

**MEMBANGUN PENGURAI DEPENDENSI UNTUK BAHASA
INDONESIA**

TESIS

**Karya tulis sebagai salah satu syarat
untuk memperoleh gelar Magister dari
Institut Teknologi Bandung**

Oleh

MIA KAMAYANI SULAEMAN

NIM : 23510084

(Program Studi Informatika)



**INSTITUT TEKNOLOGI BANDUNG
2012**

**MEMBANGUN PENGURAI DEPENDENSI UNTUK BAHASA
INDONESIA**

Oleh

Mia Kamayani Sulaeman

NIM : 23510084

(Program Studi Informasi)

Institut Teknologi Bandung

Tanggal, Maret 2012

Menyetujui

Pembimbing

DR. Eng. Ayu Purwarianti, ST. MT.
NIP. 197701272008012011

ABSTRAK

MEMBANGUN PENGURAI DEPENDENSI UNTUK BAHASA INDONESIA

Oleh

Mia Kamayani Sulaeman

NIM : 23510084

(Program Studi Informatika)

Pengurai dependensi (*dependency parser*) adalah proses analisis sintaksis dari kalimat suatu bahasa yang berprinsip bahwa suatu kata bergantung (*dependent*) pada kata lainnya (*head*). Kelebihan dependensi dibandingkan metode lain seperti konstituen adalah lebih mendekati ke level semantik karena terbentuk relasi *head* dan *dependent/modifier*. Relasi ini menunjukkan fungsi gramatikal antar kata yang tidak dapat ditunjukkan oleh pengurai berbasis konstituen yang hanya mengidentifikasi frase. Pengurai kalimat bahasa Indonesia yang sudah ada saat ini menggunakan pendekatan konstituen, sedangkan belum ada yang mengembangkan dengan metode dependensi.

Pengurai dependensi memiliki banyak pendekatan antara lain pemetaan dari struktur frase ke struktur dependensi yang digunakan di tesis ini. Metode ini terdiri dari 2 (dua) langkah yaitu *dependency extraction* dan *dependency typing*. *Dependency extraction* adalah proses mengubah struktur frase ke struktur dependensi dan *dependency typing* adalah proses pelabelan struktur dependensi sehingga menjadi pohon dependensi berlabel. Metode ini memanfaatkan perangkat pemrosesan bahasa yang sudah ada seperti *POS tagger*, pengurai konstituen (*constituency parser*). Selain itu, perangkat lainnya seperti *head rule* dan aturan pelabelan dependensi dapat diadaptasi dari bahasa Inggris yang telah dibuat oleh Stanford. Bahasa Inggris digunakan sebagai dasar *resource* karena kemiripan struktur antara kedua bahasa.

Hasil eksperimen menunjukkan bahwa tesis ini mampu memetakan struktur frase bahasa Indonesia untuk kalimat sederhana dan kalimat subordinatif relatif pendek ke struktur dependensi. Tingkat akurasi yang ditunjukkan oleh hasil eksperimen untuk 27 kalimat sederhana adalah 77 % dan untuk 13 kalimat subordinatif adalah 46 %. Pengujian dibandingkan dengan dataset yang dianotasi secara manual.

Kata kunci: *dependency parser, constituency parser, pemetaan struktur frase, dependency extraction, dependency typing, bahasa Indonesia.*

ABSTRACT

BUILDING DEPENDENCY PARSER FOR INDONESIAN LANGUAGE

Author

Mia Kamayani Sulaeman

NIM : 23510084

(Master Program of Informatics)

Dependency parser is a syntactical analysis process of a sentence which assume that a word is dependent on other words (head). It has advantages compared to others such as constituency-based parser, which is closer to semantic because relation between head and its dependent. This relation shows grammatical function between words which cannot be shown by constituency-based parser that can only identify phrase. Current Indonesian parsers use constituency-based, there isn't any published research yet in dependency parser.

Dependency parser has many approaches one of them is mapping phrase structure into dependency structure which is used in this thesis. This approach consists of 2 (two) steps, they are dependency extraction and dependency typing. This approach is quite efficient because it uses other available NLP tool for Indonesian such as POS tagger and constituency-based parser. Besides that, other tools such as head rule and dependency label adapted from English which has been built by Stanford were adapted in this thesis. The main reason this thesis uses English is because it has structure similarities with Indonesian.

Experiment result indicates that this thesis able to map Indonesia phrase structure for simple and short subordinative sentence to dependency structure. Accuracy of the experiment for simple sentences is 77% while for subordinative is 46%. Evaluation is compared with manually annotated dataset.

Keyword: *dependency parser, constituency parser, mapping phrase structure, dependency extraction, dependency typing, Indonesian language.*

KATA PENGANTAR

Puji syukur Penulis panjatkan ke hadirat Allah SWT, karena senantiasa memberikan rahmat dan karunia sehingga Penulis mampu menyelesaikan tesis yang berjudul "Membangun Pengurai Dependensi untuk Bahasa Indonesia". Laporan tesis ini disusun untuk memenuhi persyaratan kelulusan tingkat pascasarjana Program Studi Informatika ITB.

Dalam pelaksanaan tugas akhir ini, Penulis menemui banyak hambatan baik dari luar maupun dari dalam. Namun atas bantuan dari berbagai pihak, hambatan tersebut dapat diatasi sehingga tugas akhir ini dapat dilalui dengan baik. Oleh karena itu Penulis hendak memberikan ucapan terima kasih kepada:

1. Ibu DR. Eng. Ayu Purwarianti S.T., M.T., selaku pembimbing yang telah memberikan waktu dan pemikiran dalam memberikan bimbingan, saran dan perhatian dalam pelaksanaan tugas akhir ini.
2. Ibu Nur Ulfa Maulidevi dan Bapak Dwi Widyanto Hendratmo selaku penguji tesis.
3. Bapak Dr. Oerip S. Santoso M.Sc., selaku Koordinator Tesis.
4. Bapak Ade Tariyat selaku pegawai tata usaha S2 program studi Teknik Informatika ITB.
5. Ayah, ibu, nenek, kakek, adik-adik, bibi Penulis dan Faqih Akhsani, yang telah memberikan dukungan moral dan selalu menyemangati Penulis di dalam menyelesaikan tugas akhir ini juga atas perhatian dan doanya.
6. Alfian, Irfan, Filman yang telah membantu dalam pengerjaan tesis ini dengan memberikan perangkat-perangkat pemrosesan berupa data dan perangkat lunak pendukung.
7. Teman-teman IF S2 2010 yang telah mendukung penulis selama pengerjaan tesis, serta atas kebersamaannya dalam menjalani kuliah.
8. Seluruh staf Tata Usaha Teknik Informatika ITB yang telah membantu kelancaran administrasi hingga terselesaikannya tugas akhir ini.

9. Seluruh staf Perpustakaan Teknik Informatika ITB yang telah membantu kelancaran peminjaman pustaka yang dibutuhkan di dalam pelaksanaan tugas akhir.
10. Segenap staf dan karyawan Teknik Informatika ITB.
11. Semua teman dan berbagai pihak lainnya yang tidak dapat disebutkan satu persatu.

Semoga Allah SWT membalas budi baik semua pihak dengan rahmat-Nya.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan. Untuk itu, kritik dan saran berkaitan dengan tugas akhir ini sangat diharapkan. Akhir kata, Penulis berharap semoga tugas akhir ini dapat bermanfaat bagi siapapun yang membacanya.

Bandung, Maret 2012

Penulis

DAFTAR ISI

Bab I	Pendahuluan	14
I.1	Latar Belakang	14
I.2	Rumusan Masalah	16
I.3	Tujuan	16
I.4	Batasan Masalah.....	16
I.5	Metode Penelitian.....	16
I.6	Sistematika Pembahasan	17
Bab II	Tinjauan Pustaka	19
II.1	Pola Tata Bahasa Indonesia	19
II.1.1	Sintaksis	19
II.1.2	Kategori Klausa dan Frasa yang Digunakan.....	22
II.1.3	Penyesuaian Label Dependensi Bahasa Inggris ke Indonesia.....	24
II.2	Perkembangan Metoda Pengurai Kalimat Bahasa Indonesia.....	25
II.2.1	Indonesian GLR Parser (INAGP)	25
II.2.2	Pengurai Collins untuk Bahasa Indonesia.....	25
II.2.3	Pengurai Earley dan CYK untuk Bahasa Indonesia.....	26
II.3	Pengurai Dependensi (<i>Dependency Parsing</i>).....	27
Bab III	Analisis Solusi Pengurai Dependensi.....	31
III.1	Mekanisme Solusi Pengurai Dependensi	31
III.1.1	Preproses	32
III.1.2	<i>Dependency Extraction</i>	33
III.1.3	<i>Dependency Typing</i>	37
Bab IV	Pengujian.....	41
IV.1	Tujuan Pengujian	41
IV.2	Perancangan Pengujian	41
IV.3	Data Pengujian	41
IV.4	Hasil Pengujian	42
IV.4.1	Skenario 1	42
IV.4.2	Skenario 2	43

IV.5	Evaluasi Pengujian	44
IV.5.1	Kesalahan karena POS Tagger.....	44
IV.5.2	Kesalahan karena PCFG	45
IV.5.3	Kesalahan di Pengurai Dependensi.....	46
IV.5.4	Kesimpulan Pengujian	47
Bab V	Penutup.....	48
V.1	Kesimpulan	48
V.2	Saran.....	48

DAFTAR TABEL

Tabel II.1 Gatra Tambahan Yang Menerangkan Kata Kerja	22
Tabel II.2 Kategori Klausa	22
Tabel II.3 Kategori Frasa	23
Tabel II.4 Contoh Label Dependensi Non-Klausal Untuk Bahasa Indonesia.....	24
Tabel II.5 Label Non-Klausal Yang Tidak Dapat Diaplikasikan Pada Bahasa Indonesia	29
Tabel II.6 Contoh <i>Dependency Extraction</i> Pada Bahasa Inggris	30
Tabel II.7 Contoh <i>Dependency Typing</i> Pada Bahasa Inggris.....	30
Tabel III.1 Contoh Pcfg Bahasa Indonesia	32
Tabel III.2 Pohon Konstituen Keluaran Parser Earley.....	33
Tabel III.3 Aturan <i>Head</i> Untuk Frase Bahasa Indonesia	34
Tabel III.4 Daftar Cfg Yang Dibangkitkan Dari Pohon Konstituen	34
Tabel III.5 Aturan <i>Head</i>	35
Tabel III.6 Ilustrasi Pengaplikasian Aturan <i>Head</i>	35
Tabel III.7 Algoritma Pembentukan Dependensi Dari Cfg	36
Tabel III.8 Label Dependensi.....	38
Tabel III.9 Label Dependensi Klausal	39
Tabel IV.1 Data Pengujian	42
Tabel IV.2 Hasil Pengujian Skenario Kedua	42
Tabel IV.3 Hasil Pengujian Skenario Kedua	43
Tabel IV.4 Kesalahan Pos Tagger.....	44
Tabel IV.5 Kesalahan Pcfg Level Frasa.....	45
Tabel IV.6 Kesalahan Pcfg Level Klausa	45
Tabel IV.7 Label Dependensi Ambigu	46
Tabel IV.8 Contoh Kasus Label Ambigu.....	46

DAFTAR GAMBAR

Gambar III.1 Diagram Alir Proses Pengurai Dependensi	31
Gambar III.2 Pohon Dependensi Hasil <i>Dependency Extraction</i>	37
Gambar III.3 Pohon Dependensi Berlabel	38

DAFTAR LAMPIRAN

Lampiran A	Part of Speech Tag.....	52
Lampiran B	Label Dependensi Non-Klausal Adaptasi Stanford.....	53
Lampiran C	Head Rule Magerman dan Collin.....	56
Lampiran D	PCFG.....	59
Lampiran E	Aturan Pelabelan.....	63
Lampiran F	Hasil Pengujian.....	66

DAFTAR ISTILAH

Istilah	Arti
<i>Dependency Parser</i>	Pengurai kalimat berbasis dependensi, memiliki <i>head</i> dan <i>dependent</i> , dimana <i>head</i> memiliki peran penentu/dominan terhadap arti keseluruhan konstruksi.
<i>Constituency parser</i>	Pengurai kalimat berbasis konstituen atau lebih dikenal frase yang menghasilkan pohon kalimat, memiliki aturan grammar CFG.
<i>POS Tagger</i>	Memberikan informasi morfologi terhadap kata berupa label <i>part-of-speech</i> atau label fungsi grammar pada level kata.
<i>PCFG</i>	Probabilistic CFG, model hasil pelatihan CFG dari corpus dengan machine learning.
<i>CNF</i>	Conjunctive Normal Form, yaitu penulisan klausa sebagai disjungsi dari literal.
<i>CFG</i>	Context free grammar, dimana terdapat simbol non-terminal di sisi kiri rule dan simbol non-terminal atau terminal di sisi kanan.
Gatra	Jabatan fungsional dalam kalimat.
NLP	Natural Language Processing, pemrosesan bahasa alami

Bab I Pendahuluan

I.1 Latar Belakang

Analisis sintaks atau dikenal dengan parser bertujuan untuk menjelaskan struktur kalimat, terdapat banyak teknik diantaranya *constituency* dan *dependency* [NIV06]. *Constituency* yaitu dengan membagi kalimat menjadi frase dan kemudian dibagi lagi menjadi frase yang lebih kecil, sedangkan *dependency* dengan membuat keterhubungan antar kata. Pendekatan *constituency* lebih populer daripada metode *dependency*. Namun akhir-akhir ini pendekatan *dependency* lebih berkembang karena memiliki kelebihan dibandingkan *constituency* antara lain: *straightforward*, yaitu hanya menghubungkan node-node yang ada dan tidak membuat node baru. Selain itu parsing dilakukan pada kata bukan pada frase sehingga tidak perlu menunggu frase lengkap untuk membentuk relasi. Kelebihan lain adalah mempermudah proses analisis berikutnya di level semantik.

Pengurai dependensi (*dependency parsing*) menggunakan pola tata bahasa dependensi (*dependency grammar*), dimana kalimat terdiri dari kata-kata yang saling ketergantungan satu sama lain. Kedua kata dikatakan memiliki keterhubungan *dependency*, jika yang satu adalah *head* dan lainnya adalah *dependent* dan ada *link* diantara mereka. Secara umum, *dependent* adalah modifier, objek atau komplemen; *head* memiliki peran yang lebih besar dalam menentukan perilaku dari pasangan kata tersebut. Misalnya untuk frase “seorang pria tidur”, “seorang” adalah *dependent* terhadap “pria”, dan “pria” *dependent* terhadap “tidur”, sedangkan “tidur” tidak bergantung pada apapun atau disebut juga *root*. Pengurai dependensi digunakan antara lain untuk *question answering*, *text summarization* dan *information extraction*.

Pengurai dependensi memiliki banyak pendekatan [NIV06] antara lain *dynamic programming*, *constraint satisfaction* dan *deterministic*. Ketiga metode ini hanya dapat dilakukan pada kalimat *projective* sedangkan untuk *non-projective* terdapat metode tersendiri. Kalimat *projective* yaitu jika terdapat string “A B C”, dan C *dependent* terhadap A, maka B juga *dependent* terhadap A, *non-projective* yaitu B

dimungkinkan *dependent* terhadap kata lain diluar A. Metode-metode tersebut membutuhkan pendefinisian grammar dependensi yang membutuhkan keahlian dari pakar bahasa. Secara umum grammar dependensi berbentuk graf berarah [NIV06] yang terdiri dari:

1. Himpunan *node* (V), *node* ini dilabeli dengan kata (dan atau anotasi)
2. Himpunan *edge* (E), *edge* dilabel dengan jenis dependensi.
3. Urutan preseden linear dari V.

Dynamic programming menganggap dependensi sebagai konstituen, contoh pendekatannya antara lain *CYK parser*, *link grammar* (Sleator dan Temperley 1991), *bilexical grammar* (Eisner 1996). *Constraint satisfaction* menggunakan *constraint dependency grammar*, terdiri dari himpunan boolean *constraint* dan proses *parsing* merupakan *constraint satisfaction problem*, contoh metodenya *constraint grammar* (Karlson 1990), *constraint dependency grammar* (Maruyama 1990). *Deterministic parsing* menggunakan aksi sekuen deterministik dalam melakukan *parsing*, contoh metodenya antara lain algoritma *incremental* (Covington 2001) dan *shift-reduce* (Nivre 2003).

Selain metode-metode yang disebutkan, terdapat pendekatan lain yang lebih efisien yaitu pemetaan struktur frase ke struktur dependensi. Disebut efisien karena menggunakan resource dari metode lain yang sudah lebih berkembang yaitu pengurai konstituen. Pengurai dependensi untuk bahasa Inggris, Jerman, Arab, Perancis dan Cina [STA11] telah dikembangkan Stanford. Pendekatan yang digunakan adalah dengan pemetaan struktur frase ke struktur dependensi. Metode ini memanfaatkan output dari *constituency parser* kemudian dengan menggunakan aturan pemetaan mengubahnya menjadi bentuk pohon dependensi. Pengurai dependensi untuk bahasa Indonesia sampai tesis ini ditulis belum ada yang mengembangkannya. Tesis ini akan mengembangkan pengurai dependensi untuk bahasa Indonesia dengan metode pemetaan yang telah diimplementasikan Stanford untuk bahasa Inggris. Untuk mengembangkan pengurai dependensi ini

dibutuhkan *resource* diantaranya *constituency parser* seperti halnya pada pengurai dependensi Stanford.

I.2 Rumusan Masalah

Dari latar belakang yang telah dijabarkan maka dapat dirumuskan masalah utama yaitu mengadaptasi *resource* pengurai dependensi Stanford yang dibutuhkan dalam melakukan pemetaan dari struktur frase ke struktur dependensi untuk bahasa Indonesia.

I.3 Tujuan

Tujuan dari tesis ini adalah membangun pengurai dependensi untuk bahasa Indonesia dalam sebuah eksperimen yang dapat menangani kalimat bahasa Indonesia baku.

I.4 Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut:

1. Aturan pola tata bahasa dependensi yang dibangun diasumsikan adalah benar berdasarkan literatur yang dibuat oleh ahli bahasa Indonesia.
2. Bahasa Indonesia yang digunakan adalah bahasa yang baku.

I.5 Metode Penelitian

Tahapan metodologi yang akan dilakukan dalam pengerjaan penelitian ini adalah sebagai berikut:

1. **Studi literatur**, dilakukan dengan mempelajari literatur-literatur yang mendukung pelaksanaan tesis. Literatur tersebut terkait dengan tata bahasa Indonesia, *constituency grammar*, *dependency grammar*, metode pemetaan struktur frase ke struktur dependensi dan literature lain terkait pengurai dependensi.

2. **Analisis**, dilakukan dengan menganalisis masalah yang didefinisikan sebelumnya berkaitan dengan adaptasi *resource* dari bahasa Inggris ke bahasa Indonesia antara lain *head rule* dan label dependensi.
3. **Pengumpulan dataset**, dilakukan dengan mengumpulkan kalimat bahasa Indonesia yang mewakili label dependensi dan dilakukan anotasi manual terhadap dataset.
4. **Implementasi metode dan algoritma**, untuk mengimplementasikan pengurai dependensi dengan metode pemetaan struktur frase menjadi sebuah perangkat lunak.
5. **Eksperimen**, dilakukan untuk menguji metoda yang telah dipilih terhadap dataset.
6. **Analisis hasil dan kesimpulan**, dilakukan dengan menganalisis hasil pengujian, membandingkan keluaran program dengan dataset yang telah dianotasi manual.

I.6 Sistematika Pembahasan

Laporan tesis ini berisi beberapa bab yang terdiri dari Pendahuluan, Tinjauan Pustaka, Analisis Solusi Pengurai Dependensi, Pengujian, dan Penutup. Penjelasan untuk tiap bab tersebut adalah sebagai berikut:

1. Bab Pendahuluan berisi penjelasan mengenai latar belakang ide judul, rumusan masalah, tujuan, ruang lingkup, batasan masalah, metodologi, dan sistematika pembahasan. Bab ini bertujuan untuk memudahkan pemanfaatan laporan tesis ini.
2. Bab Tinjauan Pustaka berisi bahasan penelitian-penelitian yang pernah dilakukan oleh orang lain sebelumnya dan berkaitan dengan topik tesis ini. Mencakup pola tata bahasa Indonesia, aturan dependensi yang sudah ada dari bahasa Inggris, metode pemetaan struktur frase ke struktur dependensi.
3. Bab Analisis Solusi Pengurai Dependensi berisi analisis solusi pengurai dependensi untuk bahasa Indonesia mengadaptasi dari pengurai dependensi Stanford untuk bahasa Inggris.

4. Bab Pengujian berisi implementasi pengujian beserta hasil dan analisis hasil pengujian perangkat lunak. Pada bab ini diujikan kalimat bahasa Indonesia dan diukur akurasinya terhadap dataset yang dianotasi manual.
5. Bab Penutup berisi kesimpulan hasil tesis ini dan saran untuk ke depannya yang terkait tesis ini.

Bab II Tinjauan Pustaka

Bab ini membahas pola tata bahasa Indonesia, perkembangan mengenai pengurai bahasa Indonesia yang sudah ada sejauh ini, dan pengurai dependensi yang sudah ada saat ini untuk bahasa selain Indonesia. Penelitian mengenai pengurai bahasa Indonesia menggunakan metode dependensi belum ditemukan sejauh ini. Penelitian-penelitian yang dibahas pada bab ini dibagi menjadi 3 (tiga) kelompok besar yaitu penelitian mengenai pola tata bahasa Indonesia, pengurai (*parser*) dan pengurai dependensi. Penelitian mengenai pola tata bahasa Indonesia perlu dibahas agar diketahui karakteristik bahasa Indonesia beserta struktur bakunya sebagai dasar dalam membuat pengurai kalimat. Penelitian mengenai pengurai bahasa Indonesia perlu dibahas untuk mengetahui sejauh mana perkembangan pengurai untuk bahasa Indonesia beserta kelebihan dan kekurangannya. Penelitian mengenai pengurai dependensi terutama metode yang sudah dianggap *state of the art* untuk bahasa lain perlu dijadikan pertimbangan dalam membuat pengurai dependensi untuk bahasa Indonesia.

II.1 Pola Tata Bahasa Indonesia

Pada bab ini akan dibahas pola tata bahasa Indonesia pada level sintaksis dan juga penyesuaian label dependensi dari bahasa Inggris ke bahasa Indonesia.

II.1.1 Sintaksis

II.1.1.1 Kalimat Tunggal

Ada 3 macam pola dasar kalimat [KER91] yaitu:

1. Kata benda + kata kerja (gatra pangkal + gatra perbuatan). Interelasi yang membentuk pola ini terbagi atas dua macam hubungan yaitu:
 - a. Hubungan pelaku dan perbuatan
 - b. Hubungan penderita dan perbuatan
2. Kata benda + kata sifat (gatra diterangkan dan gatra menerangkan).
3. Kata benda + kata benda (gatra digolongkan + gatra penggolong).

Pada pola kalimat konvensional dikenal subjek, predikat dan objek, ada perbedaan pada penggolongan gatra yang baru. Gatra pangkal, gatra diterangkan dan gatra digolongkan merupakan rincian dari subjek, gatra perbuatan, gatra menerangkan dan gatra menggolongkan merupakan rincian dari predikat.

Kalimat tunggal adalah kalimat yang hanya terdiri dari dua unsur inti dan boleh diperluas dengan satu atau lebih unsur-unsur tambahan, selama unsur-unsur tambahan itu tidak membentuk pola baru. Kalimat tunggal dapat diuraikan menjadi fungsi gatra-gatra, jenis gatra ada dua yaitu gatra inti dan gatra tambahan.

1. Gatra inti: gatra pangkal, gatra diterangkan, gatra digolongkan, gatra perbuatan, gatra menerangkan, gatra menggolongkan.
2. Gatra tambahan: sifatnya bermacam-macam tergantung dari kata yang dijelaskannya.
 - a. Gatra tambahan yang menerangkan kata kerja
 - b. Gatra tambahan yang menerangkan kata benda, terletak persis setelah kata benda dan memiliki fungsi menerangkan gatra pangkal, gatra contoh: anak *yang nakal*, rumah *ayah*, arloji *emasnya*
 - c. Gatra tambahan yang menerangkan kata sifat, ditandai dengan kata tugas amat, sangat, lebih, kurang, hampir dll. (lihat Tabel II.1 untuk lebih lengkapnya)

II.1.1.2 Kalimat Majemuk

Batasan kalimat majemuk didefinisikan sebagai berikut [KER91]:

1. Kalimat majemuk adalah kalimat tunggal yang bagian-bagiannya diperluas sedemikian rupa, sehingga perluasan itu membentuk satu atau lebih pola kalimat yang baru di samping pola yang sudah ada.

Contoh:

“Anak itu menendang bola”

“Anak, yang kau sebut kemarin itu, menendang bola”

2. Kalimat majemuk adalah penggabungan dari dua kalimat tunggal atau lebih, sehingga kalimat yang baru ini mengandung dua pola kalimat atau lebih.

Contoh:

“Ayah menulis surat”

“Adik berdiri di sampingnya”

“Ayah menulis surat, sambil adik berdiri di sampingnya”

Berikut adalah pembagian jenis penggabungan klausa:

1. Setara (koordinatif)

Kedudukan pola-pola kalimat berada pada level yang sama, tidak ada pola-pola kalimat yang menduduki suatu fungsi dari pola yang lain. Jenis ini dibagi lagi menjadi 3 sesuai dengan fungsinya:

- a. Menggabungkan, ditandai dengan kata *dan, lagi, setelah itu, karena itu.*
- b. Memilih, ditandai dengan kata *atau.*
- c. Mempertentangkan, ditandai dengan kata *tetapi, melainkan, namun, hanya.*

2. Bertingkat (subordinatif)

Hubungan antara pola-pola kalimat tidak berada pada level yang sama, karena ada pola kalimat yang menduduki suatu fungsi dari pola yang lain. Bagian yang lebih tinggi kedudukannya disebut induk kalimat, sedangkan bagian yang lebih rendah kedudukannya disebut anak kalimat. Sesuai dengan fungsinya itu anak-anak kalimat dapat dibagi menjadi:

- a. Anak kalimat yang menduduki fungsi gatra inti
- b. Anak kalimat yang menduduki fungsi salah satu gatra tambahan

3. Campuran

Hubungan antara pola-pola kalimat itu dapat sederajat dan bertingkat. Hubungan ini terjadi kalau dalam kalimat majemuk itu terdapat paling sedikit 3 (tiga) pola kalimat, dengan 2 (dua) sederajat dan 1 (satu) bertingkat atau sebaliknya.

Tabel II.1 Gatra tambahan yang menerangkan kata kerja

Fungsi	Identifier
Keterangan tempat	di, ke, dari, pada
Keterangan waktu	kemarin, sekarang, besok, lusa
Keterangan alat	dengan + kata benda
Keterangan kesertaan	dengan + orang, bersama
Keterangan sebab	karena, sebab, oleh karena
Keterangan akibat	sehingga, sampai, akibatnya
Keterangan tujuan	untuk, guna, supaya
Keterangan perlawanan	meskipun, biarpun, walaupun, sekalipun, sungguhpun, biar
Keterangan pembatasan	kecuali, selain
Keterangan situasi	dengan + suasana
Keterangan kualitatif	dengan + cara mana/bagaimana
Keterangan kuantitatif	x kali, seberapa
Keterangan perbandingan	seperti, bagaikan
Keterangan modalitas	jikalau, kalau, seandainya, sungguh, pasti, tidak, mungkin, mari
Keterangan aspek	mulai, sudah, telah, sedang, akan, sering, kadang-kadang, tiba-tiba

Hubungan antara induk kalimat dan anak kalimat dapat dinyatakan secara eksplisit maupun implisit. Semua kata tugas yang mendahului semua anak kalimat menjadi tanda atas jenis anak kalimat tersebut. Misalnya anak kalimat yang didahului oleh kata tugas: *supaya*, *untuk*, *agar* menyatakan bahwa anak kalimat itu adalah anak kalimat keterangan tujuan.

II.1.2 Kategori Klausa dan Frasa yang Digunakan

Dari 270 aturan PCFG yang dibuat Irfan [AFI11] dan Filman [FER12] teridentifikasi kategori klausa dan frasa seperti terlihat pada Tabel II.2 dan Tabel II.3.

Tabel II.2 Kategori Klausa

No.	Kategori Klausa	Keterangan	Contoh
1.	S	Kalimat deklaratif sederhana	“Ia melempar bola”

2.	SBAR	Klausa relatif dan subordinate, dapat berfungsi sebagai komplemen verbal, sebagai keterangan kondisional, temporal dll.	“..yang melempar bola”, “..bahwa ia melempar bola”
3.	SBARQ	Kalimat pertanyaan yang mengandung kata tanya apa, siapa, bagaimana, kapan, dimana.	“siapa yang melempar bola?”
4.	SQ	<ul style="list-style-type: none"> • Klausa anak dari SBARQ setelah kata tanya. • Pertanyaan ya/tidak. 	“...yang melempar bola?”

Pengelompokan kategori frasa dan klausa menyesuaikan dengan standard Stanford dalam melakukan anotasi [BIE95].

Tabel II.3 Kategori Frasa

No.	Kategori Frasa	Keterangan	Contoh
1.	NP	Noun Phrase, frasa yang bergantung pada tipe kata benda (noun)	“...ayah saya...”
2.	VP	Verb Phrase, frasa yang bergantung pada tipe kata kerja (verb)	“...memberi uang...”
3.	PP	Prepositional Phrase, frasa yang bergantung pada kata depan	“...ke dalam rumah...”
4.	ADJP	Adjective Phrase, frasa yang bergantung pada kata sifat (adjective)	“...cantik sekali...”
5.	ADVP	Adverb Phrase, frasa yang bergantung pada kata tugas (adverb)	“...dengan cepatnya...”
6.	WHNP	Wh-Noun Phrase, menaungi klausa yang memiliki kata benda setelah identifier kata Tanya	“...berapa banyak buku...”
7.	FRAG	Bagian dari teks yang kekurangan elemen penting untuk menjadi klausa utuh, sehingga tidak dapat diekstraksi hubungan predikat-argumen nya.	“..tidak hari ini”
8.	QP	Quantifier Phrase, frasa yang mengandung tipe quantifier	“..3 kambing..”

Sedangkan hirarki kategori yang lebih rendah yaitu *part-of-speech tag* menggunakan 35 label yang dapat dilihat di LAMPIRAN A.

II.1.3 Penyesuaian Label Dependensi Bahasa Inggris ke Indonesia

Bahasa Indonesia memiliki kemiripan dengan bahasa Inggris dalam hal struktur kalimat yaitu S-V-O (subjek-predikat-objek), dengan dasar ini maka akan dibangun aturan dependensi yang diadaptasi dari aturan dependensi bahasa Inggris. Perbedaan mendasar dari pola tata bahasa Inggris dan Indonesia adalah dalam pembentukan *noun phrase* (NP), dimana pada bahasa Inggris, yang menjadi *head* terletak di belakang frase.

Hingga saat ini belum ditemukan aturan dependensi untuk bahasa Indonesia, namun untuk bahasa Inggris sudah ada yang membuatnya [MAR08]. Hal ini dapat dijadikan bahan bagi pendefinisian aturan dependensi bahasa Indonesia karena ada kemiripan struktur antara kedua bahasa dalam hal struktur kalimat S-V-O (subjek-predikat-objek).

Label dependensi adalah relasi antara *head* dan *dependent*, hal ini yang akan diadaptasi dari label dependensi bahasa Inggris [MAR08]. Hasil adaptasi label dependensi non-klausal sudah pernah dibuat (lihat LAMPIRAN B), namun aturan masih harus dilengkapi agar dapat mencakup relasi klausal. Non-klausal berarti hanya mencakup kalimat tunggal. Beberapa contoh relasi dependensi non-klausal dapat dilihat pada Tabel II.4. (Untuk lebih lengkapnya lihat LAMPIRAN B)

Tabel II.4 Contoh label dependensi non-klausal untuk bahasa Indonesia

Label	Deskripsi
acomp	Gatra pelengkap kata kerja yang berupa kata sifat
advmod	Gatra tambahan yang menerangkan (frase kata keterangan)
agent	Pelengkap dari kata kerja pasif yang ditandai dengan preposisi "oleh"
Cc	Relasi antar elemen konjungsi dan kata konjungsi seperti "dan"
Cop	Relasi antara pelengkap kata kerja kopula dan kata kerja kopula seperti "adalah"
Dobj	Frase nomina yang menjadi objek langsung dari kata kerja
Iobj	Frase nomina yang menjadi objek tidak langsung dari kata kerja

II.2 Perkembangan Metoda Pengurai Kalimat Bahasa Indonesia

Penelitian mengenai pengurai bahasa Indonesia sampai saat ini masih relatif kurang baik, hal ini terjadi karena kurangnya *resource* berupa corpus yang dianotasi manual sebagai bahan untuk metoda *data driven* atau *machine learning*. Selain itu, aturan yang dibangun untuk tata bahasa Indonesia belum lengkap dan masih memerlukan verifikasi dari pakar bahasa Indonesia. Berikut akan dijelaskan beberapa penelitian mengenai pengurai bahasa Indonesia.

II.2.1 Indonesian GLR Parser (INAGP)

Ada 3 tahap pemrosesan yang dilakukan di dalam INAGP yaitu *POS Tagger*, *parser generator*, dan *parsing* [NAS09]. *POS Tagger* yang digunakan masih memiliki kelemahan, yaitu tidak dapat menangani kata yang memiliki kelas kata yang ambigu. *Parser generator*, membaca file eksternal yang digunakan sebagai aturan *grammar* dan menghasilkan *parsing table* ke dalam sebuah file eksternal, aturan *grammar* menggunakan 120 aturan PCFG. Pada proses yang ketiga baru dilakukan proses *parsing* dengan keluaran pohon kalimat.

Dari pengujian yang dilakukan, INAGP masih memiliki banyak kekurangan. INAGP hanya mampu menangani kalimat tunggal. Performansi *POS Tagger* bawaan dari INAGP tidak dapat menangani kata yang memiliki kelas kata ambigu. INAGP juga dapat menghasilkan lebih dari satu pohon kalimat. Kesalahan struktur kata yang dihasilkan juga masih terjadi. INAGP berhasil mengeluarkan 26 pohon kalimat yang benar dari 30 kalimat tunggal yang diuji. INAGP tidak dapat menghasilkan pohon kalimat dari kalimat majemuk. Performansi *POS Tagger* yang kurang baik juga mengakibatkan pohon kalimat salah. Secara garis besar, performansi INAGP relatif masih kurang.

II.2.2 Pengurai Collins untuk Bahasa Indonesia

Pengurai kalimat bahasa Indonesia selanjutnya adalah penguraian bahasa Indonesia dengan menggunakan pengurai Collins [SUK09]. Pada penelitian Rosa, Pengurai Kalimat yang dibangun memanfaatkan pengurai Collins, yaitu pengurai kalimat yang dikenal memiliki performansi baik. Permasalahan yang ada adalah

pengurai Collins merupakan pengurai kalimat untuk bahasa Inggris. Selain itu pengurai Collins membutuhkan data latih berupa *tree bank* (sekumpulan kalimat yang telah diberi label pohon kalimat) sedangkan *tree bank* untuk bahasa Indonesia belum ada. Rosa memanfaatkan kesamaan label kata bahasa Inggris dengan bahasa Indonesia. Pengurai kalimat yang dihasilkan menerima sekumpulan label kata dari kalimat bahasa Indonesia. Dari label kata tersebut, pengurai Collins dapat menghasilkan pohon kalimat walaupun aturan yang digunakan merupakan aturan tata bahasa Inggris.

Dari hasil yang didapat, pengurai kalimat yang dibuat masih memiliki tingkat kesalahan yang relatif tinggi. Permasalahan utama yang disorot adalah kurangnya *tree bank* untuk bahasa Indonesia. Hal ini sangat mempengaruhi performansi pengurai kalimat penelitian Rosa.

II.2.3 Pengurai Earley dan CYK untuk Bahasa Indonesia

Yang terakhir adalah pengurai kalimat menggunakan metoda Earley dengan PCFG sederhana dan metoda CYK dengan CNF [AFI11]. Terbentuk 151 aturan PCFG bahasa Indonesia dari data latih 100 pohon kalimat, namun akurasi dari metoda ini masih rendah yaitu 38,89 % karena keterbatasan dari aturan yang didefinisikan. PCFG tersebut disempurnakan oleh Filman [FER12] sehingga terbentuk 270 aturan PCFG bahasa Indonesia dari data latih 200 pohon kalimat.

Walaupun performansi CYK lebih baik daripada Earley, namun hanya Earley yang dapat diaplikasikan pada tesis ini karena bentuk aturannya menggunakan CFG yaitu menghasilkan pohon konstituen. Penelitian ini masih memiliki kekurangan yaitu tidak adanya justifikasi dari ahli bahasa Indonesia dalam membangun corpus pohon kalimat.

Semua parser di atas menggunakan pendekatan *constituency*, sedangkan untuk *dependency* untuk bahasa Indonesia masih belum ada yang mengembangkannya.

II.3 Pengurai Dependensi (*Dependency Parsing*)

Pengurai dependensi menggunakan tata bahasa dependensi yang menggambarkan ketergantungan antar komponen dimana salah satu adalah *head* dan lainnya adalah *dependent*. *Head* disebut pula *modifier* karena merupakan penentu bagi pasangannya. Dalam suatu bentukan frase di bahasa Indonesia menggunakan konstruksi ditentukan-menentukan (DM).

Robinson (1970) memformulasikan 4 aksiom untuk mengatur struktur dependensi yaitu:

1. Hanya satu elemen yang independen
2. Sisanya bergantung secara langsung pada elemen lainnya
3. Tidak ada elemen yang bergantung secara langsung pada lebih dari satu elemen.
4. Jika A bergantung secara langsung pada B dan beberapa elemen C terletak diantara keduanya secara linear, maka C bergantung secara langsung pada A atau B atau elemen diantaranya.

3 aksioma pertama menyiratkan bahwa struktur dependensi adalah pohon. Aksioma 3 menjelaskan sifat *single-headness* atau *uniqueness*. Aksioma 4 disebut pula dengan *projectivity*, dan tidak membolehkan adanya *link* yang bersilangan di dalam pohon. Aksioma terakhir ini ditentang oleh Debusmann [DEB00], karena untuk bahasa Latin, Jerman dan Rusia yang memiliki derajat tinggi dalam variasi keteraturan kata (*free word order*), *projectivity* tidak berlaku. Oleh karena itu diajukan *non-projective dependency parsing*, dimana dibolehkan ada *link* dependensi yang menyilang di pohon [DEB00].

Algoritma saat ini lebih banyak berkembang untuk *projective dependency*. Untuk menentukan suatu bahasa dikategorikan ke dalam *projective* atau *non-projective* maka harus dilihat dari karakteristik tata bahasanya (*grammar*).

Untuk melakukan parsing, perlu diperhatikan hal-hal berikut [COV04]:

1. Apakah *dependent* terletak di depan atau di belakang *head*. Contohnya, dalam bahasa Jepang semua *dependent* dari kata kerja mendahului *head*.

Sebagian besar bahasa memiliki kecenderungan untuk menaruh *dependent* sebelum *head* (Jepang), atau setelah *head* (Arab). Beberapa bahasa mencampurnya (Inggris). Jika sifatnya *free word order*, maka syarat ini diabaikan. Pada bahasa Indonesia sendiri *head* terletak di depan.

2. Apakah *dependent* dari *head* yang sama harus muncul dengan urutan tertentu. Contohnya, di bahasa Inggris, objek langsung mengikuti objek tidak langsung, jika keduanya muncul. Keduanya adalah *dependent* dari kata kerja dan muncul setelah kata kerja, objek tidak langsung harus muncul lebih dahulu.
3. Apakah *projective* atau tidak. Contohnya, di bahasa Inggris dan bahasa lainnya, frase preposisi adalah *projective*, yaitu kata-kata di antara *head* dan *dependent* membentuk *dependency* yang kontinu. Pada bahasa Indonesia asumsi *projective* ini dapat berlaku, misal pada frase ayah paman saya, *head*-nya adalah ayah, paman dan saya *dependent* terhadap ayah, paman *dependent* terhadap ayah dan saya *dependent* terhadap paman.

Ketiga hal diatas merupakan hal yang harus dianalisis ketika melakukan penguraian dependensi (*dependency parsing*) untuk suatu bahasa, karena setiap bahasa memiliki kebutuhan yang berbeda. Terdapat banyak metoda pengurai dependensi, beberapa diantaranya akan dibahas berikut ini.

Pengurai dependensi untuk bahasa Inggris saat ini sudah banyak dikembangkan salah satunya Stanford parser [STA11]. Pengurai ini memanfaatkan metoda pengurai berbasis konstituen yaitu pemetaan dari struktur frase ke bentuk dependensi [MAR99]. Hal ini dimotivasi oleh tersedianya *resource* berupa corpus yang sudah dianotasi dan pengurai berbasis konstituen. Perbedaan mendasar antara struktur frase/konstituen dan struktur dependensi pada pohon parser adalah node pada pohon. Pada frase, node adalah kategori/POS tag, sedangkan pada dependensi, node adalah kata-kata. Pada tesis ini akan digunakan metode ini karena ketersediaan *resource* berupa pengurai kalimat bahasa Indonesia berbasis konstituen. Teknik ini digunakan oleh Stanford parser [STA11], *resource* yang dibutuhkan dalam Stanford parser antara lain pengurai konstituen, Collin *head rule*, label dependensi dan corpus Penn Treebank yang sudah dikonversi ke

dependensi. Label dependensi Stanford terdiri dari 40 label non klausal dan 15 label klausal. Pada LAMPIRAN B terdapat daftar label dependensi non-klausal, dan berikut pada Tabel II.5 tercantum label non-klausal yang tidak bisa diaplikasikan pada bahasa Indonesia.

Tabel II.5 Label non-klausal yang tidak dapat diaplikasikan pada bahasa Indonesia

No	Label	Keterangan	Alasan
1	<i>abbrev</i>	<i>Abbreviation modifier</i> , keterangan akronim.	Memerlukan penanganan khusus akronim.
2	<i>attr</i>	<i>Attributive</i> , pelengkap WHNP dari kata kerja kopula.	Pada bahasa Indonesia tidak ada padanannya.
3	<i>expl</i>	<i>Expletive</i> , untuk menunjukkan eksistensi yaitu "there", membutuhkan kata kerja <i>to be</i> .	Di bahasa Indonesia kata "ada" merupakan kata kerja yang berdiri sendiri.
4	<i>preconj</i>	<i>Preconjunct</i> , yaitu "both", "either", "neither"	Pada bahasa Indonesia tidak ada padanannya.
5	<i>predet</i>	<i>Predeterminer</i> , relasi antara head NP dan kata sebelum determiner. Contoh " <u>all</u> the <u>boys</u> are here"	Pada bahasa Indonesia tidak ada padanannya.
6	<i>prt</i>	<i>Phrasal verb particle</i> , contohnya "shut down"	Pada bahasa Indonesia kata kerja berdiri sendiri tanpa partikel.

Berikut dibahas bagaimana cara untuk memetakan dari struktur frase ke pohon dependensi. Langkah-langkah yang harus dilakukan untuk memetakan adalah:

1. *Dependency extraction*. Menggunakan pengurai kalimat berbasis konstituen kemudian menentukan *head* pada setiap frase. Pada beberapa kasus frase, *head* terletak di depan dan *dependent* setelahnya. Pada sedikit kasus, pencarian kata di leksikon dibutuhkan untuk menentukan *head*. Contoh aplikasi *dependency extraction* pada bahasa Inggris dapat dilihat pada Tabel II.6.

Tabel II.6 Contoh *dependency extraction* pada bahasa Inggris

Input kalimat	“my mother cooks rice”
Parse tree	(S (NP (PRP\$ my) (NN mother)) (VP (VBZ cooks) (NP (NN rice))))
Hasil ekstraksi dependensi (<i>head, dependent</i>)	(mother, my) (cooks, mother) (cooks, rice) root(cooks)

2. *Dependency typing*. Menentukan relasi gramatikal se-spesifik mungkin. Relasi ini akan menjadi label dependensi antara *head* dan *dependent*. Contoh aplikasi *dependency typing* pada bahasa Inggris dapat dilihat pada Tabel II.7.

Tabel II.7 Contoh *dependency typing* pada bahasa Inggris

Pohon dependensi	(mother, my) (cooks, mother) (cooks, rice) root(ROOT, cooks)
Hasil <i>dependency typing</i>	poss(mother, my) nsubj(cooks, mother) dobj(cooks, rice) root(ROOT, cooks)

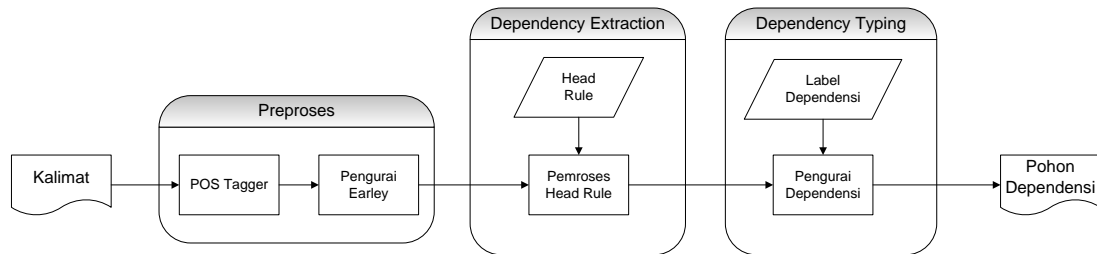
Setiap frase yang didefinisikan oleh CFG adalah flat/datar, tidak ada hirarki sedangkan pada pohon dependensi diperlukan *head* dan *dependent*. Magerman dan Collin mendefinisikan aturan *head* berdasarkan posisi *part-of-speech*-nya dalam kalimat [COL95]. Lihat LAMPIRAN C untuk mengetahui bagaimana aturan *head* ini didefinisikan.

Bab III Analisis Solusi Pengurai Dependensi

Pada Bab III ini akan dijelaskan mengenai proses-proses yang diperlukan dalam melakukan pengurai dependensi. Metode yang dipilih adalah pemetaan struktur frase ke struktur dependensi, pada pemetaan ini ada 2 tahap dan di tiap tahapnya memerlukan *resource* NLP lainnya. Pohon kalimat konstituen yang dihasilkan oleh pengurai Earley [AFI11] akan diproses melalui 2 (dua) tahap *dependency extraction* dan *dependency typing* sehingga menghasilkan pohon dependensi berlabel. Pada kedua tahap tersebut digunakan *resource* seperti *Collin head rule* dan label dependensi Stanford yang harus disesuaikan dulu terhadap bahasa Indonesia.

III.1 Mekanisme Solusi Pengurai Dependensi

Pengurai dependensi untuk bahasa Indonesia dengan menggunakan metoda pemetaan struktur frase ke pohon dependensi dapat dibuat ke dalam diagram alir proses pada Gambar III.1.



Gambar III.1 Diagram Alir Proses Pengurai Dependensi

Berikut penjelasan Gambar III.1, input proses adalah kalimat bahasa Indonesia, kemudian diproses oleh modul *Preproses* yang terdiri dari *POS Tagger* [WIC10] dan *Pengurai Earley* [AFI11]. Keluaran berupa pohon konstituen yang kemudian diproses oleh modul *Dependency Extraction* yang terdiri dari *Pemroses Head Rule* yang menggunakan *Head Rule* hasil adaptasi dari Collin head rule. Keluaran dari modul adalah pohon dependensi tanpa label, kemudian diproses oleh modul *Dependency Typing* yang terdiri dari *Pengurai Dependensi* yang menggunakan

Label Dependensi hasil adaptasi dari *Stanford Label Dependency* sebagai aturan pelabelannya. Keluaran dari sistem adalah pohon dependensi berlabel.

III.1.1 Preproses

Di antara penelitian mengenai pengurai kalimat bahasa Indonesia, sejauh ini aturan grammar PCFG dari Irfan [AFI11] dan Filman [FER12] adalah yang paling lengkap (270 aturan). Oleh karena itu, aturan tersebut digunakan sebagai *resource* dalam tesis ini dan dimodifikasi sehingga dapat menghasilkan aturan dependensi. Berikut pada Tabel III.1 adalah beberapa contoh PCFG Irfan dan Filman.

Tabel III.1 Contoh PCFG bahasa Indonesia

<SENTENCE>::=<S> 1.0
<S>::=<S><*.> 0.024691358024691357
<S>::=<NP><VP> 0.5246913580246914
<S>::=<VP><NP><*.> 0.006172839506172839
<NP>::=<*NNP> 0.1023391812865497
<VP>::=<*VBT><NP> 0.1346153846153846
<PP>::=<*IN><NP> 0.7608695652173914
<SBARQ>::=<WHNP><*SC><VP> 0.2857142857142857
<WHNP>::=<*WP> 0.8
<ADJP>::=<*JJ> 0.6571428571428571
<QP>::=<*CDP><*CDP> 0.5
<SBAR>::=<WHNP><VP> 0.5
<ADVP>::=<*RB><*JJ> 0.14285714285714285
...

Lihat LAMPIRAN D untuk lebih lengkapnya.

Preproses kalimat dilakukan terlebih dahulu dengan POS tagger dan parser:

1. Kalimat bahasa Indonesia diberi label part-of-speech Alfian [WIC10].
2. Kalimat yang sudah ditokenisasi dan diberi label part-of-speech akan menjadi input bagi Pengurai Earley Irfan [AFI11].

Keluaran dari Pengurai Earley adalah pohon konstituen seperti pada Tabel III.2.

Tabel III.2 Pohon konstituen keluaran parser Earley

S	
NP	NNP ibu
VP	VBT memasak
	NP
	NN nasi

III.1.2 Dependency Extraction

Tahap *dependency extraction* membutuhkan *resource* berupa aturan *head* bagi tiap struktur frase. Aturan *head* menggunakan *Head Rule* Collin [COL95], namun harus disesuaikan untuk bahasa Indonesia. Penyesuaian ini dibahas di subbab III.1.2.1.

III.1.2.1 Head Rule

Pada Bab II telah dibahas *head rule* yang dibuat Magerman dan Collin, namun aturan ini hanya berlaku pada bahasa Inggris. Oleh karena itu, harus dilakukan penyesuaian untuk bahasa Indonesia. Salah satu contoh penyesuaian terletak pada pencarian *head* untuk beberapa kasus NP, dimana pada bahasa Inggris *head* terletak di belakang (*head final*) sedangkan pada bahasa Indonesia *head* terletak di depan (*head initial*). Misalnya “red meat” yang menjadi *head* adalah “meat” dan *dependent*-nya adalah “red”. Selain itu, POS tag yang digunakan sudah disesuaikan dengan tag set Alfian [WIC10] dan PCFG dari Irfan [AFI11]. *Head rule* hasil adaptasi dapat dilihat pada Tabel III.3.

Adaptasi yang dilakukan dari Collin *head rule* untuk bahasa Inggris adalah penelusuran *head* untuk non terminal NP (*noun phrase*). Jika untuk simbol non terminal tidak ditemukan kategori *head* di tabelnya, maka kembalikan anak sesuai arah penelusuran, yaitu jika *Left* maka kembalikan anak pertama di paling kiri dan sebaliknya untuk *Right*.

Khusus untuk NP, ada proses tersendiri yaitu

1. Jika kata pertama berlabel *part-of-speech*, maka kembalikan kata tersebut
2. Jika bukan, cari dari kiri ke kanan anak pertama dengan tipe NNP/NN/NNG
3. Jika bukan, cari dari kiri ke kanan anak pertama dengan tipe NP

4. Jika bukan, cari dari kiri ke kanan anak pertama dengan tipe ADJP atau PRN.
5. Jika bukan, cari dari kanan ke kiri anak pertama dengan tipe CD
6. Jika bukan, cari dari kanan ke kiri anak pertama dengan tipe JJ, RB atau QP
7. Jika bukan, kembalikan kata pertama.

Tabel III.3 Aturan *head* untuk frase bahasa Indonesia

Non Terminal	Left/Right	Head
S	Right	IN VP S SBAR ADJP NP
SBAR	Right	WHNP IN DT S SBAR
ADJP	Right	QP NN ADVP JJ ADJP NP DT FW SBAR RB
ADVP	Left	RB FW ADVP CDP CDI JJ IN NP NN
CONJP	Left	CC RB IN
PP	Left	IN RP FW
PRN	Right	
QP	Right	IN NN NNP NNG JJ RB DT CDP CDI QP
SBARQ	Right	SQ S SBARQ FRAG
VP	Right	MD VBI VBT VP ADJP NN NNP NNG NP
WHNP	Right	WP WHNP

III.1.2.2 Pemroses *Head Rule*

Masukan tahap ini adalah pohon yang dihasilkan pengurai kalimat Earley, kemudian CFG (*Context Free Grammar*) dibangkitkan dari pohon kalimat tersebut seperti terlihat pada Tabel III.4.

Tabel III.4 Daftar CFG yang dibangkitkan dari pohon konstituen

S	→	NP VP
NP	→	NNP
NNP	→	ibu
VP	→	VBT NP
VBT	→	memasak
NP	→	NN
NN	→	nasi

Setelah CFG terbentuk, *head* untuk setiap frase perlu diidentifikasi. LHS pada setiap CFG akan disubstitusi dengan *head* yang terkandung di RHS-nya. Aturan *head* didefinisikan pada Tabel III.5.

Tabel III.5 Aturan *head*

Non terminal	Left/Right	<i>Head</i>
S	Right	VP
VP	Left	VBT
NP	Left	NNP

Proses pengaplikasian aturan *head* di atas akan mengubah CFG dari format LHS → RHS menjadi bentuk *Head* (word) → *Dependent* (word). Pengaplikasian aturan *head* pada suatu aturan CFG terus dilakukan hingga node berupa *part-of-speech*. Ilustrasi pengaplikasian aturan *head* dapat dilihat pada Tabel III.6.

Tabel III.6 Ilustrasi pengaplikasian aturan *head*

CFG	LHS	RHS	Dependency
0 S → NP VP	0 S	0 NP VP	0 VBT (memasak) → NNP (ibu)
1 NP → NNP	VP	NNP	
2 NNP → ibu	VBT		
3 VP → VBT NP	1 NP		
4 VBT → memasak	NNP		
5 NP → NN	3 VP	3 VBT NP	
6 NN → nasi	VBT	NN	3 VBT (memasak) → NN (nasi)
	5 NP		
	NN		

Proses pengaplikasian aturan *head* adalah sebagai berikut:

1. Jika ditemukan S maka telusuri RHS dari arah kanan. Jika menemukan VP maka itu adalah *headnya*.
2. Jika ditemukan VP maka telusuri RHS dari arah kiri. Jika menemukan VBT maka itu adalah *headnya*.
3. Jika ditemukan NP maka telusuri RHS dari arah kiri. Jika menemukan NNP maka itu adalah *headnya*.

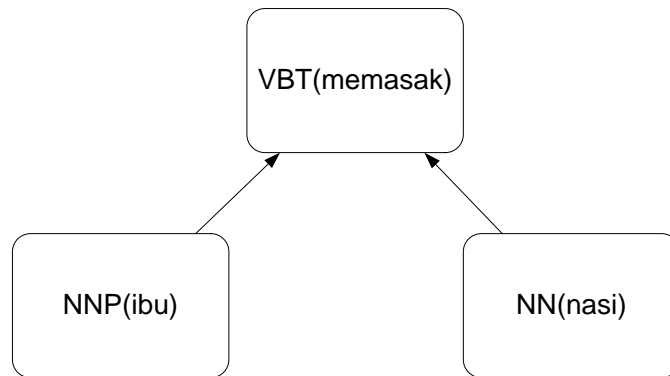
Hal ini dilakukan baik pada LHS maupun RHS. Kemudian LHS yang sudah direduksi akan menjadi *head* dan RHS yang sudah direduksi akan menjadi *dependent*. Dapat dilihat bahwa pada bentuk pohon dependensi sudah tidak ada node bertipe frase seperti NP dan VP, karena node pada dependensi adalah kata-kata itu sendiri dengan label POS-nya.

Algoritma pembentukan relasi dependensi (*dependency extraction*) dapat dilihat pada Tabel III.7.

Tabel III.7 Algoritma pembentukan dependensi dari CFG

```
for each cfg in cfg_list
    if not pos(cfg.lhs) then
        head = get_head_pos(cfg.lhs)
        for each node in cfg.rhs
            if node <> head then
                if not_pos(node) then
                    dependent = get_head_pos(node)
                else
                    dependent = node
            add_new_relation(head, dependent)
```

Hasil *dependency extraction* dari algoritma di Tabel III.7 dapat dilihat pada Gambar III.2.



Gambar III.2 Pohon dependensi hasil *dependency extraction*

III.1.3 *Dependency Typing*

Tahap *dependency typing* membutuhkan *resource* berupa label dependensi atau aturan dependensi untuk bahasa Indonesia. Modul ini menggunakan label dependensi dari Stanford yang telah disesuaikan untuk bahasa Indonesia. Namun, harus dilengkapi untuk menangani kalimat majemuk subordinatif (klausal).

III.1.3.1 Label Dependensi

Kalimat majemuk adalah gabungan dari dua atau lebih klausa. Penggabungan/koordinasi dapat terjadi pada level frase maupun klausa dan ditandai dengan kata gabung atau preposisi sesuai fungsinya. Kata gabung umumnya adalah “dan”, “atau”, “namun”, “tetapi”. Preposisi yang digunakan dalam konjungsi antara lain “dengan”, “termasuk”, dst. Penggabungan pada level frase sudah dibahas pada Bab II, pada subbab ini dibahas penggabungan pada level klausa dan label dependensi terkait. Pada Tabel III.9 adalah label klausal yang diambil dari label dependensi Stanford. Terdapat 15 label yang merupakan tambahan khusus terhadap aturan dependensi non-klausal sebelumnya untuk menangani kalimat majemuk (klausal).

III.1.3.2 Pengurai Dependensi

Setelah terbentuk *head* dan *dependent*-nya, maka selanjutnya adalah memberikan label dependensi. Label ini memiliki arah dari *head* ke *dependent* dan tidak sebaliknya. Selain informasi POS dari *head* dan *dependent*, dimungkinkan

penambahan informasi lainnya yang tidak cukup diberikan oleh POS dan dibutuhkan dalam pemberian label seperti:

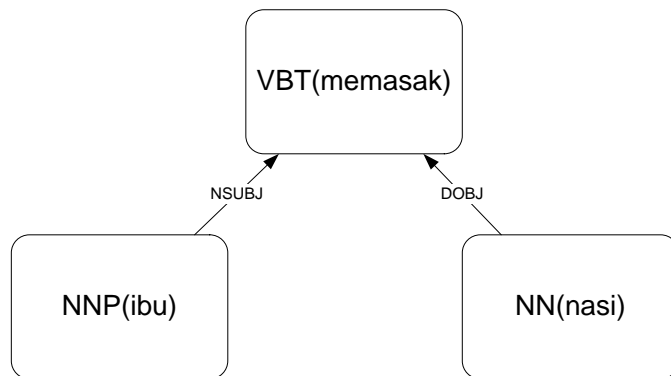
1. Posisi antara *head* dan *dependent*, apakah *head* terletak lebih dulu dibanding *dependent* dan jarak antara *head* dan *dependent*.
2. Leksikon tertentu yang harus dipenuhi oleh relasi antara *head* dan *dependent*.
3. Leksikon tertentu yang harus dipenuhi *head* atau *dependent*.

Misal untuk contoh pohon dependensi di Gambar III.2, digunakan label dependensi pada Tabel III.8.

Tabel III.8 Label dependensi

Label(<i>head</i>, <i>dependent</i>)	Label(x1, x2)
Dobj(VBT, NN)	Dobj(VBT(memasak), NN(nasi))
Nsub(VBT, NNP)	Nsubj(VBT(memasak), NNP(ibu))

Hasil dari Pengurai Dependensi adalah Pohon Dependensi Berlabel dapat dilihat di Gambar III.3.



Gambar III.3 Pohon dependensi berlabel

Tabel III.9 Label dependensi klausal

No	Label	Keterangan	Contoh
1	Advcl	<i>Adverbial clause modifier</i> , hubungan antar klausa (kondisional, temporal dll) yang menerangkan kata kerja atau subjek. Ditandai dengan kata “saat”, “ketika”, “karena”, “maka dari itu”, “oleh karena itu”	“kecelakaan itu terjadi saat malam dimulai” <i>advcl</i> (terjadi, dimulai)
2	Ccomp	<i>Clausal complement of verb</i> , kata yang menjelaskan kata kerja di klausa. Ditandai dengan kata perantara <i>bahwa</i>	“Dia berkata bahwa kamu suka berenang” <i>ccomp</i> (berkata, suka)
3	Complm	<i>Complementizer</i> , kata komplemen. Ditandai dengan kata <i>bahwa</i>	“Dia berkata bahwa kamu suka berenang” <i>complm</i> (suka, bahwa)
4	Csubj	<i>Clausal subject</i> , subjek yang berupa klausa. Ditandai dengan kata <i>yang</i>	“Apa yang dia katakan masuk akal” <i>csubj</i> (masuk, katakan)
5	Csubjpass	<i>Clausal passive subject</i> , subjek pasif yang berupa klausa	“Bahwa dia berbohong sudah disangka oleh semua orang” <i>csubjpass</i> (disangka, berbohong)
6	Mark	<i>Marker</i> , kata yang menerangkan <i>advcl</i>	“pasukan dilibatkan dalam peperangan setelah insurgent menyerang” <i>mark</i> (menyerang, setelah)
7	Parataxis	<i>Parataxis</i> , hubungan antara kata kerja utama dari klausa dan elemen lain	“Lelaki itu, dikatakan John, pergi pagi pada pagi hari” <i>parataxis</i> (pergi, dikatakan)
8	Pcomp	<i>Prepositional complement</i> , pelengkap berupa klausa dari kata-kata preposisi	“Mereka mendengar mengenai dirimu yang membolos” <i>pcomp</i> (mengenai, membolos)
9	Prepc	<i>Prepositional clausal modifier</i>	“Dia membelinya tanpa membayar premi” <i>prepc</i> (membeli, membayar)

No	Label	Keterangan	Contoh
10	Purpcl	<i>Purpose clause modifier</i> , klausa dikepalai oleh “(dalam rangka) untuk”	“Dia berbicara kepadanya dalam rangka untuk mengamankan akun” <i>purpcl</i> (berbicara, mengamankan)
11	Rcmod	<i>Relative clause modifier</i> , klausa yang menjadi modifier klausa lainnya, ditandai dengan kata “yang”, “dimana”, “tempat”	“Saya melihat laki-laki yang kamu cintai” <i>rcmod</i> (laki-laki, cintai)
12	Ref	<i>Referent</i>	“Saya melihat laki-laki yang kamu cintai” <i>ref</i> (laki-laki, yang)
13	Rel	<i>Relative</i>	“Saya melihat pria yang istrinya kamu cintai” <i>rel</i> (cintai, istri)
14	Xcomp	<i>Open clausal complement</i>	“Dia berkata bahwa kamu suka berenang” <i>xcomp</i> (suka, berenang)
15	Xsubj	Relasi antara <i>head</i> dari <i>xcomp</i> dan subjek eksternal dari klausa tsb.	“Tom suka makan ikan” <i>xsubj</i> (makan, Tom)

Bab IV Pengujian

IV.1 Tujuan Pengujian

Pengujian dilakukan untuk menguji kebenaran algoritma dalam memproses kalimat bahasa Indonesia dan menghasilkan relasi dependensi yang benar. Pengujian juga dilakukan untuk mengetahui kebenaran analisis *head rule* dan label dependensi yang diadaptasi dari aturan tata bahasa Inggris.

IV.2 Perancangan Pengujian

Pengujian yang akan dilakukan dalam tesis ini terbagi menjadi 2 skenario yaitu:

1. Masukan pengurai adalah pohon valid yang dianotasi manual
2. Masukan pengurai adalah kalimat yang kemudian diuraikan oleh pengurai Earley [AFI11].

Langkah pengujian:

1. Disiapkan dataset uji yang telah dibuat pohon dependensinya secara manual sebagai standard pengukuran akurasi.
2. Pengujian skenario 1, yaitu dataset uji yang telah dianotasi manual menjadi pohon konstituen diproses oleh pengurai dependensi.
3. Pengujian skenario 2, yaitu dataset uji berupa kalimat diproses menggunakan pengurai Earley [AFI11] dan kemudian menjadi masukan bagi pengurai dependensi.
4. Keluaran kedua skenario berupa pohon dependensi berlabel kemudian dibandingkan dengan pohon dependensi berlabel yang telah dianotasi manual dan dihitung akurasi.

IV.3 Data Pengujian

Data uji yang digunakan adalah kalimat-kalimat berita baku. Data uji yang digunakan berbeda dengan data latih pada PCFG. Variasi data uji dapat dilihat di Tabel IV.1.

Tabel IV.1 Data pengujian

Jenis Kalimat	Uji
kalimat sederhana	27
kalimat subordinatif	13

Kalimat sederhana adalah kalimat dengan menggunakan label dependensi non-klausal sedangkan kalimat subordinatif adalah kalimat dengan label dependensi klausal. Kalimat uji lengkap dapat dilihat di LAMPIRAN F. 27 kalimat sederhana tersebut sudah mencakup 29 label dependensi non-klausal yang didefinisikan di LAMPIRAN B. 13 kalimat majemuk subordinatif sudah mencakup 15 label dependensi klausal yang didefinisikan di III.2.3.1.

IV.4 Hasil Pengujian

IV.4.1 Skenario 1

Pada skenario ini, masukan sistem berupa pohon konstituen yang dianotasi manual, termasuk proses tagging dan penguraian CFG-nya. Pengujian dilakukan terhadap kalimat sederhana dan kalimat subordinatif.

Hasil pengujian kelompok pertama, dari 27 pohon konstituen berhasil diuraikan 27 pohon dependensi. Kemudian setiap kalimat mewakili 1 atau 2 label dependensi, oleh karena itu penilaian akurasi hanya akan difokuskan pada relasi yang diperuntukkan label yang diwakilinya. Hasil pengujian kelompok kedua, dari 13 kalimat berhasil diuraikan 13 kalimat. Hasil dari pengujian yang dilakukan berupa tingkat akurasi yang dihasilkan dari kelompok data uji kalimat sederhana dan kalimat majemuk subordinatif terhadap anotasi pohon dependensi secara manual. Hasil pengujian dapat dilihat pada LAMPIRAN F.

Tabel IV.2 Hasil pengujian skenario pertama

Jenis Kalimat	Akurasi
kalimat sederhana	23 benar dari 27 (85.2 %)
kalimat subordinatif	10 benar dari 13 (77 %)

Pada Tabel IV.2 dapat dilihat bahwa pengujian terhadap kalimat sederhana memberikan akurasi 85.2% dan kalimat subordinatif sebesar 77%. Kesalahan pada

pengujian kalimat sederhana disebabkan label dependensi ambigu, tidak dapat membedakan kata kerja aktif dan pasif. Kesalahan pada pengujian kalimat subordinatif juga disebabkan label dependensi yang ambigu karena kurang spesifiknya definisi aturan pelabelan dependensi.

IV.4.2 Skenario 2

Pada skenario ini, masukan sistem berupa kalimat, proses tagging dan penguraian konstituen menggunakan POS tagger Alfan [WIC10] dan Pengurai PCFG Irfan [AFI11]. Pengujian dilakukan terhadap kalimat sederhana dan kalimat subordinatif. Karena keterbatasan POS tagger dalam mengenali kata dan keterbatasan pengurai dalam menyusun kalimat panjang. Maka digunakan kalimat-kalimat sederhana yang tidak mengandung istilah baru dan relatif pendek.

Hasil pengujian kelompok pertama, dari 27 kalimat berhasil diuraikan 24 kalimat oleh pengurai Earley, sedangkan 3 sisanya dibuat pohonnya secara manual. Setiap kalimat mewakili 1 atau 2 label dependensi, oleh karena itu penilaian akurasi hanya akan difokuskan pada relasi yang diperuntukkan label yang diwakilinya. Hasil pengujian kelompok kedua, dari 13 kalimat berhasil diuraikan 13 kalimat. Hasil dari pengujian yang dilakukan berupa tingkat akurasi yang dihasilkan dari kelompok data uji kalimat sederhana dan kalimat majemuk subordinatif terhadap anotasi pohon dependensi secara manual. Hasil pengujian dapat dilihat pada LAMPIRAN F.

Tabel IV.3 Hasil pengujian skenario kedua

Jenis Kalimat	Akurasi
kalimat sederhana	21 benar dari 27 (77 %)
kalimat subordinatif	6 benar dari 13 (46 %)

Hasil pada Tabel IV.3 menunjukkan bahwa pengurai Earley dikombinasikan dengan pengurai dependensi memiliki akurasi yang lebih baik untuk jenis kalimat sederhana dibandingkan dengan kalimat majemuk subordinatif. Hal ini terjadi karena kesalahan pada pengurai kalimat Earley dalam mengurai kalimat subordinatif. Pengurai kalimat Earley masih salah mengelompokkan klausa.

IV.5 Evaluasi Pengujian

Kesulitan dalam menguji kalimat terletak pada masih terbatasnya kemampuan *tagger* dalam mengenali kata-kata sehingga banyak kata yang salah diberi tag. Selain itu, terbatasnya kemampuan parser Earley dalam melakukan penguraian kalimat yang panjang, kemungkinan karena terbatasnya aturan PCFG atau terjadinya ambiguitas dalam proses penguraian. Oleh karena itu, penyebab kesalahan dapat dibagi 3 yaitu kesalahan POS tagger, PCFG dan pengurai dependensi itu sendiri.

IV.5.1 Kesalahan karena POS Tagger

Kesalahan pelabelan POS tag tercantum pada Tabel IV.4.

Tabel IV.4 Kesalahan POS Tagger

Kalimat	POS Tag Alfam	Seharusnya
Dia terlihat <i>cantik</i>	NN(<i>cantik</i>)	JJ(<i>cantik</i>)
Sam makan daging <i>merah</i>	PRP(<i>merah</i>)	JJ(<i>merah</i>)
Reagan telah <i>mati</i>	NN(<i>mati</i>)	VBI(<i>mati</i>)
saya melihat pria yang istrinya kamu <i>cintai</i>	NN(<i>cintai</i>)	VBT(<i>cintai</i>)

Penyebab kesalahan POS tagger yaitu *out-of-vocabulary* (OOV) dan model bahasa. Kasus OOV terjadi karena kata-kata yang salah diberi tag tidak terdapat di data latih, untuk contoh kasus di Tabel IV.4 kata “cantik”, “merah”, “mati”, “cintai” tidak ada di dalam leksikon. Kesalahan lain yaitu model bahasa hasil pelatihan n-gram dari POS tagger Alfam. Untuk kasus kalimat “Dia terlihat cantik” pada Tabel IV.4, pada model bahasa POS tagger Alfam skor kombinasi VBT NN (frekuensi kemunculan 619) lebih besar daripada kombinasi VBT JJ (frekuensi kemunculan 24). Oleh karena itu, setelah kata “terlihat” (dengan label POS VBT) diikuti dengan label POS NN, sehingga kata “cantik” diberi label POS NN .

IV.5.2 Kesalahan karena PCFG

Kesalahan Earley parser terletak pada mengelompokkan frase dan klausa, contohnya dapat dilihat pada Tabel IV.5 dan

Kesalahan pada level frase terjadi karena 2 faktor yaitu kesalahan akibat POS tag dan kesalahan pendefinisian aturan CFG. Contoh di Tabel IV.5, untuk frase “terlihat cantik” komponennya diberi POS yang salah maka mengakibatkan kesalahan pengkategorian frase, sedangkan untuk frase “pria itu” walaupun frase sudah diberi label POS tag benar namun masih dikategorikan dalam frase yang salah. Hal ini terjadi karena data latih yang dianotasi kurang valid, karena tidak ada justifikasi dari pakar bahasa.

Tabel IV.6.

Tabel IV.5 Kesalahan PCFG level frase

Kalimat	Parse Tree Irfan	Seharusnya
Level frase		
<i>dia terlihat cantik</i>	NP(VBI(terlihat) NN(cantik))	VP(VBI(terlihat) JJ(cantik))
<i>pria itu telah dibunuh polisi</i>	ADJP(NN(pria) DT(itu))	NP(pria itu)

Kesalahan pada level frase terjadi karena 2 faktor yaitu kesalahan akibat POS tag dan kesalahan pendefinisian aturan CFG. Contoh di Tabel IV.5, untuk frase “terlihat cantik” komponennya diberi POS yang salah maka mengakibatkan kesalahan pengkategorian frase, sedangkan untuk frase “pria itu” walaupun frase sudah diberi label POS tag benar namun masih dikategorikan dalam frase yang salah. Hal ini terjadi karena data latih yang dianotasi kurang valid, karena tidak ada justifikasi dari pakar bahasa.

Tabel IV.6 Kesalahan PCFG level klausa

Kalimat	Parse Tree Irfan	Seharusnya
Level klausa		
<i>bahwa dia berbohong</i>	S(bahwa dia	S(SBAR(bahwa dia

sudah disangka oleh semua orang	VP(berbohong sudah disangka oleh semua orang))	berbohong) VP(sudah disangka oleh semua orang))
apa yang dia katakan masuk akal	S(SBARQ(apa yang dia katakan masuk akal))	S(SBAR(apa yang dia katakan) VP(masuk akal))

Kesalahan pada level klausa terjadi karena 2 faktor, yaitu kesalahan pengkategorian frase dan kesalahan pada pendefinisian CFG. Pada Tabel IV.6 kalimat uji “bahwa dia berbohong sudah disangka oleh semua orang” seharusnya “bahwa dia berbohong” adalah 1 klausa dan “sudah disangka oleh semua orang” adalah frase kata kerja (VP), namun karena pengelompokan frase dan klausa yang salah mengakibatkan kesalahan identifikasi pada pemotongan klausa utama. Sedangkan pada kalimat “apa yang dia katakan masuk akal”, pada aturan CFG tidak ada pendefinisian $S \rightarrow SBAR VP$ (menurut Stanford parser aturan CFG ini valid).

IV.5.3 Kesalahan di Pengurai Dependensi

Evaluasi dari sisi pengurai dependensi adalah masih perlunya penanganan ambiguitas aturan pelabelan, karena dasar pembuatan aturan antara *head* dan *dependent* adalah POS tag maka diperlukan aturan lain yang lebih spesifik untuk menangani aturan label yang memiliki kriteria POS tag yang sama. Berikut pada Tabel IV.7 tercantum beberapa aturan pelabelan yang memiliki kriteria POS tag yang sama. Hal ini terjadi karena POS tag untuk kata kerja pasif dan aktif sama yaitu VBT. Sedangkan aturan *head* yang dibuat sebagai hasil adaptasi dari Collin *head rule* sudah mencukupi. Contoh kasus label ambigu dapat dilihat pada Tabel IV.8.

Tabel IV.7 Label dependensi ambigu

Label	Head	Dependent
AUX, AUXPASS	VBI	MD
NSUBJ, NSUBJPASS	VBT/VBI	NN/NNP/NNG
CSUBJ, CSUBJPASS	VBT/VBI	VBT/VBI

Tabel IV.8 Contoh kasus label ambigu

Input	dependensi	Label dependensi	Hasil aplikasi
Dole dikalahkan Clinton	(VBT(dikalahkan) , NNP(Clinton))	NSUBJ(VBT, NNP) NSUBJPASS(VBT, NNP)	NSUBJ(dikalahkan, Dole) NSUBJPASS(dikalahkan, Dole)

IV.5.4 Kesimpulan Pengujian

Kesalahan pengujian diakibatkan oleh 3 (tiga) level kesalahan yaitu POS tag, CFG dan aturan pelabelan. Pada jenis kesalahan POS tag dan CFG, faktor utama adalah kurangnya data latih. Sedangkan pada aturan pelabelan, kesalahan adalah kurang spesifiknya aturan untuk penanganan kata kerja pasif.

Bab V Penutup

V.1 Kesimpulan

Kesimpulan dari tesis ini adalah sebagai berikut:

1. Pohon konstituen bahasa Indonesia dapat dipetakan ke pohon dependensi.
2. Collin *head rule* dapat diadaptasi untuk bahasa Indonesia dan perlu dilakukan penyesuaian penelusuran head pada non-terminal NP.
3. Sebagian label non-klausal dari dependensi Stanford dapat diadaptasi untuk bahasa Indonesia (29 dari 37), sedangkan label klausal dapat digunakan seluruhnya (15 dari 15).
4. Pemetaan struktur frase ke dependensi untuk bahasa Indonesia dapat dilakukan dengan menggunakan pengurai konstituen dan modifikasi *resource* NLP berupa label dependensi dan *head rule* dari bahasa Inggris.
5. Pengurai dependensi dengan metode pemetaan sangat bergantung dari pohon konstituen keluaran dari POS tag dan aturan PCFG yang dipakai pengurai konstituen.
6. Kesalahan pengurai dependensi dengan metode pemetaan struktur frase ke struktur dependensi terjadi di 3 (tiga) level yaitu POS tag, CFG dan aturan pelabelan dependensi. Pada POS tag kesalahan terjadi karena OOV dan keterbatasan model bahasa, pada CFG kesalahan terjadi karena kesalahan pendefinisian aturan di data latih, sedangkan pada aturan pelabelan dependensi kesalahan terjadi karena tidak dapat membedakan kata kerja aktif dan pasif.

V.2 Saran

Untuk pengembangan lebih lanjut, saran-saran yang dapat diberikan pada tesis ini adalah:

1. Salah satu kesulitan dalam mengembangkan pengurai ini adalah keterbatasan POS tag dan pengurai PCFG, sehingga kesalahan berefek domino dari preproses hingga keluaran akhir sistem. Solusi yang mungkin adalah meningkatkan kualitas POS tag dan aturan CFG.
2. Aturan *head* yang menjadi aturan pemetaan struktur frase ke struktur dependensi masih harus diuji validitasnya oleh ahli bahasa. Karena diadaptasi dari aturan *head* Collin untuk bahasa Inggris.
3. Aturan pelabelan yang lebih spesifik untuk menghindari ambiguitas, selain dilihat dari POS tag, perlu penanganan khusus untuk kata kerja pasif.

DAFTAR PUSTAKA

- [AFI11] Afif, Irfan, “Studi Perbandingan Kinerja Algoritma CYK dan Algoritma Earley pada Pengurai Kalimat Menggunakan Probabilistic Context Free Grammar Bahasa Indonesia Sederhana”. Tugas Akhir untuk Sarjana Teknik Informatika, Institut Teknologi Bandung. 2011.
- [BIE95] Bies, Ann et. al., “Bracketing Guidelines for Treebank II Style Penn Treebank Project”. Linguistic Data Consortium.1995.
- [BOH03] Bohnet, Bernd., “Mapping Phrase Structures to Dependency Structures in the Case of (Partially) Free Word Order Languages”. MTT 2003. 2003.
- [COL03] Collins, Michael, “Head-Driven Statistical Models for Natural Language Parsing”. Association for Computational Linguistics. 2003.
- [COL95] Collin Head Rule
<http://www.cs.columbia.edu/~mcollins/papers/heads>
Diakses pada: 20 Oktober 2011
- [COV04] Covington, Michael A, “Important Additional Notes about Dependency Parsing”. The University of Georgia. 2004.
- [COV03] Covington, Michael A, “A Free-Word-Order Dependency Parser in Prolog”. Research Report AI-1990-01, Artificial Intelligence Programs, The University of Georgia. 2003.
- [DEB00] Debusmann, Ralph, “An Introduction to Dependency Grammar”. Universitat des Saarlandes. 2000.
- [FIL11] Ferdian, Filman, “Implementasi Pemrosesan Semantik dengan Representasi Pengetahuan Berbasis First Order Logic pada Sistem Evaluasi Pemahaman Bacaan Bahasa Indonesia”. Tugas Akhir untuk Sarjana Teknik Informatika, Institut Teknologi Bandung. 2012.
- [KAM11] Kamayani, Mia, “Dependency Parsing for Indonesian”. ICEEI. 2011.
- [KER91] Keraf, Gorys, “Tatabahasa Indonesia untuk SMK dan Umum”. Penerbit Nusa Indah. 1991.
- [MAR99] Marneffe, Marie-Catherine et. al., “Generating Typed Dependency Parses from Phrase Structure Parses”. In Proceedings of the Fifth

- International Conference on Language Resources and Evaluation (LREC'06), pages 449—454. 1999.
- [MAR08] Marneffe, Marie-Catherine et. al., “Stanford Typed Dependencies Manual”. <http://nlp.stanford.edu/downloads/lex-parser.shtml>. 2008.
- [NAS09] Nasution, Rosalina P., “Pembangunan Parser Bahasa Indonesia dengan Pendekatan Syntactic”. Tugas untuk Akhir Program Sarjana Teknik Informatika, Institut Teknologi Bandung. 2009
- [NIV06] Nivre, Joakim, “Dependency Parsing”. Tutorial at COLING-ACL. Sydney. 2006.
- [NIV99] Nivre, Joakim, “Dependency Grammar and Dependency Parsing”. Word Journal of The International Linguistic Association. 1999.
- [SAN90] Santorini, Beatrice, “Part-of-Speech Tagging Guidelines for the Penn Treebank Project”. Technical report MS-CIS-90--47, Department of Computer and Information Science, University of Pennsylvania. 1990.
- [STA11] Stanford Parser
<http://nlp.stanford.edu/software/lex-parser.shtml>
Diakses pada: 20 Oktober 2011
- [SUK09] Sukamto, Rosa A., “Penguraian Bahasa Indonesia dengan Menggunakan Pengurai Collins”. Tesis Program Magister Informatika ITB. 2009
- [WIC10] Wicaksono, Alfian Farizki. “Part of Speech Tagger Bahasa Indonesia dengan Menggunakan Hidden Markov Model”. Tugas untuk Akhir Program Sarjana Teknik Informatika, Institut Teknologi Bandung. 2010.

LAMPIRAN A PART OF SPEECH TAG

Berikut adalah POS tag yang dibuat oleh Alfian (2010):

No	POS Tag	Nama POS	Contoh
1	OP	<i>Open Parenthesis</i>	(([
2	CP	<i>Close Parenthesis</i>)]
3	GM	Garis Miring	/
4	;	Titik Koma	;
5	:	Titik Dua	:
6	“	Kutip	“”
7	.	Titik	.
8	,	Koma	,
9	-	Tanda Pisah	-
10	...	Elipsis	...
11	JJ	Kata Sifat	Malas, murah
12	RB	Adverbia	Sedang
13	NN	Common Noun	Mobil
14	NNP	Proper Noun	Bandung
15	NNG	Genitive Noun	Bukunya
16	VBI	Kata Kerja Intransitif	Pergi
17	VBT	Kata Kerja Transitif	Membeli
18	IN	Preposisi	Di, ke, dari
19	MD	Modal	Bisa
20	CC	Konjungsi Korelatif	Dan
21	SC	Konjungsi Subordinatif	Jika
22	DT	<i>Determiner</i>	Para, ini, itu
23	UH	Kata Seru	Wah, ah
24	CDO	Kata Bilangan Bertingkat	Pertama, kedua
25	CDC	Kata Bilangan Kolektif	Bertiga
26	CDP	Kata Bilangan Utama	Satu, dua
27	CDI	Kata Bilangan Tak Beraturan	Beberapa
28	PRP	Kata Ganti Persona	Saya, kamu
29	WP	Kata Ganti WH	Apa, siapa
30	PRN	Kata Ganti Bilangan	Kedua-duanya
31	PRL	Kata Ganti Lokatif	Sini, situ, kesana
32	NEG	Negasi	Bukan, tidak
33	SYM	Simbol	%^&\$
34	RP	Partikel	Pun, lah, kah
35	FW	Kata Asing	<i>Foreign, word</i>

LAMPIRAN B LABEL DEPENDENSI NON-KLAUSAL ADAPTASI STANFORD

***acomp*: adjectival complement**

It is an adjectival phrase which functions as the complement (like an object of the verb).

“Dia terlihat cantik” *acomp*(terlihat, cantik)

***advmod*: adverbial modifier**

It is an adverb or adverbial phrase (ADVP) that serves to modify the meaning of the word.

“Makanan yang diubah secara genetik” *advmod*(diubah, secara genetik)

***agent*: agent**

It is the complement of a passive verb which is introduced by the preposition “oleh” and does the action.

“Pria itu telah dibunuh oleh polisi” *agent*(dibunuh, polisi)

***amod*: adjectival modifier**

It is any adjectival phrase that serves to modify the meaning of the NP.

“Sam makan daging merah” *amod*(daging, merah)

***appos*: appositional modifier**

it is an NP immediately to the right of the first NP that serves to define or modify that NP. It includes parenthesized examples.

“Sam, saudara saya” *appos*(Sam, saudara)

***aux*: auxiliary**

It is a non-main verb of the clause, e.g. modal auxiliary. For example, “sudah”, “bisa”.

“Reagan telah mati” *aux*(mati, telah)

***auxpass*: passive auxiliary**

It is a non-main verb of the clause which contains the passive information.

“Kennedy telah dibunuh” *auxpass*(dibunuh, telah)

***cc*: coordination**

It is the relation between an element of a conjunct and the coordinating conjunction word of the conjunct.

“Bill besar dan jujur” *cc*(besar, dan)

***conj*: conjunction**

it is the relation between two elements connected by a coordinating conjunction, such as “dan”, “atau”, etc. We treat conjunctions asymmetrically: the head of the relation is the first conjunct and other conjunctions depend on it via the *conj* relation.

“Bill besar dan jujur” *conj*(besar, jujur)

cop: copula

It is the relation between the complement of a copular verb and the copular verb.

“Dia adalah peneliti” *cop*(peneliti,adalah)

dep: dependent

A dependency is labeled as *dep* when the system is unable to determine a more precise dependency relation between two words.

“Dia yang memukul saya” *dep*(yang,memukul)

det: determiner

It is the relation between the head of an NP and its determiner.

“Pria itu disini” *det*(pria,itu)

dobj: direct object

It is the noun phrase which is the object of the verb.

“Dia memberikan saya kenaikan” *dobj*(memberikan,kenaikan)

infmod: infinitival modifier

It is an infinitive that serves to modify the meaning of the NP.

“Saya tidak memiliki apapun untuk dikatakan” *infmod*(apapun,dikatakan)

iobj: indirect object

It is the noun phrase which is the object of the verb.

“Dia memberikan saya kenaikan” *iobj*(memberikan,saya)

mwe: multi-word expression

It is used for certain multi-word idioms that behave like a single function word.

For example, baik .. maupun, maka dari itu, oleh karena itu, seperti halnya.

“Baik laki-laki maupun perempuan diundang ke pesta” *mwe*(baik,maupun)

neg: negation modifier

It is the relation between a negation word and the word it modifies.

“Bill bukan peneliti” *neg*(peneliti,bukan)

nn: noun compound modifier

It is any noun that serves to modify the head noun.

“Harga minyak naik” *nn*(naik, harga)
nn(naik, minyak)

npadvmod: noun phrase as adverbial modifier

“Usia direktur adalah 65 tahun” *npadvmod*(usia,tahun)

nsubj: nominal subject

It is a noun phrase which is the syntactic subject of a clause.

“Clinton mengalahkan Dole” *nsubj*(mengalahkan,Clinton)

***nsubypass*: passive nominal subject**

It is a noun phrase which is the syntactic subject of a passive clause.

“Dole dikalahkan oleh Clinton” *nsubypass*(dikalahkan,Dole)

***num*: numeric modifier**

It is any number phrase that serves to modify the meaning of the noun.

“Sam makan 3 kambing” *num*(kambing,3)

***number*: element of compound number**

It is a part of a number phrase or currency amount.

“Saya kalah Rp 3.2 juta” *number*(Rp,juta)

***pobj*: object of a preposition**

“Saya duduk di atas kursi”

pobj(di,kursi)

pobj(atas,kursi)

***poss*: possession modifier**

It holds between the head of an NP and its possessive determiner, or a genitive’s complement.

“kantor mereka” *possj*(kantor, mereka)

***possessive*: possessive modifier**

It appears between the head of an NP and the genitive *ku, mu, nya*.

“Baju milik Bill” *possessive*(Bill, milik)

***predet*: predeterminer**

It is the relation between the head of an NP and a word that precedes and modifies the meaning of the NP determiner.

“Semua laki-laki disini” *predet*(laki-laki , semua)

***prep*: prepositional modifier**

A prepositional modifier of a verb, adjective, or noun is any prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another preposition. In the collapsed representation, this is used only for prepositions with NP complements.

“Saya melihat kucing di dalam topi” *prep*(kucing, di)
prep(kucing, dalam)

***quantmod*: quantifier phrase modifier**

It is an element modifying the head of a QP constituent.

“Sekitar 200 orang datang ke pesta” *quantmod*(200, sekitar)

***tmod*: temporal modifier**

A temporal modifier of a VP, NP or an ADJP is a bare noun phrase constituent that serves to modify the meaning of the constituent by specifying a time.

“Tadi malam, saya berenang di danau” *tmod*(berenang, malam)

LAMPIRAN C HEAD RULE MAGERMAN DAN COLLIN

1. Head Rule Magerman

- a. Kolom pertama adalah non-terminal. Kolom kedua mengindikasikan posisi mulai ketika mencari head (left untuk kategori head-initial, right untuk kategori head-final). Sisa dari baris adalah daftar dari kategori non-terminal dan pre-terminal yang merepresentasikan head rule.
- b. ** adalah nilai wildcard. Non-terminal dengan ** pada rulanya berarti apapun dapat menjadi headnya. Jadi, untuk kategori head-initial, ** berarti kata pertama selalu menjadi head, dan untuk sebuah kategori head-final, ** berarti kata terakhir selalu menjadi head. Pada sebagian besar kasus, ** berarti belum terdefinisi head yang spesifik untuk kategori tsb dan hal ini harus diinvestigasi lebih lanjut.
- c. Tabel Pohon Head digunakan sebagai berikut:
 - i. Gunakan Pohon Head Rule berdasarkan kategori NT dari konstituen
 - ii. Untuk setiap kategori X di Pohon Head Rule, scan anak-anak dari konstituen untuk kemunculan pertama (atau terakhir, untuk head-final) dari kategori X. jika X muncul, anak tsb adalah head.
 - iii. Jika tidak ada anak yang cocok dengan kategori manapun di list, gunakan anak pertama (atau terakhir, untuk head-final) sebagai head.
- d. NP adalah kasus spesial. Sebelum menentukan head rule untuk NP, lihat anak paling kanan dengan label yang berawalan N. jika ada, gunakan anak tsb sebagai head. Jika tidak ada anak berlabel awalan N, gunakan tree head rule.

ADJP	right	% QP JJ VBN VBG ADJP \$ JJR JJS DT FW **** RBR RBS RB
ADVP	left	RBR RB RBS FW ADVP CD **** JJR JJS JJ
CONJP	left	CC RB IN
FRAG	left	**
INTJ	right	**
LST	left	LS :
NAC	right	NN NNS NNP NNPS NP NAC EX \$ CD QP PRP VBG JJ JJS JJR ADJP FW
NP	right	EX \$ CD QP PRP VBG JJ JJS JJR ADJP DT FW RB SYM PRP\$
NP\$ SYM	right	NN NNS NNP NNPS NP NAC EX \$ CD QP PRP VBG JJ JJS JJR ADJP FW
PNP	right	**

PP	left	IN TO FW
PRN	left	**
PRT	left	RP
QP	right	CD NCD % QP JJ JJR JJS DT
RRC	left	VP NP ADVP ADJP PP
S	right	VP SBAR ADJP UCP NP
SBAR	right	S SQ SINV SBAR FRAG X
SBARQ	right	SQ S SINV SBARQ FRAG X
SINV	right	S VP VBZ VBD VBP VB SINV ADJP NP
SQ	right	VP VBZ VBD VBP VB MD SQ
UCP	left	**
VP	left	VBD VBN MD VBZ TO VB VP VBG VBP ADJP NP
WHADJP	right	JJ ADJP
WHADVP	left	WRB
WHNP	right	WDT WP WP\$ WHADJP WHPP WHNP
WHPP	left	IN TO FW
X	left	**

1. Head Rule Collin

- a. Kolom pertama adalah nomor field di baris tsb.
- b. Abaikan baris untuk NP dan gunakan aturan special untuk ini. Hilangkan ADJP, QP dan juga NP yang mendominasi kepemilikan (contoh (NP (NP the man 's) telescope) menjadi (NP the man 's telescope)). Hal ini dilakukan setelah proses parsing. Kemudian digunakan aturan berikut untuk menentukan head dari NP:
 - i. Jika kata terakhir diberi label POS, kembalikan kata terakhir
 - ii. Jika bukan, cari dari kanan ke kiri anak pertama dari jenis ini: NN, NNP, NNPS, NNS, NX, POS, or JJR
 - iii. Jika bukan, cari dari kiri ke kanan anak pertama yang berupa NP
 - iv. Jika bukan, cari dari kanan ke kiri anak pertama yang berupa \$, ADJP atau PRN
 - v. Jika bukan, cari dari kanan ke kiri anak pertama yang berupa CD
 - vi. Jika bukan, cari dari kanan ke kiri anak pertama yang berupa JJ, JJS, RB atau QP
 - vii. Jika bukan, kembalikan kata terakhir

20	ADJP	0	NNS QP NN \$ ADVP JJ VBN VBG ADJP JJR NP JJS DT FW RBR RBS SBAR RB
15	ADVP	1	RB RBR RBS FW ADVP TO CD JJR JJ IN NP JJS NN
5	CONJP	1	CC RB IN
2	FRAG	1	
2	INTJ	0	
4	LST	1	LS :
19	NAC	0	NN NNS NNP NNPS NP NAC EX \$ CD QP PRP VBG JJ JJS JJR ADJP FW

8	PP	1	IN TO VBG VBN RP FW
2	PRN	0	
3	PRT	1	RP
14	QP	0	\$ IN NNS NN JJ RB DT CD NCD QP JJR JJS
7	RRC	1	VP NP ADVP ADJP PP
10	S	0	TO IN VP S SBAR ADJP UCP NP
13	SBAR	0	WHNP WHPP WHADVP WHADJP IN DT S SQ SINV SBAR FRAG
7	SBARQ	0	SQ S SINV SBARQ FRAG
12	SINV	0	VBZ VBD VBP VB MD VP S SINV ADJP NP
9	SQ	0	VBZ VBD VBP VB MD VP SQ
2	UCP	1	
15	VP	0	TO VBD VBN MD VBZ VB VBG VBP VP ADJP NN NNS NP
6	WHADJP	0	CC WRB JJ ADJP
4	WHADVP	1	CC WRB
8	WHNP	0	WDT WP WP\$ WHADJP WHPP WHNP
5	WHPP	1	IN TO FW

LAMPIRAN D PCFG

```
<SENTENCE>::=<S> 0.9849246231155779
<SENTENCE>::=<S><*CC> 0.005025125628140704
<SENTENCE>::=<S><*.> 0.005025125628140704
<S>::=<NP><VP><*.> 0.014084507042253521
<S>::=<VP><NP><*.> 0.0035211267605633804
<S>::=<*SC><S><*,><S><*.> 0.007042253521126761
<S>::=<NP><VP> 0.5915492957746479
<S>::=<*SC><S><*,><S> 0.014084507042253521
<S>::=<SBARQ><*.> 0.035211267605633804
<S>::=<S><*CC><S><*.> 0.0035211267605633804
<S>::=<S><*CC><S> 0.028169014084507043
<S>::=<SBAR><*.> 0.0035211267605633804
<S>::=<S><*SC><S> 0.028169014084507043
<S>::=<S><*CC><VP> 0.01056338028169014
<S>::=<S><*,><S> 0.01056338028169014
<S>::=<VP><*,><S> 0.0035211267605633804
<S>::=<VP><*.> 0.0035211267605633804
<S>::=<VP> 0.045774647887323945
<S>::=<*CC><VP><*,><S> 0.007042253521126761
<S>::=<SBARQ> 0.007042253521126761
<S>::=<VP><*,><*SC><NP> 0.0035211267605633804
<S>::=<PP><*,><S> 0.017605633802816902
<S>::=<PP><VP> 0.014084507042253521
<S>::=<*UH><*,><S> 0.0035211267605633804
<S>::=<ADVP><NP> 0.0035211267605633804
<S>::=<*CC><S><*,><*CC><S> 0.0035211267605633804
<S>::=<*SC><S><*,><*SC><S> 0.007042253521126761
<S>::=<NP><ADJP> 0.014084507042253521
<S>::=<NP><NP> 0.0035211267605633804
<S>::=<*SC><NP><*,><S> 0.0035211267605633804
<S>::=<ADJP><NP><VP> 0.0035211267605633804
<S>::=<PP><VP><*SC><NP> 0.0035211267605633804
<S>::=<NP><ADJP><VP> 0.007042253521126761
<S>::=<*"><S><*"> 0.01056338028169014
<S>::=<S><*.> 0.014084507042253521
<S>::=<S><SBARQ> 0.0035211267605633804
<S>::=<S><VP> 0.0035211267605633804
<S>::=<*SC><NP><VP> 0.0035211267605633804
<S>::=<S><*SC><NP> 0.0035211267605633804
<S>::=<S><*,><*CC><S> 0.007042253521126761
<S>::=<*PRP><VP> 0.0035211267605633804
<S>::=<NP><ADVP><VP> 0.0035211267605633804
<S>::=<NP><VP><PP> 0.007042253521126761
<S>::=<NP><*SC><VP> 0.0035211267605633804
<S>::=<NP> 0.0035211267605633804
<S>::=<S> 0.0035211267605633804
<S>::=<PP><VP><NP> 0.007042253521126761
<S>::=<ADVP><*,><NP><VP> 0.0035211267605633804
<S>::=<ADVP><NP><VP> 0.0035211267605633804
<S>::=<PP><NP><VP> 0.0035211267605633804
<S>::=<*NN><VP><*.> 0.0035211267605633804
<S>::=<*NN><VP> 0.0035211267605633804
<NP>::=<*NNP> 0.09195402298850575
<NP>::=<*NN> 0.28160919540229884
<NP>::=<*PRP> 0.12835249042145594
<NP>::=<NP><*DT> 0.044061302681992334
<NP>::=<*NN><*NN> 0.08045977011494253
<NP>::=<*NN><NP> 0.0210727969348659
<NP>::=<NP><QP> 0.0038314176245210726
<NP>::=<NP><NP> 0.019157088122605363
<NP>::=<NP><ADJP> 0.01532567049808429
<NP>::=<NP><PP> 0.009578544061302681
<NP>::=<NP><*CC><NP> 0.009578544061302681
<NP>::=<ADVP><NP> 0.0019157088122605363
<NP>::=<NP><*SC><ADJP> 0.011494252873563218
<NP>::=<NP><*SC><VP> 0.019157088122605363
<NP>::=<*DT><NP> 0.007662835249042145
<NP>::=<NP><*IN><ADJP> 0.0019157088122605363
<NP>::=<NP><*,><NP><*,><*CC><NP> 0.0038314176245210726
<NP>::=<*NNP><NP> 0.0019157088122605363
```

<NP>::=<NP> 0.01532567049808429
 <NP>::=<*NN><*NN><*NN> 0.01532567049808429
 <NP>::=<*VBI> 0.0038314176245210726
 <NP>::=<*NN><ADJP> 0.005747126436781609
 <NP>::=<QP><NP> 0.007662835249042145
 <NP>::=<*NNG> 0.034482758620689655
 <NP>::=<*NN><*CDO> 0.0038314176245210726
 <NP>::=<ADJP><NP> 0.0019157088122605363
 <NP>::=<*NN><PP> 0.005747126436781609
 <NP>::=<NP><*,><*SC><NP> 0.0019157088122605363
 <NP>::=<*VBT><NP> 0.0038314176245210726
 <NP>::=<*FW><*FW> 0.0019157088122605363
 <NP>::=<*NN><*NNP> 0.007662835249042145
 <NP>::=<QP><*NN> 0.0019157088122605363
 <NP>::=<*VBT> 0.0019157088122605363
 <NP>::=<*NN><*NN><*NNP><*NNP> 0.0019157088122605363
 <NP>::=<*CDI><*NN> 0.005747126436781609
 <NP>::=<*CDI><NP> 0.0019157088122605363
 <NP>::=<*FW> 0.0019157088122605363
 <NP>::=<*WP><*RP> 0.0038314176245210726
 <NP>::=<*CC><*NN> 0.0019157088122605363
 <NP>::=<*PRP><*NN> 0.0038314176245210726
 <NP>::=<*PRP><*CDC> 0.0019157088122605363
 <NP>::=<*NNP><*NNP> 0.005747126436781609
 <NP>::=<*NN><NP><ADJP> 0.0038314176245210726
 <NP>::=<*DT><*NN> 0.0019157088122605363
 <NP>::=<*NN><*DT> 0.02490421455938697
 <NP>::=<*NN><*PRP> 0.005747126436781609
 <NP>::=<NP><*SC><*CDO> 0.0019157088122605363
 <NP>::=<*NNP><*DT> 0.0019157088122605363
 <NP>::=<NP><VP> 0.005747126436781609
 <NP>::=<ADJP><*NN> 0.0019157088122605363
 <NP>::=<NP><*,><NP><*,> 0.0019157088122605363
 <NP>::=<NP><*PRP> 0.0019157088122605363
 <NP>::=<*,><NP><*,> 0.0019157088122605363
 <NP>::=<NP><*SC><NP> 0.0038314176245210726
 <NP>::=<*NN><*NNG> 0.0038314176245210726
 <NP>::=<*NN><*CC><NP> 0.0019157088122605363
 <NP>::=<*CDC> 0.0019157088122605363
 <NP>::=<*CDC><*NN> 0.0019157088122605363
 <NP>::=<*PRP><PP> 0.0019157088122605363
 <NP>::=<*NEG><VP> 0.0019157088122605363
 <NP>::=<*NN><*RP> 0.0019157088122605363
 <NP>::=<*CDP><*NN> 0.0019157088122605363
 <NP>::=<*SC><*DT> 0.0019157088122605363
 <NP>::=<*NN><*JJ> 0.007662835249042145
 <NP>::=<*VBI><NP> 0.0019157088122605363
 <NP>::=<*JJ><NP> 0.0038314176245210726
 <NP>::=<*NN><*RB> 0.0019157088122605363
 <NP>::=<*NN><*SC><VP> 0.0019157088122605363
 <NP>::=<*NN><PP><VP> 0.0019157088122605363
 <NP>::=<*DT><*NN><ADJP> 0.0019157088122605363
 <NP>::=<*NN><NP><PP> 0.0019157088122605363
 <NP>::=<*JJ><*NN> 0.0019157088122605363
 <NP>::=<*NN><*SC><ADJP> 0.0019157088122605363
 <VP>::=<*VBT><NP><PP> 0.017587939698492462
 <VP>::=<*NEG><VP> 0.04020100502512563
 <VP>::=<*VBI><*RB> 0.002512562814070352
 <VP>::=<*VBI><PP> 0.05025125628140704
 <VP>::=<*NN> 0.010050251256281407
 <VP>::=<VP><NP> 0.06532663316582915
 <VP>::=<*JJ> 0.032663316582914576
 <VP>::=<QP><*NN> 0.002512562814070352
 <VP>::=<*VBT><PP> 0.020100502512562814
 <VP>::=<*VBT><NP><NP> 0.002512562814070352
 <VP>::=<*VBT><NP> 0.12562814070351758
 <VP>::=<*IN><VP> 0.002512562814070352
 <VP>::=<*VBI> 0.10050251256281408
 <VP>::=<*MD><VP> 0.04773869346733668
 <VP>::=<*VBT><ADVP> 0.007537688442211055
 <VP>::=<*VBT> 0.09045226130653267
 <VP>::=<ADJP><*IN><VP> 0.002512562814070352
 <VP>::=<*MD><*IN><VP> 0.002512562814070352
 <VP>::=<ADJP><VP> 0.005025125628140704

<VP>::=<VP><PP> 0.03015075376884422
 <VP>::=<VP><VP> 0.007537688442211055
 <VP>::=<VP><ADVP> 0.010050251256281407
 <VP>::=<VP><*RP> 0.005025125628140704
 <VP>::=<*RB><VP> 0.01507537688442211
 <VP>::=<ADVP><*MD><VP> 0.002512562814070352
 <VP>::=<*VBI><ADVP> 0.002512562814070352
 <VP>::=<*JJ><VP> 0.005025125628140704
 <VP>::=<*NEG><VP><ADVP> 0.002512562814070352
 <VP>::=<*VBT><SBAR> 0.002512562814070352
 <VP>::=<*VBI><ADJP> 0.005025125628140704
 <VP>::=<*VBI><NP> 0.03015075376884422
 <VP>::=<VP><*SC><VP> 0.002512562814070352
 <VP>::=<ADVP><VP> 0.02512562814070352
 <VP>::=<VP><NP><PP> 0.010050251256281407
 <VP>::=<*VBT><*NN> 0.04020100502512563
 <VP>::=<*MD><VP><NP> 0.005025125628140704
 <VP>::=<VP><ADVP><PP> 0.007537688442211055
 <VP>::=<ADJP> 0.002512562814070352
 <VP>::=<*NEG><ADJP> 0.002512562814070352
 <VP>::=<NP><VP> 0.002512562814070352
 <VP>::=<*MD><NP><VP> 0.002512562814070352
 <VP>::=<VP> 0.01256281407035176
 <VP>::=<*VBI><*NN> 0.017587939698492462
 <VP>::=<*VBT><NP><ADJP> 0.005025125628140704
 <VP>::=<*SC><*VBI> 0.007537688442211055
 <VP>::=<*JJ><ADVP> 0.002512562814070352
 <VP>::=<*VBT><VP> 0.007537688442211055
 <VP>::=<ADVP><*VBI> 0.002512562814070352
 <VP>::=<*RB><*JJ> 0.007537688442211055
 <VP>::=<ADJP><PP> 0.002512562814070352
 <VP>::=<*MD><*JJ> 0.002512562814070352
 <VP>::=<*JJ><*VBI> 0.002512562814070352
 <VP>::=<*JJ><*NN> 0.005025125628140704
 <VP>::=<*MD><VP><ADVP> 0.002512562814070352
 <VP>::=<*MD><PP> 0.002512562814070352
 <VP>::=<*VBI><PP><VP> 0.002512562814070352
 <VP>::=<*JJ><PP><PP> 0.002512562814070352
 <VP>::=<*MD><*VBT> 0.005025125628140704
 <VP>::=<*RB><*VBI> 0.002512562814070352
 <VP>::=<*NEG><VP><NP> 0.002512562814070352
 <VP>::=<*RB><NP> 0.002512562814070352
 <VP>::=<*NEG><VP><PP> 0.005025125628140704
 <VP>::=<*NN><*VBT> 0.002512562814070352
 <VP>::=<*MD><*VBI> 0.005025125628140704
 <VP>::=<*VBI><VP><PP> 0.002512562814070352
 <VP>::=<*NN><*VBI> 0.002512562814070352
 <VP>::=<*MD><*NN> 0.002512562814070352
 <VP>::=<*NN><VP> 0.005025125628140704
 <VP>::=<*JJ><PP> 0.005025125628140704
 <VP>::=<*FW> 0.002512562814070352
 <VP>::=<VP><PP><PP> 0.005025125628140704
 <VP>::=<*NN><ADJP> 0.002512562814070352
 <VP>::=<*JJ><NP> 0.002512562814070352
 <VP>::=<VP><*DT> 0.002512562814070352
 <VP>::=<*PRP><*VBI> 0.007537688442211055
 <VP>::=<*PRP><*VBT> 0.002512562814070352
 <PP>::=<*NN> 0.02040816326530612
 <PP>::=<*IN><NP> 0.47959183673469385
 <PP>::=<PP><PP> 0.01020408163265306
 <PP>::=<*IN><VP> 0.07142857142857142
 <PP>::=<*IN> 0.01020408163265306
 <PP>::=<*IN><*IN><NP> 0.02040816326530612
 <PP>::=<*IN><ADJP> 0.01020408163265306
 <PP>::=<PP><*NN> 0.01020408163265306
 <PP>::=<*IN><*IN> 0.01020408163265306
 <PP>::=<*IN><*VBT> 0.02040816326530612
 <PP>::=<*RB><*NN> 0.04081632653061224
 <PP>::=<*IN><*NNP> 0.02040816326530612
 <PP>::=<*IN><NP><NP> 0.01020408163265306
 <PP>::=<*IN><*NN> 0.12244897959183673
 <PP>::=<*IN><*PRL> 0.07142857142857142
 <PP>::=<*IN><*RB> 0.01020408163265306
 <PP>::=<*IN><*PRP> 0.01020408163265306

<PP>::=<*RB><NP> 0.01020408163265306
 <PP>::=<*DT><*NN> 0.01020408163265306
 <PP>::=<*IN><*JJ> 0.01020408163265306
 <PP>::=<*IN><*VBI> 0.01020408163265306
 <PP>::=<*NN><*DT> 0.01020408163265306
 <SBARQ>::=<WHNP><NP><ADJP> 0.07692307692307693
 <SBARQ>::=<WHNP><*SC><VP> 0.15384615384615385
 <SBARQ>::=<WHNP><NP><VP> 0.15384615384615385
 <SBARQ>::=<WHNP><NP><PP> 0.07692307692307693
 <SBARQ>::=<*WP><NP><VP> 0.15384615384615385
 <SBARQ>::=<*WP><NP> 0.07692307692307693
 <SBARQ>::=<*WP><VP> 0.07692307692307693
 <SBARQ>::=<*WP><*SC><VP> 0.23076923076923078
 <WHNP>::=<*WP> 0.9
 <WHNP>::=<WHNP><*CC><WHNP> 0.1
 <ADJP>::=<*JJ> 0.5306122448979592
 <ADJP>::=<VP><*JJ> 0.04081632653061224
 <ADJP>::=<*RB><*JJ> 0.10204081632653061
 <ADJP>::=<*NN><NP> 0.02040816326530612
 <ADJP>::=<*JJ><*CDP> 0.02040816326530612
 <ADJP>::=<*JJ><NP> 0.04081632653061224
 <ADJP>::=<*CDC> 0.02040816326530612
 <ADJP>::=<*NEG><*JJ> 0.04081632653061224
 <ADJP>::=<*JJ><*JJ> 0.08163265306122448
 <ADJP>::=<*NN><*JJ> 0.02040816326530612
 <ADJP>::=<ADJP><*,><ADJP><*CC><ADJP><*,> 0.02040816326530612
 <ADJP>::=<*VBI> 0.02040816326530612
 <ADJP>::=<*SC><*JJ> 0.04081632653061224
 <QP>::=<*CDP><*CDP> 0.375
 <QP>::=<*CDP> 0.5
 <QP>::=<*CDP><*CDP><*CDP> 0.125
 <SBAR>::=<WHNP><VP> 0.5
 <SBAR>::=<WHNP><NP><VP> 0.5
 <ADVP>::=<*RB><*JJ> 0.06451612903225806
 <ADVP>::=<ADJP><NP><VP> 0.03225806451612903
 <ADVP>::=<*IN><*RB> 0.03225806451612903
 <ADVP>::=<*RB> 0.2903225806451613
 <ADVP>::=<*JJ><NP> 0.03225806451612903
 <ADVP>::=<*JJ> 0.16129032258064516
 <ADVP>::=<*MD><*RB> 0.06451612903225806
 <ADVP>::=<*MD> 0.03225806451612903
 <ADVP>::=<*NEG><*RP> 0.03225806451612903
 <ADVP>::=<*JJ><*RP> 0.03225806451612903
 <ADVP>::=<*SC><*NN> 0.03225806451612903
 <ADVP>::=<*RB><*CDC> 0.03225806451612903
 <ADVP>::=<*JJ><*JJ> 0.03225806451612903
 <ADVP>::=<*VBT><*NN> 0.03225806451612903
 <ADVP>::=<*CDI><NP><ADJP> 0.03225806451612903
 <ADVP>::=<*RB><VP> 0.03225806451612903
 <ADVP>::=<*MD><*NN> 0.03225806451612903

LAMPIRAN E ATURAN PELABELAN

Non-Klausal

No.	Label	Head	Dependent	Position	Head Lexicon	Dependent lexicon
1	acomp	VBI, VBT	JJ	H < D		
2	advmod	VBI, VBT	RB	H < D		
3	agent	VBI	PRP, NN, NNP, NNG	H < D		
4	amod	NN	JJ	H < D		
5	appos	NN	NN, NNP, NNG	H < D		
6	aux	MD	VBI	H > D		
7	auxpass	MD	VBI	H > D		
8	cc	*	CC	H > D		
9	conj	*	*	H > D		
10	cop	NNP, NN	*	H > D		
11	dep	*	*	default		
12	det	NN	DT	H < D		
13	dobj	VBI, VBT	NN, NNP, NNG	H < D		
14	infmod	NN, NNP, NNG	VBI, VBT	H < D		
15	iobj	VBI, VBT	PRP, NN, NNP, NNG	H < D		
16	neg	*	NEG	H > D		
17	nn	NN, NNP, NNG	NN, NNP, NNG	H < D		
18	npadvmod	NN, NNP	*	H < D		
19	nsubj	VBT, VBI	NN, NNP, NNG	H > D		
20	nsubjpass	VBT, VBI	NN, NNP, NNG	H > D		
21	num	NN, NNP, NNG	CD	H > D		
22	number	NN, NNP, NNG	CD	H < D	rp	ratus, ribu, juta
23	pobj	IN	NN, NNP, NNG	H < D		

24	poss	NN, NNP, NNG	PRP	H < D		
25	possessive	NN, NNP	VBT, VBI	H > D		milik
26	predet	NN, NNP, NNG	CDI, CDP	H > D		
27	prep	PRL, NN, NNP, NNG	IN	H < D		
28	quantmod	CDI	CDP	H > D		
29	tmod	VBT, VBI, NN, NNP, NNG	RB	H > D		

Klausal

No.	Label	Head	Dependent	Position	Head Lexicon	Dependent lexicon	Clause Identifier
1	advcl	VBI, VBT	VBI, VBT	H < D			saat, ketika, karena, sebab
2	ccomp	VBI, VBT	VBI, VBT	H < D			bahwa
3	complm	VBI, VBT	SC	H > D		bahwa	
4	csbj	VBI, VBT	VBI, VBT	H > D			yang
5	csubjpass	VBI, VBT	VBI, VBT	H > D			
6	mark	VBI, VBT	*	H > D		setelah, sebelum	
7	parataxis	VBI, VBT	VBI, VBT	H > D			
8	pcomp	IN	VBI, VBT	H < D			
9	preconj	*	CC	H < D	baik	maupun	
10	prepc	VBI, VBT	VBI, VBT	H < D			semua kata preposisi
11	purpcl	VBI, VBT	VBI, VBT	H < D			dalam rangka (untuk)
12	rcmod	PRP, NN, NNP, NNG	VBI, VBT	H < D			yang, dimana, tempat
13	ref	PRP, NN, NNP, NNG	SC	H < D		yang, dimana,	

						tempat	
14	rel	VBT, VBI	PRP, NN, NNP, NNG	H > D			yang, dimana, tempat
15	xcomp	VBT, VBI	VBT, VBI	H < D			bahwa
16	xsubj	VBT, VBI	PRP, NN, NNP, NNG	H > D			

LAMPIRAN F HASIL PENGUJIAN

1. Kalimat Sederhana dan Relasi Dependensinya

Skenario 1: masukan adalah pohon konstituen valid				
No	Kalimat	Relasi Dependensi (Manual)	Relasi Dependensi (Aktual)	Hasil
1	dia terlihat cantik	acomp(terlihat, cantik)	acomp(terlihat, cantik)	benar
2	makanan yang diubah secara genetik	advmod(diubah, genetik)	advmod(diubah, genetik)	benar
3	pria itu telah dibunuh polisi	agent(dibunuh, polisi)	agent(dibunuh, polisi)	benar
4	sam makan daging merah	amod(daging, merah)	amod(daging, merah)	benar
5	sam, saudara saya	appos(sam, saudara)	appos(sam, saudara)	benar
6	reagan telah mati	aux(mati, telah)	aux(mati, telah) auxpass(mati, telah)	ambigu
7	kennedy telah dibunuh	auxpass(dibunuh, telah)	auxpass(dibunuh, telah) aux(dibunuh, telah)	ambigu
8	bill besar dan jujur	cc(besar, dan) conj(besar, jujur)	cc(besar, dan) conj(besar, jujur)	benar benar
9	dia adalah peneliti	cop(peneliti, adalah)	cop(peneliti, adalah)	benar
10	dia yang memukul saya	dep(yang, memukul)	dep(yang, memukul)	benar
11	pria itu disini	det(pria, itu)	det(pria, itu)	benar
12	dia memberikan saya kenaikan	doj(memberikan, kenaikan) ioj(memberikan, saya)	doj(memberikan, kenaikan) ioj(memberikan, saya)	benar
13	saya tidak memiliki apapun untuk dikatakan	infmod(apapun, dikatakan)	infmod(apapun, dikatakan)	benar
14	bill bukan peneliti	neg(peneliti, bukan)	neg(peneliti, bukan)	benar
15	harga minyak naik	nn(harga, minyak)	nn(harga, minyak)	benar
16	usia direktur adalah 65 tahun	npadvmod(usia, tahun)	npadvmod(usia, tahun)	benar
17	clinton mengalahkan dole	nsubj(mengalahkan, Clinton)	nsubj(mengalahkan, Clinton)	ambigu
18	dole dikalahkan Clinton	nsubjpass(dikalahkan, Dole)	nsubjpass(dikalahkan, Dole)	ambigu
19	sam makan 3 kambing	num(kambing, 3)	num(kambing, 3)	benar
20	saya kalah rp 3.2 juta	number(rp, juta)	number(rp, juta)	benar

21	saya duduk di atas kursi	pobj(di, kursi) pobj(atas, kursi)	pobj(di, kursi) pobj(atas, kursi)	benar benar
22	kantor mereka	poss(kantor, mereka)	poss(kantor, mereka)	benar
23	baju milik bill	possessive(bill, milik)	possessive(bill, milik)	benar
24	semua laki-laki disini	predet(laki-laki, semua)	predet(laki-laki, semua)	benar
25	saya melihat kucing di sana	prep(kucing, di) prep(kucing, sana)	prep(kucing, di) prep(kucing, sana)	benar benar
26	sekitar 200 orang datang ke pesta	quantmod(200, sekitar)	quantmod(200, sekitar)	benar
27	tadi malam, saya berenang di pantai	tmod(berenang, malam)	tmod(berenang, malam)	benar
Average				23/27 = 85.2%

Skenario 2: masukan adalah kalimat				
No	Kalimat	Relasi Dependensi (Manual)	Relasi Dependensi (Aktual)	Benar/Salah
1	dia terlihat cantik	acomp(terlihat, cantik)	agent(terlihat, cantik)	salah
2	makanan yang diubah secara genetik	advmod(diubah, genetik)	advmod(diubah, genetik)	benar
3	pria itu telah dibunuh polisi	agent(dibunuh, polisi)	agent(dibunuh, polisi)	benar
4	sam makan daging merah	amod(daging, merah)	poss(daging, merah)	salah
5	sam, saudara saya	appos(sam, saudara)	appos(sam, saudara)	benar
6	reagan telah mati	aux(mati, telah)	aux(mati, telah) auxpass(mati, telah)	ambigu
7	kennedy telah dibunuh	auxpass(dibunuh, telah)	aux(dibunuh, telah) auxpass(dibunuh, telah)	ambigu
8	bill besar dan jujur	cc(besar, dan) conj(besar, jujur)	cc(besar, dan) conj(besar, jujur)	benar
9	dia adalah peneliti	cop(peneliti, adalah)	cop(peneliti, adalah)	benar
10	dia yang memukul saya	dep(yang, memukul)	dep(yang, memukul)	benar
11	pria itu disini	det(pria, itu)	det(pria, itu)	benar
12	dia memberikan saya kenaikan	doj(memberikan, kenaikan) ioj(memberikan, saya)	doj(memberikan, kenaikan) ioj(memberikan, saya)	benar
13	saya tidak memiliki apapun untuk dikatakan	infmod(apapun, dikatakan)	infmod(apapun, dikatakan)	benar

14	bill bukan peneliti	neg(peneliti, bukan)	neg(peneliti, bukan)	benar
15	harga minyak naik	nn(harga, minyak)	nn(harga, minyak)	benar
16	usia direktur adalah 65 tahun	npadvmod(usia, tahun)	npadvmod(usia, tahun)	benar
17	clinton mengalahkan dole	nsubj(mengalahkan, Clinton)	nsubj(mengalahkan, Clinton) nsubjpass(mengalahkan, Clinton)	ambigu
18	dole dikalahkan Clinton	nsubjpass(dikalahkan, Dole)	nsubjpass(dikalahkan, Dole) nsubj(dikalahkan, Dole)	ambigu
19	sam makan 3 kambing	num(kambing, 3)	num(kambing, 3)	benar
20	saya kalah rp 3.2 juta	number(rp, juta)	number(rp, juta)	benar
21	saya duduk di atas kursi	pobj(di, kursi) pobj(atas, kursi)	pobj(di, kursi) pobj(atas, kursi)	benar benar
22	kantor mereka	poss(kantor, mereka)	poss(kantor, mereka)	benar
23	baju milik bill	possessive(bill, milik)	possessive(bill, milik)	benar
24	semua laki-laki disini	predet(laki-laki, semua)	predet(laki-laki, semua)	benar
25	saya melihat kucing di sana	prep(kucing, di) prep(kucing, sana)	prep(kucing, di) prep(kucing, sana)	benar benar
26	sekitar 200 orang datang ke pesta	quantmod(200, sekitar)	quantmod(200, sekitar)	benar
27	tadi malam, saya berenang di pantai	tmod(berenang, tadi)	tmod(berenang, tadi)	benar
Average				21/27 = 77%

2. Kalimat Subordinatif dan Relasi Dependensinya

Skenario 1: masukan adalah pohon konstituen valid				
No	Kalimat	Relasi Dependensi (Manual)	Relasi Dependensi (Aktual)	Hasil
1	kecelakaan itu terjadi saat malam dimulai	advcl(terjadi, dimulai)	advcl(terjadi, dimulai)	benar
2	dia berkata bahwa kamu suka berenang	ccomp(berkata suka) complm(suka, bahwa)	ccomp(berkata suka) complm(suka, bahwa)	benar benar
3	apa yang dia katakan masuk akal	csbj(masuk, katakan)	csbj(masuk, katakan) csbjpass(masuk, katakan) parataxis(masuk, katakan)	ambigu

4	bahwa dia berbohong sudah disangka oleh semua orang	csubjpass(disangka, berbohong)	csubjpass(disangka, berbohong) csubj(disangka, berbohong) parataxis(disangka, berbohong)	ambigu
5	pasukan dilibatkan dalam peperangan setelah insurgent menyerang	mark(menyerang, setelah)	mark(menyerang, setelah)	benar
6	lelaki itu, dikatakan John, pergi pagi pada pagi hari	parataxis(pergi, dikatakan)	parataxis(pergi, dikatakan) csubj(pergi, dikatakan) csubjpass(pergi, dikatakan)	ambigu
7	mereka mendengar mengenai dirimu yang membolos	pcomp(mengenai, membolos)	pcomp(mengenai, membolos)	benar
8	baik laki-laki maupun perempuan ada disini	preconj(baik, maupun)	preconj(baik, maupun)	benar
9	dia membelinya tanpa membayar premi	prepc(membeli, membayar)	prepc(membeli, membayar)	benar
10	dia berbicara kepadanya dalam rangka untuk mengamankan akun	purpcl(berbicara, mengamankan)	purpcl(berbicara, mengamankan)	benar
11	saya melihat laki-laki yang kamu cintai	rmod(laki-laki, cintai) ref(laki-laki, yang)	rmod(laki-laki, cintai) ref(laki-laki, yang)	benar benar
12	saya melihat pria yang istrinya kamu cintai	rel(cintai, istri)	rel(cintai, istri)	benar
13	tom suka makan ikan	xsubj(makan, tom)	xsubj(makan, tom)	benar
Average				10/13 = 77%

Skenario 2: masukan adalah kalimat				
No	Kalimat	Relasi Dependensi (Manual)	Relasi Dependensi (Aktual)	Hasil
1	kecelakaan itu terjadi saat malam dimulai	advcl(terjadi, dimulai)	advcl(terjadi, dimulai)	benar
2	dia berkata bahwa kamu suka berenang	ccomp(berkata suka) complm(suka, bahwa)	ccomp(berkata suka) complm(suka, bahwa)	benar benar
3	apa yang dia katakan masuk akal	csubj(masuk, katakan)	csubj(masuk, katakan) csubjpass(masuk, katakan) parataxis(masuk, katakan)	ambigu

			katakan)	
4	bahwa dia berbohong sudah disangka oleh semua orang	csubjpass(disangka, berbohong)	csubjpass(disangka, berbohong) csubj(disangka, berbohong) parataxis(disangka, berbohong)	ambigu
5	pasukan dilibatkan dalam peperangan setelah insurgent menyerang	mark(menyerang, setelah)	mark(menyerang, setelah)	Benar
6	lelaki itu, dikatakan John, pergi pagi pada pagi hari	parataxis(pergi, dikatakan)	parataxis(pergi, dikatakan) csubj(pergi, dikatakan) csubjpass(pergi, dikatakan)	Ambigu
7	mereka mendengar mengenai dirimu yang membolos	pcomp(mengenai, membolos)	n.a	Salah
8	baik laki-laki maupun perempuan ada disini	preconj(baik, maupun)	preconj(baik, maupun)	Benar
9	dia membelinya tanpa membayar premi	prepc(membelinya, membayar)	n.a	Salah
10	dia berbicara kepadanya dalam rangka untuk mengamankan akun	purpcl(berbicara, mengamankan)	purpcl(berbicara, mengamankan)	Benar
11	saya melihat laki-laki yang kamu cintai	rcmod(laki-laki, cintai) ref(laki-laki, yang)	n.a	Salah salah
12	saya melihat pria yang istrinya kamu cintai	rel(cintai, istrinya)	n.a	Salah
13	tom suka makan ikan	xsubj(makan, tom)	xsubj(makan, tom)	Benar
Average				6/13 = 46 %