et al., 2010, p. 297).

In INDRA, a special lexical type for *yang* “REL” in this type of relative clause, called *yang-nya-comp-lex*, was made. It inherits from *yang-super-lex*, as explained in the previous subsection, and *basic-two-arg*. Its HEAD is of type *yang*. It modifies a noun whose SPR value is of type *canonical-synsem*. It has a feature MC whose value is minus, meaning that the possessor topic-comment relative clause is not a main clause. Its SUBJ value is empty, i.e. it does not take a subject, but its COMPS value consists of two items: the first one has HEAD value of type *+vjp* (a supertype of *verb*, *adjective*, and *adposition*) and a SUBJ,

whose value is identified with the second item. The second item, which is identified with the value of the SUBJ of the first item, has *nounnyadef* as its HEAD value, i.e. the subject of the relative clause must appear with *=nya* “DEF”. Its SPR and COMPS values are empty. Both the first and the second item in the COMPS value have a feature OPT whose value is minus, i.e. neither the subject noun nor the predicate of the relative clause is optional. Semantically, the type *yang-nya-comp-lex* adds a relation of type *arg12-ev-relation*. Its PRED is *poss_rel*, same as the one for possessive relation mentioned in Section 4.2. Its LBL is identified with the LTOP of the subject noun, i.e. the possessee. Its ARG1 is identified with the INDEX of the possessee and its ARG2 is identified with the INDEX of the head noun or the possessor.

Parse tree of Example (130c) is shown in (131). Its MRS is shown in (133). Its DMRS is shown in (133). Its semantics is similar to the semantics of Example (130a) but the LTOP is identified with the LBL of the head noun’s quantifying relation *distal_q_rel*, instead of the LBL of the *cop_v_zero_rel*. The whole thing is an individual, not an event.

(131)

(133)

However, INDRA cannot parse a phrase having many comment clauses as shown in (134).

- (134) *sopir yang namanya Ali, rumahnya di Jakarta, dan asalnya dari Bandung itu Bandung that*
 driver REL name=DEF Ali house=DEF at Jakarta and origin=DEF from
 Bandung that

“that driver whose name is Ali, whose house is in Jakarta, and whose origin is from Bandung” (based on Example 130c)

The solution is by treating the COMPS of *yang* “REL” as a coordinated clause. Each constituent in the coordinated clause has a SUBJ whose HEAD value is of type *nounnyadef*, i.e.

the subject of each constituent in the coordinated relative clause must appear with =*nya* “DEF”. This is for future work.

The head noun of the entire relative clause construction can be ellipted. Sneddon et al. (2010, p. 309) mention this as nominalized relative clauses. The *yang* phrase functions like a noun, as shown in (135), where the noun *sopir* “driver” is omitted. This can be said in a situation where it is clear that the driver is being spoken about.

(135) *yang namanya Ali itu*

REL name=DEF Ali that

“That one whose name is Ali” (based on Example 130c)

At the present stage, INDRA also cannot parse this nominalized relative clause construction. The solution is to make a unary rule which fills the value of the MOD and make the phrase as a noun.

4.8 Summary

In this chapter, I gave my analysis and implementation of the syntax and semantics of Indonesian noun phrases. Firstly, I presented analyses of noun subcategorization: common nouns, proper names, pronouns, locative and temporal nouns. Afterwards, I presented analyses of the elements that can modify Indonesian noun phrases: clitics, determiners, numerals and classifiers, adjectives and prepositions, reduplication, and two types of relative clause (defining relative clause and possessor topic-comment relative clause). INDRA can parse noun phrases modified by those elements. However, there are phenomena that have not been analyzed and implemented yet: interrogative pronouns; reflexive pronouns; personal pronouns in relation with social relationship, intimacy, politeness, and honorifics; decomposition of cardinal numbers; partitives and measurement nouns; NP having many comment clauses; and nominalized relative clause construction. In addition, there are still a lot of work need to be done on adjective subcategorization (Musgrave (2001) has done some work on adjectives related to emotion) and phenomena related to adjectives such as equative, comparative, and superlative adjective phrases. These are for future work.⁹

Other sections in this dissertation which are related to the structure of nouns and adjectives are Section 7.1 on page 183 on idiomatic compound, Section 7.3 on page 204 on noun-noun compound, Section 7.4 on page 208 on adjective-adjective compound, and Section 6.3 on negation.

⁹One of the examiners gave me a comment that the nominalized relative clause construction is very important because it is frequently used in questions, clefts, and other focus constructions and thus it is the first thing to deal with when INDRA is developed in the future. I thank the examiner for this advice.

Chapter 5

Verbs

This chapter describes my analysis of Indonesian verb phrases. The class of verbs includes lexemes which express the least time-stable concepts, e.g. events such as *lari* “run”, *makan* “eat”, and *baca* “read” (Payne, 2008, p. 47). They denote action, process, or state which is not property or quality (Alwi et al., 2014, p. 91). Semantically, they refer to events. In MRS, their predicates have `ARG0` and the value is an event, represented by `e`. Syntactically, they serve as heads of verb phrases and predicates of clauses. In Indonesian, they are negated with a negation word *tidak* “NEG”.

This chapter is structured as follows. Section 5.1 outlines each category of verbs in Indonesian. The following section (Section 5.2) describes the implementation of inflectional rules for verbs in INDRA, which mainly consist of active and passive voice rules. Two main types of passive constructions are explained in Section 5.2.2. The following sections outline the elements or modifiers in an Indonesian verb phrase, namely auxiliaries (Section 5.3), negation words (Section 5.4), and adverbs (Section 5.5). Section 5.6 concludes.

5.1 Verb subcategorization

This section focuses on verb subcategorization, including the number of arguments and their syntactic properties. Sneddon et al. (2010, p. 65) note that every verb is either intransitive which has one argument or transitive which has two arguments. Liaw (2004, pp. 111-114) and Alwi et al. (2014, pp. 95-98) mention ‘optional transitive’ which has one obligatory subject argument and one optional object argument as in *Adi makan (nasi)* “Adi eats (rice)”. Besides the number of arguments, the possibility of passivization with inflectional rules plays an important role in distinguishing intransitives from transitives in Indonesian. Example (136) and (137) show intransitive and transitive sentences respectively.

(136) *Adi tidur.*

Adi sleep

“Adi sleeps.” (own data)

(137) a. *Adi mengejar Budi.*

Adi ACT-chase Budi

“Adi chases Budi.” (MRS Test-suite No. 41)

b. *Budi dikejar Adi.*

Budi PASS-chase Adi

“Budi is chased by Adi.” (based on MRS Test-suite No. 41)

c. *Budi saya kejar.*

Budi 1SG chase

“Budi is chased by me.” (own data, based on MRS Test-suite No. 41)

In Example (137a), the transitive verb *mengejar* “chase” is formed from an active prefix *meN*-¹ and the base *kejar* (the initial sound /k/ undergoes nasalization; see Section 5.2). The active prefix *meN*- is changed to a passive prefix *di*- in passive type one (Sneddon et al., 2010, pp. 256-257), as shown in Example (137b) and without affix in passive type two (Sneddon et al., 2010, pp. 257-258), as shown in Example (137c). Sneddon et al. (2010, pp. 256-257) state that in passive type one, the actor is third person or a noun, while in passive type two, the agent is a pronoun or pronoun substitute and it comes before the unprefix base (see Section 5.2.2 on passive voice).

In addition to transitive, intransitive, and optional transitive verbs, Sneddon et al. (2010, pp. 272-274) state that *ada* is an intransitive verb which differs significantly from other verbs and deserves special comment. The more detailed subcategorization into other groups such as ditransitives was done semi-automatically (see Section 5.1.1) as well as manually (manual extension), following the type hierarchy for *verb*, as shown in (138).

¹It is not the case that all verbs having a prefix *meN*- are transitive verbs. The intransitive verb *meng-gonggong* “bark” has a prefix *meN*- and it cannot be passivized (see Section 5.1.2 on intransitive verbs).

The head type *verb* in INDRA is subtyped according to: first, whether a verb is an auxiliary or not (see Section 5.3 on auxiliaries which are regarded as verbs); second, whether it can occupy the position of verb complements in control and raising constructions; and third, whether it can participate in verb-verb compounding (see Section 7.2 on page 185 on verb-verb compound, including control and raising constructions).

The type *verb* has two subtypes: *aux* for auxiliaries and *non-aux* for non-auxiliaries. The type *non-aux* has four subtypes: *existential* for *ada* “EXIST” (see Section 5.1.4), *pass2* for passive type two verbs, *copula* for copula verbs (which is further subtyped into *copula-stative* and *copula-dynamic*, see Chapter 6 on page 165 on copula constructions), and *control-raising-comp-verb* for verb complements in control and raising constructions. The type *pass2* is further subtyped into *pass-two*, *pass-ku*, and *pass-kau*, as shown in (139) (see Section 5.2.2 on passive type two).

The type *control-raising-verb* has three subtypes: *copula-dynamic* (which inherits also from *copula*), *compound-verb*, and *locative* (see Section 6.1.3 on page 168 on copula clauses with prepositional phrase predicates). The type *compound-verb* is further subtyped into *intransitive*, *transitive*, and *pass1*.

Verbal lexical type hierarchy is shown in (141). The type *verb-lex* inherits from *non-mod-lex-item* (*non-mod-lex-item* is a *lex-item* whose HEAD cannot modify anything), whose HEAD is of type *verb*. It has two subtypes: *indonesian-verb-lex* and *aux-lex*. The type *indonesian-verb-lex* has the following constraints: its HEAD is of type *non-aux* and the value of feature AUX is minus, its SUBJ has empty value for the SPR and COMPS with the INDEX identified with the XARG. The type *aux-lex* has the following constraints: its HEAD is of type *aux* and the value of AUX is plus. The type *main-verb-lex* inherits from *indonesian-verb-lex* and *basic-verb-lex* (from matrix file) with an additional information: its SUBJ is

of type *subj-noun* (see Section 4.1 on page 72 on noun subcategorization). The type *subj-raise-aux* inherits from *aux-lex* and *trans-first-arg-raising-item* (from *matrix* file) with the following constraints: its *SUBJ* is of type *subj-noun* and its *COMPS* is of type *+vjp* (a supertype of *verb*, *adjective*, and *adposition*). Section 5.3 details the type *subj-raise-aux*.

(141)

Most verbal lexical types inherit from *main-verb-lex*. Type hierarchy (142) shows all lexical types which inherit from *main-verb-lex*. The type *clausal-comp-verb-lex*, which inherits from *clausal-second-arg-trans-lex-item* and *main-verb-lex*, is described in Section 5.1.3. The type *intransitive-verb-lex*, which inherits from *intransitive-lex-item* and *main-verb-lex*, is described in Section 5.1.2. The type *cop-verb-lex*, which inherits from *transitive-lex-item* and *main-verb-lex*, is described in Chapter 6 on page 165 on copula constructions. The type *transitive-verb-lex*, which also inherits from *transitive-lex-item* and *main-verb-lex*, is described in Section 5.1.5. The type *ditransitive-verb-lex*, which also inherits from *ditransitive-lex-item* and *main-verb-lex*, is described in Section 5.1.5.

(142)

Table 5.1 on the next page shows verb types in INDRA, with information on the *HEAD* value of the verb lexical type, its subject, complements, and example. More than half of the verbs in INDRA were semi-automatically acquired from a combination of the English grammar ERG and the English and Indonesian wordnets, as explained in the following subsection.

5.1.1 Verb acquisition and subcategorization

The lexicon is important to the robustness of the grammar. Since inputting words or lexical entries manually into the grammar is labor intensive and time consuming, doing lexical acquisition (semi-)automatically is vital. In order to do this, we need good lexical resources. This subsection describes my attempts to extract Indonesian verbs from Wordnet Bahasa (Nurriil Hirfana Mohamed Noor et al., 2011, p. 258) and group them based on syntactic types in the English Resource Grammar (ERG), such as intransitive, transitive and ditransitive, using Python 3.4 and Natural Language Toolkit (NLTK) (Bird et al.,

Table 5.1: Verb subcategorization in INDRA (as of January 2018)

Verb type	HEAD	Subject	Complement 1		Complement 2		Example	Freq.
		HEAD	HEAD	OPT	HEAD	OPT		
<i>intr-verb-lex</i>	<i>intransitive</i>	<i>subj-noun</i>	(none)		(none)		<i>tidur</i> “sleep”	843
<i>v_ada_np_lex</i>	<i>existential</i>	<i>nounbare</i>	(none)		(none)		<i>ada</i> “EXIST”	1
<i>tr-verb-lex</i>	<i>verb</i>	(unspec.)	(unspec.)		(none)		<i>kejar</i> “chase”	651
$\hookrightarrow$ <i>transitive-verb-lex</i>	<i>transitive</i>	<i>subj-noun</i>	<i>comp-noun</i>	-	(none)		<i>mengejar</i> “chase”	
$\hookrightarrow$ <i>passive-one-verb-lex</i>	<i>pass1</i>	<i>subj-noun</i>	<i>pass1agent</i>	+	(none)		<i>dikejar</i> “chased”	
$\hookrightarrow$ <i>passive-two-verb-lex</i>	<i>pass2</i>	<i>subj-noun</i>	<i>pass2agent</i>	-	(none)		<i>kejar</i> “chased”	
<i>non-opt-tr-verb-lex</i>	<i>transitive</i>	<i>subj-noun</i>	<i>subj-noun</i>	-	(none)		<i>menyerupai</i> “resemble”	23
<i>opt-tr-np-comp-verb-lex</i>	<i>transitive</i>	<i>subj-noun</i>	<i>subj-noun</i>	+	(none)		<i>makan</i> “eat”	97
<i>relational-verb-lex</i>	<i>transitive</i>	<i>subj-noun</i>	<i>nonoleh-adp</i>	+	(none)		<i>berteman</i> “be friends”	2
<i>aspect_perf-aux-lex</i>	<i>aux</i>	<i>subj-noun</i>	<i>+vjp</i>	-	(none)		<i>sudah</i> “PRF”	2
<i>aspect_prog-aux-lex</i>	<i>aux</i>	<i>subj-noun</i>	<i>+vjp</i>	-	(none)		<i>sedang</i> “PROG”	3
<i>modal-aux-lex</i>	<i>aux</i>	<i>subj-noun</i>	<i>+vjp</i>	-	(none)		<i>harus</i> “must”	7
<i>modal_fut-aux-lex</i>	<i>aux</i>	<i>subj-noun</i>	<i>+vjp</i>	-	(none)		<i>akan</i> “FUT”	1
<i>v_pp*_dir_le</i>	<i>transitive</i>	<i>subj-noun</i>	<i>nonoleh-adp</i>	+	(none)		<i>pergi</i> “go”	76
<i>v_np_cop_noasp_le</i>	<i>cop-stat</i>	<i>subj-noun</i>	<i>subj-noun</i>	-	(none)		<i>adalah</i> “COP”	1
<i>v_np_cop_3_le</i>	<i>cop-stat</i>	<i>subj-noun</i>	<i>subj-noun</i>	-	(none)		<i>ialah</i> “COP”	1
<i>v_np_cop_common_le</i>	<i>cop-stat</i>	<i>subj-noun</i>	<i>commonnoun</i>	-	(none)		<i>merupakan</i> “COP”	1
<i>v_np_cop_le</i>	<i>cop-dyn</i>	<i>subj-noun</i>	<i>subj-noun</i>	-	(none)		<i>menjadi</i> “COP”	1
<i>verb-adj-lex</i>	<i>cop-dyn</i>	<i>subj-noun</i>	<i>adj</i>	-	(none)		<i>menjadi</i> “COP”	1

Continued on next page

Table 5.1: Verb subcategorization in INDRA (continued)

Verb type	HEAD	Subject	Complement 1		Complement 2		Example	Freq.
		HEAD	HEAD	OPT	HEAD	OPT		
<i>cop-adalah-adj-lex</i>	<i>cop-stat</i>	<i>subj-noun</i>	<i>adjective</i>	-	(none)		<i>adalah</i> “COP”	1
<i>cop-adalah-pp-lex</i>	<i>cop-stat</i>	<i>subj-noun</i>	<i>nonoleh-adp</i>	-	(none)		<i>adalah</i> “COP”	1
<i>cop-ialah-pp-lex</i>	<i>cop-stat</i>	<i>subj-noun</i>	<i>nonoleh-adp</i>	-	(none)		<i>ialah</i> “COP”	1
<i>locative-verb-lex</i>	<i>locative</i>	<i>subj-noun</i>	<i>nonoleh-adp</i>	-	(none)		<i>berada</i> “be (at)”	2
<i>n+cop+n-lex</i>	<i>copula</i>	<i>subj-noun</i>	<i>subj-noun</i>	-	(none)		<i>bukan</i> “NEG”	1
<i>n+cop+p-lex</i>	<i>copula</i>	<i>subj-noun</i>	<i>adp</i>	-	(none)		<i>bukan</i> “NEG”	1
<i>trans-first-arg-raising-verb</i>	<i>verb</i>	<i>subj-noun</i>	<i>ctrl-rais-cmp-v</i>	-	(none)		<i>tampak</i> “seem”	6
<i>trans-first-arg-control-verb</i>	<i>verb</i>	<i>subj-noun</i>	<i>ctrl-rais-cmp-v</i>	-	(none)		<i>mencoba</i> “try”	67
<i>ques-comp-verb-lex</i>	<i>intransitive</i>	<i>subj-noun</i>	<i>comp</i>	-	(none)		<i>bertanya</i> “ask”	4
<i>prop-comp-verb-lex</i>	<i>intransitive</i>	<i>subj-noun</i>	<i>+vc</i>	-	(none)		<i>berkata</i> “say”	83
<i>v_np_noarg3_le</i>	<i>verb</i>	(unspec.)	(unspec.)		(unspec.)		<i>bawakan</i> “bring”	9
$\hookrightarrow$ <i>ditransitive-n-n-verb-lex</i>	<i>transitive</i>	<i>subj-noun</i>	<i>comp-noun</i>	-	<i>subj-noun</i>	-	<i>membawakan</i> “bring”	151
$\hookrightarrow$ <i>pass-one-ditr-n-n-v-lex</i>	<i>pass1</i>	<i>subj-noun</i>	<i>pass1agent</i>	+	<i>subj-noun</i>	-	<i>dibawakan</i> “brought”	
$\hookrightarrow$ <i>pass-two-ditr-n-n-v-lex</i>	<i>pass2</i>	<i>subj-noun</i>	<i>pass2agent</i>	-	<i>subj-noun</i>	-	<i>bawakan</i> “brought”	
<i>v_np_pp_prop_le</i>	<i>verb</i>	(unspec.)	(unspec.)		(unspec.)		<i>taruh</i> “put”	
$\hookrightarrow$ <i>ditransitive-n-p-verb-lex</i>	<i>transitive</i>	<i>subj-noun</i>	<i>comp-noun</i>	-	<i>nonoleh-adp</i>	+	<i>menaruh</i> “put”	
$\hookrightarrow$ <i>pass-one-ditr-n-p-v-lex</i>	<i>pass1</i>	<i>subj-noun</i>	<i>pass1agent</i>	+	<i>nonoleh-adp</i>	-	<i>ditaruh</i> “put”	
$\hookrightarrow$ <i>pass-two-ditr-n-p-v-lex</i>	<i>pass2</i>	<i>subj-noun</i>	<i>pass2agent</i>	-	<i>nonoleh-adp</i>	-	<i>taruh</i> “put”	

2009). The grouping of verbs in Wordnet (Fellbaum, 1998b), known as *verb frames*, is employed to be the bridge between the English and Indonesian grammar. A verb frame or a subcategorization frame for the verb (Jurafsky & Martin, 2009) is the possible sets of complements. The assumptions are words with similar meaning have similar subcategorization frames and words with the same translations have similar meanings (Fujita & Bond, 2007). In other words, the number of arguments of verbs with similar meaning should be the same across languages.

Each verb synset in PWN (also in Wordnet Bahasa) contains a list of sentence frames specified by the lexicographer illustrating the types of simple sentences in which the verbs in the synset can be used (Fellbaum, 1998b). There are 35 verbal sentence frames in Wordnet, some of them are shown in 143, preceded by their frame numbers:

- (143) 1 Something ----s
 8 Somebody ----s something
 21 Somebody ----s something PP

Frame 1 is a typical intransitive verbal sentence frame, as in *the book fell*; frame 8 is a typical (mono)transitive verbal sentence frame, as in *he chases his friend*; and frame 21 is a ditransitive verbal sentence frame, as in *she put a book on a table*. A verb may have more than one synset and each synset may have more than one verb frame, for example the verb *eat* has six synsets with each synset having different verb frames. Three of the six synsets, together with their definitions and verb frames, are presented in Table 5.2.

Table 5.2: Three of six synsets of the verb *eat* and their verb frames in Wordnet

Synset	Definition	Verb frame
01168468-v	Take in solid food	8 Somebody ----s something
01166351-v	Eat a meal, take a meal	2 Somebody ----s
01157517-v	Use up (resources or materials)	11 Something ----s something 8 Somebody ----s something

These verb frames can be employed as a bridge between the verb types (also verb lexical items) in ERG and those in INDRA so that automatic lexical acquisition can be done. Out of 354 verb types in ERG,² the top eleven most frequently used types in the Redwoods corpus were chosen, excluding the specific English verb types such as ‘be’-type verbs (e.g. *is*, *be* and *was*), ‘have’-type verbs, verbs with prepositions (e.g. *depend on*, *refer to* and *look after*) and modals (e.g. *would*, *may* and *need*). The chosen eleven verb types are given in Table 5.3. The third, fifth and eighth type (*v__unacc_le*, *v__le* and *v_pp_unacc_le*) are regarded as the same type, i.e. intransitive verb type, in INDRA at the present stage. Similarly, the sixth and ninth type are regarded as the same type, i.e.

²http://compling.hss.ntu.edu.sg/ltodb/cgi/ERG_1214//ltypes.cgi

ditransitive verb type; and the seventh and tenth type are regarded as the same verb type with propositional complementizer in INDRA.

Table 5.3: The eleven most frequently used ERG verb types in the corpus

	Verb type	Freq		Examples of verb
		Corp	Lex	
1	<i>v_pp*_dir_le</i>	7,079	204	go, come, hike
2	<i>v_vp_seq_le</i>	3,921	105	want, like, try
3	<i>v_-_unacc_le</i>	3,144	334	close, start, end
4	<i>v_np_noarg3_le</i>	2,723	5	make, take, give
5	<i>v_-_le</i>	2,666	486	arrive, occur, stand
6	<i>v_np-pp_e_le</i>	2,439	334	compare, know, relate
7	<i>v_pp*-cp_le</i>	2,360	154	think, add, note
8	<i>v_pp_unacc_le</i>	2,307	44	rise, fall, grow
9	<i>v_np-pp_prop_le</i>	1,861	135	base, put, locate
10	<i>v_cp_prop_le</i>	1,600	80	believe, know, find
11	<i>v_np_ntr_le</i>	1,558	10	get, want, total

The first type contains verbs expressing movement or direction having optional prepositional phrase (PP) complements expressing direction, as in *B crept into the room*. The verbs in the second type are subject control verbs, the subject of which in the main clause is the same as in the subordinate verb phrase (VP) complement clause, as in *B intended to win*. The third type consists of unaccusative verbs without complements as in *The plate gleamed*. The fourth type contains verbs having two arguments only or monotransitive verbs although they have a potential to be ditransitive as in *B took the book*. The fifth type contains intransitive (unergative) verb as in *B arose*. The verbs in the sixth type have obligatory noun phrase (NP) and PP complements as in *B compared C with D*. The verbs in the seventh type are verbs with optional PP complements and obligatory subordinate clauses as in *B said to C that D won*. Unaccusative verbs with optional PP complements as in *The seed grew into a tree* belong to the eighth type. Ditransitive verbs with obligatory NPs and PPs with state result as in *B put C on D* belong to the ninth type. The tenth type consists of verbs with optional complementizers as in *B hoped (that) C won* and the eleventh type consists of verbs with obligatory NP complements which cannot be passivized as in *B remains C*.

Based on the syntactic information of each verb type mentioned above, the corresponding verb frames in Wordnet were manually chosen. Table 5.4 shows the eleven verb types in ERG and their corresponding Wordnet verb frames.

The first type includes verb frame 2 and verb frame 22 since it has optional PP complements and contains verbs expressing movement or direction. The intransitive verbs include verb frame 1 in which the subject is a thing or verb frame 2 in which the subject is a human. Monotransitive verbs with obligatory NP complements include verb frame 8 in which the subject is a human or verb frame 11 in which the subject is a thing. Ditransitive

Table 5.4: The eleven most frequently used ERG verb types in the corpus and their corresponding Wordnet verb frames

Verb type	Verb frame
<i>v_pp*_dir_le</i>	2 Somebody ----s AND 22 Somebody ----s PP
<i>v_vp_seq_le</i>	28 Somebody ----s to INFINITIVE
<i>v_-_unacc_le</i>	1 Something ----s
<i>v_-_le</i>	OR 2 Somebody ----s
<i>v_pp_unacc_le</i>	
<i>v_np_noarg3_le</i>	8 Somebody ----s something OR 11 Something ----s something
<i>v_np-pp_e_le</i>	15 Somebody ----s something to somebody
<i>v_np-</i>	OR 17 Somebody ----s somebody with something
<i>pp_prop_le</i>	OR 20 Somebody ----s somebody PP OR 21 Somebody ----s something PP OR 31 Somebody ----s something with something
<i>v_pp*-cp_le</i>	26 Somebody ----s that CLAUSE
<i>v_cp_prop_le</i>	
<i>v_np_ntr_le</i>	8 Somebody ----s something OR 11 Something ----s something

verbs have verb frame 20 in which the direct NP object is a human or verb frame 21 in which the direct NP object is a thing. Verbs such as *think* and *believe* with optional or obligatory complementizer include verb frame 26 with *that* clause.

Each verb in each verb type in Table 5.3 was firstly checked whether it is in Wordnet or not. If it could be found in Wordnet, the next step was to check whether the verb includes the verb frames mentioned in Table 5.4 or not. This step had to be done in order to find out the right synset since a verb can have many synsets but different verb frames as shown in Table 5.2. After the right synset was found, the corresponding Indonesian lemmas or translations were checked. One synset may have more than one Indonesian lemma and also may not have Indonesian lemmas at all.

The next important step is to check one by one the Indonesian lemmas belonging to the same synset and verb frames whether each can be grouped in the same verb type or not. This manual step has to be done because grouping verbs in a particular language into types is language-specific work. Arka (2000) states that languages vary with respect to their lexical stock of “synonymous” verbs that may have different argument structures. He gives an example that the verb corresponding to *know* can be both intransitive and transitive in Indonesian: *tahu* and *ketahui* respectively, transitive only with an obligatory NP in Balinese³ *tawang*, and transitive with optional NP in English *know*. Lastly, after

³Balinese (ISO 639-3: ban) is a Western Malayo-Polynesian language of the Austronesian language

the Indonesian verbs were extracted and grouped into their corresponding verb types, a new lexicon file for INDRA was made, in which the verbs are alphabetically sorted.

A few verbs in ERG (240 verbs) could not be found in Wordnet, such as *basejump*, *bird feed*, *carpool*, *counter attack*, *defuel*, *entwist*, *flip flop*, *fuzz*, *gust*, *ice skate*, *increment*, *misfeed*, *multitask*, *roller skate*, *self insure*, *tap dance*, *unfrost* and *water ski*. 181 synsets do not have Indonesian lemmas, such as 01948659-v *balloon*, 00883635-v *gloat*, 01525295-v *malfunction*, 01977421-v *plop* and 00420549-v *tauten*. 372 verbs in Wordnet were found not having the same verb frames as in Table 5.4. For example, there are three types of *afford* in ERG, one of them belongs to the subject control verbs *v_vp_seq_le*, which is supposed to have a verb frame 28 “Somebody - - - s to INFINITIVE” in Wordnet. However, among the four synsets found in Wordnet for *afford*, none of them has the verb frame 28. In total, 939 Indonesian verbs were extracted and grouped into seven verb types as presented in Table 5.5. One verb may belong to more than one verb type. This lexical acquisition is useful to extract lexical items (semi-)automatically through linguistic resources such as Wordnet Bahasa. The generated lexicon can be used to improve the grammar’s coverage.

Table 5.5: New verb types extracted from ERG and their corresponding names in INDRA

ERG verb type	INDRA verb type	Number of verb
<i>v_pp*_dir_le</i>	<i>v_pp*_dir_le</i>	76
<i>v_vp_seq_le</i>	<i>trans-first-arg-control-verb</i>	49
<i>v_-_unacc_le</i>	<i>intr-verb-lex</i>	594
<i>v_np_noarg3_le</i>	<i>v_np_noarg3_le</i>	5
<i>v_np_pp_e_le</i>	<i>v_np_pp_prop_le</i>	126
<i>v_pp*_cp_le</i>	<i>prop-comp-verb-lex</i>	76
<i>v_np_ntr_le</i>	<i>non-opt-tr-verb-lex</i>	13
Total		939

5.1.2 Intransitive verbs

An intransitive clause has two nuclear components: a subject which indicates the actor or agent and a predicate which contains an intransitive verb (Sneddon et al., 2010, p. 250). Sneddon et al. (2010, pp. 65-71) classify Indonesian intransitive verbs into three groups based on the form (with or without affixes) of the verbs:

- Intransitive verbs without affixes. For example, *bangun* “wake up”, *duduk* “sit”, *hidup* “live”, *tenggelam* “sink”, *gagal* “fail”, *terbang* “fly”, and *tidur* “sleep”.
- Intransitive verbs with prefix *ber-*, which is necessary to produce a well-formed verb. For example, *berenang* “swim”, *berdiri* “stand”, and *bekerja* “work”.

family. It belongs to the Malayo-Sumbawan branch. It is mainly spoken in the island of Bali in the Republic of Indonesia as a regional language (Lewis, 2009).

- Intransitive verbs with prefix *meN-*, which is required to produce a well-formed verb. It has a number of different meanings with noun stems. For example, *menangis* “cry”, *meledak* “explode”, *melompat* “jump”, and *menggonggong* “bark”. Sneddon et al. (2010, pp. 70-71) note that with adjective stems, *meN-* usually forms verbs meaning “become”, such as *menghangat* “warm up” or “become warm”, *mengering* “dry” or “become dry”, and *menguning* “grow yellow” or “become yellow”.

Intransitive verbs are simply listed in the INDRA lexicon, together with their affixes. They are of the type *intransitive-verb-lex*, which specifies that they take a subject (the value of the feature SUBJ has one item) and no complements (the value of COMPS is empty). The subject’s INDEX value is linked to the value of ARG1 in the elementary predication contributed by the verb to the semantics. The semantics of an intransitive sentence *anjing menggonggong* “dogs bark” is shown in 19 on page 28.

5.1.3 Verbs with clausal complementizer

Verbs with clausal complementizer in INDRA are either of the type *prop-comp-verb-lex* or of the type *ques-comp-verb-lex*. Both types are subtypes of *clausal-comp-verb-lex*, as shown in (144).

The type *prop-comp-verb-lex* includes intransitive report verbs which can have *bahwa* “COMP” clauses as complements. This group of verbs includes *tahu* “know”, *percaya* “believe”, *berpikir* “think”, and *berkata* “say”. The *bahwa* clauses most frequently occur as objects or complements of verbs which report something that happened or report what someone said (Sneddon et al., 2010, pp. 277-278, 301). The complementizer *bahwa* can be omitted when its clause is object or complement. Example (145) shows a sentence with an optional propositional complementizer *bahwa* “COMP”.

- (145) *Saya tahu (bahwa) dia pencuri.*
 1SG know COMP 3SG thief
 “I know (that) he is a thief.” (Sneddon et al., 2010, p. 301)

The type *ques-comp-verb-lex* includes verbs which can have indirect question clauses with *apakah* “COMP”, translated “if, whether”, as complements (Sneddon et al., 2010, p. 278), as shown in Example (146).

(146) *Saya tidak tahu apakah dia akan datang.*

1SG NEG know COMP 3SG FUT come

“I don’t know whether he will come.” (Sneddon et al., 2010, p. 278)

Complementizer *bahwa* belongs to lexical type *propcomp-lex-item* and *apakah* is of type *qpart-lex-item*. Both lexical types are subtypes of *complementizer-lex-item*, as shown in (147). They contribute semantic information of the feature SF (SENTENTIAL FORCE) in the INDEX but do not introduce any relations (semantically empty). The value of SF in *propcomp-lex-item* is of type *prop* (*proposition*) and the one in *qpart-lex-item* is of type *ques* (*question*). The type *complementizer-lex-item* inherits from *raise-sem-lex-item* and *basic-one-arg*. The type *raise-sem-lex-item* identifies the value of the HOOK of the complementizer (i.e. INDEX | SF) with the HOOK of the first item in the COMPS. Its HEAD is of type *comp* (*complementizer*). Its COMPS has one item, whose HEAD is of type *verb* and its MC (MAIN CLAUSE) value is plus, with both SUBJ and COMPS value are saturated. Both complementizers can take only a full, saturated sentence as its only complement.

The type hierarchy of *clausal-comp-verb-lex* in (144) is shown with each type’s constraint in (148). The type *clausal-comp-verb-lex* inherits from *main-verb-lex* and *clausal-second-arg-trans-lex-item*. It can take either *verb* or *complementizer* as the head of its sole complement (the value of its COMPS’ HEAD is of type *+vc*). The type *prop-comp-verb-lex* adds one constraint that the value of SF in the COMPS’ INDEX is of type *prop*. All verbs belong to this lexical type can combine with either a sentence or a complementizer whose SF is of type *prop*, i.e. *bahwa*. The type *ques-comp-verb-lex* adds two constraints in the value of its COMPS: the SF of the INDEX is of type *ques* and the HEAD is of type *comp*. All verbs belong to this lexical type can combine only with a complementizer whose SF is of type *ques*, i.e. *apakah*.

(148)

clausal-comp-verb-lex

Phrase rule *head-comp* combines complementizers with clauses and main verbs with the complementizers, as shown in (149), parse tree of Example 145. Its DMRS is shown in (150). The verb *tahu* “know” takes two arguments. Its ARG1 is coindexed with the ARG0 of its subject *saya* “1sg” and its ARG2 is *qeq*-ed with the LBL of *cop_v_zero*.

(150)

Similar to *bahwa*, *apakah* is combined with a clause and also with the verb *tahu* “know” using *head-comp* rule, as shown in the parse tree of Example 146 in (151). The ARG2 of *tahu* “know” is *qeq*-ed with the LBL of the verb *datang* “come”, as shown in (152).

(151)

(152)

5.1.4 Existential verb *ada*

Sneddon et al. (2010, p. 272) note that *ada* is an intransitive verb which differs significantly from other verbs and deserves special comment. Tjung (2003) mentions three functions of *ada* corresponding to different word orders of the clause in which it occurs: existential or presentational, locational, and possessive. Sneddon et al. (2010) add one more function: intensifying.

The first one is existential, asserting the existence of an entity (either indefinite or definite) or an event, as shown in Example (153). Usually the noun phrase is indefinite (Alwi et al., 2014, p. 373) or non-specific (Mintz, 2002, p. 29), following *ada*, but a definite noun phrase may occur, as shown in Example (154), preceding *ada*.

(153) *Ada koran.*

EXIST newspaper

“There is a newspaper.” (Sneddon et al., 2010, p. 272)

(154) *Tuhan ada.*

God EXIST

“God exists.” (Sneddon et al., 2010, p. 274)

Tjung (2003) states that *ada* in this first function is similar in its meaning and usage to the construction containing the lexical (unaccusative) verb *exist*, rather than the *there be*-existential construction. It contains only the theme argument and it may have a locative constituent an adverbial adjunct.

The second function is locational, asserting the location of an entity, as shown in Example (155). The verb *ada* is optional and can be substituted with *berada* (Sneddon et al., 2010, p. 273; Mintz, 2002, p. 31). The real predicate is the prepositional phrase (Mintz, 2002, p. 31), which takes one theme argument and one locative argument. Usually the noun phrase (theme) is definite or specific (Alwi et al., 2014, p. 373; Mintz, 2002, p. 30).

(155) *Ayah (ada) di kantor.*

father LOC at office

“Father is in the office.” (Sneddon et al., 2010, p. 273)

The third function is possessive, asserting the individual’s possession of an entity, as shown in Example (156). It takes two arguments: one noun phrase as subject (the possessor) and one noun phrase as complement (the theme). It is more common in colloquial style (Sneddon et al., 2010, p. 274).

(156) *Saya tidak ada uang kecil.*

1SG NEG have money small

“I don’t have any small change.” (Sneddon et al., 2010, p. 274)

The fourth function is intensifying. The word *ada* having this function cannot be regarded as a verb (Sneddon et al., 2010, p. 274). It has high or emphatic (focal) intonation (Hopper, 1972, p. 136). Example (157) shows *ada* having this fourth function.

(157) *Ia ada menerima surat itu.*

3SG do receive letter that

“He did receive the letter.” (Sneddon et al., 2010, p. 274)

Existential, locational, and possessive *ada*, like other verbs, can be preceded by negation word, temporal markers, and modals. Based on the four functions mentioned above, *ada* can be grouped into two: *ada* as a predicate (semantically not empty) and *ada* not as a predicate (semantically empty). Predicative *ada* can be further grouped based on the

number of the arguments: existential *ada* which has only one argument and two possibilities of word order (*ada* precedes or follows the noun phrase argument), and possessive *ada* which has two arguments. Also possibly the intensifying *ada*, but this has not been analyzed and implemented in INDRA. Non-predicative *ada* is the locational *ada*, which is explained in Section 6.2.3 on page 175 on copula clauses with prepositional phrase predicate. The possessive *ada* is treated as monotransitive verb with an obligatory complement but cannot be passivized (see Section 5.1.5). The existential *ada* with the word order NP+*ada*, as in Example (154), is treated as an intransitive verb (see Section 5.1.2).

The existential *ada* with the word order *ada*+NP, as in Example (153), is given a special lexical type *v_exist-lex* which inherits from *basic-one-arg-no-hcons*, *verb-lex*, and *basic-verb-lex*. Its HEAD is of type *existential*. The COMPS has empty value, but the SUBJ has value of one item, whose OPT is minus. The *head-subj* rule unifies the predicate *ada* with the noun phrase argument, as shown in (158), the parse tree of Example (153). It introduces a relation *_exist_v_rel*, the ARG0 is of type *e*, and the ARG1 is coindexed with the INDEX of the noun, as shown in (159).

5.1.5 Transitive verbs

A transitive clause minimally consists of a subject, a predicate, and an object/complement (Sneddon et al., 2010, p. 251). The lexical type *transitive-verb-lex* is the supertype of all transitive verbs in INDRA as shown in (160). Its HEAD is of type *transitive*. Its COMPS has value, containing one item whose POSTHEAD is plus (it occurs after the transitive verb) and its SPR and COMPS have empty value (the noun phrase complement is saturated). It has two subtypes: *opt-tr-verb-lex* whose complement is optional (the value of OPT in the COMPS is plus) and *non-opt-tr-verb-lex* which has an obligatory complement (the value of OPT in the COMPS is minus).

(160)

transitive-verb-lex

This subsection discusses these two subtypes: transitive verbs with an optional complement and those with an obligatory complement, which can further be divided into two groups (those which cannot be passivized and those which can), as well as ditransitive verbs.

Monotransitive verbs with an optional complement

Some transitive verbs do not require the object to be expressed when it is obvious or unimportant (Sneddon et al., 2010, pp. 251-252). Some grammarians, such as Sneddon et al. (2010, pp. 251-252), call them *pseudo-intransitive* verbs, while some grammarians, such as Alwi et al. (2014, pp. 96-97), call them *semitransitive* verb.

Sneddon et al. (2010, pp. 65-66, 251-252) note that based on their forms (with or without affix), these verbs can be classified into three groups:

- Verbs without affixes, such as *makan* “eat” and *minum* “drink”.
- Verbs with prefix *ber-*, such as *belajar* “study”.
- Verbs with prefix *meN-*, such as *menyanyi* “sing” and *memasak* “cook”.

Some verbs can take suffix *-kan* or *-i*. Where suffix *-kan* and *-i* are optional, only the variant without the suffix can occur without an object (Sneddon et al., 2010, p. 252), as shown in Example (161).

- (161) a. *Anak-anak sedang menyanyikan lagu.*
 child-REDUP PROG ACT-sing-APPL song
 “The children are singing a song.” (Sneddon et al., 2010, p. 252)
- b. **Anak-anak sedang menyanyikan.*
 (Intended: “The children are singing.”)

- c. *Anak-anak sedang menyanyi (lagu).*

child-REDUP PROG ACT-sing song

“The children are singing (a song).” (Sneddon et al., 2010, p. 252)

In addition to optional noun phrase complements, some verbs may take optional prepositional phrase complements. Verbs expressing movement or direction such as *berangkat* “depart” and *pergi* “go” can have an optional prepositional phrase complements expressing direction, as shown in Example (162).

- (162) *Mereka belum berangkat (ke Jakarta).*

3PL not.yet depart to Jakarta

“They haven’t left (to Jakarta) yet.” (based on Sneddon et al., 2010, p. 209)

Verbs with prefix *ber-* and noun stems denoting a reciprocal relationship can appear without a prepositional phrase, as shown in Example (163a). They can also occur with a prepositional phrase, as shown in Example (163b), indicating that two people stand in such relationship to each other.

- (163) a. *Ali dan Tomo bertetangga.*

Ali and Tomo ber-neighbour

“Ali and Tomo are neighbours (with each other).” (Sneddon et al., 2010, p. 67)

- b. *Ali bertetangga dengan Tomo.*

Ali ber-neighbour with Tomo

“Ali is neighbour with Tomo.” (based on Sneddon et al., 2010, p. 67)

Based on the type (part-of-speech) of the optional complements mentioned above, *opt-tr-verb-lex* is subtyped into *opt-tr-np-comp-verb-lex* which has an optional nominal complement and *opt-tr-pp-comp-verb-lex* which has an optional prepositional complement, as shown in (164).

- (164)

The type *opt-tr-np-comp-verb-lex* has one item in COMPS whose HEAD is of type *subj-noun* (see Section 4.1 on page 72 on noun subcategorization). On the other hand, the type *opt-tr-pp-comp-verb-lex*'s COMPS is of type *nonoleh-adp* (the head type *adposition* in INDRA is roughly divided into *oleh-adp* and *nonoleh-adp*, see Section 5.2.2 on passive type one with *oleh* “by”). The type *opt-tr-pp-comp-verb-lex* is further subtyped into *relational-verb-lex* and *v_pp*_dir_le*. The former is for verbs denoting a reciprocal relationship, as shown in Example (163), and the later is for verbs expressing movement or direction, as shown in Example (162). Both subtypes have not been constrained yet. Future work is to constraint these two subtypes.

Monotransitive verbs with an obligatory complement, cannot be passivized

A few verbs do not have a passive form and have an obligatory complement (Sneddon et al., 2010, pp. 254-255), such as *menyerupai* “resemble”, also some verbs with prefix *ber-* and *meN-* mentioned in Sneddon et al. (2010, pp. 274-277). Some grammarians, such as Liaw (2004, pp. 113-114), call these verbs *semi-transitive verbs* and *linking verbs*. In INDRA, the verb *ada* which functions as a possessive verb with the meaning “have” or “possess” also belongs to this type. Verbs in this category are of type *non-opt-tr-verb-lex* whose COMPS has a feature OPT with a value minus and the HEAD is of type *subj-noun*.

Monotransitive verbs with an obligatory complement, can be passivized

Monotransitive verbs take two obligatory noun phrase arguments. Included in this group are transitive verbs such as *membuka* “open”, *memukul* “hit”, and *mengejar* “chase”, also verbs with applicative suffixes *-i* and *-kan*, with the exception of *merupakan* (see Section 6.1.1 on copula clauses with noun phrase predicates) (Sneddon et al., 2010, pp. 73-97).

In INDRA lexicon, these monotransitive verbs are listed as lexemes, i.e. in their forms without prefix *meN-*. Thus, *buka* for *membuka* “open”, *pukul* for *memukul* “hit”, and *kejar* for *mengejar* “chase”. They are of type *tr-verb-lex* which serves as daughter of *active-voice-lex-rule-super* and *passive-voice-lex-rule-super*, two inflectional rules which transform transitive verb lexemes into active verbs with prefix *meN-* and passive verbs with prefix *di-* respectively. The type *active-voice-lex-rule-super* inherits from *transitive-verb-lex* which takes two indices as arguments (its ARG1 is linked to the ARG0 of the subject and its ARG2 to the complement), in addition to the ARG0 event variable corresponding to the verb itself. The type hierarchy for *tr-verb-lex* is shown in (167). The details of the inflectional rules are explained in Section 5.2. Example 137 on page 118 shows a sentence with a monotransitive verb and an obligatory complement which can be passivized. Its parse tree is shown in (171) and its DMRS in (172).

Ditransitive verbs

Ditransitive verbs are verbs which require two objects or complements, as shown in Example (165). Example (165a) shows both objects are noun phrases and Example (165b) shows the object immediately following the verb is a noun phrase and the other one is a prepositional phrase.

- (165) a. *Mereka mengirimi kami surat.*
 3PL ACT-send-APPL 1PL.EXCL letter
 “They sent us a letter.” (Sneddon et al., 2010, p. 251)
- b. *Sopir memasukkan mobil ke dalam garasi.*
 driver ACT-put.in car to inside garage
 “The driver put the car into the garage.” (Sneddon et al., 2010, p. 101)

Ditransitive verbs in INDRA lexicon are listed as lexemes, i.e. in their forms without prefix *meN-*. Thus, *kirimi* for *mengirimi* “send” and *kirim* for *mengirim* “send”. Ditransitive verb lexemes taking two noun phrase objects, similar to the one in Example 165a, are of type *v_np_noarg3_le* and the ones taking a noun phrase and a prepositional phrase, as in Example 165b, belong to type *v_np-pp_prop_le* (see type hierarchy 167).

The type *v_np_noarg3_le* serves as daughter of *active-voice-lex-rule-ditrans-n-n* and *passive-voice-lex-rule-ditrans-n-n*, two inflectional rules which transform ditransitive verb lexemes into active voice with prefix *meN-* and passive voice with prefix *di-* respectively. Similarly, the type *v_np-pp_prop_le* serves as daughter of *active-voice-lex-rule-ditrans-n-p* and *passive-voice-lex-rule-ditrans-n-p*. Section 5.2 explains the inflectional rules and Section 5.2.2 deals with passive voice. The type *active-voice-lex-rule-ditrans-n-n* and

active-voice-lex-rule-ditrans-n-p inherit from *ditransitive-n-n-verb-lex* and *ditransitive-n-p-verb-lex* respectively, which are subtypes of *ditransitive-verb-lex*, as shown in (166).

The type *ditransitive-verb-lex* inherits from *ditransitive-lex-item* which has three arguments: its ARG1 is linked to the subject and its ARG2 and ARG3 linked to the two items in the COMPS slot. The HEAD of *ditransitive-verb-lex* is of type *transitive* and the HEAD of the first item in the COMPS slot is of type *comp-noun*. Both items in the COMPS slot have POST-HEAD whose value is plus (they occur after the head verb) and the values of both SPR and COMPS are empty. The HEAD of the second item in the COMPS of *ditransitive-n-n-verb-lex* is of type *subj-noun* and the one of *ditransitive-n-p-verb-lex* is of type *nonoleh-adp*.

5.2 Inflectional rules

Active transitive verbs have prefix *meN-*, as shown in Example 137a on page 118. The construction is called an active clause or said to be in active voice. The subject is the actor or agent and the other participant is expressed by the object. The Indonesian passive verb either has prefix *di-* (passive type one) or has no prefix (passive type two), as shown in Example 137b and Example 137c on page 118. The construction is called a passive clause or is said to be in passive voice. In passive type two, the verb may be attached to *ku-* “1sg” and *kau-* “2sg” denoting the first and the second singular pronoun agent, respectively, as shown in Example 178 on page 144. This section discusses the inflectional rules taking part in word formation processes of active monotransitive and ditransitive verbs (with *meN-*), as well as passive monotransitive and ditransitive verbs (both with and without *di-*, *ku-*, and *kau-*).

Copestake (2002, p. 124) notes that there is a variety of unary rule in the LKB: *morphological rules* are used if affixation is needed and *lexical rules* are used if no affix is involved. INDRA uses morphological rules to encode inflection. Inflectional rules map between *lexemes* and *words*. The lexical entries or lexemes in INDRA’s lexicon which undergo inflectional rules are not always the root word (*kata dasar*). For example, the root word for both *mengirim* and *mengirimi* in Example (165) is *irim*. However, their stems are different: the former has the stem *irim* and the later *irimi*. INDRA, at the present stage, lists the verb stems, instead of roots in the lexicon. No affix is attached to the stem in the word formation process for passive type two.

Verb lexemes taking part in inflectional rules in INDRA are categorized into three lexical types: *tr-verb-lex* for monotransitive verbs, *v_np_noarg3_le* for ditransitive verbs with two noun phrase complements, and *v_np_pp_prop_le* for ditransitive verbs with one noun phrase and one prepositional phrase complement, as shown in (167) and discussed in Section 5.1.5. All three lexical types have the same constraint: they have a feature INFLECTED which has a feature VOICE-FLAG whose value is minus. The supertypes of these

three types are defined in `matrix.tdl`.⁴

(167)

5.2.1 Active voice

In an inflectional rule with *meN*- “ACT”, a number of nasalization (sound changes) occurs when it combines with the stems.⁵ Table 5.6 shows the patterns of sound changes when *meN*- “ACT” combines with stems. A stem loses its initial consonant if the consonant is one of the following voiceless consonants: /p/, /t/, /s/ and /k/. It retains its initial consonant otherwise. The patterns of sound changes of every possible combination of consonant clusters in Alwi et al. (2014, pp. 67-78) were manually examined using KBBI (Alwi et al., 2008; Sugono et al., 2014) and summarized.

The active transitive word *mengejar* “ACT-chase”, as shown in Example 137a on page 118, can be constructed by applying the *act-prefix* to the lexeme *kejar* “chase”. The rule *act-prefix* is specified in (168). Copestake (2002, pp. 127-130) describes the TDL implementation for morphological rules. The spelling rule introduced by % that comes immediately after the := has two parts: the first is simply either *prefix* or *suffix* and the second is the affixation pattern which contains a set of subrules in the form of matched simple partial regular expressions. Subrules are checked from right to left when generating.

(168)

```
act-prefix :=
%prefix (p mem) (pl mempl) (pr mempr) (ps memps) (pt mempt) (b memb)
(m mem) (f memf) (v memv) (w mew) (t men) (tr mentr) (ts ments)
(d mend) (n men) (r mer) (s meny) (sl mensl) (sr mensr) (sp mensp)
(sm mensm) (sn mensn) (sk mensk) (sy mensy) (sw mensw) (st menst)
(z menz) (l mel) (c menc) (j menj) (ny meny) (y mey) (k meng)
(kh mengkh) (kl mengkl) (kr mengkr) (g mengg) (ng meng) (q mengq)
(h mengh) (a menga) (i mengi) (u mengu) (e menge) (o mengo)
act-lex-rule &
[ SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.OPT - ].
```

⁴I treat voice morphology as part of inflectional morphology, although it is theoretically debatable whether voice alternation is indeed “inflectional”.

⁵Unlike *meN*- “ACT”, the passive voice prefix *di*- “PASS”, as well as *ku*- “1SG” and *kau*- “2SG”, however, does not undergo sound changes.

Table 5.6: Sound changes in inflectional rule with *meN-*

Allomorph of <i>meN-</i>	Initial orthography of the base		Example	
<i>mem-</i>	p	(lost)	<i>meN-</i> + <i>pakai</i>	→ <i>memakai</i> “use”
	pl, pr, ps, pt, b, bl, br, f, fl, fr, v	(retained)	<i>meN-</i> + <i>beli</i>	→ <i>membeli</i> “buy”
<i>men-</i>	t	(lost)	<i>meN-</i> + <i>tanam</i>	→ <i>menanam</i> “plant”
	tr, ts, d, dr, c, j, sl, sr, sy, sw, sp, st, sk, sm, sn, z	(retained)	<i>meN-</i> + <i>cari</i>	→ <i>mencari</i> “look for”
<i>meny-</i>	s	(lost)	<i>meN-</i> + <i>sewa</i>	→ <i>menyewa</i> “rent”
<i>meng-</i>	k	(lost)	<i>meN-</i> + <i> kirim</i>	→ <i>mengirim</i> “send”
	kh, kl, kr, g, gl, gr, h, q, a, i, u, e, o	(retained)	<i>meN-</i> + <i>ganti</i>	→ <i>mengganti</i> “replace”
<i>me-</i>	m, n, ny, ng, l, r, w, y	(retained)	<i>meN-</i> + <i>lempar</i>	→ <i>melempar</i> “throw”
<i>menge-</i>	(base with one syllable)		<i>meN-</i> + <i>cek</i>	→ <i>mengecek</i> “check”

Besides the consonant clusters, when the stem consists of only one syllable, which usually are loanwords, *meN-* becomes *menge-* with no sound changes in the base. Every possible combination of one syllable stem with *meN-* which forms a transitive verb (can be passivized) in KBBI (Alwi et al., 2008) was listed up. There were 44 one syllable words in total. In addition to the consonant clusters and one syllable stems, a manual extension was also done for the exceptions. The sound /p/ is usually lost when combined with *meN-* but it is retained when it is a prefix *per-* as in *pertinggi* (from *per-* and *tinggi* “high”). There are also stems such as *punyai* “have” which do not undergo the common sound changes (Sneddon et al., 2010, pp. 16-17). All stems containing one syllable and prefix *per-*, as well as those which do not undergo the common morphological rule for inflection with *meN-* are treated specially in a separate file named `irregs.tab`, a special file used to define irregular forms. Some irregular forms listed in `irregs.tab` are shown in (169). Each irregular form entry consists of a triplet: inflected form, rule specification, and stem (Copestake, 2002, p. 200).

(169)

mengecek	act-prefix	cek
mengebel	act-prefix	bel
mengeposkan	act-prefix	poskan

mengesahkan	act-prefix	sahkan
memperkenalkan	act-ditrans-n-p-prefix	perkenalkan
mempersembahkan	act-ditrans-n-p-prefix	persembahkan

The *act-prefix* rule inherits from *act-lex-rule*, which inherits from *active-voice-lex-rule-super*, as shown in (170). This rule inherits from *infl-lex-rule* and *transitive-verb-lex*. It takes *tr-verb-lex* as DTR and operates like a unary grammar rule in that a lexeme, e.g. *kejar* “chase”, can instantiate the single element of the ARGS list of the rule to give the *word*, with added orthographic information describing affixation, e.g. *mengejar* “ACT-chase”, whose HEAD is of type *transitive*, and a constraint stating that the object whose POSTHEAD value is plus, is obligatory. In addition, it changes the value of the feature VOICE-FLAG from minus to plus, i.e. from not-inflected from (lexeme) to the inflected form (word).

(170)

```
act-lex-rule := active-voice-lex-rule-super.

active-voice-lex-rule-super := add-only-no-ccont-rule &
  infl-lex-rule & transitive-verb-lex &
  [ INFLECTED.VOICE-FLAG +,
    DTR tr-verb-lex ].
```

Parse tree of Example (137a) is shown in (171). It shows the rule *act-prefix* changes the lexeme *kejar* “chase” into an active transitive word *mengejar* “ACT-chase”. The *head-comp* rule combines it with the object.

The semantics is shown in (172). The value of ARG0 of named_rel “adi” and named_rel “budi” refer to the value of the ARG1 and ARG2 feature of the semantic head daughter respectively.

(172)

Ditransitive verb lexemes are treated in a similar way. The *active-voice-lex-rule-ditrans-n-n* rule transforms its daughter, of type *v_np_noarg3_le*, a lexeme (the value of VOICE-FLAG is minus) to a word (VOICE-FLAG is plus) with prefix *meN-* through *act-ditrans-n-n-prefix* rule. The result is an active ditransitive word, of type *ditransitive-n-n-verb-lex*. The *active-voice-lex-rule-ditrans-n-p* rule transforms its daughter, of type *v_np_pp_prop_le*, a lexeme (the value of VOICE-FLAG is minus) to a word (VOICE-FLAG is plus) with prefix *meN-* through *act-ditrans-n-p-prefix* rule. The result is an active ditransitive word, of type *ditransitive-n-p-verb-lex*. Parse tree of Example (165a) is shown in (173). It shows the rule *act-ditrans-n-n-prefix* changes the lexeme *kirimi* “send-APPL” into an active ditransitive word *mengirimi* “ACT-send-APPL”. The *head-comp* rule combines it with the first noun phrase object, and again with the second noun phrase object.

The semantics is shown in (174). The ARG0 of the pronoun *mereka* “3PL”, *kami* “1PL.EXCL”, and *_surat_n_rel* refer to the value of the ARG1, ARG2, and ARG3 feature of the semantic head daughter *_kirimi_v_rel* respectively.

(174)

5.2.2 Passive voice

A passive is a semantically transitive (two-participant) clause. Typically, the agent is either omitted or demoted to an oblique role, the other core participant possesses all properties of subjects, and the verb possesses formal properties of intransitive verbs (Payne, 2008, p. 204). Passive constructions are far more frequent in Indonesian than in English; an Indonesian passive is often naturally translated into English by an active construction (Sneddon et al., 2010, pp. 256, 263-264). Passive constructions in Indonesian are used in imperatives and for politeness, as well as in relative clauses which can only relativize subjects on defining relative clauses. Research on Indonesian passives has been done by many linguists, such as McCune (1979), Voskuil (2000), Arka & Manning (2008), Cole et al. (2008), and Nomoto (2013). There has been a lot of linguistic work on Indonesian voice, in particular the status of passive-like structures in Indonesian and Austronesian languages (Musgrave, 2001; Riesberg, 2014).

There are two main types of passive in Standard Indonesian, following Sneddon et al. (2010, pp. 256-260) and Alwi et al. (2014, pp. 352-356). They are called ‘passive type 1’ (P1) and ‘passive type 2’ (P2).⁶ Both types are available for monotransitive and ditransitive verbs. They promote an object to subject. If there are two objects in an active ditransitive clause, only the one immediately following the verb (which has semantic role as patient or recipient) can be promoted to subject of the passive (Sneddon et al., 2010, p. 260).⁷ P1 and P2 are in (near) complementary distribution. P1 takes only a third person agent, while P2 may take first, second, and third person agent. P1 and P2 overlap with respect to the third person agent (Sneddon et al., 2010, p. 256).

Passive type 1

The verb in P1 is morphologically built by attaching a prefix *di-* to a transitive verbal stem (lexeme) in the lexicon. The subject (which usually has semantic role as agent) in the active sentence becomes an optional complement, immediately follows the passive verb (post-verbal), and it is optionally marked by a semantically empty preposition *oleh* ‘by’. Its PERNUM is third person, i.e. pronoun *dia* ‘3SG’, *mereka* ‘3PL’, enclitic *=nya* ‘3SG’, (common) noun, or proper name (Sneddon et al., 2010, p. 256-257) (see the type hierarchy 60 on page 74 for *pass1agent* which is the type for agent in passive type one). The position of the components of the predicate, such as auxiliaries and temporal markers, as well as the negative word *tidak* ‘NEG’ (see Section 5.3 on auxiliaries and Section 5.4 on negation), remain unchanged, i.e. they immediately precede the verb predicate both in active and passive voice.

⁶Other types such as passives with prefix *ter-* and circumfix *ke-...-an* have not been analyzed and implemented in INDRA. They are for future work.

⁷This study only describes passives for monotransitive verbs. However, the analysis proposed here can be applied to ditransitive verbs as well.

Example (175a) shows a transitive sentence in active voice. An aspect marker *sudah* “PRF” immediately precedes the active voice verb *menjemput* “ACT-pick.up”. Its corresponding P1 constructions are shown in Example (175b) to (175e). The position of the aspect marker is the same in all example sentences in (175). Example (175b), (175c), and (175d) show the optional preposition *oleh* “by”. Example (175c) and (175d) show that the enclitic *=nya* “3SG” can attach directly to the passive verb or to the preposition *oleh* “by”. Example (175e) shows that a P1 construction may occur without a complement.

- (175) a. *Dia sudah menjemput Budi.*
 3SG PRF ACT-pick.up Budi
 “He has met Budi.” (lit. “He has picked Budi up.”) (based on Sneddon et al., 2010, p. 256)
- b. *Budi sudah dijemput (oleh) dia.*
 Budi PRF PASS-pick.up by 3SG
 “Budi has been met by him.” (lit. “Budi has been picked up by him.”) (based on Sneddon et al., 2010, p. 257)
- c. *Budi sudah dijemputnya.*
 Budi PRF PASS-pick.up=3SG
 “Budi has been met by him.” (lit. “Budi has been picked up by him.”) (based on Sneddon et al., 2010, p. 257)
- d. *Budi sudah dijemput olehnya.*
 Budi PRF PASS-pick.up by=3SG
 “Budi has been met by him.” (lit. “Budi has been picked up by him.”) (based on Sneddon et al., 2010, p. 257)
- e. *Budi sudah dijemput.*
 Budi PRF PASS-pick.up
 “Budi has been met.” (lit. “Budi has been picked up.”) (based on Sneddon et al., 2010, p. 257)

In a coordinative construction with two or more passive verbs, the agent (both full forms and the bound form or enclitic *=nya*) can appear only once, following the last passive verb, as shown in (176).

- (176) *Budi sudah ditunggu dan dijemputnya.*
 Budi PRF PASS-wait and PASS-pick.up=3SG
 “Budi has been waited and picked up by him.”

Passive type 2

The verb in P2 is morphologically built by not attaching any affixes to a transitive verb lexeme in the lexicon. The verbs appear in bare stem form. Different from P1, the subject (agent) in the active sentence becomes an obligatory complement (argument), immediately preceding the verb (pre-verbal), without any prepositions such as *oleh* “by”. The agent is a pronoun such as *aku* “1sg”, *engkau* “2sg”, *dia* “3sg” etc. or *pronoun substitute*, i.e. kinship terms such as *bapak* “father”, *ibu* “mother”, and personal names which can refer to the addressee, meaning “you”, or to the speaker, meaning “I” (Sneddon et al., 2010, pp. 257, 259) (see the type hierarchy (57) for *pass2agent* which is the head type for agent in passive type two). No other component of the clause, such as negative and temporal marker, can come between the NP agent and the P2 verb (Sneddon et al., 2010, p. 258). They must occur before the agent.

Example (177) shows the corresponding P2 construction of Example (175a). The aspect marker *sudah* “PRF” precedes the agent *dia* “3sg”.

(177) *Budi sudah dia jemput.*

Budi PRF 3sg pick.up

“He has met Budi.” (lit. “Budi has been picked up by him.”) (based on Sneddon et al., 2010, p. 257)

If the agent is *aku* “1sg” or *engkau* “2sg”, the bound forms (also called as *proclitics* by some grammarians) *ku-* “1sg” and *kau-* “2sg” usually occur (Sneddon et al., 2010, p. 258), as shown in (178).

(178) *Budi sudah kujemput.*

Budi PRF 1sg-pick.up

“I have met Budi.” (lit. “Budi has been picked up by me.”) (based on Sneddon et al., 2010, p. 257)

In a coordinative construction with two or more passive verbs, the bound forms usually occur before each passive verb, as shown in (179a).

(179) a. *Budi sudah kutunggu dan kujemput.*

Budi PRF 1sg-wait and 1sg-pick.up

“Budi has been waited and picked up by me.”

b. ??*Budi sudah kutunggu dan jemput.*

Budi PRF 1sg-wait and pick.up

$$(180) \quad \left[\begin{array}{c} \text{INPUT} \quad \left\langle X, \left[\begin{array}{c} \text{lexeme (tr-verb-lex)} \\ \text{ARG-ST} \left\langle [\dots], [\dots] \right\rangle \end{array} \right] \right\rangle \\ \\ i\text{-rule :} \\ \\ \text{OUTPUT} \quad \left\langle (\text{di-/ku-/kau-})X, \left[\begin{array}{c} \text{word (passive-transitive-lex-item)} \\ \text{ARG-ST} \left\langle \left[\begin{array}{c} \text{INDEX} \quad \boxed{1} \\ \text{ICONS-KEY} \quad \boxed{3} \end{array} \right], \left[\begin{array}{c} \text{INDEX} \quad \boxed{2} \\ \text{ICONS-KEY} \quad \boxed{4} \end{array} \right] \right\rangle \\ \text{LKEYS.KEYREL} \quad \left[\begin{array}{c} \text{ARG1} \quad \boxed{2} \\ \text{ARG2} \quad \boxed{1} \end{array} \right] \\ \text{ICONS} \left\langle ! \quad \boxed{3} \left[\begin{array}{c} \text{focus-or-topic} \\ \text{IARG2} \quad \boxed{1} \end{array} \right], \boxed{4} \left[\begin{array}{c} \text{non-topic} \\ \text{IARG2} \quad \boxed{2} \end{array} \right] ! \right\rangle \end{array} \right] \right\rangle \end{array} \right]$$

Analysis of passive constructions

I treat passive as an inflectional rule, as shown in (180). The input is a lexeme, of type *tr-verb-lex*, which has two arguments. The output is a word, of type *passive-transitive-lex-item* which adds the semantic information for passives, i.e. its ARG1 is coindexed with the ARG0 of the complement (agent) and its ARG2 with the subject. The prefix *di-*, *ku-*, or *kau-* may be attached. Following Song (2017, pp. 211-214), I added information in the ICONS. The promoted argument or the subject is marked as *focus-or-topic*, while the demoted argument is marked as *non-topic*.

We treat *ku-* “1sg”, *kau-* “2sg”, and *=nya* “3sg” differently because of the difference in their occurrence in coordinative constructions and their optionality. Following Zwicky & Pullum (1983) who distinguish clitics from inflectional affixes, I tokenize *=nya*, treating it as a word which belongs to a type *encl-3pers*. One of the reasons is because *=nya* can attach both to the verb or to a preposition. On the other hand, I do not tokenize *ku-* and *kau-* and treat them as inflectional affixes.

I made four lexical rules for P1 and P2, as shown in (181). The first rule is for P1 (having an optional complement) without *oleh* “by” and the second one is with *oleh*. The third rule is for P2 (having an obligatory complement) without affixes and the fourth one is for P2 with a saturated complement and a prefix *ku-* or *kau-*. The details of each rule will be discussed in the next section.

(181)

Analysis of P1

Similar to the active voice inflectional rule mentioned in Section 5.2.1, the passive voice inflectional rule *pas-one-prefix* which adds prefix *di-* is defined in `irules.tdl`, as shown in (182). Unlike prefix *meN-*, no sound changes occur in the inflection with prefix *di-*. The *pas-one-prefix* rule inherits from *pas-one-lex-rule*, which inherits from *passive-voice-lex-rule-super* and *passive-one-verb-lex*. The *passive-voice-lex-rule-super* inherits from *infl-lex-rule*, a unary inflectional lexical rule, which takes a transitive verb lexeme *tr-verb-lex* as its only daughter. It transforms the *tr-verb-lex*, from an uninflected lexeme (the value of `VOICE-FLAG` is minus) to an inflected passive word (`VOICE-FLAG` is plus), of type *passive-one-verb-lex*, with an optional complement.

(182)

```

pas-one-prefix :=
%prefix (* di)
pas-one-lex-rule &
  [ SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.OPT + ].

pas-one-lex-rule := passive-voice-lex-rule-super & passive-one-verb-lex.

passive-voice-lex-rule-super := add-only-no-ccont-rule & infl-lex-rule &
  [ INFLECTED.VOICE-FLAG +,
    DTR tr-verb-lex ].

```

The *passive-one-verb-lex* inherits from *passive-transitive-lex-item* and *main-verb-lex*. It contributes the value of the HEAD, which is of type *pass1* and the COMPS, which has one item as its value. The HEAD of the COMPS is of type *pass1agent*. Its POSTHEAD's value is plus and it is saturated. The *passive-transitive-lex-item* contributes the semantic information, i.e. its ARG1 is coindexed with the ARG0 of the optional complement (agent) and its ARG2 with the subject, as shown in (183).

The parse tree of (175c) is shown in (184). It shows the *pas-one-prefix* rule changes the lexeme *jemput* “pick.up” to an inflected passive word *dijemput* “PASS-pick.up”. The inflected passive word is combined with its optional complement via *head-comp* rule. The semantics of the passive sentences in examples (175b) to (175d) look very much like the semantics of their active sentence counterpart in (175a), as shown in (185), with additional information on the information structure. The ARG1 is linked to the optional agent complement and the ARG2 linked to the subject.

(184)

(185)

I treat *oleh* “by” as a semantically empty preposition. It adds nothing to the meaning except the information that the **COMPS** of the passive verb is coindexed with the one of *oleh*, as shown in (186). The semantics of the PP headed by *oleh* is identical to that of *oleh*’s NP complement.

In a similar way, passive ditransitive verbs are built from the inflectional rule *passive-voice-lex-rule-ditrans-n-n*, as shown in (187). This rule takes *v_np_noarg3_le* as daughter, changes the value of the feature **VOICE-FLAG** from minus to plus, add a prefix *di-* “PASS”, and makes it a passive word of type *passive-one-ditrans-n-n-verb-lex* whose first complement (agent) is optional and second complement is obligatory.

(187)

```

pas-one-ditrans-n-n-prefix :=
%prefix (* di)
pas-one-ditrans-n-n-lex-rule &
[ SYNSEM.LOCAL.CAT.VAL.COMPS < [OPT +], [OPT -] > ].

```

```

pas-one-ditrans-n-n-lex-rule := passive-voice-lex-rule-ditrans-n-n &
    passive-one-ditrans-n-n-verb-lex.

passive-voice-lex-rule-ditrans-n-n := add-only-no-ccont-rule &
    infl-lex-rule &
    [ INFLECTED.VOICE-FLAG +,
      DTR v_np_noarg3_le ].

```

The type hierarchy of *passive-one-ditrans-n-n-verb-lex* is shown in (188). The supertype is *passive-ditransitive-lex-item*, which has two subtypes: *passive-one-ditransitive-lex-item* and *passive-two-ditransitive-lex-item* (will be explained in the next subsection). The lexical type *passive-one-ditrans-n-n-verb-lex* has a feature **HEAD** whose value is of type *pass1*. It has three saturated noun phrase arguments: the first one is the subject, its **HEAD** is of type *subj-noun*; the second one is the first complement (the agent) whose **HEAD** is of type *pass1agent*; and the third one is the second complement, of type *subj-noun*. Both complements have a feature **POSTHEAD** whose value is *plus*. The semantics is defined in the supertype (*passive-ditransitive-lex-item*). The **ARG1** is linked to the first complement (the agent), the **ARG2** is linked to the subject, and the **ARG3** is linked to the second complement. The lexical type *passive-one-ditrans-n-p-verb-lex* is similar to *passive-one-ditrans-n-n-verb-lex* but the second complement is of type *nonoleh-adp*.

(188)

Analysis of P2

I made a morphological rule *pas-two-no-prefix*, a lexical rule *pas-two-lex-rule*, and an inflectional rule *passive-voice-lex-rule-super* for passive type two, as shown in (189). The morphological rule *pas-two-no-prefix* says that it inherits from *pas-two-lex-rule* without any affixes and the complement is obligatory. The lexical rule *pas-two-lex-rule* inherits from *passive-voice-lex-rule-super*, an inflectional rule which takes *tr-verb-lex* as its daughter, changes it from a lexeme to a word of type *passive-two-verb-lex* whose HEAD is of type *passive-two*.

(189)

```

pas-two-no-prefix :=
pas-two-lex-rule &
  [ SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.OPT - ].

pas-two-lex-rule := passive-voice-lex-rule-super & passive-two-verb-lex &
  [ SYNSEM.LOCAL.CAT.HEAD passive-two ].

```

```

passive-voice-lex-rule-super := add-only-no-ccont-rule & infl-lex-rule &
[ INFLECTED.VOICE-FLAG +,
  DTR tr-verb-lex ].

```

The type hierarchy for *passive-two-verb-lex* is shown in (190). It takes two saturated noun phrase arguments: the first argument is the subject whose HEAD's value is of type *subj-noun* and the second argument is the sole item in the COMPS whose HEAD's value is of type *pass2agent* and it has a feature POSTHEAD whose value is minus, i.e. it must occur before the head verb. The coindexation of the arguments is defined in *passive-transitive-lex-item*. The ARG1 is coindexed with the INDEX of the complement (agent) and the ARG2 is coindexed with the subject.

In addition, I made a new phrase rule called *complement-head* rule which inherits from *basic-head-1st-comp-phrase* and *head-final*. It is constrained to lexical P2 head only. The HEAD value of its HEAD-DTR is of type *passive-two*. Parse tree of (177) is shown in (191). The complement (agent) and P2 verb are combined by *complement-head* rule, the result is combined with the aspect marker by *head-complement* rule. Its semantics is similar to the one shown in (185).

(191)

Similar to the rule for passive monotransitive verbs type two, passive ditransitive verbs type two are built from a morphological rule *pas-two-ditrans-n-n-no-prefix*, a lexical rule *pas-two-ditrans-n-n-lex-rule*, and an inflectional rule *passive-voice-lex-rule-ditrans-n-n*, as shown in (192).

(192)

```

pas-two-ditrans-n-n-no-prefix :=
pas-two-ditrans-n-n-lex-rule &
  [ SYNSEM.LOCAL.CAT.VAL.COMPS < [OPT -], [OPT -] > ].

pas-two-ditrans-n-n-lex-rule := passive-voice-lex-rule-ditrans-n-n &
  passive-two-ditrans-n-n-verb-lex.

passive-voice-lex-rule-ditrans-n-n := add-only-no-ccont-rule &
  infl-lex-rule &
  [ INFLECTED.VOICE-FLAG +,
    DTR v_np_noarg3_le ].

```

The inflectional rule *passive-voice-lex-rule-ditrans-n-n* takes a lexeme of type *v_np_noarg3_le* as daughter and makes it an inflected passive word of type *passive-two-ditrans-n-n-verb-lex*. Its type hierarchy together with the constraints is shown in (193).

(193)

For P2 with *ku-* “1sg”, we made a rule *pas-two-prefix-ku* which adds *ku-*. It adds the semantic information that the PERNUM of the ARG1 is first person singular. The COMPS is saturated. The result is a passive verb with *ku-* whose COMP’s value is empty (saturated) but still needs a subject. The verb’s ARG2 is coindexed with the INDEX of the subject, whose HEAD’s value is of type *subj-noun*.

Parse tree of (178) is shown in (194). It shows *pas-two-prefix-ku* rule makes the lexeme *jemput* “pick.up” become *kujemput* “1sg-pick.up”. The result is the verb *kujemput* “1sg-pick.up” which has *aku* “1sg” in the semantics, coindexed with the ARG1 of the verb. This verb is then combined with an aspect marker *sudah* “PRF” by *head-complement* rule. Its semantics is similar to the one shown in (185). For P2 with *kau-*, I treat it similarly as for P2 with *ku-*. I made a rule *pas-two-prefix-kau* which adds *kau-* with the PERNUM of ARG1 is second person singular.

Summary

I made four rules for two types of passive (P1 and P2) and type hierarchies for the complement nouns (agent). Due to the optionality of the complements in coordinative constructions, the bound pronouns *-nya* “3sg”, *ku-* “1sg”, and *kau-* “2sg” are treated differently: *-nya* is treated as a word, while *ku-* and *kau-* are treated as affixes. I made a *complement-head* rule which combines a complement with a P2 verb without affixes. The next section discusses auxiliaries, such as *sudah* “PRF”.

5.3 Auxiliaries

Auxiliaries are verbs in that they satisfy the morphosyntactic definition of verbs, e.g. they occur in the position of a verb and they carry at least some of the inflectional information normally associated with verbs. However, they are often semantically empty and do not embody the major conceptual relation, state, or activity expressed by the clause; or they express auxiliary information such as tense, aspect, or mode (Payne, 2008, p. 84).

(195)

subj-raise-aux

subj-raise-aux-no-pred (199) *subj-raise-aux-with-pred* (203)

All auxiliaries in INDRA inherit from *subj-raise-aux*, as shown in (195). The supertype *subj-raise-aux* inherits from *aux-lex* and *trans-first-arg-raising-lex-item* (see type hierarchy 141 on page 120). Auxiliaries are regarded as raising verbs. *subj-raise-aux* takes one subject and one complement. The subject is a saturated noun phrase, its head is of type *subj-noun*. Its INDEX is linked to the auxiliary's XARG. The complement's head is of type *+vjp*, i.e. it can take a verb, adjective, or preposition as its complement. Its POSTHEAD is plus, so the verb, adjective, and preposition must occur after the auxiliary. The SUBJ in the COMPS has value of one item, i.e. the complement is not saturated and still looking for a subject.

The auxiliary lexical supertype *subj-raise-aux* has two subtypes: *subj-raise-aux-no-pred* which does not contribute any semantic predicate and *subj-raise-aux-with-pred* which does. This section is divided into two subsections. Section 5.3.1 describes temporal markers which are semantically empty (do not add any relations to the semantics) but adds the ASPECT OR TENSE value. Section 5.3.2 describes modals which adds a semantic relation.

5.3.1 Temporal markers

Temporal markers are words modifying a predicate which indicates that an action or state has occurred, is occurring, or is yet to occur in relation to the moment of utterance or in relation to some other event referred to (Sneddon et al., 2010, p. xxxvi).

Some Indonesian temporal markers which contribute information about the feature

ASPECT are *sudah* “PRF”,⁸ *telah* “PRF”, *sedang* “PROG”, and *tengah* “PROG”. The type hierarchy for these aspect markers is shown in (196).

The perfect aspect markers *sudah* and *telah* usually indicate that an action has occurred before another action or that a state has been achieved. *telah* is usually used in writing and very formal speech, while *sudah* is used in all registers (Sneddon et al., 2010, pp. 204-205, 207). The progressive aspect marker *sedang* and *tengah* indicate that an action is in progress. *tengah* is less frequently used than *sedang* (Sneddon et al., 2010, pp. 205-206).

Another temporal marker is *akan* which indicates a future event or state (Sneddon et al., 2010, p. 206). Arka (2013a) states that there is future tense in Indonesian, expressed by *akan*. Other grammarians such as Mintz (2002, pp. 72-73) and Liaw (2004, p. 67) mention that tense is indicated by temporal noun phrases or auxiliary verbs. I follow Arka (2013a) in that Indonesian denotes future tense by auxiliary verb *akan* but at the same time, tense is underspecified if there is no *akan* (see Example 197a and 197b).

- (197) a. *Anjingnya akan menggonggong.*
 dog=DEF FUT bark
 “The dog will bark.” (MRS Test-suite No. 392)
- b. *Anjingnya menggonggong.*
 dog=DEF bark
 “The dog barked/barks/will bark.” (based on MRS Test-suite No. 392)

The type hierarchy for *tense* is shown in (198).

All temporal markers in INDRA inherit from *subj-raise-aux-no-pred*, which inherits from *subj-raise-aux*, *raise-sem-lex-item*, and *no-icons-lex-item*. The *subj-raise-aux-no-pred* has three subtypes: *aspect_perf-aux-lex* (for *sudah* “PRF” and its variants), *aspect_prog-aux-lex* (for *sedang* “PROG” and its variants), and *modal_fut-aux-lex* (for *akan* “FUT”), as shown in (199).

⁸Olsson & Moeljadi (2014) found little overlap between the perfect and *sudah*

(199)

The *aspect_perf-aux-lex* takes a complement having a feature **FORM** whose value is *form* (underspecified) and makes a phrase, in which the **FORM**'s value is *finite*. It contributes the value of feature **ASPECT**, of type *perf*. Similarly, the *aspect_prog-aux-lex* takes a complement of underspecified type of form and makes the value of the **FORM** of type *finite*. It contributes the value of the **ASPECT**, of type *prog*. The feature **FORM** is needed so that *sudah* “PRF” and *sedang* “PROG” cannot co-occur. On the other hand, *modal_fut-aux-lex* only contributes the **TENSE** value, of type *future*, without adding value of **FORM** of the phrase. The temporal marker *akan* “FUT” can occur with other temporal markers *sudah* “PRF”, *sedang* “PROG”, and their variants.

- (200) a.
$$\left[\begin{array}{l} \text{HEAD} \mid \text{FORM } \textit{finite} \\ \text{VAL} \mid \text{COMPS} \left\langle \left[\begin{array}{l} \text{HEAD} \mid \text{FORM } \textit{form} \\ \text{INDEX} \mid \text{E} \mid \text{ASPECT } \textit{perf} \end{array} \right] \right\rangle \end{array} \right]$$
- b.
$$\left[\begin{array}{l} \text{HEAD} \mid \text{FORM } \textit{finite} \\ \text{VAL} \mid \text{COMPS} \left\langle \left[\begin{array}{l} \text{HEAD} \mid \text{FORM } \textit{form} \\ \text{INDEX} \mid \text{E} \mid \text{ASPECT } \textit{prog} \end{array} \right] \right\rangle \end{array} \right]$$
- c.
$$\left[\begin{array}{l} \text{VAL} \mid \text{COMPS} \left\langle \left[\begin{array}{l} \text{HEAD} \mid \text{FORM } \textit{form} \\ \text{INDEX} \mid \text{E} \mid \text{TENSE } \textit{future} \end{array} \right] \right\rangle \end{array} \right]$$

5.3.2 Modals

Modals refer to concepts such as possibility, ability, and necessity (Sneddon et al., 2010, p. 208). The main Indonesian modals are *dapat* “can”, *bisa* “can”, *boleh* “may”, *harus* “must”, and *perlu* “need”. Some modals can occur with adjectives and prepositions, as shown in Example (201).

- (201) a. *Polisi harus tinggi.*
 police must tall
 “Policemen must be tall.”
- b. *Saya boleh ke sana.*
 1SG may to there
 “I may go there.”

Alwi et al. (2014, p. 164-165), Macdonald (1976, p. 93) mention that multiple auxiliaries can occur in a clause. Example (202) shows the head verb *menyelesaikan* “solve” is modified by *akan* “FUT”, *harus* “must”, and *dapat* “can”.

- (202) *Dia akan harus dapat menyelesaikan soal itu segera.*
 3SG FUT must can ACT-solve problem that soon
 “S/he will have to be able to solve that problem soon.” (Alwi et al., 2014, p. 165)

All modals in INDRA belong to a lexical type *modal-aux-lex*. Its supertype is *subj-raise-aux-with-pred*, as shown in (203), which inherits from *subj-raise-aux*, *norm-sem-lex-item*, and *trans-first-arg-raising-lex-item-1*. The *modal-aux-lex* does not add any information on the FORM of the phrase with modals in order to enable multiple modals to co-occur in a sentence.

- (203)
<i>subj-raise-aux-with-pred</i>
<i>modal-aux-lex</i>
[VAL COMPS < [HEAD FORM <i>form</i>] >]

Parse tree of Example (202) is shown in (204). The HEAD-COMP rule combines all auxiliaries, both temporal markers and modals, with their complements. Its DMRS is shown in (205). Only the modals introduce semantic predicates. Their ARG1 is *req*-ed to the LBL of the corresponding complements.

(204) a.

b.

(205)

5.4 Negation

The negation word *tidak* (and its variant *tak* which has a literary flavor) negates all types of predicate other than nouns, including verbs, adjectives, and prepositional phrases (Sneddon et al., 2010, pp. 202-203) (see also Section 6.3 on page 177). It takes the whole clause as its scope and it can also modify a single predicate (Kroeger, 2014), as shown in Example (206).

- (206) a. *Dia tidak boleh hadir.*
 3SG NEG may present
 “He is not allowed to be present.” (Sneddon et al., 2010, p. 208)
- b. *Dia boleh tidak hadir.*
 3SG may NEG present
 “He is allowed to be absent.” (Sneddon et al., 2010, p. 209)

The meanings of the sentences in (206) can be represented as follows:⁹

(206a) TIDAK BOLEH [dia hadir]

(206b) BOLEH TIDAK [dia hadir] or BOLEH [dia [TIDAK hadir]]

The negation word *tidak* “NEG” belongs to a lexical type *neg-adv-lex* which inherits from *basic-scopal-adverb-lex*, as shown in (207). It modifies a phrase whose HEAD is of type *+vjp*, i.e. *tidak* “NEG” can negate verb phrases, adjective phrases, and prepositional phrases. It precedes the negated phrase, so the POSTHEAD is minus. The negated phrase’s ASPECT value is of type *perf-and-prog*, so it cannot negate copulas such as *adalah* “COP” and *ialah* “COP”.

⁹I thank one of my external examiners for pointing this out.

Semantically, the ARG1 of *tidak* “NEG” is *qeq*-ed with the LBL of the predicate. DMRS of Example (206a) is shown in (208) and the one of Example (206b) is shown in (209).

Besides *tidak* and *bukan* (see Section 6.3 on page 177), there is another negation word *belum* which corresponds to “not yet” in English. It combines the meanings of *tidak* “NEG” plus temporal marker *sudah* “PRF” and it is regarded the same as a substitute for **tidak sudah* (Sneddon et al., 2010, p. 202; Alwi et al., 2014, p. 166), similar to *bukan* which is analyzed as a substitute for **tidak adalah/ialah* (see Section 6.3 on page 177 on copula negation).¹⁰

¹⁰Similar to copula *adalah* and *ialah* which can precede the negation word *tidak*, *sudah* “PRF” can precede *tidak* and *bukan*, the combination indicating that what follows is no longer the case (Sneddon et al., 2010, pp. 209-210).

- (210) a. *Dia belum profesor.*
 3SG not.yet professor
 “He is not a professor yet.” (Sneddon et al., 2010, p. 209)
- b. *Mereka belum berangkat.*
 3PL not.yet depart
 “They have not left yet.” (Sneddon et al., 2010, p. 209)

In INDRA, the negation word *belum* “not.yet” is of lexical type *neg+perf-adv-lex* which inherits from *neg-adv-lex*. It contributes the value of the ASPECT of the negated phrase to be of type *perf*.

5.5 Adverbs

Adverbs are words used to modify verbs, adjectives, other adverbs, or the whole sentence (Alwi et al., 2014, p. 203; Liaw, 2004, p. 119). Adverb modification can be intersective or scopal.

From a lexical-semantic viewpoint, Indonesian intersective adverbs can be grouped into several categories:

- Adverbs of manner, e.g. *cepat-cepat* “quickly”
- Adverbs of time, e.g. *besok* “tomorrow”
- Adverbs of place, e.g. *kemari* “hither”
- Adverbs of frequency, e.g. *kadang-kadang* “sometimes”, indicating the frequency with which an action occurs (Sneddon et al., 2010, p. 227)
- Adverbs of degree, e.g. *sangat* “very”, which occur with adjectives (see Section 4.5 on page 100 on adjectives)

Scopal adverbs typically function on the clause or discourse level, i.e. their semantic effect (scope) is relevant to entire clauses or larger units rather than simply to phrases (Payne, 2008, p. 69). Example (211) illustrates intersective and scopal adverbial modification in one sentence.

- (211) *Dia mungkin akan segera menyelesaikan soal itu.*
 3SG maybe FUT soon ACT-solve problem that
 “S/he maybe will solve that problem soon.” (based on Alwi et al., 2014, p. 165)

Semantically, both intersective and scopal adverbs are one-place predicates (both of them have LBL, ARG0, and ARG1). Their ARG0 is of type *event*. The difference is in their

LBL and ARG1. The ARG1 of intersective adverbs is restricted to events and coindexed with the ARG0 of the modified verbs, adjectives, or prepositions. The ARG1 of a scopal adverb is a handle, which is *qeq*-ed to the LBL of the modified predicate.

Parse tree of Example (211) is shown in (212) and its DMRS in (213). The *adj-head-int* rule combines the intersective adverb *segera* “soon” with the modified verb phrase and the *adj-head-scop* rule combines the scopal adverb *möglich* “maybe” with the modified verb phrase. The syntactic head daughter is the verb *menyelesaikan* “ACT-solve” and the semantic head daughter is the scopal adverb *möglich* “maybe”.

(212) a.

b.

(213)

5.6 Summary

This chapter details the handling of verbs in INDRA, including their subcategorization, inflectional rules, auxiliaries, negation, and adverbs. The subcategorization types of verbs together with their properties in each subcategorization pattern have been outlined. The inflectional rules for active voice, passive voice type one, passive voice type two without affixes, and passive voice type two with bound morphemes *ku-* “1sg” and *kau-* “2sg” in monotransitive and ditransitive clauses have been described. Auxiliaries (temporal markers and modals), negation, and adverbs have been analyzed and implemented in INDRA. Although the rules for passive voice have been implemented, passive with *ter-* and *ke-...-an* have not been analyzed and implemented yet. In addition, the passive voice rules at present only work with monotransitive and ditransitive verbs, not with control verbs such as *mencoba* “ACT-try”. Future work is to make other verbs such as control verbs be able to be passivized.

Chapter 6

Copula constructions

In this chapter I show how basic copula clauses in Indonesian are analyzed and implemented in INDRA. Three types of basic copula clauses in Indonesian were analyzed and implemented in INDRA: copula clauses with noun phrase complements (NP) expressing the notions of ‘proper inclusion’ and ‘equation’, adjective phrases (AP) expressing ‘attribution’, and prepositional phrases (PP) expressing relationships such as ‘location’.

In the following sections I introduce the three types of basic copula clauses in Indonesian (Section 6.1). I then give an analysis for each type (Section 6.2) and the negation for copula clauses (Section 6.3). Finally, I describe the generation (Section 6.4) and write a brief conclusion (Section 6.5). Sections of this chapter have appeared in Moeljadi et al. (2016).

6.1 Introduction

Every language has a copula clause type, which may take a copula verb (Dryer, 2007). Some languages lack a copula verb; the copula slot is left blank and we have ‘verbless clauses’. In addition, some languages have more than one kind of copula verb. Most commonly, one will just refer to ‘a state’ and the other to ‘coming into a state’, similar to *be* and *become* in English (Dixon, 2009, p. 175). Indonesian has multiple copula verbs, distributed over different semantic relations in addition to ‘verbless clauses’. I give an analysis that covers both multiple copula verbs and verbless clauses.

Analyses of Indonesian copulas can be found in reference grammars, such as Alwi et al. (2014), Mintz (2002), and Sneddon et al. (2010). A syntactic analysis in Lexical Functional Grammar (LFG) (Kaplan & Bresnan, 1982; Dalrymple, 2001) was done by Arka (2013a). However, to the best of my knowledge, no work has been done on modeling Indonesian copula clauses in HPSG (Pollard & Sag, 1994; Sag et al., 2003) and MRS (Copestake et al., 2005). This chapter aims to fill in this gap, referring to existing HPSG literature on copulas, such as Bender (2001) and Van Eynde (2009).

Table 6.1 shows an outline of the three types of basic copula clauses in Indonesian.

Relation	Subject	Predicate	
Proper inclusion	<i>Budi</i>	(<i>adalah</i>)	<i>guru</i> (NP)
	Budi	is	a teacher
Attribution	<i>Budi</i>	∅	<i>pandai</i> (AP)
	Budi	is	clever
Location	<i>Tempatnya</i>	(<i>adalah</i>)	<i>di rumah</i> (PP)
	place=DEF	is	at home

Table 6.1: Three types of basic copula clauses in Indonesian

All three types of basic copula clauses in Table 6.1 can appear without a copula verb. In fact, ‘attribution’ is typically expressed without a copula verb. The copula verb shown in Table 6.1 is *adalah*³ for both ‘proper inclusion’ and ‘location’. However, as mentioned before, there are more than one copula for some semantic relations. These other types will be discussed in the following subsections.

6.1.1 Copula clauses with noun phrase predicates

Copula clauses with an NP as predicate may or may not have a copula verb *adalah*, *ialah*,⁴ or *merupakan*⁵ (Alwi et al., 2014, p. 358-359). These clauses express the notions of ‘proper inclusion’ and ‘equation’. Indonesian does not distinguish these notions syntactically, as shown in Example (214a) and Example (214b). The three copula verbs behave the same way.

- (214) a. *Budi (adalah/ialah/merupakan) guru.*
 Budi COP teacher
 ‘Budi is a teacher.’

¹Copulas in Indonesian can be followed by VPs and CPs. They have not been analyzed and implemented in INDRA. Copula constructions with VP and CP predicates are for future work. I thank one of the external examiners for pointing this out.

²‘Proper inclusion’ is when a specific entity is asserted to be among the class of items specified in the nominal predicate, as in English sentence “He is a teacher”. Usually the subject is specific (“he”) and the nominal predicate is non-specific (“a teacher”). ‘Equation’ is when a particular entity is identical to the entity specified in the predicate nominal, e.g. “He is my father” (Payne, 2008, p. 114).

³adalah “COP” is derived from *ada* “EXIST” and a focus particle *-lah*.

⁴*ialah* ‘COP’ is derived from *ia* ‘3SG’ and a focus particle *-lah*.

⁵ *merupakan* is derived from a noun *rupa* ‘form, figure, appearance, sort’, prefix *meN-* ‘ACT’, and suffix *-kan* ‘APP’. The original meaning is ‘to form, to shape, to constitute’.

- b. *Budi (adalah/ialah/merupakan) guruku.*
 Budi COP teacher=1sg
 “Budi is my teacher.”

Since *ialah* is historically derived from *ia* “3sg”, it only occurs with a third person subject (Sneddon et al., 2010; Mintz, 2002). Example (215) shows that *saya* “1sg” cannot be the subject of a copula clause with *ialah*.

- (215) *Saya (adalah/*ialah/merupakan) guru.*
 1sg COP teacher
 “I am a teacher.”

The copula verb *merupakan* is a verb which is in the process of becoming a copula (see Footnote 5). At its present stage it cannot appear if the NP predicate is a specific referent, such as a proper name, demonstrative, or pronoun, as shown in Example (216). However, it can precede a unique referent NP with a definite marker or a possessive marker as shown in Example (214b).

- (216) *Orang itu (adalah/ialah/*merupakan) Budi.*
 person that COP Budi
 “That person is Budi.”

In addition, it can take an aspect or tense marker, while *adalah* and *ialah* cannot, as shown in Example (217a) and Example (217b).

- (217) a. **Ini sudah/akan adalah/ialah hal yang luar biasa.*
 this PRF/FUT COP case REL beyond ordinary
 (Intended meaning: “This has been/will be an extraordinary case.”) (based on NTU-MC sentence ID 11938)
- b. *Ini sudah/akan merupakan hal yang luar biasa.*
 this PRF/FUT COP case REL beyond ordinary
 “This has been/will be an extraordinary case.” (based on NTU-MC sentence ID 11938)

These have been confirmed in the Indonesian section of NTU-MC (Tan & Bond, 2012), a parallel corpus containing 2,975 translated sentences from three sources: Singapore Tourism Board website,⁶ a Sherlock Holmes short story “The Adventure of the Speckled Band”, and a Japanese short story “The Spider’s Thread” written by Akutagawa Ryunosuke.

⁶www.yoursingapore.com

To the best of my knowledge, there is no meaning difference among the three copula verbs. Sneddon et al. (2010), Alwi et al. (2014), Macdonald (1976), and Mintz (2002) note that *adalah* and *ialah* are interchangeable and most common in noun clauses where either the subject or predicate is long or structurally complex in formal, written language. Alwi et al. (2014) mention that *adalah* can also be changed with *merupakan*.

6.1.2 Copula clauses with adjective phrase predicates

Copula clauses which express the notion of ‘attribution’ are the ones which have an AP as the main semantic content and are called ‘predicate adjectives’ (Payne, 2008, p. 120-121). A copula is usually absent in predicate adjectives, as shown in Example (218a). As Sneddon et al. (2010, p. 246-247) note, a copula *adalah* may be used by some speakers in adjective clauses, as illustrated in Example (218b).

- (218) a. *Budi pandai.*
 Budi clever
 “Budi is clever.”
- b. *Pernyataan itu (?adalah/??ialah/*merupakan) benar.*
 statement that cop true
 “‘That statement is true.’ (based on Sneddon et al., 2010, p. 247)

According to my intuition, a copula *ialah* may be less commonly used than *adalah*, but *merupakan* cannot occur with predicate adjectives. In addition, a copula occurs if the subject NP in a sentence is fairly long and if a formal register is considered. Not all speakers agree with the grammaticality of this and I did not find any occurrence of predicate adjectives with copulas in the NTU-MC; further Arka (2013a, p. 31, 33) states that a copula cannot precede an adjective. Even so, I do provide an analysis for copula with adjective phrase predicates in INDRA.

6.1.3 Copula clauses with prepositional phrase predicates

Copula clauses which express the notion of ‘location’ are the ones which have a PP as the main semantic content and are called ‘predicate locatives’ (Payne, 2008, p. 121-123). An existential verb *ada* or *berada* may be used optionally in predicate locatives, as illustrated in (219a). The copulas *adalah* or *ialah* may appear, too, as shown in (219b). Both in (219a) and (219b), the PP is a complement, not an adjunct.

- (219) a. *Budi (ada/berada) di rumah.*
 Budi EXIST at home
 “Budi is at home.”

- b. *Satu-satunya air yang ada (adalah/ialah/*merupakan) dari telaga.*
 one-RED=DEF water REL EXIST COP from lake
 “The only water there is is from the lake.” (based on Sneddon et al., 2010, p. 247)

There is another ‘benefactive’ clause in which the main semantic content of the predication is realized in a PP, marked by a preposition *untuk* “for”, and its syntactic pattern usually follows the one of predicate locatives (Payne, 2008, p. 122).⁷ In Indonesian, an optional copula verb *adalah* or *ialah* may appear in this ‘benefactive’ clause, as shown in Example (220a).

- (220) a. *Ini (adalah/ialah/*merupakan) untuk Budi.*
 this COP for Budi
 “This is for Budi.”
- b. **Engkau ialah untukku.*
 2SG COP for=1SG
 (Intended meaning: “You are for me.”)
- c. *Presiden ialah untuk rakyat.*
 president COP for people
 “The president is for the people.”

Regarding *ialah*, for the same reason mentioned in Section 6.1.1, it can only appear with a third person subject. Example (220b) shows that *engkau* “2SG” cannot be the subject of *ialah*, while Example (220c) shows that 3SG subject *presiden* “the president” can be the subject.

6.2 Analysis

This section contains my analysis for each type of copula clause introduced in the previous section.

6.2.1 Copula clauses with noun phrase predicate

The copula verbs *adalah*, *ialah*, and *merupakan* take two arguments, syntactically similar to simple transitive verbs. Our analysis follows the Montagovian treatment as presented in Van Eynde (2009, p. 368), in the sense that this analysis treats the copula as a transitive verb, covering both the predicating and identifying uses.

⁷Copula clauses with PPs can express many meanings other than locatives and benefactives, e.g. source, goal, reason, and purpose. They have not been implemented in INDRA. I thank one of the external examiners for pointing this out.

In order to model the shared meaning of the various copulas, I use a simple type hierarchy, as illustrated in (221). The supertype of all the NP copulas *cop-verb-lex* inherits from *transitive-verb-lex* and *main-verb-lex* whose HEAD is of type *copula*, with a subject whose HEAD is of type *subj-noun* and an obligatory, saturated complement whose HEAD is also of type *subj-noun*. This then has two children. The copula *adalah* is an instance of *v_np_cop_noasp_le* which inherits from *cop-verb-lex* with an additional constraint: it cannot occur with any aspect or tense marker (see Example (217a)). The copula *merupakan* also inherits from *cop-verb-lex*, but with a different constraint: the head of the complement should be a common noun, not a proper name, a personal pronoun, or a demonstrative pronoun. The copula *ialah* (*v_np_cop_3_le*) inherits from *v_np_cop_noasp_le* with another constraint: the subject should be third person.

The MRS representation is the same as the one for transitive sentences, as shown in

(222).⁸ The predicate `_cop_v_ialah_rel` represents an event,⁹ its ARG1 has a constraint: the value of the `PNG.PERNUM` is *3sg*. So `named_rel("Budi")` (ARG1) must be third person, while there is no constraint on the ARG2, `_guru_n_rel`. The parse tree of Example (214a) with a copula *adalah* is shown in (223).

The attribute value matrix (AVM) for *zero-cop-rule* in (225) shows that it inherits from *unary-phrase* which employs *unary rule*, i.e. a rule with one daughter. In this rule, a pumping rule¹⁰ was made which pumps (or converts) an NP to a VP for zero copula clauses. Figure 224 shows the parse tree result. This pumping rule introduces a predicate `cop_v_zero_rel` with the subject as the first argument (ARG1) and the NP predicate as the second argument (ARG2), denoting a relation of coreference between the indices of the

⁸In the simplified version of the graph shown in this dissertation, properties of the predicates such as semantic type, aspect, tense and number are not shown. If they are important to the analysis they will be discussed in the text. Referential individuals will be in the restriction of a quantifier (shown with the link `RSTR/H`). All other predicates are events.

⁹Currently I do not distinguish between dynamic and stative meanings, referring to both as events.

¹⁰A unary rule that changes the type (Copestake, 2002, p. 120).

subject and the predicative complement. A new feature was added to the HEAD, called EMPTY. The NP predicate has EMPTY minus (-) and the value is changed to plus (+) after it changes to VP. This is needed to prevent every NP in constructions other than zero copula construction being pumped into VP. COORD is a boolean that is used to enforce the syntax of coordination. COORD must be minus for all phrases that are not participating in coordination. The MRS is similar to that produced by the copula verbs *adalah*, *ialah*, or *merupakan*.

This syntactic structure is similar to the one in Arka (2013a, p. 38) where any lexical category (VP, NP, AP, and PP) can be a predicate XP; the NP subject takes this XP to make an Indonesian clause. Our analysis corresponds to ‘Constructional analysis II’ in Bender (2001, pp. 101-118). There are three kinds of facts which make such an analysis unsuccessful to deal with African American Vernacular English (AAVE) copula absence:

(1) the possibility of copulaless existentials, (2) a curious interaction of negation and ellipsis, and (3) the possibility of complement extraction (Bender, 2001, p. 107), as shown in the examples below. These three things do not exist in Indonesian: Indonesian has an obligatory intransitive existential verb *ada*, compared with AAVE which has an NP *there* and a zero copula in existential sentences, as shown in Example (226);

(226) *Ada mobil yang menghalangi jalanku.*

EXIST car REL ACT-block way=1SG

“There a car blocking my way.” (AAVE based on Bender, 2001, p. 107)

Copula absence in AAVE is not possible if the complements of the copula are elided but it turns out that ellipsis is possible just in case it strands *not*, while Indonesian uses a negation marker *bukan*, treated as a non-modifier verb which does not occur with any copula,¹¹ as shown in Example (227).

(227) *Mereka berkata mereka adalah sahabat, tetapi mereka (*adalah) bukan.*

3PL say 3PL COP best.friend but 3PL COP NEG

“They say they’re best friends, but they (are) not.” (AAVE based on Bender, 2001, p. 115)

Finally, AAVE has a long distance dependency in which the complement of the silent copula can be extracted, as shown in (228a) and (228b) with the Indonesian translations. However, different from Bender (2001) who analyzes this as complement extraction, I analyze this as a copula clause in which the predicate is focused and precedes the NP subject.¹²

(228) a. *Di mana mobilmu?*

at where car=2SG

“Where your car?” (AAVE based on Bender, 2001, p. 117)

b. *Di mana menurutmu mobilmu?*

at where according.to=2SG car=2SG

“Where you think your car?” (AAVE based on Bender, 2001, p. 117)

In short, because of differences in syntactic structure and analysis, the constructional analysis which does not work for AAVE can be implemented for Indonesian.

¹¹see Section 6.3 for negation.

¹²Wh-question constructions have not been implemented in INDRA.

6.2.2 Copula clauses with adjective phrase predicate

As mentioned in Section 6.1.2, the predicate and the main semantic content of copula clauses with AP predicates is the AP. Predicative APs take one argument (NP as the subject), similar to intransitive predicates.¹³ Figure 229 shows the parse tree of Example (218a).

As mentioned in Section 6.1.2, a copula may or may not precede AP. In this section, I provide an analysis for copula + AP, too. The copula *adalah* is treated as a raising auxiliary which does not introduce a predicate and links its subject to the subject of its complement (the adjective). Figure 230 shows the parse tree of Example (218a) with *adalah*.

The MRS representation is the same as the one for intransitive sentences, as shown in (231). The MRS of the clauses with and without *adalah* are the same. The event, `_pandai_a_rel` is the semantic head and hook for composition. Its ARG1 is linked to the subject: `named_rel("Budi")`.

6.2.3 Copula clauses with prepositional phrase predicate

Predicate locatives have a PP as the main semantic content and an optional verb *ada* or *berada*, or a copula *adalah* or *ialah*. Predicative prepositions, such as *di* “in/on/at”, take two arguments, similar to transitive predicates, as shown in Figure 232.

¹³There is a lexical rule that converts these to attributive adjectives for the modifier use.

When appearing with PPs, I also treat *ada*, *berada*, *adalah*, and *ialah* as auxiliaries which do not introduce a predicate of their own. The head of the subject is a noun and the head of the complement is a preposition. Figure 233 shows the parse tree of Example (219a) with an existential verb *ada*.

The MRS of predicate locatives with *ada*, *berada*, *adalah*, and *ialah* is exactly the same as the one without, as shown in the dependency MRS representation in Figure 234.

In the MRS representation, the semantic head daughter and hook for composition is the event *_di_p_rel*. Its ARG1 and ARG2 are linked to *named_rel("Budi")* and *_rumah_n_rel* respectively.

Regarding ‘benefactive’ clauses, my analysis is the same as the one for predicate locatives. I treat *adalah* and *ialah* in these clauses as auxiliaries which do not introduce a predicate. The MRS (and DMRS) representation is similar to the one in (234).

6.3 Negation

Indonesian has two main negation markers for clauses, placed before the negated element. The following examples summarize the interaction of negation with copula verbs in Indonesian, for NP, AP, and PP, respectively. The standard negation marker *tidak* is used when the predicate is verbal, including the copula verb *merupakan* and existential verbs *ada* and *berada*, as shown in Example (235a), (235b), and (235c), or adjectival, as in Example (235e), and with PP predicates, as in Example (235d).

- (235) a. *Budi tidak *adalah/*ialah/merupakan guru.*
 Budi NEG COP teacher
 “Budi is not a teacher.”
- b. *Mereka tidak/*bukan menolong kami.*
 3PL NEG help 1PL.EXCL
 “They didn’t help us.” (Sneddon et al., 2010, p. 202)
- c. *Budi tidak/*bukan ada/berada di rumah.*
 Budi NEG EXIST at home
 “Budi is not at home.”
- d. *Tempatnya tidak/bukan di sini.*
 place=DEF NEG at here
 “The place is not here.”
- e. *Budi tidak/*bukan pandai.*
 Budi NEG clever
 “Budi is not clever.”

It cannot negate copula *adalah* or *ialah*, as illustrated in Example (235a), (236a), (236b), and (236c).

- (236) a. **Guru itu tidak adalah/ialah/merupakan Ali.*
 teacher that NEG COP Ali
 Intended meaning: “That teacher is not Ali.”
- b. **Budi tidak adalah/ialah/merupakan pandai.*
 Budi NEG COP clever
 Intended meaning: Budi is not clever.
- c. *Ini tidak/bukan *adalah/*ialah/*merupakan untuk Budi.*
 this NEG COP for Budi
 “This is not for Budi.”

In Example (236a), *tidak* is not compatible with *adalah* and *ialah*; also *merupakan* is ruled out because the NP predicate is a proper name (see also Example (216)).

The special negation marker *bukan* “be not” is used when the predicate is nominal, as in Example (237) (Kroeger, 2014, p. 137),¹⁴ or prepositional,¹⁵ as shown in Example (235d).

- (237) a. *Budi bukan/*tidak guru.*
 Budi NEG teacher
 “Budi is not a teacher.”
- b. *Guru itu bukan/*tidak Ali.*
 teacher that NEG Ali
 “That teacher is not Ali.” (Arka, 2011b, p. 85)

However, it cannot negate copula *adalah* or *ialah*, as illustrated in Example (236c), or existential verbs *ada* and *berada*, as in Example (235c). The AVM of *bukan* “NEG” is shown in (240). Sneddon et al. (2010, p. 202) note that a number of prepositions, including *karena* “because”, *untuk* “for”, and *seperti* “like” can be negated by either *bukan* or *tidak*.

I treat *tidak* as an adverb modifying VP, AP, or PP, as shown in Figure 238.

It is represented as *neg_rel* in the MRS. The value of its *ARG1* is equated with the *LBL* of the VP, AP, or PP predicate, as illustrated in (239). Here, the negation is represented by the scopal predicate *neg_rel*. The *ARG1* of this predicate is a handle, which is linked through via *HCONS* to the *LBL* of the preposition predicate.

¹⁴Kroeger (2014, p. 137) notes that in certain kinds of contexts, *bukan* can be used to negate verbal clauses and argues that it is a marker of ‘external’ (sentential) negation. I will not discuss it because this is beyond the scope of this chapter.

¹⁵Sneddon et al. (2010, p. 202) mention that a number of prepositions can be negated by either *bukan* or *tidak*.

(239)

In the AVM (240), I treat *bukan* as a non-modifier verb, a combination of *copula_v_rel* as the head (KEYREL) and *neg_rel* as the daughter (ALTKEYREL), which takes an NP subject and an NP or PP complement, as shown in Figure 241 and 242.

adalah and *ialah*'s ASPECT is *non-perf-and-prog*, which means they cannot take aspect markers, *tidak* and *adalah* or *ialah* are not compatible.

6.4 Generation

I model similarities, in this case of meaning, using a type hierarchy, as illustrated in (243).

I can use this to underspecify the input to the generator. For example, for `copula_v_rel` it will then try to generate all predicates that are subsumed by it, i.e. all copula constructions, and only succeed for the grammatical ones.

Input:

Budi merupakan guru

Output:

Budi guru
 Budi ialah guru
 Budi adalah guru
 Budi merupakan guru
 ...

Input:

Budi ada di rumah

Output:

Budi di rumah
 Budi ada di rumah
 Budi berada di rumah
 Budi adalah di rumah
 ...

6.5 Summary

My analyses of Indonesian copula clauses are similar to Arka (2013a)'s LFG analysis but cover more copula verbs with a refined type hierarchy. Because of differences in syntactic structure between AAVE and Indonesian, the analysis that builds a VP out of a predicative NP, which does not work for AAVE, can be successfully implemented for Indonesian. Copula clauses with (nominalized) VP and CP predicates have not been analyzed and implemented in INDRA. They are for future work.

Chapter 7

Compounds

In the previous chapter I have talked about basic copula clauses in INDRA with NP, AP, and PP complements. In this chapter I will show how compounds in Indonesian can be analyzed and implemented in INDRA. A compound is a combination of two or more simple words which come together to form a complex word, it may be a compound noun, a compound adjective, or a compound verb (Sneddon et al., 2010, p. 26; Payne, 2008, p. 92; Liaw, 2004, p. 278). Three types of compounds in Indonesian were analyzed and implemented in INDRA: Verb-Verb compound (or Serial Verb Constructions), Noun-Noun compound, and Adjective-Adjective compound. Idiomatic compounds were listed in the lexicon and were not analyzed based on their components.

In the following sections I explain how I dealt with idiomatic compound (Section 7.1), Verb-Verb compound (Section 7.2), Noun-Noun compound (Section 7.3), and Adjective-Adjective compound (Section 7.4) in Indonesian. Analysis for each type of compound is given in each section and Section 7.5 summarizes. Section 7.2 discussing about Verb-Verb compound or serial verb constructions will appear in Moeljadi & Ow (2018).

7.1 Idiomatic compound

The term ‘idiomatic compound’ in this section refers to ‘compound’, including both ‘idiomatic compound’ and ‘semi-idiomatic compound’, also some ‘non-idiomatic compound’ in Kridalaksana (1989) and ‘compound with idiomatic meaning’ in Sneddon et al. (2010).

Kridalaksana (1989, pp. 107-108) gives definitions and examples as follows:

- Idiomatic compound is a construction, of which the whole meaning is not the same as the meanings of its components, e.g. *buah bibir* “topic of conversation” (lit. “lip fruit”), *bulan madu* “honeymoon”, *tanah air* “fatherland” (lit. “water land”)
- Semi-idiomatic compound is a construction, in which one of its components has a special meaning which exists only in that construction, e.g. *kereta api* “train” (lit.

“fire wagon”), *harga diri* “self-esteem” (lit. “self price”), *mata kaki* “ball of ankle” (lit. “foot eye”)

- Non-idiomatic compound, e.g. *anak cucu* “descendant” (lit. “child(ren) (and) grand-child(ren)”), *kereta kuda* “horse carriage”, *gaji pokok* “basic salary” (lit. “base salary”).

Regarding the characteristics of compounds, Kridalaksana (1989, pp. 104-105) notes that nothing can be inserted between components in compounds, e.g. *kereta api* “train” (lit. “fire wagon”), while a common phrase such as *orang Indonesia* “Indonesian” (lit. “Indonesia person”) or *buku sejarah* “history book” can be inserted with *dari* “from”, e.g. *orang dari Indonesia* “person from Indonesia”, or *tentang* “about”, e.g. *buku tentang sejarah* “book about history”.¹

Sneddon et al. (2010, p. 27) note that some compounds have an idiomatic meaning, e.g. *rumah sakit* “hospital” (lit. “sick house”). They are ambiguous, either compounds with idiomatic meaning, or just phrases with no idiomatic meaning, depending on the context. The combination can only be separated by *yang* “REL” if it is not a compound, e.g. *kamar kecil* “toilet, small room” and *kamar yang kecil* “small room”, *orang tua* “parents, old people” and *orang yang tua* “old people” (Sneddon et al., 2010, p. 151).

In INDRA, these idiomatic compounds are listed as part of the lexicon in `lexicon.tdl`. For example, *kamar kecil* “toilet” is an inanimate noun (see Section 4.1) and thus it belongs to *inanimate-noun-noun-lex* and is treated as a single unit, having a PRED name `_toilet_n_rel`, same as the one for the inanimate noun *toilet* “toilet”, as shown in (244).

(244)

```
kamar+kecil := inanimate-noun-noun-lex &
[ STEM < "kamar", "kecil" >,
  SYNSEM.LKEYS.KEYREL.PRED "_toilet_n_rel",
  TRAITS native_token_list ].
```

Since both *kamar* “room” and *kecil* “small” are also listed in the lexicon, INDRA gives two parse trees for the NP *kamar kecil* “toilet, small room” and hence can tell that it is ambiguous, as shown in (245) and (246).²

¹The term common phrase used by Kridalaksana (1989) is what I regard as Verb-Verb compound, Noun-Noun compound, and Adjective-Adjective compound in the next sections.

²Actually, INDRA gives three parse trees. Apart from the idiomatic and non-idiomatic NP, INDRA also analyzes it as a sentence meaning “rooms are small”.

7.2 Verb-verb compound (serial verb construction)

This section discusses syntactic and semantic properties of Verb-Verb compound or Serial Verb Constructions (hereafter SVCs) in standard Indonesian. Analyses of Indonesian SVCs can be found in Englebretson (2003, pp. 127-151), as well as in reference grammars such as Sneddon et al. (2010, pp. 279-286). A syntactic analysis for some SVCs in LFG (Kaplan & Bresnan, 1982; Dalrymple, 2001) was done by Arka (2000). However, to the best of my knowledge, no work has been done on modeling Indonesian SVCs within the HPSG framework (Pollard & Sag, 1994) and MRS (Copestake et al., 2005). This section aims to fill in this gap, referring to existing HPSG literature on SVCs, such as Müller & Lipenkova (2009). As for the data source, I employ the Indonesian section of NTU-MC (Tan & Bond, 2012), a parallel corpus of English, Chinese, Japanese, and Indonesian. A Python script was written to extract the Indonesian SVCs in NTU-MC. The HPSG analysis is implemented and tested in the INDRA. Some parts, especially the Python script written to extract SVCs from NTU-MC and the PARSEME guidelines, were done by Viola Ow, and described in Moeljadi & Ow (2018).

7.2.1 Introduction

I follow the term *serial verb constructions* in Englebretson (2003). Englebretson (2003, pp. 128-133) defines serial verbs in Indonesian as ‘at least two adjacent verbs without intervening material, which refer to closely related events, occur in the same intonation unit, and share at least one argument’. A single intonation unit refers to ‘the intonation properties of a single verb clause, and not a sequence of clauses’ (Aikhenvald, 1999, p. 470). Three criteria are established by Englebretson (2003) as a characterization of prototypical serial verb constructions in Indonesian, namely occurring contiguously, encompassing a single intonation unit, and sharing at least one argument.

The Serial Verb Constructions (SVCs) have been described by many authors. They are a type of syntactic feature with a sequence of two or more juxtaposed verbs in a single monoclausal structure, neither of which is an auxiliary, referring to a single (possibly

complex) event, which are not separated by any conjunctions, share at least one semantic argument, and has a single intonation contour (Kroeger, 2004; Aikhenvald & Dixon, 2006). This phenomenon has been mentioned in various ways in many reference grammars of Indonesian. Sneddon et al. (2010, p. 279) note that some Indonesian verbs can have a verbal clause as a complement. The main verb and the complement verb have the same subject. Alwi et al. (2014, pp. 169-170) mention that some verbs and verb phrases can function as complements (*pelengkap*) or adjuncts (*keterangan*) in verbal predicate sentences. Mintz (2002, pp. 381-383) notes that some verb sequences in Indonesian lack conjunctions or prepositions meaning “to” or “for” in between, thus appear side by side, while in English prepositions are required. Payne (2008, p. 308) mentions three formal factors for serial verb constructions:

1. There is no independent marking of the subject of the second verb
2. There is no independent tense or aspect marking of the second verb
3. The intonation is characteristic of a single clause

SVCs encompass a broad range of semantic relationships which is strictly inferential, based on verb meaning and context. There is no overt syntactic marking, conjunction, or other morpheme to indicate the semantic relation between the verbs. The meaning of SVCs is determined by both semantic compositionality and extra-lexical meaning components. Kroeger (2004) states that different languages impose different restrictions as to which specific combinations of verbs are permissible, and that these restrictions are sometimes due to cultural factors.

It may be more proper and less misleading not to use the term SVC, but to use a more inclusive term such as *verb series*. However, this section follows the term *serial verb constructions* in Englebretson (2003), focuses on the surface form of the constructions, and considers constructions with juxtaposed verb predicates as SVCs. However, different from Englebretson (2003), I analyze SVCs in the standard ‘High’ variety of Indonesian. The previous work done by Englebretson (2003) uses a corpus of colloquial ‘Low’ variety of Indonesian (see Section 2.1.1 on page 7 regarding the ‘High’ and ‘Low’ variety of Indonesian). Section 7.2.2 contains mainly the classification of Indonesian SVCs done by Englebretson (2003) with additional data from other previous works, such as Arka (2000). The process of extracting Indonesian SVCs from the Nanyang Technological University Multilingual Corpus (NTU-MC) (Tan & Bond, 2012) and some extracted data are shown and explained in Section 7.2.3. I propose my analyses in Section 7.2.4.

7.2.2 Previous works

Analyses of Indonesian SVCs can be found in Englebretson (2003), Polinsky & Potsdam (2008), as well as in reference grammars such as Sneddon et al. (2010), as mentioned in

In this section, I summarize Englebretson (2003)'s analysis based on semantic relations and give my syntactic analysis in terms of headedness and types of verbs or transitivity. Englebretson (2003) classifies SVCs in Indonesian into two big groups, i.e. 'serial verb as putative complements' and 'serial verbs with other semantic relationships'. I add one more group, i.e. 'other semantic relationships not mentioned in Englebretson (2003)'.

1. *Serial verbs as putative complements.* This is the most common type of Indonesian SVCs. The second verbs in the SVCs (V2s) are reduced complements and the first verbs (V1s) are *complement-taking predicates* (CTP) which semantically can be classified as *modality verbs* in Englebretson (2003) or *commitment verbs* and *orientation verbs* in Arka (2000).³

Arka (2000) notes that the commitment verbs are those in which the committer commits himself or herself to bring about some state of affairs. For example, *mencoba* "try", *menolak* "refuse", *berusaha* "attempt", and *mulai* "begin". The orientation verbs are typically experiencer verbs. For example, *mau* "want", *ingin* "desire", *berhak* "to have rights", *perlu* "need", *suka* "like (to do something)", and *bisa/tahu* "know how to". Arka (2000)'s commitment verbs and orientation verbs are classified as modality verbs in Englebretson (2003).⁴

These are at the top or close to the top of the binding hierarchy defined by Givón (1980) and tend to be tightly integrated, reflecting a close conceptual bond between events. According to Givón (1980), the binding hierarchy refers to the strength of the semantic bond between two events, which contributes to the syntactic integration of the two events into a single event. Modality and manipulative verbs are at the top of this binding hierarchy, and perception-cognition-utterance verbs are lower in the hierarchy. The following is one example. In this and all subsequent examples, serial verbs are underlined.

(247) Budi mencoba mengejar Adi.

Budi ACT-try ACT-chase Adi

"Budi tries to chase Adi."

³Englebretson (2003) also includes *manipulative verbs* like *suruh* "order" and *paksa* "force" as CTP which correspond to the *influence verbs* in Arka (2000). Arka (2000) states that they are characterized by having the influenced argument, i.e. the person asked, as controller. Sneddon et al. (2010, p. 281) notes that the object of the main verb is simultaneously subject of the complement verb. Thus, the verbs are not juxtaposed in active voice because the object of V1 (or the subject of V2) appears between V1 and V2. I will not discuss this in this dissertation since at this stage I limit my research to SVCs with juxtaposed non-passive verbs. SVCs with passive voice are for future work.

⁴Englebretson (2003) includes *malas* "lazy" as a modality verb. On the other hand, Sneddon et al. (2010, pp. 285-286) regard this as one of a limited number of adjectives which can take an active verb as a complement. This is also for future work.

This construction in Example (247) above is regarded as a control construction in Arka (2000) which can be defined as a relation of referential dependence between an unexpressed argument in an embedded clause (controlled argument) and an expressed or unexpressed argument (the controller) in a matrix clause. In Example (247), the unexpressed subject of *mengejar* “ACT-chase”, i.e. Budi, is said to be controlled (or shared). The head of this construction is V1 which belongs to a group called *control verbs*. V2 can be intransitive, e.g. *Budi mencoba tidur* “Budi tries to sleep”.

In addition, Arka (2000) mentions another complement clause called *raising*, in which the argument that is ‘thematically’ associated with an embedded clause is syntactically expressed as the argument of the matrix verb, where the matrix verb does not assign any thematic role to the ‘raised’ argument. The following is an example of raising. V1 is the head, it belongs to *raising verbs*. V2 can be intransitive, e.g. *Budi tampak tidur* “It appears that Budi sleeps”.

- (248) *Budi tampak mengejar Adi.*
 Budi appear ACT-chase Adi
 “It appears that Budi chases Adi.”

2. *Serial verbs with other semantic relationships*. Englebretson (2003) states that verb serialization also encodes four other relationships, i.e. manner, purpose, causation, and coordinated action. The relationships between the verbs must be inferred on the basis of verb semantics and context.

- (a) *Manner serialization*. V2 expresses how V1 is done (V1 is the head). Example (249) shows that V1 is intransitive and V2 is transitive. V2 can be intransitive, e.g. *Budi pergi bersepeda* “Budi goes by riding a bicycle”. However, V1 cannot be transitive because the object of V1 may appear before V2, e.g. *Budi mengejar Adi menggunakan mobil* “Budi chases Adi by using a car”.

- (249) *Budi berjalan menggunakan tongkat.*
 Budi walk ACT-use stick
 “Budi walks using a stick.”

- (b) *Purpose serialization*. This type denotes mandatory temporal sequence. V1 enables V2 to happen, and V1 must be done first (V1 is the head). In Example (250), V1 is intransitive and V2 is transitive. Intransitive verbs can fill in the position of V2, e.g. *Budi pergi tidur* “Budi goes (away) to sleep”. Transitive verbs cannot fill in the position of V1 for the same reason mentioned in *manner serialization* above.

(250) *Budi pulang mengambil uang.*

Budi go.home ACT-take money

“Budi goes home to get money.”

(c) *Periphrastic causative construction.* V1 constructs an individual complex predicate with V2, as illustrated in Example (251). This is usually non-standard in Indonesian and does not occur in formal Indonesian, except for the lexicalized *beri tahu* “inform” (*beri* “give”, *tahu* “know”). Since this type is periphrastic (non-standard) and this dissertation only deals with the standard Indonesian, I will not discuss this type further. In addition, I did not find any example of this type in the data in Section 7.2.3.

(251) *Budi kasih mati mereka.*

Budi give die 3PL

“Budi kills them.”

(d) *Coordinated actions.* V1 and V2 occur rapidly and repetitively, appearing to be simultaneous, as shown in the following example. The order of V1 and V2 is interchangeable, e.g. *Mereka mengeroyok memukul Budi* “they gang up and beat up Budi”. In Example (252), both V1 and V2 are transitive verbs. V1 and V2 can be intransitive, e.g. *Budi tertawa bertepuk tangan* “Budi laughs and claps hands”, also *Budi bertepuk tangan tertawa* “Budi claps hands and laughs” (*bertepuk tangan* “clap hands” is an intransitive verb in Indonesian).

(252) *Mereka memukul mengeroyok Budi.*

3PL ACT-beat ACT-gang.up Budi

“They beat up and gang up Budi.”

3. *Other semantic relationships not mentioned in Englebretson (2003).* In addition to the above semantic relationships, Alwi et al. (2014, p. 170) notes the following example which has an ‘origin’ or ‘source’ meaning.

(253) *Budi pulang bertamasya.*

Budi go.home have.a.picnic

“Budi goes home from picnic.”

Example (253) is ambiguous. Depending on the context, Example (253) can be interpreted as *purpose serialization* and thus the meaning is “Budi goes home to

have a picnic”. Regarding the headedness and transitivity of V1 and V2, the behavior is the same as the one in *purpose serialization*.

There is also another construction which has a *resultative* meaning, as shown in the following example. Kridalaksana (1989, p. 139) mentions that this type is a combination of an action (predicated by V1, the head) and a result (predicated by V2) which is caused by that action. In this construction, V1 is transitive and V2 is intransitive, as shown in Example (254), which consists of an action *Budi membunuh Adi* “Budi kills Adi” and a result *Adi mati* “Adi dies”.

- (254) *Budi membunuh mati Adi.*
 Budi ACT-kill die Adi
 “Budi kills Adi until Adi dies.”

Englebretson (2003) mentions that SVCs in Indonesian may appear with connectors between the verbs, e.g. *untuk* “in order to” which may appear after a manipulative or modality verb in the first type above. However, he notes that while *untuk* potentially indicates some sort of relationship between a series of verbs, how to characterize the relationship is not clear and he found only three examples of *untuk*, compared with 517 SVCs in his corpus with no connective.

Englebretson (2003) also states that there is apparent grammaticization of some of the Indonesian SVCs into auxiliaries marking aspect and modality, e.g. *mau* “want” as a modality verb in SVC type one and as a temporal auxiliary marking future. Polinsky & Potsdam (2008) analyze *mau* followed by a passive predicate which generates an unexpected, ambiguous interpretation. Sneddon et al. (2010, pp. 280-281) also note that *mau* with a passive complement raises ambiguity as to whether it is the subject or the agent of the complement who wants the action to occur. This is an interesting phenomenon. How to analyze and model this in HPSG as well as the implementation is left for future work.

7.2.3 Indonesian data

A Python script was written to extract SVCs from a Sherlock Holmes short story and a Japanese short story in the Indonesian database in the Nanyang Technological University — Multilingual Corpus (NTU-MC) Tan & Bond (2012), a parallel English-Chinese-Japanese-Indonesian corpus containing 2,975 sentences from three sources: Singapore Tourism Board website (www.yoursingapore.com), a Sherlock Holmes short story “The Adventure of the Speckled Band”, and a Japanese short story written by Akutagawa Ryunosuke: “The Spider’s Thread”. The Sherlock Holmes short story and the Japanese short story are originally written in English and Japanese, respectively, and translated into

standard Indonesian. Both the original texts and the translations are part-of-speech (POS) tagged.

The NTU-MC data is organized according to sentence IDs (SID), word IDs (WID) for V1, WID for V2, V1, V2, parts-of-speech (POS), and sentence. After extracting all possible Indonesian SVCs, the PARSEME annotation guidelines were used to determine if an extracted SVC is an SVC. After which, SVCs following the patterns observed were automatically tagged and the remaining complex verbs were manually tagged. Automatic tags were then hand checked for errors.

Extracting SVCs

The PARSEME annotation guidelines are state-of-the-art for annotation of verbal multi-word expressions (Candito et al., 2016). They are written with the assumption that a person, not a computer, is doing the annotation. The annotators can use the guidelines to write scripts to automate the annotation. Since PARSEME does not have specific guidelines for Indonesian, the guidelines for English were modified and suggested PARSEME guidelines were written for the extraction and identification of Indonesian SVCs to ensure consistent annotation and identification.

These suggested PARSEME guidelines for Indonesian SVCs were written based on the findings in Section 7.2.2. The guidelines describe a series of tests to identify and classify Indonesian SVCs (see Table 7.1).

1. If V1 is a *control verb* like *ingin* “desire”, *mau* “want”, *mencoba* “try”, *tahu* “know”, and *mampu* “be able to”, and the meaning of V1+V2 is compositional, V1+V2 is a *control SVC*.
2. If V1 is a *raising verb* like *terlihat* “seems”, *terasa* “feels like”, and *tampak* “looks”, and the meaning of V1+V2 is compositional, V1+V2 is a *raising SVC*.
3. If V2 expresses the manner of V1, and V1 is the head of the SVC, V1+V2 is a *manner SVC*.
4. If V2 indicates the purpose of V1, V1 is the head of the SVC, and V1 and V2 are bound by a temporal sequence (V1 happens before V2), V1+V2 is a *purpose SVC*.
5. If V1 and V2 occur rapidly and seem to be simultaneous actions, and the meaning of V1+V2 is compositional, V1+V2 is a *coordinated action SVC*.
6. If V2 happens before V1 and V1 is the head of the SVC, V1+V2 is a *source SVC*.
7. If V1 indicates an action, V2 is the result of V1 (V1 happens before V2), and V1 is the head of the SVC, V1+V2 is a *resultative SVC*.

8. If *untuk* “for/to” can be inserted between V1 and V2, V1+V2 is a *control* or *purpose* SVC, where *control* SVCs are not constrained by temporal sequence, but *purpose* SVCs require the V1 to occur before V2.
9. If a pronoun like *dia* “3sg” can be inserted between V1 and V2, V1+V2 is *NOT an* SVC because it does not fulfill two of the three criteria for prototypical SVCs, i.e. occurring contiguously and encompassing a single intonation unit, as mentioned in Section 7.2.1. For example,

(255) *Dia mengatakan (dia) mau ke kota...*
 3SG ACT-say (3SG) want to city
 “He says he wants to go to the city...” (SID: 10255)

10. If a complementizer like *bahwa* “that” or *apakah* “whether” can be inserted between V1 and V2, where V1 is a *saying* verb like *mengatakan* “ACT-say” or an *asking* verb like *menanyakan* “ACT-ask”, V1+V2 is *NOT an* SVC, for the same reason mentioned above. For example,

(256) *Dia mengatakan (bahwa) (dia) mau ke kota...*
 3SG ACT-say (that) (3SG) want to city
 “He says that he wants to go to the city...” (SID: 10255)

All possible Indonesian SVCs are where two words tagged as “verb” appear beside each other in the corpus. Following the suggested PARSEME guidelines for Indonesian SVCs, a script was written to semi-automatically assign the tags. First, each sentence in the corpus was checked whether the POS tag V and V occur side-by-side, afterwards it was checked whether V1 is a raising or a control verb, if V1 is neither a raising nor a control verb, the tags were manually checked and assigned. Instances where V1s or V2s turn out to be segmentation errors, nouns, adverbs, or prepositions that were wrongly tagged in the corpus were then marked and tagged as “NOT VV”. Semi-automatically tagged SVCs were then checked for errors.

Result

Out of 45 candidates which have the POS tags V-V side-by-side and can be automatically extracted, only 29 are SVCs. Out of the 29, 21 are *control* SVCs, three are *manner* SVCs, two are *purpose* SVCs, two are *raising* SVCs, and one is a *coordinated action* SVC. Regarding the remaining 16 candidates, 13 of them have incorrect POS tags (e.g. the word *masalah* “problem” was tagged V, it should have been tagged N), two of them are incorrectly segmented (have segmentation errors), and the rest (one candidate) is the one in

Type	Tag	V1	V2
Control SVC	control	control verb	(in)transitive
Raising SVC	raising	raising verb	(in)transitive
Manner SVC	manner	intransitive	(in)transitive
Purpose SVC	purpose	intransitive	(in)transitive
Coord. action SVC	coord. action	(in)transitive	(in)transitive
Source SVC	source	intransitive	(in)transitive
Resultative SVC	resultative	transitive	intransitive

Table 7.1: Indonesian SVCs and the corresponding tags

Example (256), its V1 is a *saying* verb. Table 7.2 shows the distribution of Indonesian SVCs from the extracted data. All the extracted SVCs in the data comprise two juxtaposed verbs. SVCs having more than two juxtaposed verbs were not found in the corpus.

A test-suite containing the 29 SVC sentences was made.⁵ The sentences were slightly edited to accommodate INDRA, focusing on the SVCs only.

Tag	Number	Percentage
control	21	72.4%
raising	3	10.3%
manner	2	6.9%
purpose	2	6.9%
coordinated action	1	3.4%
source	0	0.0%
resultative	0	0.0%
Total	29	100.0%

Table 7.2: Distribution of Indonesian SVCs in the corpus

For *control* SVCs, V1s are the head of the SVCs, and are control verbs like *jadi* “manage (to)”, *ingin* “wish”, and *bermaksud* “intend”. The V2s attached to these V1s are complements, and the meaning of V1+V2 is compositional for all control SVCs in the data, as shown in the following example.

(257) ...*Holmes* ...*mencoba membuka palang itu*...

Holmes ACT-try ACT-open shutter that

“...Holmes ...tries to open that shutter ...” (SID: 10417)

For Indonesian *manner* SVCs, V2s indicate the manner and direction of V1. For example, in *pulang* “return home” + *melalui* “pass through”, the act of returning home is done by passing through a place, as illustrated in Example (258).

⁵The test-suite can be accessed in INDRA repository in GitHub: <https://github.com/davidmoeljadi/INDRA/blob/master/testsuites/SVC.txt>

- (258) ...*saya pulang melalui halaman yang berumput itu.*
 1sg return.home pass.through yard REL grassy that
 "...I go home by passing through that lawn." (SID: 10500)

For the *raising SVC* category, all V1s are raising verbs and are the heads of each SVC, and V2s are complements. The meaning of V1+V2 is compositional, as shown in the following example.

- (259) *Waktu terasa berlalu dengan lambat sekali.*
 time feel pass with slow very
 "It feels that time passes very slowly." (SID: 10585)

For *purpose SVCs*, V1 is the head of the SVC and V2 indicates the reason for doing V1, and V1 has to happen before V2. The meaning of every V1+V2 is compositional. For example, in *bersiap* "prepare" + *pergi* "go", the act of preparing was done for the purpose of going somewhere, as shown in the following example. The other purpose SVC found in the corpus is *menggapai-gapai* "reach out" + *mencari* "search". However, one can argue that *bersiap pergi* "prepare to go" is a control SVC and *menggapai-gapai mencari* "reach out and search" is a coordinated action SVC. Some SVCs have ambiguous semantic relations (see also Example (253)) and I keep in mind about this when making my analysis in Section 7.2.4.

- (260) ...*Holmes ...bersiap pergi.*
 Holmes prepare go
 "...Holmes ...is prepared to go." (SID: 10587)

The only *coordinated action SVC* in the extracted data is *berlari* "run" + *menuju* "head towards", meaning "run and go towards". V1 and V2 happen rapidly and repetitively to describe the seemingly simultaneous action of running and going towards somewhere.

- (261) *Saya segera berlari menuju kamar ayah tiri kami ...*
 1sg soon run head.towards room father step- 1PL.EXCL ...
 "I soon ran towards our stepfather's room ..." (SID: 10193)

7.2.4 Analysis and computational implementation

This section describes an HPSG analysis of Indonesian SVCs and its implementation. As mentioned in Section 7.2.2, my analysis departs from Englebreton (2003) who classifies semantic relationships of Indonesian SVCs into two big groups, i.e. serial verbs as putative complements and serial verbs with other semantic relationships, as shown in Figure 7.1. Based on Arka (2000), I further divide the first group into control and raising constructions.

Figure 7.1: Analysis of Indonesian SVC

Serial verbs as putative complements

This group contains control and raising constructions, following Arka (2000). Control verbs assign a semantic role to their subject, while raising verbs do not. Sag et al. (2003, p. 376) also state that subject control verbs express a relation between an individual and a situation, while subject raising verbs express properties of situations.

Control SVC

The head daughter of this construction is V1. V1's predication involve two things, the first one is the subject (a noun) which is the shared argument of V1 and V2 and the second one is some situation or state of affairs predicated by V2 (a verb). V2 is a semantic argument of V1's predication. Its INDEX is linked to both V1 and V2. V2's semantic external argument (XARG) is identified with its subject's semantic index. That index is identified with the ARG1 in the lexical relation introduced by the verb. This is illustrated in (262). I put a constraint [AUX -] on V2 because V2 cannot take auxiliaries or aspect markers. See also (263) for the DMRS representation of the control construction.

The head-complement rule unifies V1's constraints on its complement with those of V2 which results in the identification of the XARG value of V2 with the index of the subject. The head-complement rule also propagates up the constraints on the subject from the head daughter *mencoba* “tried” to the verb phrase *mencoba membuka palang itu* “tried to open that shutter”. The semantics of this verb phrase preserve the semantic properties of its daughters, including the desired reentrancies with the subject index. The subject-head rule combines *Holmes* with *mencoba membuka palang itu* and identifies the ARG0 value of *Holmes* with the ARG1 values in both V1 and V2. This is illustrated in (264).

Raising SVC

The raising phenomenon is where a syntactic argument's semantic index is not linked to any semantic argument position in a given lexical entry's semantic relation, but is instead assigned to a role in the semantics of another syntactic argument of the lexical entry. In a raising construction, the head daughter (V1) can have two syntactic arguments, an NP and a VP, as in Example (259). V1 does not do anything semantically with its subject. Sag et al. (2003, p. 366) use 'active-passive pairs' which have essentially the same meaning. V1 identifies the external argument (XARG) of V2 with the subject's index. However, it takes the proposition introduced by V2 as its only semantic argument, rather than assigning a second semantic role to the subject's index. This is illustrated in (265). Unlike the control construction, there is no constraint [AUX -] on V2 because V2 can take auxiliaries or aspect markers.

The head-complement rule unifies V1's constraints on its complement with those of V2 and identifies the subject's index with the XARG value of V2. The subject-head rule combines the subject *waktu* "time" with the VP *terasa berlalu* "feels passing" and identifies the ARG0 value of the subject with the ARG1 in V2. This is illustrated in (266).

Serial verbs with other semantic relationships

The result in Section 7.2.3 shows us that V1 in the manner, purpose, and coordinated action SVCs can be any verbs other than control and raising verbs and there are ambiguities in the semantic relationships. Because of this, I made rules based on the transitivity and introduced a general semantic relation `svc_p_rel`, `svc_coord_p_rel`, and `svc_result_p_rel`.⁶

(268)

V1 is intransitive

In an SVC where the V1 is intransitive, V1 and V2 share the same subject but not the same object. The object, if there is any, is the argument of V2. The head daughter of this construction is V1. Here I introduce a new relation `svc_p_rel` having two arguments. Its first argument (ARG1) is identified with the index of the head daughter (V1), while its second argument (ARG2) is identified the index of V2. V2 cannot take auxiliaries or aspect markers, I put a constraint [AUX -] on it. This is illustrated in (269).

⁶The other relations `compound_p_rel` and `adj_coord_p_rel` are for noun-noun compound and adjective-adjective compound.

The head-complement rule unifies V2 with its object. The SVC rule unifies V1 and V2 (with its object argument) and propagates up the constraints on the subject from the head daughter *pulang* “return home” to the verb phrase *pulang melalui halaman itu* “go home by passing through that yard”. The semantics of this verb phrase retain the semantic properties of its daughters and the *svc_p_rel* predicate relation. The subject-head rule combines *saya* “1sg” with *pulang melalui halaman itu* and identifies the ARG0 of the subject with the ARG1 of V1 and V2. This is illustrated in (270).

(270)

(271)

V1 and V2 are transitive with a shared object

In this SVC, both V1 and V2 share the same subject and the same object. The L-INDEX and R-INDEX of `svc_coord_p_rel` are both identified with the index of V1 and V2. The semantics is similar to “V1 and V2”. This is illustrated in (272). As mentioned in Section 7.2.2, illustrated in Example (252), V1 and V2 occur simultaneously and the order of V1 and V2 is interchangeable. I put a constraint [AUX -] on V2 because it cannot take auxiliaries or aspect markers.

(274)

The SVC rule unifies V1 *memukul* “beat up” and V2 *mengeroyok* “gang up” and propagates up the constraints on the object *Budi* and the subject *mereka* “3PL”. The head-complement rule unifies V1+V2 with their shared object and identifies the ARG0 of the object with the ARG2 of V1 and V2. The subject-head rule combines the shared subject with *memukul mengeroyok Budi* and identifies the ARG0 of the subject with the ARG1 of V1 and V2. This is illustrated in (273).

V1 is transitive and V2 is intransitive, having a resultative meaning

This SVC is a particular construction, with regard to the transitivity of V1 and V2 and the semantic relationship. V1 is a transitive verb, its subject and object are the same as the subject and the object of the SVC, respectively. V2 is an intransitive verb, its subject corresponds to the object of the SVC. In terms of semantic relation, it is resultative and thus I posit a new relation *svc_result_p_rel*. Its ARG1 and ARG2 are identified with the indices of V1 and V2, respectively. I put a constraint [AUX -] because V2 cannot take auxiliaries or aspect markers. This is illustrated in (275).

The SVC rule unifies V1 *membunuh* “kill” and V2 *mati* “die”. The head-complement rule unifies V1 with its object (also V2 with its subject argument) and identifies the ARG0 of the SVC’s object with the ARG1 of V2 and the ARG2 of V1. The subject-head rule combines the SVC’s subject with the VP *membunuh mati Adi* and identifies the ARG0 of the subject with the ARG1 of V1. This is illustrated in (276).

(277)

7.2.5 Evaluation and conclusion

The analyses of Indonesian SVCs depart from Englebretson (2003) with references from Arka (2000) and other reference grammars. A Python script was written to extract SVCs from a Sherlock Holmes short story and a Japanese short story, both translated into standard Indonesian, in NTU-MC (Tan & Bond, 2012). A test-suite containing representative set of sentences designed to show SVC phenomena for Indonesian was created based on the 29 sentences extracted from NTU-MC (all positive/grammatical test items). Indonesian SVCs were analyzed and modeled using HPSG and MRS and the analyses were implemented in INDRA. I posit lexical types for control and raising verbs. For verbs with other semantic relationships, I made rules based on the transitivity of V1 and V2 and the shared arguments, as well as the semantic relationships. After the addition of lexical types and SVC rules, as well as new lexical items in the test-suite, I did the evaluation by checking the coverage of parsed sentences in the test-suite using gTest, a DELPH-IN Grammar Testing Tool.⁷ 27 out of 30 sentences in the test-suite can be parsed (overall coverage 90%). This coverage was got after the addition of new lexical items (lexical acquisition) and SVC rules. In the future, I will get more data from different corpora and do more work on verb subcategorization, I will analyze which verbs in particular SVCs are likely to appear in which semantic relations (manner, purpose, coordinated action, source, and resultative).

7.3 Noun-noun compound

This section discusses non-idiomatic noun-noun compound in Indonesian (cf. Section 7.1), which is regarded as common phrase by Kridalaksana (1989) or as non-lexicalized compound by Spärck Jones (1983).⁸ In this compound, a noun can be followed by another, modifying noun without any overt morphemes, which gives more specific information about the head noun.

⁷<https://github.com/goodmami/gtest>

⁸Spärck Jones (1983) divides noun-noun compounds into the lexicalized and the non-lexicalized. The lexicalized compound nouns can be regarded as single words, developing meaning extensions having no reference to their underlying structure. They correspond to the idiomatic compounds mentioned in Section 7.1.

Table 7.3: The six relation classes for compound nouns (Ó Séaghdha, 2007)

Relation	Example
BE	<i>pisau besi</i> “steel knife”
HAVE	<i>pisau ibu</i> “mother’s knife”
IN	<i>pisau dapur</i> “kitchen knife”
INST	<i>pisau roti</i> “bread knife”
ACTOR	<i>sopir bus</i> “bus driver”
ABOUT	<i>buku sejarah</i> “history book”

Noun-noun compounding is productive in Indonesian. Some modifying nouns can themselves take a modifier. For example, *awak bis kota* “city bus crew”. Here the head of the noun phrase is *awak* “crew”. The modifying noun *bis* “bus” has its own modifying noun, *kota* “city” (Sneddon et al., 2010, p. 148). Long compound nouns are common in Indonesian, for example *Badan Penyelidik Usaha-Usaha Persiapan Kemerdekaan Indonesia* (BPUPKI) “the Investigating Committee for Preparatory Work for Indonesian Independence” and *Badan Pembinaan Pendidikan Pelaksanaan Pedoman Penghayatan dan Pengamalan Pancasila* (BP7) “the Education Development Board Implementation Guidance and Practice of Pancasila”. In these examples we see that in the Indonesian version, the nouns are simply juxtaposed, while in the corresponding English translations, prepositions such as “for” and “of” are sometimes used.

Regarding the interpretation of compound nouns, Spärck Jones (1983) states that three elements can be employed to provide a meaning representation for compound nouns. Firstly, the sense of the constituent words have to be identified; secondly, the structure or syntactic bracketing has to be determined (see Figure 279 and 280); and thirdly, the semantic relations linking the words have to be established. Establishing the semantic relations linking the constituent words has been researched by many scholars such as Ó Séaghdha (2007) who created a new set of six semantic relations for compound nouns in English, as shown in Table 7.3. The relationship can take a wide range of meanings: *pisau besi* “steel knife” is knife made of steel, *pisau dapur* “kitchen knife” is knife used in the kitchen, and *pisau roti* “bread knife” is knife for cutting bread. Possession relation can also be expressed by juxtaposing the nouns: the possessor noun follows the head noun, which can be a common noun, a pronoun, or a proper name, e.g. *pisau ibu* “mother’s knife”, *pisau saya* “my knife”, and *pisau Budi* “Budi’s knife”.

However, as Ó Séaghdha (2007) mentions, compounds are often highly ambiguous and a large degree of ‘world knowledge’ seems necessary to understand them. A given compound type can have a different meaning in different context. For example, *buku sekolah* “school book” can be “a book used in school” (IN relation), “a book belonging to a school” (HAVE relation), or “a book about a school” (ABOUT relation) (Ó Séaghdha

& Copestake, 2007). Because of this, the best way to deal with the semantic relations is to create an underspecified semantics predicate.

My analysis of noun-noun compound is sketched in Figure 278. This rule inherits from *head-initial* rule. The head daughter is N1 and is of type *noun*. The non-head daughter is N2 and is of type *nondemnoun*, i.e. demonstrative pronouns such as *ini* “this” and *itu* “that” cannot modify N1 (see Section 4.1 for noun subcategorization). The underspecified semantics is represented by *compound_p_rel*, which takes as its two arguments the modified noun and the modifying noun. The letter p in *compound_p_rel* represents ‘preposition’. Its ARG1 is identified with the index of the head daughter (N1), while its ARG2 is identified with the index of N2. The LBL of the compound is identified with the one in the head daughter. The SPR of the head daughter (N1) is identified with the SPR of the compound. For example, in *toko buku itu* “that bookstore”, the determiner *itu* “that” modifies the N1 *toko* “store, shop”, not the N2 *buku* “book”.

Using the rule in (278), INDRA can recognize the syntactic ambiguity in the compound *toko buku Jepang* which can mean “Japan’s book shop” and “Japanese book shop”. In the first interpretation, *Jepang* “Japan” modifies *toko* “shop”, its parse tree is shown in (279) and the semantic representation is shown in (281). In the second interpretation,

Jepang “Japan” modifies *buku* “book”, its parse tree is shown in (280) and the semantic representation is shown in (282).

N1 may be modified by an adjective and N2 modifies and follows the N1+Adjective. For example,

- (283) *sikap tegas bapak*
 attitude firm father
 ‘father’s firm attitude’ (Sneddon et al., 2010, p. 160)

7.4 Adjective-adjective compound

In Indonesian two adjectives may be juxtaposed to form a kind of coordination, without *dan* ‘and’ occurring between them. For example,

- (284) *Perawakannya tinggi langsing, pembawaannya tegap gesit.*
 figure=3SG tall slender bearing=3SG firm agile
 ‘Her figure was tall and slender, her bearing firm and agile.’ (Sneddon et al., 2010, p. 181)

The rule for adjective-adjective compound is illustrated in (285). In this rule, both the first adjective (A1) and the second adjective (A2) in the compound share the same subject (SUBJ). A1 is the syntactic head, but semantically the predicate *adj_coord_p_rel* is the head. The left index and the right index of the compound predicate are identified with the index of A1 and A2, treating the semantics the same as ‘A1 and A2’.

Using the rule in (285), INDRA can parse the sentence in Example (284), as shown in (286) with the semantic representation in (287).

(286)

(287)

7.5 Summary

This chapter has shown how compounds in Indonesian are analyzed and implemented in INDRA. Idiomatic or lexicalized compounds are regarded as single words and simply listed in the lexicon. The analyses of verb-verb compounds depart from Englebretson (2003) with references from Arka (2000), which includes control and raising. I made lexical types for control and raising verbs. For compound verbs with other semantic relationship, I made rules based on the transitivity of V1 and V2, as well as the semantic relationships. However, which verbs in which particular SVCs are likely to appear in which semantic relations have not been analyzed yet and they are for future work. For compound nouns, I created an underspecified semantics predicate because a compound noun may mean differently in different context. As Spärck Jones (1983) notes, the meaning of compound nouns has to be constructed by the reader or hearer. Lastly, for adjective-adjective compound having a coordinative relation, I analyzed the semantics as “A1 and A2”.

PART III:

JATI

Chapter 8

KBBI database

In the previous part I have talked about basic structures and several grammatical phenomena in INDRA. In this third part, I talk about the JATI treebank and the corpus I used to build JATI. The corpus data is definition sentences from the official dictionary of the Indonesian language, Kamus Besar Bahasa Indonesia (KBBI). In this chapter, I explain about my work on creating a database for KBBI. KBBI is published by Badan Pengembangan dan Pembinaan Bahasa (The Language Development and Cultivation Agency) or Badan Bahasa, under the Ministry of Education and Culture of the Republic of Indonesia. The fourth edition of KBBI (Sugono, 2008) has more than 92,000 entries and 100,000 senses and contains a wealth of linguistic information and cultural diversity of Indonesia. However, the data was available only in Microsoft Excel and Word files in exactly the same format as the one in the printed dictionary. Its online edition was only meant for basic word search by entry words. Thus, in order to create an online dictionary application which has advanced search capabilities, building a database is very vital: the data structure needs to be identified and the data itself needs to be cleaned so that it can be broken down based on its components. Atkins & Rundell (2008, p. 114) state that a database is one of the three main components of Dictionary Writing System (DWS). This chapter describes my effort in building the KBBI database in SQLite¹ using Python programming language² and presents some applications for lexicographic and linguistic research and analysis. The KBBI database is employed for the online DWS application called “KBBI Dalam Jaringan” or “KBBI Daring”³ (Kamajaya et al., 2017), the offline KBBI mobile applications in Android and iOS, and the printing of the latest, fifth edition of KBBI (Amalia, 2016).

In the following sections I introduce the situation before the KBBI database was created (Section 8.1). I then describe the dictionary format (Section 8.2) and mention in details the process of the cleaning-up and database creation (Section 8.3). Finally, I look

¹www.sqlite.org

²www.python.org

³<https://kbbi.kemdikbud.go.id>

at the current state of the database, describe some of its applications (Section 8.4), summarize, and propose some future work (Section 8.5). Sections of this chapter have appeared in Moeljadi et al. (2017).

8.1 Introduction

Kamus Besar Bahasa Indonesia (KBBI) is the official dictionary for Indonesian, published by Badan Pengembangan dan Pembinaan Bahasa (The Language Development and Cultivation Agency) or Badan Bahasa under the Ministry of Education and Culture of the Republic of Indonesia. Up until present, KBBI is the most comprehensive and the most authoritative reference for the Indonesian language. The first edition of KBBI, published in 1988, has 62,000 entries. The number of entries increased to 72,000 or about 10,000 entries over three years in the second edition (1991). The third edition of KBBI, published in 2001, contains 78,000 entries and seven years later, the fourth edition of KBBI's number of entries increased to more than 92,000. The latest, fifth edition of KBBI was released for the first time in 2016 in three formats: printed, online, and offline versions. These three versions are launched to meet the needs of all users.

The online KBBI before 28 October 2016 used the data from the third edition of KBBI and allowed searches only by headwords. The search results were presented exactly in the same format as the one in the printed version, i.e. using bold or italic typefaces and different punctuations, such as colons and semicolons (see Section 8.2 for the details of the formatting effects). These formatting effects serve only as stylistic presentations and do not distinguish the fields or their structure explicitly. For example, to look up *mengacang*, a user must first look up the basic form *kacang*, as shown in Figure 8.1. This may present some difficulties if the user is not familiar with Indonesian morphological rules. The users cannot perform more targeted searches and computer applications cannot utilize the data fully. This can be overcome by identifying the data structure, cleaning the data, and breaking it down based on its components or structure of dictionary entries.

I identified the data structure and broke it down using regular expressions in Python programming language. The results are converted into an SQLite database to facilitate more specific and targeted word lookup and analysis. Lim et al. (2016) mention the categorization of lexical resources in terms of their digital readiness for natural language processing (NLP) work, from paper dictionaries, machine-readable dictionaries, machine-tractable dictionaries, to semantic rich resources, as illustrated in Figure 8.2 on page 215. Paper dictionaries are traditional dictionaries printed on paper. They are only for human consumption. The contents are presented with text formatting effects and organized by headwords. Machine-readable dictionaries (MRDs) are digitized versions of the original paper-printed versions and are the most common form of electronic dictionaries, which retain the text formatting styles. The previous online KBBI, as shown in Figure 8.1, was

Figure 8.1: Screenshot of the online KBBI before 28 October 2016

an MRD. Machine-tractable dictionaries are MRDs with machine-tractable structures, i.e. all fields and hierarchy of the entries are specifically marked and delineated, such that different information can be identified and extracted. My work was to bring the KBBI to the level of this digital-readiness. Semantic rich resources are machine-tractable dictionaries with semantic information for each sense entry. They are very useful for NLP tasks, such as text categorization, sentiment analysis, and information extraction. However, this is outside the scope of my work.

Figure 8.2: Types of lexical resources, based on digital readiness (Lim et al., 2016)

Figure 8.3: Example entry *mereka* “they” Figure 8.4: Example entry *AD*

8.2 The KBBI dictionary format

KBBI is a general dictionary whose macrostructure has a hierarchical order. The schematic is arranged by placing the basic form as the headword or the lemma. The information fields in an entry structure include a headword or lemma; variant forms; pronunciations; labels: parts-of-speech, styles, languages, domains, figurative expressions, abbreviations; sublemmas/subentries: derived words, multiword expressions (MWEs) including compounds, figurative expressions, and proverbs; definitions; cross-references; examples; scientific names; and chemical formulas.

Figure 8.3 shows that the headword or the lemma is in bold type with periods for syllabification, followed by the pronunciation, surrounded by slashes. The part-of-speech label is written in italic type, following the pronunciation. Figure 8.4 shows that if there is more than one definition phrase in one sense, the definition phrases are separated by semicolons. If there is an example, a colon is put after the definition and followed by a space; the example is in italic type. The lemma is represented by two hyphens in the example. If a lemma is an abbreviation, a label for abbreviation is placed before the definitions. If there is more than one sense, a polysemy number is written in bold type before each sense and senses are separated by semicolons. If the definition is in a foreign language, it is written in italics. Figure 8.5 shows that if a lemma has a chemical formula or a scientific name, it is written after the definition and preceded by a semicolon. The scientific name is written in italic type. Some numbers in the chemical formula are subscripted.

Figure 8.6 shows that if a lemma is homonymous, a homonymy number is placed before the lemma in superscript bold type. If a particular label is appropriate for every sense, it is written before the first polysemy number. If a label is appropriate only for a particular sense, it is written after the respected polysemy number for that sense. Subentries, such as compounds and derived words, are in bold type. Subentries are separated

Figure 8.5: Example entries *digitalin* and *digitalis* Figure 8.6: Example entry *kaul* “vow”

by semicolons. If a derived word has an example, it is represented by a tilde in the example. The derived forms are not listed as main entries unlike English dictionaries. They are listed below the basic forms. There is a special feature which distinguishes KBBI from other Indonesian monolingual dictionaries, i.e. the order of the derived words is not arranged alphabetically, but in accordance with the paradigm of word formation. For example, *tinju* “boxing”, as a lemma, has *meninju* “to box” as a transitive verb followed by *peninju* “boxer”, *peninjuan* “act/process of boxing”, and *tinjuan* “the result of boxing”. This sequence of verbs, actors, acts/processes, and results is called the paradigm of word formation.

Figure 8.7 shows that if a lemma or a sublemma appears in proverbs or figurative expressions, the lemma is represented by two hyphens, while the sublemma is represented by a tilde, same as in the example field. Both the proverbs and the figurative expressions are in italics, followed by a comma and *pb* for proverbs (*pb* stands for *peribahasa* “proverb”) or *ki* for figurative expressions (*ki* stands for *kiasan* “figurative”). For cross-references, Figure 8.8 shows that if a lemma is non-standard, a right arrow is placed after it, followed

Figure 8.7: Example entry *karam* “shipwrecked”

ke.ron.sang → **kerongsang**
ke.ron.tang *lihat* ¹**kering**

Figure 8.8: Example entries *keronsang* and *kerontang*

by the standard lemma in bold type. If a lemma is a part of an idiomatic compound, it is followed by *lihat* “see” and the cross-referenced lemma printed in bold type. Up until the fourth edition of KBBI, the dictionary data with the formatting effects mentioned above was available only in Word and Excel files. The following section describes my work in breaking down the components based on the formatting effects.

8.3 Cleaning-up, conversion, and database creation

In January 2016 during the Wordnet Bahasa workshop at NTU I asked Badan Bahasa to share their KBBI data (fourth edition) for the improvement of Wordnet Bahasa. They agreed to share the data with one condition: I have to build a database because the data is available only in Word and Excel.

The cleaning-up and conversion processes of data in Word and Excel files are quite tricky. This is because the data available in Word and Excel are formatted text (i.e. some of the data are in bold type, italic type, superscripted, subscripted, or any combination of those, as described in Section 8.2). In addition, I want to keep the format as I transfer the data into the database. Figure 8.9 shows a part of the Word and Excel files. The format, not just the text, is a part of the lexicographic information and must not be removed during the conversion process. Hence, simple data extraction from Excel cells and Word paragraphs

	A
1	<u>A, a n 1 huruf pertama abjad Indonesia; 2 nama huruf a; 3 penanda pertama di urutan (mutu, nilai, dsb)</u>
2	<u>à 1 kira-kira; lebih kurang (antara dua angka untuk memperkirakan panjang, besar, dsb sesuatu): ular itu panjangnya 6 — 7 m; lama perjalanan 2 — 3 jam; 2 harga tiap-tiap satuan: ia membeli bahan itu 5 m — Rp20.000,00</u>
3	<u>a- bentuk terikat 1 kekurangan: anemia; 2 tidak atau bukan: aseksual; 3 tanpa: anonim</u>
4	<u>aa Sd n akang</u>
5	<u>¹ab n wadah kecil dr timah untuk candu; hap</u>
6	<u>²ab ark n ayah</u>
7	<u>ab- bentuk terikat dari: jauh dr: abnormal</u>
8	<u>aba n ayah; bapak</u>
A	
aco, meng.a.co v	1 berkata tidak keruan; memberi keterangan asal berkata saja; 2 mengigau; 3 berjalan tidak betul (tt mesin, arloji, dsb): sudah beberapa hari ini arlojiku ~ saja; ~ belo mengacau tidak keruan;
aco.an a	sembroto; ugal-ugalan; serampangan;
aco-aco.an a	ugal-ugalan; sembroto
ae.ros.kop /aéroskop/ n	alat untuk menangkap debu, bakteri, spora, dsb dr udara untuk tujuan tes (percobaan, pengujian)
² agon n	Lay garis di peta yg menghubungkan
	diterima oleh panitia untuk seminar pd bulan Desember yang akan datang; 2 usul; anjuran;
	peng.a.ju.an n proses, cara, perbuatan mengajukan; pengusulan: -- usulmu itu terlambat
ak.ro.me.ter /akrométer/ n	Tek alat untuk mengukur kerapatan minyak
am.bi.li.ngu.al n	orang atau masyarakat yg mempunyai kemampuan seimbang di dua bahasa
ame.ta.bo.la /amétabola/ n	Zool serangga yg tidak menunjukkan adanya metamorfosa di perkembangannya

Figure 8.9: A part of KBBI Fourth Edition in Microsoft Excel and Word files

to the database entries cannot be done as it would not retain the text format.

In order to retain the text format in the conversion, the formatting effects in the Word and Excel files must be read, too. Therefore, a Windows Form application named “KB-BICleaner” (see Figure 8.10) was created using .Net Framework by Ian Kamajaya to help me complete the task. The program uses Word-and-Excel-compatible Microsoft-created dynamic link libraries (.dll), namely Microsoft.Office.Interop.Word and Microsoft.Office.Interop.Excel, to extract the data from Word and Excel in the Rich Text Format (RTF). Figure 8.11 shows a part of the RTF file. Furthermore, to ease the cleaning-up process, the program is designed with three additional main functionalities: (a) File and string manipulation, (b) List of text filter and conversion, and (c) List of regular expression filter and conversion, explained in the following subsections.

8.3.1 File and string manipulation

This function helps me determine which portions of the Excel file to be cleaned using the program. This is used primarily for cleaning up the Excel file as it contains multiple sheets with different (inconsistent) “Range” to be cleaned up (for example, in one sheet, there might be three columns of data while in another sheet there might be four columns). It also helps me determine which starting and ending strings (in RTF format) can be used as division of cells, lines, or paragraphs.

Figure 8.10: Screenshot of “KBBICleaner”

```

\trowd \trgaph30\trleft-30\trrh317\cellx1040\clmgf \cellx2351\clmrg \cellx18546\pard \intbl
\qc \f5\fs22 \cf55 1\cell \ql \f6\fs22 \b A\fs5\fs22 \b0 , \f6\fs22 \b a \f7\fs22 \i \b0 n\fs
\fs22 \i0 \f6\fs22 \b 1\fs5\fs22 \b0 huruf pertama abjad Indonesia; \f6\fs22 \b 2\fs5\fs22
\b0 nama huruf \f7\fs22 \i a\fs5\fs22 \i0 ; \f6\fs22 \b 3\fs5\fs22 \b0 penanda pertama dl
urutan (mutu, nilai, dsb) \cell \qr \f0\fs22 \cell
\pard \intbl \row\trowd \trgaph30\trleft-30\trrh317\cellx1040\clmgf \cellx2351\clmrg
\cellx18546\pard \intbl \qc \f5\fs22 2\cell \ql \f6\fs22 \b \u224\ 'e0\fs5\fs22 \b0 \f6\fs22
\b 1\fs5\fs22 \b0 kira-kira; lebih kurang (antara dua angka untuk memperkirakan panjang,
besar, dsb sesuatu): \f7\fs22 \i ular itu panjangnya 6 \u8212\ '97 7 m\fs5\fs22 \i0 ; \f7\fs22
\i lama perjalanan 2 \u8212\ '97 3 jam\fs5\fs22 \i0 ; \f6\fs22 \b 2\fs5\fs22 \b0 harga tiap-
tiap satuan: \f7\fs22 \i ia membeli bahan itu 5 m \u8212\ '97 Rp20.000,00\fs5\fs22 \i0 \cell
\qr \f0\fs22 \cell

```

Figure 8.11: A part of the RTF file of KBBI

8.3.2 List of text filter and conversion

This function helps me process the file with a list of text filter and conversion. Some data in the RTF format are not needed (for example, the header of the file and the unused format code) and some need to be changed (for example, indicators of bold type, italic type, superscript, and subscript formats). Thus, a list of text filter and conversion will greatly help me process such data. All entries in the list will be applied to the original text in a sequential fashion, i.e. from the top entry to the bottom entry.

If a user needs to remove a certain consistently unused string, he or she needs to specify it in the filter with no conversion value. In addition, if a user needs to replace a certain consistently appearing string, he or she needs to specify it and the desired conversion value in the list. Moreover, if a user needs to change two or more different formats into a single final format, he or she can exploit the sequential behavior of the filter to convert the earlier format(s) to the uniformed format in a sequential fashion and convert the uniformed format to the desired (single) final format.

This list of filter can be applied with both (real-time) user input values and predefined, loaded .txt file containing the filter information to further help a user save his or her filter midway whenever he or she finds the list of filter non-final and wants to continue to do it conveniently next time.

8.3.3 List of regular expression filter and conversion

Similar to the list of text filter and conversion above, this function helps a user with a list of regular expression (regex) —instead of text— filter and conversion. Regular expression or regex is a language for specifying text search strings which requires a pattern that we want to search for and a corpus of texts to search through (Jurafsky & Martin, 2009). The regex filter behaves the same way as the text filter: it obtains and converts the filtered text according to the given list in a sequential fashion. However, it filters and converts the filtered text using regex patterns instead of doing direct conversion. Thus, this function can simply be perceived as a more powerful version of its text filter counterpart.

Naturally, however, being made of a set of regexes, this regex filter and conversion are significantly slower than the text filter and conversion. For cleaning-up process of a text data as large-sized as dictionary data, the time difference can be significant. Thus, this function is meant to help us process unused or to-be-converted data which form certain patterns. For statically written data, although they can be processed by this filter, they should be efficiently processed using the text filter instead of this regex filter. Table 8.1 shows some examples of the conversion from Excel to RTF and Hypertext Markup Language (HTML) and Figure 8.12 shows a part of the HTML file, as a result of the filter and conversion process using “KBBICleaner”.

Table 8.1: Some examples of the conversion

Field	Excel	RTF	HTML
Lemma	A, a	\b A\f5\fs22 \b0 , \f6 \fs22 \b a \f7\fs22 \b0	A, a
Label	<i>n</i>	\i n\f5\fs22 \i0	<i>n</i>
Homonymy number and lemma	¹ab	\b \super 1\f6\fs22 \nosupersub ab\f5\fs22 \b0	ab (1)
Chemical formula	Cu ₃	Cu\f9\fs22 \sub 3\f5\fs22 \nosupersub	Cu₃

```

<b>A, a</b><i>n</i><b>1</b>..huruf.pertama.abjad.Indonesia;..<b>2</b>..
nama.huruf.<i>a</i>;..<b>3</b>..penanda.pertama.dl.urutan.(mutu,
nilai,dsb)
<b>a</b>..<b>1</b>..kira-kira;lebih.kurang.(antara.dua.angka.untuk.
memperkirakan.panjang,.besar,.dsb.sesuatu):<i>ular.itu.panjangnya.6.
\u8212\'97.7.m</i>;<i>lama.perjalanan.2.\u8212\'97.3.jam</i>;..<b>2</b>
..harga.tiap-tiap.satuan:<i>ia.membeli.bahan.itu.5.m.\u8212\'97.
Rp20.000,00</i>
<b>a</b><i>bentuk.terikat</i><b>1</b>..kekurangan:<i>anemia</i>;..<
b>2</b>..tidak.atau.bukan:<i>aseksual</i>;..<b>3</b>..tanpa:<i>anonim</i>
<b>aa</b><i>Sd</i><i>n</i>.akang
<b>ab.(1)</b><i>n</i>.wadah.kecil.dr.timah.untuk.candu;.hap
<b>ab.(2)</b><i>ark</i>.<i>n</i>.ayah
<b>ab</b><i>bentuk.terikat</i>.dari;.jauh.dr:<i>abnormal</i>
<b>aba</b><i>n</i>.ayah;.bapak

```

Figure 8.12: A part of the HTML file of KBBI

8.3.4 Cleaning-up

After I converted the RTF file to a HTML file using “KBBICleaner”, I found some inconsistencies in the formatting effects and I did some cleaning-up for the data. I observe that these inconsistencies in formatting are sporadic and are due to the manual formatting work by hand. Table 8.2 shows some of the inconsistencies we found. In addition, I modified some definitions in order to make the formatting more consistent and to extract more information, such as chemical formulas, scientific names, and examples. Some examples are shown in Table 8.3.

8.3.5 Breaking down the components and creating a database

I wrote a Python script to break down the components or fields for each dictionary entry based on the patterns and formatting effects described in Section 8.2, using regex. Figure 8.13 illustrates the algorithm I used to extract a number of fields in an entry. To facilitate easier manipulation of the data, all broken-down components such as lemmas, definitions, and examples were exported to an SQLite database.

Table 8.2: Some inconsistencies in the KBBI format and the cleaning-up process

Type	Example	
	Before cleaning-up	After cleaning-up
incomplete syllabification	(ke)ro.boh.an	(ke.)ro.boh.an
a semicolon should be a colon before an example	...pangkat dsb); <i>~nya sbg duta...	...pangkat dsb): <i>~nya sbg duta...
a comma should be a semicolon separating examples	...spt air mengalir: <i>-- udara, -- lalu lintas</i>;...	...spt air mengalir: <i>-- udara</i>; <i>-- lalu lintas</i>;...
pronunciations should precede labels	... <i>n</i> /gadéh/ belanak	... /gadéh/ <i>n</i> belanak
<i>ki</i> should be written in an figurative expression	<i>tidak -- tidak bukan</i>, yg ...	<i>tidak -- tidak bukan, ki</i> yg ...
a comma should precede <i>pb</i> in a proverb	...tak lapuk oleh hujan pb</i> ...	...tak lapuk oleh hujan, pb</i> ...
Scientific names should not be put inside brackets	...; <i>(Aquilaria malaccensis)</i>;	...; <i>Aquilaria malaccensis</i>;

Table 8.3: Some modifications in the KBBI definitions

(1) Move chemical formulas to the end of the definitions, preceded by a semicolon	
Before	nit.rat <i>n</i> <i>Kim</i> garam asam nitrat HNO ₃ , dipakai dl campuran pupuk
After	nit.rat <i>n</i> <i>Kim</i> garam asam nitrat, dipakai dl campuran pupuk; HNO ₃
(2) Change <i>rumus kimia</i> “chemical formula” to a semicolon	
Before	kam.fa.na <i>n</i> <i>Kim</i> kristal ... pd suhu 158–159°C dan rumus kimia C ₁₀ H ₁₈
After	kam.fa.na <i>n</i> <i>Kim</i> kristal ... pd suhu 158–159°C; C ₁₀ H ₁₈
(3) Move scientific names to the end of the definitions, preceded by a semicolon	
Before	tal (1) <i>n</i> l tumbuhan palem ; <i>Borassus fiabellifer</i> daunnya ...; batang lontar;
After	tal (1) <i>n</i> l tumbuhan palem, daunnya ...; batang lontar; <i>Borassus fiabellifer</i>
(4) Change <i>msl</i> “e.g.” before examples to a colon	
Before	(dl bentuk kata kerja ber-...-an); msl <i>bersikutat</i>, berikutat-kutatan; <i>bersipandang</i>, berpandang- pandangan
After	(dl bentuk kata kerja ber-...-an): <i>bersikutat</i>, berikutat-kutatan; <i>bersipandang</i>, berpandang- pandangan

```

For each entry in each line in the HTML file,
  IF <b> is at the beginning of that line,
    extract the lemma between <b> and </b>
    IF a lemma contains an opening bracket and a closing bracket,
    or IF there is a comma in the lemma,
      extract the variant form(s)
    IF there is a slash after </b>,
      extract the pronunciation(s) between slashes
      IF <i> appears after the second slash,
        extract the label(s) for POS, language, domain, etc.
      IF there is a number surrounded by <b>...</b>,
        split and extract the senses
        for each sense,
          IF <i> appears after </b>,
            extract the label(s)
          extract the definition
          IF there is <i>...</i> after a colon,
            extract the example(s)
          IF there is <i>...</i> after a semicolon,
            extract the scientific name(s)
          IF there is some chemical formula(s),
            extract the chemical formula(s)
  IF there is an arrow,
  or IF there is lihat,
    extract the cross-reference

```

Figure 8.13: A part of the algorithm used to extract a number of fields in an entry

8.3.6 Dictionary data structure

The data structure of KBBI consists of four types of data: entry, sense, example, and category. The relationship between entry and sense, as well as the one between sense and example are one-to-many. The category is a list of descriptions or a metadata for entry, sense, and example. Figure 8.14 illustrates the KBBI data structure. An entry can be a fixed expression (*ungkapan*) or a basic form (*bentuk dasar*).⁴ A fixed expression has at least one sense and one example. In this case, one fixed expression may have one to multiple senses and one sense may have one to multiple examples. A basic form has at least one cross-reference, one sense, one compound, or one derived word. In this case, one basic form may have zero to multiple senses and one sense may have zero to multiple examples. A basic form may also have variant(s), proverb(s), and figurative expression(s). A proverb or a figurative expression has at least one sense. A compound has at least one cross-reference or one sense. One sense may have zero to multiple examples. Similar to the basic form, a derived word has at least one cross-reference, one sense, or one compound. It may also have variant(s), proverb(s), and figurative expression(s). The basic form can be in the form of compound if it can be affixed and have derived word(s).

⁴It is also called *kata dasar* “basic word” in the grammars of Indonesian.

Figure 8.14: The data structure for KBBI entry

8.4 The current state of the KBBI database and its applications

After the data was broken down into its components, the number of each component in the database can be checked. As of 10 August 2018, the KBBI database contains 109,213 entries which consist of (excluding definitions and examples):

- 48,748 *bentuk dasar* “basic forms”
- 26,312 *kata turunan* “derived words”
- 30,625 *gabungan kata* “compounds”
- 2,040 *peribahasa* “proverbs”
- 268 *kiasan* “figurative expressions”
- 127,775 *makna* “definitions”
- 29,495 *contoh* “examples”

The database is not really open to public at present. The license and legal aspect of the data was discussed at the Indonesian Lexicography Seminar (Seminar Leksikografi Indonesia) on August 1-3, 2018, organized by Badan Bahasa. For the time being, one can get access to the database by asking permission from Badan Bahasa, only for research purpose. Appendix B shows a letter of permission from Badan Bahasa to use KBBI V data for building a treebank.

Many applications can be made using the KBBI database. This section will provide some examples of those applications, especially for lexicography and linguistics field.

8.4.1 Targeted look-ups

A user can search for all definitions for a word which may originate from two different headwords, e.g. *mereka*, using the following search procedure. The word *mereka* has two meanings: “they” and “contrive”. The latter is derived from a root *reka*. The results are shown in Table 8.4.

```
SELECT entri, jenis, induk, lafal, kelas, makna FROM baseview WHERE entri="mereka";
```

The task of looking up phrases and MWEs such as figurative expressions and proverbs is also made simpler, as a user would no longer need to find out which headword to look up first, e.g. the following search procedure can be used to look up a proverb *sedia payung*

Table 8.4: Search results for all definitions of words with orthographic form *mereka*

Entri “entry”	Jenis “type”	Induk “root”	Lafal “pro- nunciation”	Kelas “word class”	Makna “definition”
mereka	root	(null)	merèka	pron	<i>orang ketiga jamak...</i> “third person plural...”
mereka	derived	reka	(null)	v	<i>menyusun (mengatur,...</i> “set (arrange,...”
mereka	derived	reka	(null)	v	<i>mencari akal (ikhtiar,...</i> “make efforts (endeavors,...”
mereka	derived	reka	(null)	v	<i>memikirkan (sesuatu);...</i> “think about (something);...”
mereka	derived	reka	(null)	v	<i>membayangkan (dl angan-...</i> “imagine (in delusion...”
mereka	derived	reka	(null)	v	<i>menduga; mengira-ngirakan</i> “guess; surmise”

Table 8.5: Search result for the proverb *sedia payung sebelum hujan*

Entri	Jenis	Makna
<i>sedia payung sebelum hujan</i> “prepare the umbrella before it rains”	proverb	<i>bersiap sedia sebelum terjadi yg kurang baik</i> “get ready before things go wrong”

sebelum hujan (lit. “prepare the umbrella before it rains”) (the headword is *payung* “umbrella”). Table 8.5 shows the result.

```
SELECT entri, jenis, makna FROM baseview WHERE entri="sedia payung sebelum hujan";
```

Linguists and etymologists can also search specific entries by their labels. For example, a user can search archaic (*ark*) lemmas originating from Javanese (*Jw*) using the following search procedure. Table 8.6 shows the search results.

```
SELECT entri, ragam, bahasa, makna FROM baseview WHERE ragam="ark" and bahasa="Jw";
```

We can also build a specific corpus from a subset of definition sentences or examples in the KBBI database. For example, we can search all definition sentences starting with the word *makanan* “food” and build a corpus about food from the search result (see Table 8.7). This is what I did in order to build a corpus data for the JATI treebank, which is explained in the next chapter.

```
SELECT entri,makna FROM baseview where makna GLOB"makanan*";
```

Table 8.6: Search results for all lemmas with labels *ark* (archaic) and *Jw* (Javanese)

Entri “entry”	Ragam “style”	Bahasa “language”	Makna “definition”
cutel	ark	Jw	<i>tamat; habis (tt cerita dsb); berakhir</i> “come to an end (about story etc); finish”
gundang	ark	Jw	<i>lekum; tenggorok</i> “Adam’s apple; throat”
pembarap	ark	Jw	<i>anak sulung</i> “firstborn child”
sikep	ark	Jw	<i>orang dr desa yg mempunyai kewajiban melakukan...</i> “a person from village who has a duty to do...”
ubel-ubel	ark	Jw	<i>tentara Inggris asal India</i> “British soldier from India”
wiyata	ark	Jw	<i>pengajaran; pelajaran</i> “teaching, lesson”

Table 8.7: Search results for lemmas having definition sentences starting with *makanan*

Entri	Makna
abon	<i>makanan yg dibuat dr daging atau ikan rebus yg diserat-seratkan, dibumbui,...</i> “food made from meat or stewed fish which is shredded, seasoned...”
acar	<i>makanan yg terbuat dr buah-buahan atau sayuran, diberi rempah dan diasamkan</i> “food made from fruits or vegetables, seasoned and pickled”
agropangan	<i>makanan yg berasal dr hasil pertanian</i> “food which comes from agricultural product”
alas perut	<i>makanan yg pertama sekali dimakan pd pagi hari sekadar pencegah masuk angin</i> “food which is firstly eaten in the morning just to prevent from catching a cold”
alimen	<i>makanan atau gizi</i> “food or nutrition”

Table 8.8: Twenty most frequent words in KBBI definitions

Word	Freq.	Word	Freq.	Word	Freq.	Word	Freq.
<i>yang</i> “REL”	43,613	<i>untuk</i> “for”	10,312	<i>pada</i> “at”	6,793	<i>dapat</i> “can”	3,020
<i>dan</i> “and”	26,221	<i>dalam</i> “in”	8,638	<i>orang</i> “person”	6,110	<i>tempat</i> “place”	2,970
<i>atau</i> “or”	14,414	<i>di</i> “at”	8,537	<i>tentang</i> “about”	4,746	<i>sebagai</i> “as”	2,917
<i>sebagainya</i> “others”	12,410	<i>tidak</i> “NEG”	7,756	<i>seperti</i> “like”	3,422	<i>oleh</i> “by”	2,910
<i>dengan</i> “with”	12,016	<i>dari</i> “from”	7,280	<i>ke</i> “to”	3,247	<i>sesuatu</i> “something”	2,851

Table 8.9: Twenty most frequent genus words in KBBI definitions

Word	Freq.	Word	Freq.	Word	Freq.	Word	Freq.
<i>orang</i> “person”	2,703	<i>tempat</i> “place”	806	<i>keadaan</i> “state”	526	<i>ilmu</i> “science”	401
<i>proses</i> “process”	1,858	<i>hasil</i> “result”	656	<i>ikan</i> “fish”	521	<i>fobia</i> “phobia”	350
<i>alat</i> “tool”	1,595	<i>sesuatu</i> “something”	573	<i>hal</i> “case”	512	<i>nama</i> “name”	337
<i>bagian</i> “part”	835	<i>kata</i> “word”	557	<i>tumbuhan</i> “plant”	443	<i>zat</i> “matter”	300
<i>perihal</i> “subject”	823	<i>pohon</i> “tree”	547	<i>tiruan</i> “imitation”	413	<i>penyakit</i> “disease”	297

8.4.2 Lexicography analysis

The definitions and examples in KBBI can be regarded as a corpus which can be employed for various analyses and give further insights to the Indonesian language. I extracted the twenty most frequent words in definitions using the Python NLTK library⁵ (see Table 8.8). These frequent words can be used as a part of a lexical set for the Indonesian language learner’s dictionary for non-Indonesian speakers Badan Bahasa is making now which uses limited words in the definitions and examples. We can also look for the genus words whose result is shown in Table 8.9. Lim et al. (2016) present the fifty most frequent words and genus words used in definitions in Kamus Dewan, the authoritative dictionary for Standard Malay. With these data, we can make a comparison of the vocabularies of Indonesian and Standard Malay.

⁵<http://www.nltk.org>

Table 8.10: Some derived words in KBBI entries, grouped by affixes

Affix/Redup.	Example	Number	Percentage
meN-	mengabdi	5,185	21.1%
meN-...-kan	mengabadikan	2,884	11.7%
ber-	berabang	2,704	11.0%
-an	abaian	1,873	7.6%
peN-...-an	pengabaian	1,780	7.2%
peN-	pengabai	1,552	6.3%

Table 8.11: Some examples of aligned KBBI entries and the Wordnet synsets

KBBI entry	Scientific name	Wordnet lemma	Wordnet synset
abaka	musa textilis	abaca	12353431-n
abalone	haliotis	Haliotis	01942724-n
abrikos	prunus armeniaca	common apricot	12641007-n
acerang	coleus amboinicus	country borage	12845187-n
adas	foeniculum vulgare	common fennel	12939282-n
adas manis	pimpinella anisum	anise, anise plant	12943049-n

8.4.3 Linguistic analysis

The Indonesian language has a very rich morphology for word derivation process. It has a rich affixation system, including a variety of prefixes, suffixes, circumfixes, non-productive infixes; and a variety of reduplications. Most of the affixes are derivational (Sneddon et al., 2010). Using regular expressions in Python, we made a table of more than 100 patterns of word formation based on affixes and various types of reduplication in Indonesian. Table 8.10 shows a part of it. It has been used in a linguistics research for analyzing the difference between *meN-...-i* and *meN-...-kan* (Nur Amirah Binte Khairul Anuar et al., 2017). There are many possibilities we can do with the data, such as analyzing other affixes and reduplications.

8.4.4 Linking to other lexical resources

KBBI contains a number of scientific names for flora and fauna. Using them as a pivot, we aligned more than 600 entries in KBBI to the entries in other lexical resources, such as Wordnet Bahasa (Bond et al., 2014). Table 8.11 shows some examples of aligned entries via scientific names.

Figure 8.15: Screenshot of the online KBBI homepage

8.4.5 Online and offline applications

KBBI database serves as the vital part in building the online DWS⁶, called “KBBI Dalam Jaringan” or “KBBI Daring”, launched on 28 October 2016 (Kamajaya et al., 2017) and offline mobile applications, both for Android⁷ and iOS⁸, launched on 17 November 2016. Figure 8.15 shows the homepage of the online KBBI and Figure 8.16 shows both the screenshots of the Android and iOS applications. In order to facilitate the workflow of the editorial staff for the online application and the online public participation to add, edit, and deactivate lemmas, definitions, and examples, the KBBI database is equipped with tables for proposals.

8.5 Summary and future work

I have described my work in creating a database for KBBI from Microsoft Excel and Word files by converting them to a RTF file and a HTML file, identifying its structure, cleaning up the data, and breaking it down based on the structure. The broken-down components were then exported to SQLite database. The database allows lexicographers, linguists, and researchers in NLP field to access the rich lexicographic and linguistic contents in the Indonesian language in more flexible ways, opening up possibilities in discovering new insights into the language, as well as helping the KBBI editorial staff work on the dictionary more effectively.

In the near future, the database will be expanded with etymological information. Badan Bahasa has been compiling and editing the etymological information since 2015. They have finished working on lemmas from Sanskrit and are working on lemmas originating from Old Javanese and Dutch (Dora Amalia p.c.). In addition, the database will be con-

⁶<https://kbbi.kemdikbud.go.id>

⁷<https://play.google.com/store/apps/details?id=yuku.kbbi5>

⁸<https://itunes.apple.com/us/app/kamus-besar-bahasa-indonesia/id1173573777>

Figure 8.16: Screenshots of the Android (left) and iOS (right) mobile applications

nected to a corpus. The source of the corpus is from scientific publications. Badan Bahasa is now adding about five million words per year.

Chapter 9

JATI

The previous chapter explains my work on creating a database for the official dictionary of the Indonesian language (KBBI), which is employed for building a treebank. This chapter introduces and describes the construction of a new lexical resource for Indonesian: “the JATI treebank”. It was built from a subset of parsed dictionary definition sentences in the fifth edition of KBBI (Amalia, 2016), the official and the most comprehensive dictionary for the Indonesian language. The dictionary definition sentences are parsed using the INDRA.

In the following sections, I give a brief introduction about JATI. I then mention some related works and describe the development process and the result. Finally I summarize and propose some future work. Most sections of this chapter have appeared in Moeljadi (2017).

9.1 Introduction

A treebank is a linguistically annotated corpus that includes some grammatical analysis beyond the part-of-speech level (Nivre, 2008). It is used for empirical linguistic research, as well as natural language processing. It enables more precise queries and reduces the noise in the answer set. The treebank data can be used in qualitative research, such as finding an example of a certain linguistic construction or a counter-example to a claim about syntactic structure, as well as in quantitative research, as a source of information about frequencies and cooccurrences. In addition, treebanks are indispensable in order to achieve robust broad-coverage parsing.

However, to date, there are few open-source treebanks for Indonesian, annotated with both syntactic and semantic information. One such treebank is the Indonesian Treebank in ParGramBank (Sulger et al., 2013), a parallel treebank which is based on LFG (Kaplan & Bresnan, 1982; Dalrymple, 2001). Similar to the Indonesian Treebank in ParGramBank, the new treebank is being built based on a computational grammar for Indonesian.

The motivation is to develop a broad-coverage grammar together with the treebank. Treebanking allows us to immediately identify problems in the grammar and improving the grammar directly improves the quality of the treebank (Oepen et al., 2004). The broad-coverage grammar used to annotate syntactic and semantic structure in the new treebank is the INDRA, which is explained in the previous part of this dissertation. It is developed using open-source tools in the Deep Linguistic Processing with HPSG Initiative (DELPH-IN). The DELPH-IN research community has built one treebank for English, called “The LinGO Redwoods Treebank” (Oepen et al., 2002), and one treebank for Japanese, called “Hinoki” (Bond et al., 2004).¹ The new Indonesian treebank introduced and described in this chapter is named “JATI”, the Indonesian word for “teak”, which is the national tree of Indonesia.² Teak wood is considered as a first-class wood because of its strength, durability, and beauty. It is used for building furniture and carvings. Since it is not easily deformed by climate change, it is also used as a material for boat decks, bridges, rail pads, and warships. The heartwood is brownish red in color. I wish JATI could serve as a strong and durable source for linguistics research and various NLP applications.

In this chapter I present the details of the treebank construction. Section 9.2 mentions some related work in treebanks, especially for Indonesian. Section 9.3 contains a brief explanation about the source of the corpus, i.e. the KBBI dictionary data. Section 9.4 explains the method for building JATI from the parsed KBBI dictionary definitions and its development process. Section 9.5 reports the treebanking result and evaluation. Finally, Section 9.6 summarizes and outlines the future research.

9.2 Related work

Most previous work on Indonesian treebanks focuses on syntactic annotation, rather than semantic annotation, for example: the Indonesian Dependency Treebank developed by Charles University in Prague (Green et al., 2012), a treebank with manually annotated dependency structures for Indonesian; the Indonesian treebank developed by the Faculty of Computer Science of Universitas Indonesia (Dinakaramani et al., 2016) which uses a part-of-speech (POS) tagged corpus as a starting point and adopts Penn Treebank (Marcus et al., 1993) bracketing guidelines; and the Indonesian treebank in the Asian Language Treebank (ALT) which was built by the Agency for the Assessment and Application of Technology (BPPT) (Riza et al., 2016), comprises about 20,000 sentences originally sampled from the English Wikinews in 2014, and uses tools such as POS tagger, syntax tree generator, shallow parser, and word alignment.

The Indonesian Treebank in the ParGram Parallel Treebank (ParGramBank) is based on a non-open-source LFG computational grammar called “IndoGram” and covers a di-

¹The DELPH-IN tradition is to name the treebank after a species of tree (Francis Bond p.c.).

²https://en.wikipedia.org/wiki/List_of_national_trees

verse set of phenomena, implemented using XLE that includes a parser, a generator, and a transfer system. It is publicly available via the INESS treebanking environment³ (Sulger et al., 2013) and contains 79 sentences and 433 words. The building of JATI is similar to this in terms of syntactic and semantic annotation using a computational grammar for Indonesian (grammar-based corpus annotation). However, the annotation process and the tools used are similar to the ones employed in the LinGO Redwoods Treebank of English (Oepen et al., 2002) and the Hinoki treebank of Japanese (Bond et al., 2004), in which sentences are parsed and the annotator selects the best parse from the full analyses derived by the grammar. Moreover, JATI uses an open-source HPSG computational grammar where the syntax and semantics are represented in the same structure, same as in LinGO Redwoods Treebank and Hinoki.

9.3 The corpus

The main corpus data for JATI comes from the fifth edition of KBBI, the official Indonesian dictionary published by Badan Pengembangan dan Pembinaan Bahasa (The Language Development and Cultivation Agency) or Badan Bahasa under Ministry of Education and Culture, Republic of Indonesia (Amalia, 2016), as explained in the previous chapter. Appendix B shows a letter of permission from Badan Bahasa to use the KBBI V data for treebanking.

KBBI is the most comprehensive Indonesian dictionary and contains a wealth of linguistic information, biodiversity (flora and fauna), and cultural diversity of Indonesia, which consists of various domains or fields such as traditional food, clothing, and weapons. For the purpose of building JATI, only a subset of the KBBI database containing entries related to a certain domain, i.e. food and beverages, is extracted by looking up definitions started with genus terms such as *makanan* “food”, *masakan* “dish”, *kue* “snack”, and *minuman* “drink”. By restricting to a certain domain, an ontology can be relatively easy to obtain. The definition sentences of the extracted entries are semi-automatically edited and rewritten to make them more consistent. An example is given in Table 9.1. Definition sentences in KBBI paper dictionary use abbreviations such as *yg*, *dr*, and *dng*. These abbreviations can be automatically changed to their full forms: *yang* “REL”, *dari* “from”, and *dengan* “with”.⁴ The second example in Table 9.1 shows the omission of the relative clause marker *yang*. To make consistent, the relative clause marker should be manually added. The definition of the entry *bakso* “meatball” in the third example uses *terbuat* instead of *dibuat* and there is no conjunction which connects the ingredients. To make consistent, *terbuat* should be changed to *dibuat* and a conjunction *atau* “or” should

³<http://clarino.uib.no/iness>

⁴Another approach would be to add the abbreviations to INDRA’s lexicon with the same predicate as the full form (Francis Bond p.c.).

Table 9.1: Some KBBI definition sentences, before and after editing

Before	After
minuman keras yg dibuat <i>dr</i> nira yg telah disuling	minuman keras <i>yang</i> dibuat <i>dari</i> nira <i>yang</i> telah disuling
kue kering, dibuat <i>dr</i> sagu dan dibungkus <i>dng</i> daun nipah	kue kering <i>yang</i> dibuat <i>dari</i> sagu dan dibungkus <i>dengan</i> daun nipah
makanan <i>terbuat dr</i> daging, udang, ikan yg dicincang	makanan <i>yang dibuat dari</i> daging, udang, <i>atau</i> ikan <i>yang</i> dicincang

be added. One can argue that a modified or edited corpus is not a real, natural corpus. However, editing definition sentences in a dictionary is a part of lexicography process. It is easier for the readers to read if the definition sentences are consistent. It also accommodates the current state of the INDRA which covers a limited number of phenomena. In addition, it depends on what we are going to do with the corpus. If the purpose is to build a statistical model based on an annotated treebank, then a modified or edited corpus can increase the consistency and the precision of the statistical model. JATI, at this stage, is built for this purpose. The total number of sentences is 2,003, containing 23,129 words. The average length is 11.5 words per sentence (the shortest sentence contains only one word and the longest one contains 51 words). The corpus is in GitHub.⁵

Comparing with other commonly used text for corpora such as newspaper text, dictionary sentences are shorter, contain more fragments, especially noun phrases as single utterances, and fewer quoted sentences and proper names. The dictionary definition sentences are noun phrases rather than sentences. However, they are valid examples of naturally occurring texts and a native speaker can read and understand them without special training (Fujita et al., 2007).

9.4 Treebank development

Treebanking is a part of the grammar development process (Bender et al., 2011). Firstly, a collection of sentences called test-suite is developed. JATI uses a natural test-suite from KBBI, as mentioned in Section 9.3. Linguistic phenomena in the test-suite are then identified and analyzed based on reference grammars and other linguistics literature. The analyses are modeled in HPSG and implemented in INDRA. The test-suite is parsed using INDRA and then the annotator selects the correct analysis (or rejects all analyses) using Full Forest Treebanker (FFTB). Selection is done through a choice of discriminants. The system selects features that distinguish between different parsers and the annotator selects

⁵https://github.com/davidmoeljadi/INDRA/tree/master/tsdb/skeletons/jati_sorted

Figure 9.1: Screenshot of FFTB main page for JATI

or rejects the features until only one parse is left. The choices made by the annotators are saved and thus, it is possible to update the treebank when the grammar changes (Oepen et al., 2004). If a sentence is ungrammatical or if INDRA cannot parse the sentence, no discriminants will be found. However, if a sentence is grammatical and no correct tree is found, all the possible trees should be rejected and the grammar has to be modified or debugged. Sentences for which no analysis had been implemented in the grammar or which fail to parse are left unannotated. Figure 9.1 shows the FFTB main page for JATI using a KBBI test-suite. Using FFTB, we can note some interesting findings or linguistic analyses item by item.

Figure 9.2 shows the annotation page for sentence number 1331 in the KBBI test-suite *nasi yang biasanya dimasak dengan daging kambing dan diberi bumbu* “rice which is usually cooked with mutton and seasoned”. This sentence is syntactically ambiguous: the first interpretation is that the adverb *biasanya* “usually” modifies *dimasak dengan daging kambing* “cooked with mutton”, the second interpretation is that it also modifies *diberi bumbu* “seasoned”. This sentence is the definition sentence for *nasi kebuli* “an Indonesian style spicy steamed rice dish cooked in goat broth, milk, and ghee” and the first interpre-

nasi yang biasa dimasak dengan daging kambing dan diberi bumbu

4 remaining.

Exact Unary Chain ▾

adj-head-int

biasanya dimasak dengan daging kambing

2 trees

vp1-top-coord

dimasak dengan daging kambing dan diberi bumbu

2 trees

extracted-subj

adj-head-int

biasanya dimasak dengan daging kambing dan diberi bumbu

2 trees

extracted-subj

vp1-top-coord

biasanya dimasak dengan daging kambing dan diberi bumbu

5 new manual

bare-np @n-n-compound	=	5 to 7	[x]
head-comp	=	8 to 10	[x]
head-adj-int	=	3 to 7	[x]
inanimate-noun-lex	=	0 to 1	[x]
adposition-mod-lex	=	4 to 5	[x]

/ 1331 -- accepted
[prev](#) | [next](#) | [accept](#) | [reject](#) | [list](#) | [exit](#) | [show](#) | [hide ignored text](#)

Figure 9.2: Screenshot of the annotation page for sentence number 3 in the KBBI test-suite

nasi yang biasa dimasak dengan daging kambing dan diberi bumbu

1 remaining.

/ 1331 -- accepted
[prev](#) | [next](#) | [accept](#) | [reject](#) | [list](#) | [exit](#) | [show](#) | [hide ignored text](#)

7 new manual

bare-np @n-n-compound	=	5 to 7	[x]
head-comp	=	8 to 10	[x]
head-adj-int	=	3 to 7	[x]
inanimate-noun-lex	=	0 to 1	[x]
adposition-mod-lex	=	4 to 5	[x]
adj-head-int	=	2 to 7	[x]
bare-np @head-adj-int	=	0 to 10	[x]

Figure 9.3: Screenshot of the annotation result with parse tree for sentence number 3 in the KBBI test-suite

tation is the correct one, illustrated as a phrase structure tree in the annotation result (see Figure 9.3). After the correct parse tree is selected, the annotator selects “accept” to save the result and continues annotating the next sentence. The annotator may select “list” to go back to the FFTB main page or “exit” to exit the FFTB. If “Show MRS” is selected, the MRS semantic representation will be shown (see Figure 9.4).

9.5 Result and evaluation

JATI can be evaluated by measuring the number of coverage, i.e. how many sentences or how many percent of total sentences INDRA can parse and how many of them are actually good (having correct parse trees and semantics). Figure 9.5 shows the increase

TOP: **h0**
 INDEX: **x2**
 RELS:
h4:" nasi_n_rel"(ARG0: **x2**)
h5:" biasanya_a_rel"(ARG0: **e6**,ARG1: **e7**)
h5:" masak_v_rel"(ARG0: **e7**,ARG1: **x8**,ARG2: **x2**)
h5:" dengan_p_rel"(ARG0: **e9**,ARG1: **e7**,ARG2: **x10**)
h11:" daging_n_rel"(ARG0: **x10**)
h12:" kambing_n_rel"(ARG0: **x13**)
h14:"exist_q_rel"(ARG0: **x13**,RSTR: **h15**,BODY: **h16**)
h11:compound_p_rel(ARG0: **e17**,ARG1: **x10**,ARG2: **x13**)
h18:"exist_q_rel"(ARG0: **x10**,RSTR: **h19**,BODY: **h20**)
h4:" and_coord_rel"(L-HNDL: **h5**,R-HNDL: **h21**,C-ARG: **e22**,L-INDEX: **e7**,R-INDEX: **e23**)
h21:" beri_v_rel"(ARG0: **e23**,ARG1: **x24**,ARG2: **x2**,ARG3: **x25**)
h26:" bumbu_n_rel"(ARG0: **x25**)
h27:"exist_q_rel"(ARG0: **x25**,RSTR: **h28**,BODY: **h29**)
h1:"exist_q_rel"(ARG0: **x2**,RSTR: **h30**,BODY: **h31**)
 HCONS: **h0** =_q **h1**, **h15** =_q **h12**, **h19** =_q **h11**, **h28** =_q **h26**, **h30** =_q **h4**
 ICONS: **e32** non-focus **x8**

Figure 9.4: Screenshot of the MRS semantic representation of the annotation result

in coverage from stage one to stage ten.

At the first stage, I changed the abbreviations in the definition sentences to their full forms, as mentioned in Section 9.3. Testing INDRA on the full set of 2,003 definition sentences from KBBI at this first stage gave a coverage of 17.5%. At the second stage, I edited the definition sentences and I got an increase of coverage of 18.3%. The first big increase in coverage (to 34.8%) was from expanding the lexicon or lexical acquisition, at the third stage. As stated in Bond et al. (2004), the first big increase in coverage in Hinoki (from the initial coverage of around 10% to 55%) came from automatically expanding the lexicon. At the fourth stage, I added new rules for quantifiers and adverbs modifying adjectives and I got a coverage of 40.5%. I was able to bring the coverage to over 40.9% at the fifth stage after I added a new rule for transitive verbs with non-optional complements. At the sixth stage, I got an increase in the coverage to 41.7% after I added subordinate conjunctions. Afterwards, I added a rule for possessor topic-comment relative clause and words with *ber-* prefix, at the seventh stage, and the coverage increased to 53.6%. I added a small number of lexical items at the eighth stage and I got a bit of increase in the coverage to 56.6%.

I then added a rule for active and passive voice for ditransitive verbs as well as a rule for PP modifying noun at the ninth stage and got 62.5% coverage, at which point I started treebanking. During the treebanking process, more phenomena were identified and analyzed and especially, a number of homographs, such as *tahu* “tofu”, and compounds, such as *kuning telur* “egg yolk” (lit. “egg yellow”) were added to increase the coverage. The word *tahu* “tofu” was not detected in the previous lexical acquisition because INDRA

Figure 9.5: Evolution of coverage

already had *tahu* but it was a verb meaning “know”. Similarly, the compound *kuning telur* “egg yolk” was not detected because INDRA had *kuning* “yellow” and *telur* “egg” separately, not as a compound. At this tenth stage, I got a big increase in coverage (to 84.5%). The increasing of the coverage increased the ambiguity or the number of readings for each sentence. The main sources of ambiguity are coordinate constructions and relative clause constructions. This is similar to the result in the Hinoki corpus consisting of around 95,000 dictionary definition sentences parsed using JACY (Fujita et al., 2007). On average I needed five decisions to get the correct reading, i.e. the correct parse tree and MRS.

At the present stage, testing INDRA on the set of 2,003 definition sentences from KBBI gave a coverage of 84.5%, as mentioned above. However, not all 84.5% have correct parses and semantics. After treebanking, I got 62.6% or 1,253 out of 2,003 sentences having correct syntactic trees and semantics. The average length of the sentences that have correct parses and semantics is 8.14 words per sentence (the shortest one contains only one word and the longest one contains 31 words). This is slightly lower than the average length of all sentences (11.5 words per sentence), as mentioned in Section 9.3. The reason is because most of the long definition sentences are possessor topic-comment relative clauses with many relative clauses, which have not been implemented yet, as mentioned in Section 4.7.2 on page 112. In addition to the possessor topic-comment relative clause construction, around 730 sentences (36.4%) could not be parsed or had no correct parse because some phenomena are not covered yet, such as equative, comparative, and superlative adjectives; adjectives modifying verbs (as adverbs); adjectives preceding verbs (e.g. *enak dimakan* “delicious to eat”) or nouns (e.g. *bebas kolesterol* “cholesterol free”); coordinate constructions with constituents having different parts-of-speech (e.g.

VP constituents and AP constituents); and coordinate constructions without any conjunction marking. The rest 20 sentences could not be fully resolved. The treebanking result is stored in plain text files.⁶ They can be easily edited to accommodate the changes made in the grammar (INDRA).

9.6 Summary and future work

This chapter has described the construction of JATI which was created from a subset of KBBI dictionary definition sentences, parsed using INDRA and annotated using FFTB. At the present stage, the number of items that can be parsed is 84.5% and the number of items which have correct parses and semantics is 62.6% (1,253 sentences out of 2,003 sentences). This can be improved by analyzing and implementing more phenomena such as equative, comparative, and superlative adjectives, relative clause constructions, as well as coordinate constructions.

To simplify grammar development, a snapshot of the grammar used to treebank in each development cycle will be taken. From this, information about lexical items and their types from both the grammar and treebank can be extracted and converted into an electronically accessible structured database (the lexical-type database) (Hashimoto et al., 2005). This allows grammar developers and treebankers to see comprehensive up-to-date information about lexical types, including documentation, syntactic properties, usage examples from the treebank, and links to other dictionaries. The treebank data can be employed to build a Feature Forest-based Maximum Entropy Model Trainer using the tools made by Woodley Packard.⁷

⁶https://github.com/davidmoeljadi/INDRA/tree/master/tsdb/gold/jati_sorted

⁷<http://sweaglesw.org/linguistics/>

PART IV:

SUMMARY

Chapter 10

Conclusions

“[E]xploring the structure of language [...] is like working on a gigantic puzzle – one so large that it could occupy many lifetimes.”

Sag et al. (2003: 9)

In this dissertation, I have described the creation and the development of a broad-coverage Indonesian computational grammar, called INDRA, using the framework of HPSG (Pollard & Sag, 1994; Sag et al., 2003) with MRS (Copestake et al., 2005). INDRA is the first Indonesian HPSG which aims to encode precise morphological, syntactic, and semantic information in feature structures. It also aims to be a resource grammar which can be applied to various NLP applications. A final evaluation on naturally occurring data was done over 2,003 definition sentences from the Indonesian monolingual dictionary (KBBI) (Amalia, 2016). These were parsed and the best parse selected to form the JATI treebank (Moeljadi, 2017).

Currently INDRA is being developed within the DELPH-IN community, together with other computational grammars such as the English grammar ERG (Flickinger et al., 2010), the Japanese grammar JACY (Siegel et al., 2016), and the Chinese grammar Zhong (Fan et al., 2015). The development of INDRA constitutes an important test of the cross-linguistic validity of the HPSG and MRS and it can be employed to build semantic-transfer-based machine translation systems within the DELPH-IN community, as well as other NLP applications.

Compared with IndoGram (Arka, 2012), a computational grammar for Indonesian LFG (Kaplan & Bresnan, 1982; Dalrymple, 2001) written in XLE, INDRA has more refined type hierarchies and more precision in the parsing for some phenomena. Further, it is more open, has more potential to be employed in various multilingual NLP applications, and has a bigger open-source treebank.

10.1 INDRA

I have developed INDRA from a starter grammar I built using the LinGO Grammar Matrix (Bender & Fokkens, 2010) to a medium-sized HPSG computational grammar. This was done through a combination of manual extensions and semi-automatic lexical acquisition via Wordnet Bahasa (Nurhil Hirfana Mohamed Noor et al., 2011; Bond et al., 2014). Currently INDRA covers a number of phenomena in standard, formal Indonesian such as noun and verb subcategorization, copula constructions, and compounds. It can parse and generate sentences and phrases having those phenomena. As of 15 January 2018, it contains 11 lexical rules, 62 grammar rules, 2,050 types, and 16,608 lexical entries. The coverage of the current version of INDRA is measured using the test-suites. The MRS test-suite contains 172 positive (grammatical) items, translated from the original set of 107 English sentences (one English sentence may have two or more Indonesian translations), that are meant to cover basic semantic phenomena.¹ The coverage of the MRS test-suite is 70%.² The other test-suites contain a collection of phenomena that are specific to Indonesian such as control and raising constructions, serial verb constructions, copula constructions, and *ada* sentences. The details are shown in Appendix A. It aims to be a resource grammar, for various NLP applications. In addition, INDRA is equipped with utilities for dealing with unknown words and a proof-of-concept transfer grammar for Indonesian-English machine translation. The current version of INDRA is open-source, on GitHub, and under the MIT license.

There are still many phenomena not yet analyzed and implemented in INDRA, such as adjective subcategorization, equative, comparative, superlative constructions, nominalized relative clause construction, and copula clauses with CP predicates. Even for phenomena which have been covered, such as noun and verb subcategorization, there is still room for improvement. At the present stage, INDRA covers only standard, formal Indonesian. One of the future directions is to develop INDRA into a shared, integrated grammar for both formal/standard and informal/colloquial Indonesian, similar to Zhong (Fan et al., 2015). Formal and informal Indonesian share a large amount of structure, though they may differ in morphology, lexicon, and syntax. It is advantageous to cover both the common parts of the grammars and the language specific descriptions. If INDRA can reach this stage, it can be applied to parse not only newspaper articles written in formal Indonesian, but also tweets written in informal Indonesian. A computational grammar can be employed to obtain higher accuracy and better results in building a sentiment analysis system for tweets, as noted in Le et al. (2016). In conclusion, this work is

¹<http://moin.delph-in.net/MatrixMrsTestSuiteIndonesian>

²As a comparison, the coverage of the MRS test-suite in Jacy is 93% (Siegel et al., 2016, p. 249). Jacy was firstly built in 1998, INDRA in 2014. Taking this into account, the 70% coverage of the MRS test-suite in INDRA can be considered good.

10.2 JATI

JATI is a treebank, built from a corpus of (a subset of) the KBBI dictionary definition sentences (or more precisely, noun phrases). I extracted 2,003 definition sentences about food and beverages from the KBBI database, which was created using Python and SQLite (Moeljadi et al., 2017). I used INDRA to parse this subset and the full forest treebanker (FFTB) (Packard, 2014) to treebank by selecting discriminants or features that distinguish between parses. Using INDRA as a base makes the treebanking consistent. The annotated sentences have well-formed parses. Out of 2,003 sentences, 62.6% or 1,253 sentences have well-formed parses and have been treebanked. The others could not be parsed or no correct parse could be found because some phenomena are not covered yet or not fully covered in INDRA at the present stage. The coverage in JATI can be increased by improving INDRA, i.e. analyzing phenomena not yet covered in INDRA and adding more rules for those phenomena. Well-documented annotation guidelines should be written in order to ensure consistency, i.e. that the same or similar linguistic phenomena are annotated in the same or similar ways throughout the corpus, as noted in Nivre (2008). The guidelines can also serve as a source of information for future users of JATI. In the future, JATI will be used for training a statistical model in order to parse other definition sentences and extract an ontology.

PART V:

BIBLIOGRAPHY AND APPENDICES

Bibliography

- Abas, Husen. 1987. *Indonesian as a unifying language of wider communication: a historical and sociolinguistic perspective* (Pacific Linguistics Series D 73). Canberra.
- Adelaar, Alexander. 2010. Structural Diversity in The Malayic Subgroup. In *The Austronesian languages of Asia and Madagascar*, 202–226. London and New York: Routledge Language Family Series.
- Aikhenvald, Alexandra Y. 1999. Serial constructions and verb compounding: Evidence from Tariana (North Arawak). *Studies in Language* 23(3). 469–498.
- Aikhenvald, Alexandra Y. & R. M. W. Dixon. 2006. *Serial verb constructions: A cross-linguistic typology*. Oxford University Press.
- Alwi, Hasan, Soenjono Dardjowidjojo, Hans Lapoliwa & Anton M. Moeliono. 2014. *Tata Bahasa Baku Bahasa Indonesia*. Jakarta: Balai Pustaka 3rd edn.
- Alwi, Hasan, Dendy Sugono & Sri Sukesu Adiwimarta. 2008. *Kamus Besar Bahasa Indonesia Dalam Jaringan (KBBI Daring)*. 3rd edn. URL <http://badanbahasa.kemdikbud.go.id/kbbi/>.
- Amalia, Dora (ed.). 2016. *Kamus Besar Bahasa Indonesia*. Jakarta: Badan Pengembangan dan Pembinaan Bahasa 5th edn.
- Arka, I Wayan. 2000. Control and argument structure: explaining control into subject in Indonesian. In *Fourth international symposium on malay/indonesian linguistics*, Jakarta.
- Arka, I Wayan. 2010a. Categorical multifunctionality in Indonesian: linguistic and implementation issues.
- Arka, I Wayan. 2010b. Dynamic and stative passives in Indonesian and their computational implementation.
- Arka, I Wayan. 2011a. On modality and finiteness in Indonesian: complexities in =nya nominalisation. In *Proceedings of the International Workshop on TAM and Evidentiality in Indonesian Languages*, 73–89.
- Arka, I Wayan. 2011b. On modality and finiteness in Indonesian: complexities of =nya nominalisation. In *Workshop on TAM markers and evidentiality in Indonesian languages*, 17–18. Tokyo University of Foreign Studies.

- Arka, I Wayan. 2012. Developing a Deep Grammar of Indonesian within the ParGram Framework: Theoretical and Implementational Challenges. In *26th Pacific Asia Conference on Language, Information and Computation*, 19–38.
- Arka, I Wayan. 2013a. Nonverbal predicates in Austronesian and Papuan languages: an LFG perspective, 30–46. Unud, Bali.
- Arka, I Wayan. 2013b. On the typology and syntax of TAM in Indonesian 55. 23–40.
- Arka, I Wayan. 2014. Computational implementation of crossed control structures in Indonesian. In *Cahaya Bahasa: a festschrift in honour of Prof. Sutjaja*, 21–27. Denpasar: Swasta Nulus.
- Arka, I Wayan, Mary Dalrymple, Mistica Meladel, Surriel Mofu & J. Simpson. 2009. A linguistic and computational morphosyntactic analysis for the applicative -i in Indonesian. In *Proceedings of the LFG 09 Conference*, Cambridge.
- Arka, I Wayan & C. D. Manning. 2008. Voice and grammatical relations in Indonesian: a new perspective. In *Voice and Grammatical Relations in Austronesian Languages*, 45–69. Stanford: CSLI Publications.
- Arka, I Wayan & Mistica Meladel. 2011. Aspects of negation in Indonesian.
- Atkins, B.T.S. & Michael Rundell. 2008. *The Oxford Guide to Practical Lexicography*. Oxford: Oxford University Press.
- Bender, Emily M. 2001. *Syntactic Variation and Linguistic Competence: The Case of AAVE Copula Absence*: Stanford University dissertation.
- Bender, Emily M., Scott Drellishak, Antske Fokkens, Laurie Poulson & Safiyyah Saleem. 2010. Grammar customization. In *Research on Language and Computation*, 23–72. Netherlands: Springer.
- Bender, Emily M., Dan Flickinger & Stephan Oepen. 2002. The grammar matrix: an open-source starter-kit for the rapid development of cross-linguistically consistent broad- coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 8–14. Taipei.
- Bender, Emily M., Dan Flickinger & Stephan Oepen. 2011. Grammar Engineering and Linguistic Hypothesis Testing: Computational Support for Complexity in Syntactic Analysis. In *Language from a Cognitive Perspective: Grammar, Usage and Processing*, 5–29. Stanford: CSLI Publications.

- Bender, Emily M. & Antske Sibelle Fokkens. 2010. The LinGO Grammar Matrix: Rapid Grammar Development for Hypothesis Testing. URL <http://www.delph-in.net/matrix/hpsg2010/hpsg-tutorial.pdf>.
- Bird, Steven, Edward Loper & Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc. URL <http://www.nltk.org/book/>.
- Bond, Francis & Ryan Foster. 2013. Linking and extending an open multilingual wordnet, 1352–1362. Sofia.
- Bond, Francis, Sanae Fujita, Chikara Hashimoto, Kaname Kasahara, Shigeko Nariyama, Eric Nichols, Akira Ohtani, Takaaki Tanaka & Shigeaki Amano. 2004. The Hinoki Treebank: A Treebank for Text Understanding. In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, 158–167. Springer Verlag Lecture Notes in Computer Science.
- Bond, Francis, Lian Tze Lim, Enya Kong Tang & Hammam Riza. 2014. The combined wordnet bahasa. *NUSA: Linguistic studies of languages in and around Indonesia* 57. 83–100.
- Bond, Francis & Kentaro Ogura. 2007. Combining linguistic resources to create a machine-tractable Japanese-Malay dictionary. *Language Resources and Evaluation* 42(2). 127–136.
- Bond, Francis & Kyonghee Paik. 2012. A survey of wordnets and their licenses. In *Proceedings of the sixth Global Wordnet Conference (GWC 2012)*, 64–71. Matsue.
- Callmeier, Ulrich. 2000. PET - a platform for experimentation with efficient HPSG processing techniques 6(1). 99–107.
- Candito, Marie, Fabienne Cap, Silvio Cordeiro, Vassiliki Foufi, Polona Gantar, Voula Giouli, Carlos Herrero, Mihaela Ionescu, Verginica Mititelu, Johanna Monti, Joakim Nivre, Mihaela Onofrei, Carla Parra Escartín, Manfred Sailer, Carlos Ramisch, Monica-Mihaela Rizea, Agata Savary, Ivelina Stonayova, Sara Stymne & Veronika Vincze. 2016. PARSEME shared task - annotation guidelines. Accessed on 2017-01-11 (ver 1.6b). URL <http://parsemefr.lif.univ-mrs.fr/guidelines-hypertext/>.
- Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory* 2(3). 113–124.
- Chu, Benjamin, Qiang Liu, Rohana Mahmud, Arun Anand, Weng Onn Kow & Dickson Lukose. 2014. Malay semantic text processing engine. In *eKNOW 2014, the Sixth International Conference on Information, Process, and Knowledge Management*, 38–43. Barcelona.

- 254
- Cole, Peter, Gabriella Hermon & Yanti. 2008. Voice in Malay/Indonesian. *Lingua* 118(10). 1500–1553.
- Copestake, Ann. 2000. Appendix: Definitions of Typed Feature Structures 6. 109–112.
- Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. Stanford: CSLI Publications.
- Copestake, Ann & Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second Conference on Language Resources and Evaluation (LREC-2000)*, Athens.
- Copestake, Ann, Dan Flickinger, Carl Pollard & Ivan A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation* 3(4). 281–332.
- Dalrymple, Mary. 2001. Lexical-Functional Grammar. *Syntax and Semantics* 34. Academic Press.
- Daudé, Jordi, Lluís Padró & German Rigau. 2003. Validation and tuning of wordnet mapping techniques. In *Proceedings of The International Conference on Recent Advances in Natural Language Processing (RANLP'03)*, Borovets, Bulgaria.
- Dinakaramani, Arawinda, Fam Rashel, Andry Luthfi, Bayu Distiawan & Ruli Manurung. 2016. Developing (and Utilizing) an Indonesian Treebank. In *The Second Wordnet Bahasa Workshop*, Nanyang Technological University, Singapore.
- Dixon, R. M. W. 2009. *Basic Linguistic Theory 2: Grammatical Topics*. Oxford: OUP Oxford.
- Drellishak, Scott. 2009. *Widespread but Not Universal: Improving the Typological Coverage of the Grammar Matrix*. Seattle: University of Washington PhD dissertation.
- Drellishak, Scott & Emily M. Bender. 2005. A Coordination Module for a Crosslinguistic Grammar Resource. In Stefan Müller (ed.), *Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar*, 108–128. Stanford, CA: CSLI Publications.
- Dryer, Matthew S. 2007. Clause types. *Language Typology and Syntactic Description* 1. 224–275. Cambridge University Press.
- Englebretson, Robert. 2003. *Searching for structure: The problem of complementation in colloquial Indonesian conversation*. John Benjamins Publishing.

- BIBLIOGRAPHY 233
- Fan, Zhenzhen, Sanghoun Song & Francis Bond. 2015. An HPSG-based Shared-Grammar for the Chinese Languages: Zhong []. In *Proceedings of the Grammar Engineering Across Frameworks (GEAF) Workshop, 53rd Annual Meeting of the ACL and 7th IJCNLP*, 17–24.
- Fellbaum, Christiane (ed.). 1998a. *WordNet: An Electronic Lexical Database*. MIT Press.
- Fellbaum, Christiane. 1998b. *WordNet: an electronic lexical database*. Cambridge: MIT Press. URL <http://wordnet.princeton.edu/man/wninput.5WN.html>.
- Fellbaum, Christiane. 2005. WordNet and wordnets. In *Encyclopedia of language and linguistics*, 665–670. Oxford: Elsevier 2nd edn. URL <http://wordnet.princeton.edu>.
- Fellbaum, Christiane & Piek Vossen. 2007. Connecting the universal to the specific: Towards the global grid. In *First International Workshop on Intercultural Collaboration (IWIC-2007)*, 2–16. Kyoto.
- Ferguson, Charles A. 1959. Diglossia 15. 325–340.
- Flickinger, Dan. 2000. On Building a More Efficient Grammar by Exploiting Types 6(1). 15–28.
- Flickinger, Dan, Yi Zhang & Valia Kordoni. 2010. Grammar Engineering for Deep Linguistic Processing: Project Seminar 2010. URL <http://www.coli.uni-saarland.de/~yzhang/ge-ss10/lecture-01.pdf>.
- Fokkens, Antske, Emily M. Bender & Varya Gracheva. 2012. LinGO Grammar Matrix Customization System Documentation.
- Fokkens, Antske Sibelle. 2014. *Enhancing Empirical Research for Linguistically Motivated Precision Grammars*. Saarbrücken: Department of Computational Linguistics, Universität des Saarlandes PhD dissertation.
- Fujita, Sanae & Francis Bond. 2007. A method of creating new valency entries. *Machine Translation* 21(1). 1–28.
- Fujita, Sanae, Francis Bond, Stephan Oepen & Takaaki Tanaka. 2007. Exploiting Semantic Information for HPSG Parse Selection. In *Proceedings of the Workshop on Deep Linguistic Processing*, 25–32. Association for Computational Linguistics.
- Givón, Talmy. 1980. The binding hierarchy and the typology of complements. *Studies in Language* 4(3). 33–77.

- Green, Nathan, Septina Dian Larasati & Zdeněk Žabokrtský. 2012. Indonesian Dependency Treebank: Annotation and Parsing. In *26th Pacific Asia Conference on Language, Information and Computation*, 137–145.
- Hammarström, Harald, Robert Forkel, Martin Haspelmath & Sebastian Bank (eds.). 2013. *Glottolog 2.4*. Leipzig: Max Planck Institute for Evolutionary Anthropology. URL <http://glottolog.org/>.
- Hashimoto, Chikara, Francis Bond, Takaaki Tanaka & Melanie Siegel. 2005. Integration of a lexical type database with a linguistically interpreted corpus. In *6th International Workshop on Linguistically Integrated Corpora (LINC-2005)*, 31–40. Cheju, Korea.
- Hopper, Paul J. 1972. Verbless stative sentences in Indonesian. *The verb 'be' and its synonyms* 115–152.
- Ikehara, Satoru, Masahiro Miyazaki, Akio Yokoo, Satoshi Shirai, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama & Yoshihiko Hayashi. 1997. *Goi-Taikei – A Japanese Lexicon. 5 volumes/CDROM*.
- Johns, A. H. (ed.). 2000. *Kamus Inggeris Melayu Dewan*. Kuala Lumpur: Dewan Bahasa dan Pustaka.
- Jurafsky, Daniel & James H. Martin. 2009. *Speech and Language Processing*. New Jersey: Pearson Education, Inc. 2nd edn.
- Kamajaya, Ian, David Moeljadi & Dora Amalia. 2017. KBBI Daring: A Revolution in The Indonesian Lexicography. In *Electronic lexicography in the 21st century. Proceedings of eLex 2017 conference.*, 513–530. Leiden. URL <https://elex.link/elex2017/wp-content/uploads/2017/09/paper31.pdf>.
- Kang, In-Su, Sin-Jae Kang, Se-Jin Nam & Key-Sun Choi. 2010. Linking CoreNet to WordNet through KorLex—some aspects and interim consideration. In Pushpak Bhat-tacharyya, Christiane Fellbaum & Piek Vossen (eds.), *5th Global Wordnet Conference: GWC-2010*, Mumbai: Narosa Pub.
- Kaplan, Ronald & Joan Bresnan. 1982. Lexical Functional Grammar: A formal system for grammatical representation. In *The Mental Representation of Grammatical Relations*, 173–281. Cambridge: the MIT Press.
- Kridalaksana, Harimurti. 1989. *Pembentukan kata dalam bahasa Indonesia*. Gramedia.
- Krieger, Hans-Ulrich & Ulrich Schäfer. 1994. TDL—A type description language for constraint-based grammars. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, 893–899. Kyoto.

- BIBLIOGRAPHY 237
- Kroeger, Paul. 2004. *Analyzing syntax: A lexical-functional approach*. Cambridge University Press (Cambridge).
- Kroeger, Paul. 2014. External negation in Malay/Indonesian. *Language* 90(1). 137–184.
- Lafourcade, M., G. Sérasset, L. Metzger, A. Rahman & C. K. Chuah. 2003. Dictionnaire Français-Anglais-Malais (FeM) – version 2. *CD-ROM, Dictionnaire en version XML et Application Java*. (online at <http://www-clips.imag.fr/cgi-bin/geta/fem/fem.pl?lang=en>).
- Larasati, Septina Dian, Vladislav Kuboň & Daniel Zeman. 2011. Indonesian Morphology Tool (MorphInd): Towards an Indonesian Corpus. *Springer CCIS proceedings of the Workshop on Systems and Frameworks for Computational Morphology* 119–129.
- Le, Tuan Anh, David Moeljadi, Yasuhide Miura & Tomoko Ohkuma. 2016. Sentiment analysis for low resource languages: A study on informal Indonesian tweets. In *Proceedings of the 12th Workshop on Asian Language Resources*, 123–131.
- Lewis, M. Paul. 2009. *Ethnologue: Languages of the World*. Dallas, Texas: SIL International 16th edn. URL <http://www.ethnologue.com>.
- Liaw, Yock Fang. 2004. *Indonesian Grammar Made Easy*. Singapore: Times Editions.
- Lim, Lian Tze, Ruoh Tau Chiew, Enya Kong Tang, Abdul Ghani Rusli & Yusof Naimah. 2016. Digitising a machine-tractable version of Kamus Dewan with TEI-P5. *PeerJ Preprints* 4. e2205v1. doi: 10.7287/peerj.preprints.2205v1. URL <https://doi.org/10.7287/peerj.preprints.2205v1>.
- Lim, Lian Tze & Nur Hussein. 2006. Fast prototyping of a Malay wordnet system. In *Proceedings of the Language, Artificial Intelligence and Computer Science for Natural Language Processing (laics-nlp) Summer School Workshop*, 13–16.
- Macdonald, R. Ross. 1976. *Indonesian Reference Grammar*. Washington: Georgetown University Press 2nd edn.
- Marcus, Mitchell P, Mary Ann Marcinkiewicz & Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics* 19(2). 313–330.
- McCune, Keith. 1979. Passive Function and the Indonesian Passive. *Oceanic Linguistics* 18(2). 119–169.
- Meladel, Mistica, I Wayan Arka, T. Baldwin & A. Andrews. 2009. Double Double, Morphology and Trouble: Looking into Reduplication in Indonesian. In *Proceedings of the Australasian Language Technology Workshop (ALTW 2009)*, 44–52. Sydney.

- Mintz, Malcolm W. 2002. *An Indonesian and Malay Grammar for Students*. Perth: Indonesian / Malay Texts and Resources 2nd edn.
- Mistica, Meladel. 2013. *An investigation into deviant morphology: Issues in the implementation of a deep grammar for Indonesian*. Canberra: The Australian National University PhD dissertation.
- Moeljadi, David. 2014a. How can we improve the Wordnet Bahasa? In *Workshop/hackathon for the wordnet bahasa*, Nanyang Technological University, Singapore.
- Moeljadi, David. 2014b. Usage of Indonesian Possessive Verbal Predicates: A Statistical Analysis Based on Storytelling Survey. *Tokyo University Linguistic Papers* 35. 155–176.
- Moeljadi, David. 2017. Building JATI: A Treebank for Indonesian. In *Proceedings of The 4th Atma Jaya Conference on Corpus Studies (ConCorps 4)*, 1–9. Jakarta.
- Moeljadi, David & Francis Bond. 2016. Identifying and Exploiting Definitions in Wordnet Bahasa. In *Proceedings of the Eighth Global WordNet Conference*, 226–232. Bucharest.
- Moeljadi, David & Francis Bond. 2018. HPSG Analysis and Computational Implementation of Indonesian Passives. In *Proceedings of the 25th International Conference on Head-Driven Phrase Structure Grammar*, 129–139. Tokyo.
- Moeljadi, David, Francis Bond & Luís Morgado da Costa. 2016. Basic copula clauses in Indonesian. In *Proceedings of the Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar*, 442–456. Warsaw.
- Moeljadi, David, Francis Bond & Sanghoun Song. 2015. Building an HPSG-based Indonesian Resource Grammar (INDRA). In *Proceedings of the Grammar Engineering Across Frameworks (GEAF) Workshop, 53rd Annual Meeting of the ACL and 7th IJCNLP*, 9–16. URL <http://aclweb.org/anthology/W/W15/W15-3302.pdf>.
- Moeljadi, David, Ian Kamajaya & Dora Amalia. 2017. Building the Kamus Besar Bahasa Indonesia (KBBI) Database and Its Applications. In Hai Xu (ed.), *Proceedings of the 11th International Conference of the Asian Association for Lexicography*, 64–80. the Asian Association for Lexicography Center for Linguistics and Applied Linguistics, Guangdong University of Foreign Studies.
- Moeljadi, David, Lian Tze Lim, Hannah Choi & Francis Bond. forthcoming. Wordnet Bahasa. *Linguistic Issues in Language Technology (LiLT)*.
- Moeljadi, David & Viola Ow. 2018. Serial verb constructions in Indonesian: An HPSG analysis and its computational implementation. *Journal of the Southeast Asian Linguistics Society*.

- Montolalu, Lucy R. & Leo Suryadinata. 2007. National Language and Nation-Building: The Case of Bahasa Indonesia. In *Language, Nation and Development in Southeast Asia*, 39–50. Singapore: Institute of Southeast Asian Studies.
- Muhammad Zulhelmy bin Mohd Rosman, Francis Bond & František Kratochvíl. 2014. Bringing together over- and under-represented languages: Linking Wordnet to the SIL semantic domains. In *Proceedings of the 7th Global Wordnet Conference (GWC 2014)*, 40–48. Tartu.
- Müller, Stefan & Janna Lipenkova. 2009. Serial verb constructions in Chinese: A HPSG account. In *Proceedings of the 16th international conference on head-driven phrase structure grammar*, 234–254. CSLI Publications.
- Musgrave, Simon. 2001. *Non-subject arguments in Indonesian*. Melbourne: The University of Melbourne PhD dissertation.
- Nivre, Joakim. 2008. Treebanks. In Anke Lüdeling & Merja Kytö (eds.), *Corpus linguistics: An international handbook*, vol. 1, chap. 13, 225–241. Berlin: Walter de Gruyter.
- Nomoto, Hiroki. 2013. On the optionality of grammatical markers: A case study of voice marking in Malay/Indonesian. *Voice variation in Austronesian languages of Indonesia* 121–143.
- Nomoto, Hiroki, Hannah Choi, David Moeljadi & Francis Bond. 2018a. MALINDO Morph: Morphological dictionary and analyser for Malay/Indonesian. In *Proceedings of the LREC 2018 Workshop “The 13th Workshop on Asian Language Resources”*, 36–43.
- Nomoto, Hiroki, Kenji Okano, David Moeljadi & Hideo Sawada. 2018b. TUFs Asian Language Parallel Corpus (TALPCo). In *Proceedings of the Twenty-fourth Annual Meeting of the Association for Natural Language Processing*, 436–439.
- Nur Amirah Binte Khairul Anuar, Hannah Choi & František Kratochvíl. 2017. Verb sub-categorization: -kan and -i suffixing verbs in Malay and Indonesian. Presented at The Twenty-First International Symposium On Malay/Indonesian Linguistics (ISMIL 21).
- Nurhil Hirfana Mohamed Noor, Suerya Sapuan & Francis Bond. 2011. Creating the open Wordnet Bahasa. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation (PACLIC 25)*, 258–267. Singapore.
- Ó Séaghdha, Diarmuid. 2007. Annotating and learning compound noun semantics. In *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop*, 73–78. Association for Computational Linguistics.

- Ó Séaghdha, Diarmuid & Ann Copestake. 2007. Co-occurrence contexts for noun compound interpretation. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, 57–64. Association for Computational Linguistics.
- Oepen, Stephan, Dan Flickinger & Francis Bond. 2004. Towards holistic grammar engineering and testing—grafting treebank maintenance into the grammar revision cycle. In *Beyond Shallow Analyses—Formalisms and Statistical Modelling for Deep Analysis (Workshop at IJCNLP-2004)*, Hainan Island.
- Oepen, Stephan, Dan Flickinger, Kristina Toutanova & Christopher D. Manning. 2002. LinGO Redwoods: A Rich and Dynamic Treebank for HPSG. In *Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT2002)*, Sozopol, Bulgaria.
- Oepen, Stephan & Daniel Flickinger. 1998. Towards systematic grammar profiling: Test suite technology ten years after 12(4). 411–436.
- Oepen, Stephan, Erik Velldal, Jan Tore Lønning, Paul Meurer, Victoria Rosén & Dan Flickinger. 2007. Towards hybrid quality-oriented machine translation: On linguistics and probabilities in MT. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, 144–153. Skövde, Sweden.
- Olsson, Bruno & David Moeljadi. 2014. A parallel corpus approach to sudah. Presented at The Eighteenth International Symposium On Malay/Indonesian Linguistics (ISMIL 18).
- Paauw, Scott H. 2009. *The Malay contact varieties of Eastern Indonesia: A typological comparison*. State University of New York at Buffalo PhD dissertation.
- Packard, Woodley. 2013. ACE, the Answer Constraint Engine. URL <http://sweaglesw.org/linguistics/ace/>.
- Packard, Woodley. 2014. FFTB: the full forest treebanker. URL <http://moin.delph-in.net/FftbTop>.
- Payne, Thomas E. 2008. *Describing Morphosyntax: A Guide for Field Linguists*. New York: Cambridge University Press.
- Petrolito, Tommaso & Francis Bond. 2014. A survey of wordnet annotated corpora. In *Proceedings of the Global WordNet Conference (GWC-2014)*, 236–245.
- Polinsky, Maria & Eric Potsdam. 2008. The syntax and semantics of wanting in Indonesian. *Lingua* 118(10). 1617–1639.
- Pollard, Carl & Ivan A Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

- Quah, Chiew Kin, Francis Bond & Takefumi Yamazaki. 2001. Design and construction of a machine-tractable Malay-English lexicon. In *Asialex 2001 Proceedings*, 200–205. Seoul.
- Rashel, Fam, Andry Luthfi, Arawinda Dinakaramani & Ruli Manurung. 2014. Building an Indonesian Rule-Based Part-of-Speech Tagger, Kuching.
- Riesberg, Sonja. 2014. *Symmetrical Voice and Linking in Western Austronesian Pacific Linguistics* 646. Berlin: Mouton de Gruyter.
- Riza, Hammam, Budiono & Chairil Hakim. 2010a. Collaborative work on Indonesian wordnet through Asian wordnet (AWN). In *Proceedings of the 8th Workshop on Asian Language Resources*, 9–13.
- Riza, Hammam, Budiono & Chairil Hakim. 2010b. Collaborative work on Indonesian WordNet through Asian WordNet (AWN). In *Proceedings of the 8th Workshop on Asian Language Resources*, 9–13. Beijing, China: Asian Federation for Natural Language Processing.
- Riza, Hammam, Michael Purwoadi, Gunarso, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thái, Vichet Chea, Rapid Sun, Sethserey Sam, Sopheap Seng et al. 2016. Introduction of the Asian Language Treebank. In *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Technique (O-COCOSDA)*, 1–6. IEEE.
- Roskov, Y., L. Abucay, T. Orrell, D. Nicolson, T. Kunze, A. Culham, N. Bailly, P. Kirk, T. Bourgoin, R.E. DeWalt, W. Decock & A. De Wever (eds.). 2015. *Species 2000 & ITIS Catalogue of Life, 2015 annual checklist*. Leiden, the Netherlands: Species 2000: Naturalis. URL <http://www.catalogueoflife.org/annual-checklist/2015>.
- Sag, Ivan A. 1997. English relative clause constructions. *Journal of linguistics* 33(2). 431–483.
- Sag, Ivan A., Thomas Wasow & Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. Stanford: CSLI Publications 2nd edn.
- Sagot, Benoît & Darja Fišer. 2008. Building a free French wordnet from multilingual resources. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, .
- Saloot, Mohammad Arshi, Norisma Idris & Rohana Mahmud. 2014. An architecture for Malay Tweet normalization. *Information Processing & Management* 50(5). 621–633.

- Schlangen, David & Alex Lascarides. 2003. A compositional and constraint-based approach to non-sentential utterances. In Stefan Müller (ed.), *Proceedings of the 10th International Conference on Head-Driven Phrase Structure Grammar*, 380–390. Stanford, CA: CSLI Publications.
- Seah, Yu Jie & Francis Bond. 2014a. Annotation of pronouns in a multilingual corpus of Mandarin Chinese, English and Japanese. In *10th Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, Reykjavik.
- Seah, Yu Jie & Francis Bond. 2014b. Annotation of Pronouns in a Multilingual Corpus of Mandarin Chinese, English and Japanese.
- Siegel, Melanie, Emily M. Bender & Francis Bond. 2016. *Jacy: An Implemented Grammar of Japanese*. Stanford: CSLI Publications.
- Slayden, Glenn. 2012. *Array TFS Storage for Unification Grammars*: University of Washington Masters Thesis.
- Sneddon, James Neil. 2006. *Colloquial Jakartan Indonesian*. Canberra: Pacific Linguistics.
- Sneddon, James Neil, Alexander Adelaar, Dwi Noverini Djenar & Michael C. Ewing. 2010. *Indonesian Reference Grammar*. New South Wales: Allen & Unwin 2nd edn.
- Soderberg, Craig D. & Kenneth S. Olson. 2008. Indonesian. *Journal of the International Phonetic Association* 38(2). 209–213.
- Song, Sanghoun. 2015a. JacyYYMode. URL <http://moin.delph-in.net/JacyYYMode>.
- Song, Sanghoun. 2015b. ZhongYYMode. URL <http://moin.delph-in.net/ZhongYYMode>.
- Song, Sanghoun. 2017. *Modeling information structure in a cross-linguistic perspective*.
- Spärck Jones, Karen. 1983. Compound noun interpretation problems. Tech. rep. University of Cambridge, Computer Laboratory.
- Sproat, Richard, Christer Samuelsson, Jennifer Chu-Carroll & Bob Carpenter. 2004. Computational Linguistics. In *The Handbook of Linguistics* Blackwell Handbooks in Linguistics, 608–636. Oxford: Blackwell Publishing.
- Sugono, Dendy (ed.). 2008. *Kamus Besar Bahasa Indonesia Pusat Bahasa*. Jakarta: PT Gramedia Pustaka Utama 4th edn.

- Sugono, Dendy, Sugiyono & Meity Taqdir Qodratillah (eds.). 2014. *Kamus Besar Bahasa Indonesia Pusat Bahasa*. Jakarta: PT Gramedia Pustaka Utama 4th edn.
- Sulger, Sebastian, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh M Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt et al. 2013. Pargrambank: The pargram parallel treebank. In *Acl (1)*, 550–560.
- Tan, Liling & Francis Bond. 2012. Building and annotating the linguistically diverse NTU-MC (NTU-multilingual corpus). In *International Journal of Asian Language Processing*, vol. 22 4, 161–174.
- Tjung, Yassir Nasanius. 2003. On Ada-Constructions in Indonesian. *Cakrawala Baru: Liber Amicorum untuk Prof. Soenjono Dardjowidjojo Ph.D.* 261–281.
- Van Eynde, Frank. 2009. On the copula: from a Fregean to a Montagovian treatment. In Stefan Müller (ed.), *Proceedings of the 16th International Conference on HPSG*, 359–375. URL <http://cslipublications.stanford.edu/HPSG/2009/vaneynde.pdf>.
- Voskuil, Jan E. 2000. Indonesian voice and A-bar movement. In *Formal issues in austronesian linguistics*, 195–213. Springer.
- Vossen, Piek (ed.). 1998. *Euro wordnet*. Kluwer.
- Wang, Shan & Francis Bond. 2014. Building the sense-tagged multilingual parallel corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, 2403–2409.
- Wasow, Thomas. 2004. Generative Grammar. In *The Handbook of Linguistics* Blackwell Handbooks in Linguistics, 295–318. Oxford: Blackwell Publishing.
- Xu, Renjie, Zhiqiang Gao, Yingji Pan, Yuzhong Qu & Zhisheng Huang. 2008. An integrated approach for automatic construction of bilingual Chinese-English WordNet. In *3rd Asian Semantic Web Conference (ASWC 2008)*, 302–314.
- Zwicky, Arnold M. & Geoffrey K. Pullum. 1983. Cliticization vs. Inflection: English N'T. *Language* 59(3). 502–513.

Appendix A

INDRA Meta-information

(as of 23 January 2018)

Maintainer	David Moeljadi
Contributors	David Moeljadi, Sanghoun Song, Francis Bond, Michael Goodman, Dan Flickinger, Luis Morgado da Costa
Contact email	davidmoeljadi@gmail.com
Grammar name	Indonesian Resource Grammar
Short grammar name	INDRA
Language name	Indonesian
ISO code	ind
Repository	github.com/davidmoeljadi/INDRA
Latest revision	23 January 2018
License	MIT (svn.delph-in.net/erg/trunk/LICENSE)
Grammar type	Medium-sized grammar
Required external resources	Indonesian POS Tagger (for unknown word handling)
Lexical items	16,751 items, 939 of which were extracted from ERG via Wordnet Bahasa
Lexical rules	12
Grammar rules	63
Features	168
Types	2,057 types
Associated resources	transfer grammars for machine translation, unknown word handling with YY mode
Test-suites	
matrix test-suite	2 positive items and 1 negative item 100% coverage

(as of 23 January 2018)

lab3 test-suite	66 positive items and 40 negative items 77.3% coverage
mrs test-suite	172 positive items 70.9% coverage
controlraising test-suite	8 positive items and 2 negative items 37.5% coverage
SVC test-suite	30 positive items 90.0% coverage
TBBI_copula test-suite	38 positive items and 5 negative items 47.4% coverage
Sneddon_ada test-suite	21 positive items 57.1% coverage
Sneddon_clauses test-suite	59 positive items 28.8% coverage
ntumc test-suite	2197 positive items 5.9% coverage
JATI test-suite	2003 positive items 84.5% coverage

Appendix B

Letter of permission

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

BADAN PENGEMBANGAN DAN PEMBINAAN BAHASA

Jalan Daksinapati Barat IV, Rawamangun, Jakarta Timur 13220

Telepon: (021) 4706287, 4706288, 4896558; Faksimile: (021) 4750407

Laman: www.badanbahasa.kemdikbud.go.id; Pos-el: badan.bahasa@kemdikbud.go.id

Nomor: 8775 / G2.1 / B5 / 2017

9 Agustus 2017

Hal : Tanggapan Permohonan Penggunaan
Data KBBI V

Yth. David Moeljadi
Division of Linguistics and Multilingual Studies
School of Humanities and Social Sciences
Nanyang Technological University
14 Nanyang Drive, HSS Level 3
Singapore 637332

Dengan hormat,

Sehubungan dengan permohonan penggunaan data KBBI V yang Saudara ajukan melalui pos-el tertanggal 21 Juli 2017 untuk keperluan penelitian S-3 Saudara, kami mengizinkan Saudara untuk menggunakan data KBBI berupa lema (*headword*) yang berkaitan dengan makanan dan minuman beserta definisinya. Data tersebut akan digunakan untuk mengembangkan *treebank* bahasa Indonesia dengan INDRA (*Indonesian Resource Grammar*). Kami mempersilakan Saudara memublikasikan hasil penelitian *treebank* tersebut dengan lisensi MIT dan hak cipta tetap menjadi milik Badan Pengembangan dan Pembinaan Bahasa, Kementerian Pendidikan dan Kebudayaan Indonesia.

Berkaitan dengan hal itu, kami mengimbau agar data tersebut dipergunakan sebaik mungkin dan semata-mata untuk tujuan penelitian agar dapat berguna untuk penelitian linguistik bahasa Indonesia.

Salam kami,
Kepala Bidang Pengembangan,

Dr. Dora Amalia
Dr. Dora Amalia
NIP 197107292002122001

Appendix C

List of publications and presentations

Papers

Journal articles

- Moeljadi, David, Lian Tze Lim, Hannah Choi & Francis Bond. forthcoming. Word-net Bahasa. *Linguistic Issues in Language Technology (LiLT)*
- Moeljadi, David & Viola Ow. 2018. Serial verb constructions in Indonesian: An HPSG analysis and its computational implementation. *Journal of the Southeast Asian Linguistics Society*
- Moeljadi, David. 2014b. Usage of Indonesian Possessive Verbal Predicates: A Statistical Analysis Based on Storytelling Survey. *Tokyo University Linguistic Papers* 35. 155–176

Conference proceedings

- Moeljadi, David & Francis Bond. 2018. HPSG Analysis and Computational Implementation of Indonesian Passives. In *Proceedings of the 25th International Conference on Head-Driven Phrase Structure Grammar*, 129–139. Tokyo
- Nomoto, Hiroki, Hannah Choi, David Moeljadi & Francis Bond. 2018a. MALINDO Morph: Morphological dictionary and analyser for Malay/Indonesian. In *Proceedings of the LREC 2018 Workshop “The 13th Workshop on Asian Language Resources”*, 36–43
- Nomoto, Hiroki, Kenji Okano, David Moeljadi & Hideo Sawada. 2018b. TUFSA Asian Language Parallel Corpus (TALPCo). In *Proceedings of the Twenty-fourth Annual Meeting of the Association for Natural Language Processing*, 436–439

- Kamajaya, Ian, David Moeljadi & Dora Amalia. 2017. KBBI Daring: A Revolution in The Indonesian Lexicography. In *Electronic lexicography in the 21st century. Proceedings of eLex 2017 conference.*, 513–530. Leiden. URL <https://elex.link/elex2017/wp-content/uploads/2017/09/paper31.pdf>
- Moeljadi, David. 2017. Building JATI: A Treebank for Indonesian. In *Proceedings of The 4th Atma Jaya Conference on Corpus Studies (ConCorps 4)*, 1–9. Jakarta
- Moeljadi, David, Ian Kamajaya & Dora Amalia. 2017. Building the Kamus Besar Bahasa Indonesia (KBBI) Database and Its Applications. In Hai Xu (ed.), *Proceedings of the 11th International Conference of the Asian Association for Lexicography*, 64–80. the Asian Association for Lexicography Center for Linguistics and Applied Linguistics, Guangdong University of Foreign Studies
- Le, Tuan Anh, David Moeljadi, Yasuhide Miura & Tomoko Ohkuma. 2016. Sentiment analysis for low resource languages: A study on informal Indonesian tweets. In *Proceedings of the 12th Workshop on Asian Language Resources*, 123–131
- Moeljadi, David, Francis Bond & Luís Morgado da Costa. 2016. Basic copula clauses in Indonesian. In *Proceedings of the Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar*, 442–456. Warsaw
- Moeljadi, David & Francis Bond. 2016. Identifying and Exploiting Definitions in Wordnet Bahasa. In *Proceedings of the Eighth Global WordNet Conference*, 226–232. Bucharest
- Moeljadi, David, Francis Bond & Sanghoun Song. 2015. Building an HPSG-based Indonesian Resource Grammar (INDRA). In *Proceedings of the Grammar Engineering Across Frameworks (GEAF) Workshop, 53rd Annual Meeting of the ACL and 7th IJCNLP*, 9–16. URL <http://aclweb.org/anthology/W/W15/W15-3302.pdf>

Invited presentations

Keynotes

- Moeljadi, David (2017) Building JATI: A Treebank for Indonesian. Presented at The 4th Atma Jaya Conference on Corpus Studies (ConCorps 4), Pusat Kajian Bahasa dan Budaya, Program Studi Linguistik Terapan Bahasa Inggris, Fakultas Pendidikan dan Bahasa, Universitas Katolik Indonesia Atma Jaya, 21 July 2017.

Workshops

- Moeljadi, David and Takayuki Kuribayashi (2018) Introduction and demo of Jacy: an implemented HPSG grammar of Japanese. Presented at the Workshop on the Clause Structure of Japanese and Korean in The 25th International Conference on Head-Driven Phrase Structure Grammar (HPSG 2018), the University of Tokyo, Komaba Campus, Tokyo, 2 July 2018.
- Moeljadi, David (2016) WordNet (Bahasa): Pangkalan data leksikal (bahasa Indonesia/Melayu) [WordNet (Bahasa): (Indonesian/Malay) lexical database]. Presented at Wordnet Bahasa Workshop in Indonesian Lexicography Seminar, Jakarta, Indonesia, 28 July 2016.

Others

- Moeljadi, David (2016) Kamus Besar Bahasa Indonesia (KBBI): Dari Word+Excel ke Pangkalan Data, Aplikasi Daring+Luring (Dari KBBI IV ke KBBI V). Presented at Diseminasi Kamus Besar Bahasa Indonesia Edisi V, Badan Bahasa, Jakarta, 13 December 2016.
- Moeljadi, David (2016) Kamus Besar Bahasa Indonesia (KBBI): Dari Word+Excel ke Pangkalan Data, Aplikasi Daring+Luring (Dari KBBI IV ke KBBI V). Presented at Sosialisasi Kamus Besar Bahasa Indonesia (KBBI) Edisi Kelima bagi seluruh staf Badan Bahasa, Ruang Rapat Gedung Samudra, Rawamangun, Jakarta Timur, 7 December 2016.
- Moeljadi, David (2016) Pendaringan KBBI. Presented at Lokakarya Pemutakhiran Kamus Besar Bahasa Indonesia, Best Western Premier The Hive Hotel, Jakarta, 22 September 2016.
- Moeljadi, David (2016) Wordnet: Jejaring kata / Pangkalan data leksikal [Wordnet: Net of words / Lexical database]. Presented at Musyawarah Sekretariat Ke-22 Majelis Bahasa Brunei Darussalam-Indonesia-Malaysia (MABBIM), Millenium Hotel Sirih, Jakarta Pusat, Indonesia, 23 August 2016.
- Moeljadi, David (2016) Building an open-source computational grammar for Indonesian “INDRA“. Presented at The first LingDy Forum in 2016 at ILCAA, Tokyo, 5 April 2016.