

**PEMBELAJARAN RELASI NON-TAKSONOMI
DARI DOKUMEN WEB**



Disusun Oleh:

N a m a : Elang Cergas Pembrani
NIM : 14523290

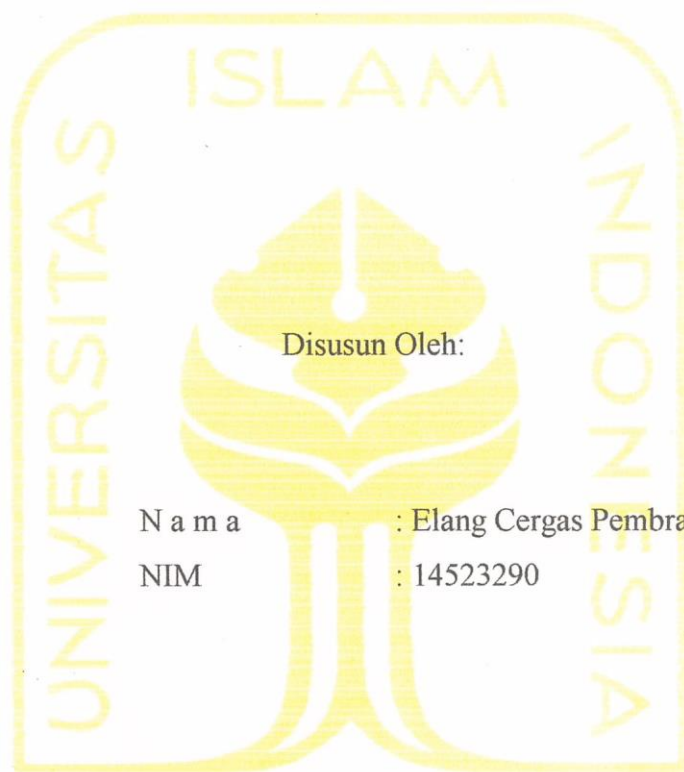
**PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2018**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PEMBELAJARAN RELASI NON-TAKSONOMI

DARI DOKUMEN WEB

TUGAS AKHIR



Yogyakarta, 7 Januari 2019

Pembimbing,

(Dhomas Hatta Fudholi S.T., M.Eng., Ph.D.)

HALAMAN PENGESAHAN DOSEN PENGUJI**PEMBELAJARAN RELASI NON-TAKSONOMI
DARI DOKUMEN WEB****TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 7 Januari 2019

Tim Penguji

Dhomas Hatta Fudholi S.T., M.Eng., Ph.D.

Anggota 1

Ahmad Fathan Hidayatullah, S.T., M.Cs.

Anggota 2

Chanifah Indah Ratnasari, S.Kom., M.Kom.

Mengetahui,

Ketua Program Studi Teknik Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Elang Cergas Pembrani

NIM : 14523290

Tugas akhir dengan judul:

**PEMBELAJARAN RELASI NON-TAKSONOMI
DARI DOKUMEN WEB**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 7 Januari 2019


(Elang Cergas Pembrani)

HALAMAN PERSEMBAHAN



Dengan mengucap syukur *Alhamdulillah*, karya ini kupersembahkan untuk orang-orang yang kusayangi:

Kedua orang tua tersayang,
Ayahanda Arman Yulianta
Ibunda Ary Retnowati

Motivator terbesar dalam hidupku yang selalu memberikan doa, semangat, nasehat dan motivasi demi kelancaran semuanya. Semoga dengan prestasi kecil ini dapat membuat bangga orang tua tercinta, *Aamiin*.

Ketiga adik tersayang,
Gagas Akar Ilalang
Kalia Bening Alirya
Senyumi Cahaya Pagi

Saudara sekaligus sahabat yang selalu memberikan semangat dan bersedia menjadi tempat untuk bercerita. Semoga tujuan kita membahagiakan kedua orang tua kita dan orang yang kita cintai tercapai, *aamiin*.

HALAMAN MOTO

“Love all, trust a few, do wrong to none”

(William Shakespeare)

“Live as if you were to die tomorrow.

Learn as if you were to live forever.”

(Mahatma Gandhi)

“The only true wisdom is in knowing you know nothing.”

(Socrates)

*“Ada dua kenikmatan yang banyak manusia tertipu,
yaitu nikmat sehat dan waktu senggang”*

(HR. Bukhari no. 6412, dari Ibn ‘Abbas)

*“Allah tidak akan membebani seorang manusia
melainkan sesuai dengan kemampuan manusia itu sendiri”*

(Q.S. Al Baqarah: 286)

KATA PENGANTAR



Assalamu 'alaikum warahmatullahi wabarakatuh.

Alhamdulillahirabbil'alamiin, puji syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya kepada kita semua. Sholawat serta salam semoga selalu tercurah kepada junjungan Nabi Muhammad SAW beserta keluarga dan sahabatnya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “PEMBELAJARAN RELASI NON-TAKSONOMI DARI DOKUMEN WEB” dengan baik dan lancar.

Laporan Penelitian Tugas Akhir ini disusun sebagai salah satu syarat yang harus dipenuhi untuk menyelesaikan pendidikan pada jenjang Strata 1 (S1) Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Penulis menyadari bahwa dalam penulisan dan penyusunan laporan ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih sebesar-besarnya kepada:

1. Bapak Hendrik, S.T., M.Eng., selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Sarjana Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Dhomas Hatta Fudholi S.T., M.Eng., Ph.D., selaku dosen pembimbing tugas akhir yang telah sabar dalam memberikan ilmu dan saran yang sangat bermanfaat serta meluangkan waktunya untuk membimbing penulis dalam menyelesaikan tugas akhir ini.
4. Bapak dan Ibu Dosen Jurusan Teknik Informatika yang telah memberikan ilmunya kepada penulis.
5. Orang tua tersayang Bapak Arman Yulianta dan Ibu Ary Retnowati, serta ketiga adik tersayang Gagah Akar Ilalang, Kalia Bening Alirya, dan Senyumi Cahaya Pagi atas segala dukungan, arahan, pengorbanan, kasih sayang, nasihat, serta doa yang tidak pernah putus kepada penulis.
6. Semua pihak yang dicintai dan disayangi penulis, yang tidak dapat disebutkan satu persatu, terima kasih atas bantuan dan doanya.

Penulis menyadari bahwa Penelitian Tugas Akhir ini tidaklah sempurna dan tidak dapat terlepas dari banyaknya kekurangan karena keterbatasan kemampuan dan pengalaman penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun sebagai bahan evaluasi agar dapat lebih baik lagi ke depannya. Semoga Tugas Akhir ini dapat diterima dan memberikan manfaat bagi orang-orang yang menggunakannya. *Aamiin.*

Wassalamu 'alaikum warahmatullahi wabarakatuh.

Yogyakarta, 7 Januari 2019

(Elang Cergas Pembrani)

SARI

Ontologi memiliki peran yang sangat penting dalam sistem ilmu pengetahuan modern karena ontologi memberikan standar formal dalam penyusunan struktur basis pengetahuan serta dalam penggunaan kembali (*reuse*) dan berbagi pakainya (*sharing*). Pembangunan ontologi skala besar di mana terdapat sangat banyak data dan informasi, penentuan nama relasi yang ideal dilakukan secara otomatis tanpa keterlibatan manusia. Penentuan nama relasi secara otomatis inilah yang menjadi tantangan dalam pembangunan ontologi secara otomatis. Salah satu alternatif solusi untuk memecahkan masalah ini adalah dengan memanfaatkan data dan informasi berlimpah dari Web untuk menemukan kemungkinan nama relasi yang sesuai. Dengan memanfaatkan pengolahan bahasa alami, komputer dapat diprogram untuk mencari kemungkinan nama relasi yang sesuai dari kelas-kelas yang diberikan. Dalam penelitian ini, dibangun rancangan alur kerja dan program untuk melakukan pembelajaran relasi non-taksonomi (*Learning Non-Taxonomic Relationships of Ontologies* atau LNTRO) secara otomatis dengan bersumber data dari dokumen Web menggunakan mesin pencari. Untuk mencapai hal tersebut, penelitian dilakukan dimulai dengan studi pustaka untuk meneliti penyelesaian masalah LNTRO terdahulu, kemudian merancang alur kerja pembelajaran relasi non-taksonomi, dilanjutkan dengan implementasi program, dan terakhir pengujian alur kerja dan program yang dibangun. Ada dua metode ekstraksi kata yang diusulkan, yaitu metode MAANS (*Maximum of Absolute Average of Normalized Score*) dan metode MFLR (*Most Frequent Local Recommendation*). Proses penelitian tersebut menghasilkan suatu program yang dapat digunakan untuk melakukan pembelajaran relasi non-taksonomi secara otomatis. Pengujian dilakukan dengan dua data uji yang berbeda dengan empat parameter pembelajaran yang berbeda. Dari hasil pembelajaran dan pengujian, nama relasi non-taksonomi yang dihasilkan sudah sesuai dan relevan dengan penalaran peneliti. Hasil evaluasi kata menunjukkan nama relasi non-taksonomi juga cukup relevan.

Kata kunci: ontologi, relasi non-taksonomi, LNTRO, web, mesin pencari

GLOSARIUM

<i>bottom-up</i>	Pendekatan penyelesaian masalah dari yang paling rinci ke yang umum
<i>bugs</i>	Kecacatan/kekurangan program atau perangkat lunak
<i>cache</i>	Basisdata komputer yang digunakan untuk menyimpan suatu data dalam waktu sementara sehingga lebih mudah dan cepat diakses
<i>corpus annotation</i>	Proses pendeteksian jenis kata dari setiap kata yang ada dalam suatu korpus
<i>corpus construction</i>	Proses pembangunan suatu korpus sebagai sumber data pembelajaran
<i>error</i>	Kesalahan fatal pada program yang mungkin menyebabkan program berhenti berjalan
<i>heuristic</i>	Metode atau algoritma di mana komputer dapat mempelajari sendiri dan memperkirakan solusi yang mungkin dari permasalahan yang dihadapi
<i>instance</i>	Suatu individu tunggal dari suatu kerangka objek atau kelas data
<i>keyword</i>	Kata kunci (biasa digunakan pada mesin pencari)
<i>knowledge engineer</i>	Ahli yang bekerja dengan data, informasi dan pengetahuan yang disimpan dan diolah oleh komputer
korpus	dokumen yang terdiri dari himpunan kalimat-kalimat yang merepresentasikan sumber data tekstual dan biasanya berukuran besar
LNTRO	<i>Learning Non-Taxonomic Relationships of Ontologies</i> , istilah yang mengacu pada proses pembelajaran relasi non-taksonomi pada ontologi
<i>machine learning</i>	Bidang kecerdasan buatan yang bertujuan agar komputer dapat mempelajari dan menyelesaikan suatu masalah berdasarkan data
<i>machine-processable</i>	Suatu objek atau teks dengan bentuk atau format sedemikian rupa sehingga dapat diolah oleh komputer (mesin)
<i>market share</i>	Pangsa pasar
<i>Mining</i>	proses pengolahan untuk menemukan makna baru dari suatu objek, misal <i>text mining</i>
<i>noun</i>	Kata benda

<i>overhead</i>	Pengeluaran atau sumber daya tambahan yang tidak diperlukan dan membebani sistem
<i>Part-of-Speech tag</i>	Penanda pada suatu kata yang menunjukkan jenis kata verbal dari kata tersebut (kata benda, kata kerja, kata sifat dan lain-lain)
<i>pipeline</i>	Istilah pemrograman untuk fungsi-fungsi yang dirantai untuk mengolah data secara beruntun
<i>POS Tagging</i>	<i>Part-of-Speech Tagging</i> , proses untuk mengenali jenis kata dalam suatu korpus/kalimat
<i>preprocessing</i>	Pengolahan awal dari suatu data objek atau teks sehingga data dapat dipahami dengan baik oleh komputer dan siap diolah lebih lanjut
<i>refinement</i>	Proses perbaikan untuk meningkatkan kualitas atau akurasi dari suatu objek atau data
<i>regular expression</i>	Sintaks pemrograman untuk merepresentasikan suatu pola tekstual
<i>search engine</i>	Mesin pencari di Internet yang digunakan untuk mencari data atau dokumen berdasarkan kata kunci (query)
<i>snippet</i>	Ringkasan dari suatu teks
<i>state-of-the-art</i>	Teknologi atau pengetahuan paling canggih dan mutakhir dalam suatu bidang tertentu
<i>stopwords</i>	Himpunan kata yang dianggap tidak relevan dalam konteks pengolahan bahasa alami
<i>template</i>	Contoh dari suatu objek atau data
<i>term evaluation</i>	Proses evaluasi kata berdasarkan kriteria tertentu
<i>term extraction</i>	Proses ekstraksi kata dari suatu korpus berdasarkan mekanisme penyaringan dan validasi tertentu
<i>top-down</i>	Pendekatan penyelesaian masalah dari yang umum ke yang paling rinci
<i>tuple</i>	Objek yang merepresentasikan satu atau beberapa data sekaligus dalam satu variabel atau dokumen
<i>verb</i>	Kata kerja
<i>wordnet</i>	Suatu basisdata besar yang berisi setiap kata dari suatu bahasa beserta hubungan antar katanya

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian.....	4
1.7. Sistematika Penulisan	5
BAB II LANDASAN TEORI.....	6
2.1. Akuisisi Pengetahuan.....	6
2.1.1. Definisi Akuisisi Pengetahuan	6
2.1.2. Metode Akuisisi Pengetahuan	6
2.2. Web Semantik.....	7
2.3. Ontologi	8
2.3.1. Definisi Ontologi.....	8
2.3.2. Komponen Ontologi	9
2.3.3. Relasi Taksonomi dan Non-Taksonomi	9
2.4. Metode Pembelajaran Relasi Non-Taksonomi	10
2.4.1. <i>LNTRO based on the Extraction of Association Rules</i>	10

2.4.2. <i>LNTRO based on the Extraction of Generalized Association Rules</i>	11
2.4.3. <i>LNTRO based on Queries on Web Search Engines</i>	12
2.4.4. <i>LNTRO based on logistic regression</i>	14
BAB III RANCANGAN ALUR KERJA	15
3.1. Alur Kerja Pembelajaran Relasi Non-Taksonomi	15
3.2. Pembangunan Korpus (<i>Corpus Construction</i>).....	17
3.2.1. Mesin Pencari Web (Web Search Engine)	18
3.2.2. Pembangunan Kata Kunci (<i>Keyword</i>).....	22
3.2.3. Proses Pencarian pada Mesin Pencari	22
3.2.4. <i>Preprocessing</i>	23
3.2.5. Ekstraksi Kalimat Relevan dalam Korpus.....	24
3.3. Anotasi Korpus (<i>Corpus Annotation</i>)	26
3.3.1. <i>Part-of-Speech Tagging</i>	26
3.3.2. Seleksi Kata Kerja	28
3.4. Ekstraksi Kata (<i>Term Extraction</i>)	29
3.4.1. Perhitungan Frekuensi Kemunculan Kata	30
3.4.2. <i>Maximum of Absolute Average of Normalized Score (MAANS)</i>	30
3.4.3. <i>Most Frequent Local Recommendation (MFLR)</i>	32
3.4.4. Penyaringan Nama Relasi Taksonomis	33
3.4.5. Validasi Kata Kerja Menggunakan Kamus	34
3.5. Pengembangan Kata (<i>Term Refinement</i>)	35
3.5.1. Deteksi Kata Kerja Aktif dan Pasif	35
3.5.2. Pemilihan Nama Relasi Non-Taksonomi	37
3.6. Evaluasi Kata (<i>Term Evaluation</i>).....	37
3.6.1. Sampling Data untuk Evaluasi	38
3.6.2. Seleksi Kata Berdasarkan Skor Tertinggi	38
3.6.3. Proses Pencarian Jumlah Hit pada Mesin Pencari.....	38
BAB IV IMPLEMENTASI PROGRAM	40
4.1. Implementasi Program Utama	40
4.1.1. Kebutuhan Perangkat Lunak Program Utama.....	40
4.1.2. Pengembangan Model POS Tagger	43
4.1.3. Layanan Eksternal	47
4.1.4. Spesifikasi Alamat URL API	48

4.2. Implementasi Antarmuka Pengguna	49
4.2.1. Kebutuhan Perangkat Lunak Antarmuka Pengguna	49
4.2.2. Tampilan Antarmuka Pengguna	49
BAB V HASIL DAN PEMBAHASAN	60
5.1. Metodologi Pengujian	60
5.1.1. Langkah Pengujian	60
5.1.2. Data Uji	61
5.1.3. Parameter Pembelajaran	61
5.2. Hasil Pengujian	63
5.2.1. Pengujian Parameter Pembelajaran Pertama (Google).....	63
5.2.2. Pengujian Parameter Pembelajaran Kedua (Yahoo)	64
5.2.3. Pengujian Parameter Pembelajaran Ketiga (Ecosia)	65
5.2.4. Pengujian Parameter Pembelajaran Keempat (Semua)	66
5.3. Pembahasan Pengujian.....	67
5.3.1. Analisis Akurasi Hasil Pembelajaran	68
5.3.2. Analisis Metode Ekstraksi.....	71
5.3.3. Temuan Tambahan yang Berpotensi Mempengaruhi Hasil Pembelajaran	72
5.3.4. Kendala Pengujian.....	84
BAB VI KESIMPULAN DAN SARAN	85
6.1. Kesimpulan	85
6.2. Saran	85
DAFTAR PUSTAKA	86
LAMPIRAN.....	88

DAFTAR TABEL

Tabel 2.1 Solusi <i>LNTRO based on the Extraction of Association Rules</i>	11
Tabel 2.2 Solusi <i>LNTRO based on the Generalized Extraction of Association Rules</i>	12
Tabel 2.3 Solusi <i>LNTRO based on Queries on Web Search Engines</i>	13
Tabel 2.4 Solusi <i>LNTRO based on logistic regression</i>	14
Tabel 3.1 Pengubahan data ke dalam bentuk kata kunci pada mesin pencari	22
Tabel 3.2 Korpus awal hasil pencarian melalui mesin pencari.....	22
Tabel 3.3 Korpus yang sudah diolah melalui tahapan <i>preprocessing</i> dan pemangkasan	25
Tabel 3.4 Korpus yang sudah diolah melalui tahapan <i>POS Tagging</i> dan seleksi kata	28
Tabel 3.5 Korpus yang sudah diubah menjadi daftar kata kerja beserta nilainya.....	30
Tabel 3.6 Himpunan frekuensi kata kerja dari beberapa sampel data (MAANS)	31
Tabel 3.7 Himpunan kata kerja unik dari semua sampel data (MAANS)	31
Tabel 3.8 Himpunan frekuensi kata kerja untuk salah satu sampel data	31
Tabel 3.9 Penjumlahan skor pada himpunan kata kerja unik dari semua data	31
Tabel 3.10 Perhitungan skor total setiap kata kerja metode MAANS	32
Tabel 3.11 Himpunan frekuensi kata kerja diurutkan berdasarkan frekuensi	32
Tabel 3.12 Himpunan pasangan data dengan kata kerja terpilih	33
Tabel 3.13 Perhitungan skor total setiap kata kerja metode MFLR	33
Tabel 3.14 Contoh daftar <i>stopwords</i> nama relasi taksonomis	33
Tabel 3.15 Kategori penentuan akurasi hasil prediksi/klasifikasi	34
Tabel 3.16 Contoh himpunan pasangan data sampel dari hasil <i>random sampling</i>	38
Tabel 3.17 Himpunan kata kerja terpilih untuk evaluasi masing-masing metode.....	38
Tabel 3.18 Hasil pencarian setiap data sampel untuk setiap kata kerja	39
Tabel 3.19 Hasil penjumlahan jumlah <i>hit</i> setiap mesin pencari	39
Tabel 3.20 Hasil akhir penjumlahan jumlah <i>hit</i> total setiap kata kerja.....	39
Tabel 5.1 Parameter pembelajaran yang diujikan.....	62
Tabel 5.2 Pengujian 1.a - Data [herbal_penyakit] pada Google	63
Tabel 5.3 Pengujian 1.b - Data [tempat_wisata] pada Google	64
Tabel 5.4 Pengujian 2.a - Data [herbal_penyakit] pada Yahoo	64
Tabel 5.5 Pengujian 2.b - Data [tempat_wisata] pada Yahoo.....	65
Tabel 5.6 Pengujian 3.a - Data [herbal_penyakit] pada Ecosia	65
Tabel 5.7 Pengujian 3.b - Data [tempat_wisata] pada Ecosia.....	66

Tabel 5.8 Pengujian 4.a - Data [herbal_penyakit] pada semua mesin pencari	66
Tabel 5.9 Pengujian 4.b - Data [tempat_wisata] pada semua mesin pencari.....	67
Tabel 5.10 Ringkasan Hasil Pengujian LNTRO	67
Tabel 5.11 Perbandingan alur kerja antara menggunakan dengan tidak menggunakan titik...	73
Tabel 5.12 Perbandingan statistik penggunaan titik pada ID pengujian (4.a)	73
Tabel 5.13 Perbandingan ekstraksi kata penggunaan titik pada ID pengujian (4.a).....	74
Tabel 5.14 Perbandingan statistik penggunaan titik pada ID pengujian (4.b)	74
Tabel 5.15 Perbandingan ekstraksi kata penggunaan titik pada ID pengujian (4.b).....	75
Tabel 5.16 Perbandingan alur kerja menggunakan <i>stopword</i> kata kerja bantu	76
Tabel 5.17 Perbandingan statistik penggunaan <i>stopword</i> pada ID pengujian (4.a).....	77
Tabel 5.18 Perbandingan ekstraksi kata penggunaan <i>stopword</i> pada ID pengujian (4.a)	77
Tabel 5.19 Perbandingan statistik penggunaan <i>stopword</i> pada ID pengujian (4.b).....	78
Tabel 5.20 Perbandingan ekstraksi kata penggunaan <i>stopword</i> pada ID pengujian (4.b)	78
Tabel 5.21 Perbandingan alur kerja menggabungkan frekuensi berdasarkan kata dasar	81
Tabel 5.22 Perbandingan statistik penggabungan frekuensi pada ID pengujian (4.a).....	81
Tabel 5.23 Perbandingan ekstraksi kata penggabungan frekuensi pada ID pengujian (4.a) ...	82
Tabel 5.24 Perbandingan statistik penggabungan frekuensi pada ID pengujian (4.b).....	82
Tabel 5.25 Perbandingan ekstraksi kata penggabungan frekuensi pada ID pengujian (4.b) ...	83

DAFTAR GAMBAR

Gambar 2.1 Metodologi pembelajaran relasi non-taksonomi berbahasa Inggris	13
Gambar 3.1 Alur kerja pembelajaran relasi non-taksonomi	16
Gambar 3.2 Formula matematis sederhana pembangunan korpus	17
Gambar 3.3 Ekstraksi kalimat dari setiap butir hasil pencarian.....	18
Gambar 3.4 Halaman awal mesin pencari Google.....	19
Gambar 3.5 Halaman awal mesin pencari Bing.....	19
Gambar 3.6 Halaman awal mesin pencari DuckDuckGo	20
Gambar 3.7 Halaman awal mesin pencari Yahoo.....	20
Gambar 3.8 Halaman awal mesin pencari Ecosia.....	21
Gambar 3.9 Halaman awal mesin pencari Dogpile	21
Gambar 3.10 Alur <i>preprocessing</i> (berbentuk <i>pipeline</i>)	24
Gambar 3.11 Contoh pemangkasan setiap kalimat dalam korpus	25
Gambar 3.12 Formula matematis sederhana anotasi korpus	26
Gambar 3.13 Contoh proses <i>POS Tagging</i> oleh <i>POS Tagger</i>	27
Gambar 3.14 Alur seleksi kata.....	28
Gambar 3.15 Formula matematis sederhana ekstraksi kata.....	29
Gambar 3.16 <i>Regular expression</i> untuk <i>stemming</i> (sintaks menyesuaikan <i>library</i>)	36
Gambar 3.17 Formula matematis sederhana evaluasi kata.....	37
Gambar 4.1 Halaman situs Stanford NLP Group (bagian <i>POS Tagger</i>)	41
Gambar 4.2 Data yang disimpan pada basisdata MongoDB	42
Gambar 4.3 Parameter pelatihan model <i>POS Tagger</i>	44
Gambar 4.4 Perintah pelatihan model <i>POS Tagger</i> dengan <i>library</i> Stanford CoreNLP	45
Gambar 4.5 Keluaran CLI dari pelatihan model <i>POS Tagger</i>	45
Gambar 4.6 Perintah pengujian model <i>POS Tagger</i> dengan <i>library</i> Stanford CoreNLP	46
Gambar 4.7 Keluaran CLI dari pengujian model <i>POS Tagger</i>	46
Gambar 4.8 Halaman situs Kamus Besar Bahasa Indonesia	47
Gambar 4.9 Halaman Home yang merupakan halaman awal program	50
Gambar 4.10 Daftar hasil pembelajaran relasi non-taksonomi terdahulu.....	50
Gambar 4.11 Halaman Form LNTRO (a).....	52
Gambar 4.12 Halaman Form LNTRO (b).....	52
Gambar 4.13 Deskripsi dan parameter hasil pembelajaran relasi non-taksonomi.....	53

Gambar 4.14 Rangkuman keluaran dari hasil pembelajaran relasi non-taksonomi.....	53
Gambar 4.15 Data masukan pembelajaran yang disajikan dalam bentuk tabel.....	54
Gambar 4.16 Hasil pencarian data masukan pada mesin pencari	54
Gambar 4.17 Hasil pembangunan korpus melalui mesin pencari setelah diolah	55
Gambar 4.18 Hasil anotasi korpus oleh POS Tagger beserta informasi tag.....	55
Gambar 4.19 Hasil perhitungan frekuensi kemunculan kata	56
Gambar 4.20 Hasil perhitungan skor akhir menggunakan metode MFLR.....	56
Gambar 4.21 Hasil perhitungan skor akhir menggunakan metode MAANS	57
Gambar 4.22 Hasil pemilihan nama relasi non-taksonomi.....	57
Gambar 4.23 Data sampel yang dipakai pada saat evaluasi	58
Gambar 4.24 Hasil evaluasi kata dengan pencarian melalui mesin pencari	58
Gambar 4.25 Tombol <i>download</i> ontologi dan informasi <i>benchmark</i>	59
Gambar 5.1 Jumlah <i>hit</i> hasil pencarian yang dihasilkan oleh mesin pencari	61
Gambar 5.2 Ontologi data [herbal_penyakit] dengan nama relasi “mengobati”	69
Gambar 5.3 Ontologi data [tempat_wisata] dengan nama relasi “terletak”.....	71

BAB I

PENDAHULUAN

1.1. Latar Belakang

Jaringan Internet telah mengubah cara manusia berkomunikasi satu sama lain. Dengan adanya Internet, komputer-komputer dari berbagai belahan dunia dapat terhubung dan saling berbagi data dan informasi. Setiap orang yang terhubung ke jaringan Internet memiliki kesempatan untuk berbagi informasi. Berbagai halaman web pun tercipta untuk menunjang kegiatan berbagi informasi ini. Alhasil, informasi yang dihasilkan sampai sekarang pun sangat melimpah dan ada dalam berbagai bentuk, seperti teks, gambar, suara, video, dan lain-lain. Hal ini berlaku terutama untuk informasi yang berbentuk teks. Namun, informasi berbentuk teks yang dihasilkan oleh begitu banyak orang ini belum disusun secara terstruktur, sehingga sulit bagi komputer untuk mencari informasi tertentu yang relevan dan saling berkaitan. Data dan informasi yang ada juga memiliki bentuk dan format yang beragam, sehingga sulit untuk diproses. Untuk itulah organisasi World Wide Web Consortium (W3C) membuat suatu gerakan untuk mendorong agar data dan informasi yang ada di Internet menjadi lebih terstruktur secara semantik (Edition dkk., 2008). Penyusunan semantik membuat setiap data dan informasi memiliki arti yang dapat dipahami, baik oleh manusia maupun oleh komputer. Dengan demikian, maka data dan informasi menjadi lebih mudah dikelola dan dimanfaatkan. Data semantik kemudian dapat digunakan untuk mengembangkan sistem yang lebih kompleks, misalnya suatu basis pengetahuan yang menyimpan pengetahuan manusia secara lebih terstruktur. Dalam penggunaannya, data semantik digunakan sebagai sumber data untuk mengembangkan ontologi.

Ontologi adalah spesifikasi eksplisit dan formal dari suatu konseptualisasi bidang keilmuan tertentu yang mendefinisikan kumpulan konsep dan relasinya untuk merepresentasikan pengetahuan pada bidang yang tersebut (Gruber, 1995). Ontologi memiliki peran yang sangat penting dalam sistem ilmu pengetahuan modern karena ontologi memberikan standar formal dalam penyusunan struktur basis pengetahuan serta dalam penggunaan kembali (*reuse*) dan berbagi pakainya (*sharing*) (Alexiev dkk., 2005). Karena peran ontologi yang sangat penting inilah maka banyak usaha yang dilakukan untuk dapat membangun ontologi berbantuan komputer, baik menggunakan metode pembelajaran otomatis

yang tidak melibatkan peran manusia maupun metode pembelajaran semi-otomatis yang melibatkan peran manusia.

Salah satu bagian terpenting dari suatu struktur ontologi adalah kelas dan relasinya. Kelas dari ontologi digunakan untuk mengelompokkan data individu berdasarkan konsep dasarnya. Sedangkan relasi dari ontologi digunakan untuk menghubungkan antara kelas satu dengan kelas lainnya dan memberikan makna dari hubungan tersebut. Terdapat dua jenis relasi ontologi, yaitu relasi taksonomis dan relasi non-taksonomis. Relasi taksonomis sendiri adalah relasi yang menghubungkan kelas-kelas yang bersifat taksonomi, yang secara konsep memiliki hubungan umum ke khusus. Sedangkan relasi non-taksonomis sendiri adalah relasi yang tidak menghubungkan kelas-kelas ke dalam bentuk umum ke khusus atau sebaliknya.

Pembangunan ontologi dimulai dari data individu dalam suatu bidang keilmuan tertentu yang dikelompokkan dalam kelas-kelas, kemudian menentukan nama relasi antara kelas-kelas yang sudah terbentuk untuk memberikan makna hubungan antar kelas. Berkat Web yang memiliki data dan informasi yang sangat melimpah, banyak kelas-kelas dan data individu dalam bidang apapun yang bisa didapatkan. Salah satu tantangan dalam pembangunan ontologi adalah penentuan nama relasi antar kelas dan individu yang sudah terbentuk. Meskipun seorang ahli dalam bidang keilmuan yang bersangkutan dapat dengan mudah menentukan nama relasi berdasarkan data dan kelas yang ada, namun pembangunan ontologi yang melibatkan manusia masuk ke dalam kategori sistem pembelajaran semi-otomatis dan tidak ideal untuk diterapkan dalam skala besar. Untuk pembangunan ontologi skala besar di mana terdapat sangat banyak data dan informasi, penentuan nama relasi yang ideal dilakukan secara otomatis tanpa keterlibatan manusia. Penentuan nama relasi secara otomatis inilah yang menjadi tantangan dalam pembangunan ontologi secara otomatis. Penentuan nama relasi secara otomatis sulit dilakukan karena relasi menentukan konteks dari kelas-kelas atau konsep-konsep yang saling dihubungkan. Hal ini berlaku terutama untuk relasi yang bersifat non-taksonomis, karena dalam ilmu pengetahuan lebih banyak konsep yang terhubung secara non-taksonomis, di mana hubungan yang ada bukanlah hubungan dari umum ke khusus atau sebaliknya.

Salah satu alternatif solusi untuk memecahkan masalah ini adalah dengan memanfaatkan data dan informasi berlimpah dari Web untuk menemukan kemungkinan nama relasi yang sesuai. Dengan memanfaatkan pengolahan bahasa alami, komputer dapat diprogram untuk mencari kemungkinan nama relasi yang sesuai dari kelas-kelas yang diberikan. Namun bahasa yang digunakan sangat mempengaruhi masukan dan keluaran dari program. Meskipun program komputer yang dapat melakukan hal ini sudah banyak dikembangkan (Sánchez & Moreno,

2008), namun program-program tersebut tidak dikembangkan untuk memproses data yang menggunakan Bahasa Indonesia. Selain itu, proses pengolahan bahasa alami yang dijalankan juga sangat dipengaruhi oleh bahasa yang diproses, sehingga bahasa yang berbeda pun perlu pendekatan pengolahan yang berbeda pula. Oleh karena itulah penelitian ini dilakukan, yaitu untuk menentukan nama relasi yang mungkin dari data kelas ontologi yang menggunakan Bahasa Indonesia secara otomatis. Kemungkinan nama relasi didapatkan dari dokumen Web yang jumlahnya melimpah, yang kemudian diproses secara otomatis untuk menentukan satu nama relasi yang dapat menghubungkan kelas-kelas yang diberikan.

1.2. Rumusan Masalah

Bagaimana menentukan nama relasi non-taksonomis yang tepat untuk dua data kelas ontologi dari dokumen dan data dari Web?

1.3. Batasan Masalah

Adapun batasan masalah dalam penelitian ini antara lain:

- a. Metode yang dikembangkan hanya dapat menentukan relasi non-taksonomis saja dan tidak menentukan relasi taksonomis.
- b. Bahasa yang diproses adalah Bahasa Indonesia. Oleh karena itu maka data yang digunakan, baik data kelas ontologi maupun data dokumen Web, harus menggunakan Bahasa Indonesia.
- c. Bahasa pemrograman yang digunakan untuk mengimplementasikan metode yang dikembangkan dalam bentuk program komputer adalah bahasa Java.
- d. Program komputer yang dikembangkan dibuat berbasis Web, yang dapat diakses melalui antarmuka halaman Web dan melalui Layanan Web API (Application Programming Interface).
- e. Data masukan harus dalam format file TSV atau CSV dan dalam bentuk list yang terdiri atas pasangan-pasangan data kelas ontologi yang seragam. Misalnya untuk kelas ontologi [Tanaman Herbal] - [Penyakit] maka contoh isi datanya:

[jahe] - [batuk]
 [kencur] - [pilek]
- f. Mesin pencari yang digunakan untuk mencari dokumen Web yang relevan adalah Google, Bing, Yahoo, DuckDuckGo, Ecosia, dan Dogpile. Untuk itu, format keyword disesuaikan dengan format keyword dari masing-masing mesin pencari.

- g. Meskipun proses pembelajaran relasi non-taksonomi yang diusulkan dapat digunakan pada domain mana pun, namun data uji yang digunakan hanya terbatas pada dua domain saja, yaitu domain pengetahuan kesehatan (pasangan data tanaman herbal - penyakit) dan domain pariwisata (pasangan data tempat wisata - lokasi).

1.4. Tujuan Penelitian

Penelitian ini dilakukan untuk menghasilkan metode yang dapat menentukan nama relasi non-taksonomis yang tepat untuk dua data kelas ontologi. Sumber data yang digunakan untuk menerapkan metode yang dikembangkan adalah data teks dari Web. Dengan adanya penelitian ini, maka nama relasi non-taksonomis dapat ditentukan secara otomatis sehingga dapat membantu proses pembangunan ontologi secara otomatis. Dengan demikian, maka pembangunan ontologi dapat dilakukan menggunakan skala data yang besar, karena penentuan nama relasi tidak lagi harus melibatkan ahli dan dapat dilakukan secara otomatis sepenuhnya.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat berupa memudahkan *knowledge engineer* untuk membangun ontologi berbahasa Indonesia secara otomatis berdasarkan data-data kelas/objek yang sudah ada

1.6. Metodologi Penelitian

Metodologi penelitian dirancang agar proses penelitian dapat dilaksanakan sesuai dengan rencana dan mendapatkan hasil yang diharapkan. Adapun metodologi yang diterapkan dalam penelitian ini yaitu:

a. Studi Pustaka

Pada tahap ini dilakukan analisis untuk mengetahui secara rinci permasalahan yang diangkat, yaitu pembelajaran relasi non-taksonomi dari dokumen web. Analisis dilakukan dengan cara melakukan studi pustaka untuk mempelajari solusi-solusi penelitian terdahulu yang menyelesaikan atau menghadapi masalah serupa.

b. Perancangan Alur Kerja

Pada tahap ini dibuat rancangan solusi berupa formula masukan-proses-keluaran. Formula tersebut digambarkan dalam bentuk diagram alur yang menggambarkan tahapan-tahapan logis yang ditempuh untuk mengolah data dari bentuk masukan (data kelas) menjadi bentuk keluaran yang diinginkan (suatu kata yang merepresentasikan relasi non-

taksonomi). Dengan demikian, solusi dapat didefinisikan secara jelas dengan mendefinisikan satu per satu setiap tahapan yang ditempuh.

c. Implementasi Sistem

Pada tahap ini dilakukan implementasi dari rancangan sistem yang telah dibuat pada tahap sebelumnya ke dalam bentuk program yang dapat dijalankan.

d. Pengujian

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dikembangkan. Pengujian diperlukan untuk memeriksa apakah sistem telah berjalan sesuai dengan rancangan solusi yang dibuat. Selain itu, pengujian juga dilakukan untuk melakukan perbaikan apabila terdapat kesalahan (*error*) atau kecacatan program (*bugs*).

1.7. Sistematika Penulisan

Sistematika penulisan ditujukan untuk memetakan pembahasan penelitian ini. Secara garis besar, sistematika penulisan laporan ini terbagi menjadi enam bab. Adapun uraian dari masing-masing bab yaitu:

BAB I Pendahuluan, berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan laporan.

BAB II Landasan Teori, berisi teori-teori dasar dan kajian pustaka yang menjadi landasan penelitian. Teori-teori yang dibahas adalah teori yang berkaitan dengan proses pembelajaran relasi, seperti akuisisi pengetahuan, web semantik, ontologi, dan seterusnya. Penelitian-penelitian sebelumnya yang membahas tentang pembelajaran relasi juga dibahas sebagai bagian dari kajian pustaka.

BAB III Rancangan Alur Kerja, berisi rancangan konsep alur kerja pembelajaran relasi, mulai dari pembangunan korpus (*corpus construction*), anotasi korpus (*corpus annotation*), ekstraksi kata (*term extraction*), hingga evaluasi kata (*term evaluation*).

BAB IV Implementasi Program, berisi rincian implementasi program untuk menerapkan rancangan alur kerja yang sudah disusun beserta antarmuka pengguna yang dibuat.

BAB V Hasil dan Pembahasan, berisi hasil pengujian program dan pembahasannya untuk mengevaluasi kemampuan program dalam melakukan pembelajaran relasi non-taksonomi.

BAB VI Kesimpulan dan Saran, berisi kesimpulan dari hasil dan pembahasan pengujian program serta saran untuk penelitian-penelitian berikutnya yang bertujuan untuk meneliti hal serupa, yaitu pembelajaran relasi non-taksonomi.

BAB II

LANDASAN TEORI

2.1. Akuisisi Pengetahuan

2.1.1. Definisi Akuisisi Pengetahuan

Menurut Potter (2003), akuisisi pengetahuan adalah proses untuk memperoleh informasi (baik dari manusia maupun sumber lain) dan struktur bakunya, yang memungkinkan pekerjaan tertentu dapat dikerjakan oleh sistem komputer. Dalam bidang kecerdasan buatan, informasi terstruktur semacam ini biasa disebut dengan istilah “pengetahuan.”

2.1.2. Metode Akuisisi Pengetahuan

a. Metode Konvensional

Pada metode konvensional, akuisisi pengetahuan dilakukan oleh seorang *knowledge engineer* (insinyur pengetahuan) yang bertugas untuk mengumpulkan pengetahuan dari berbagai sumber. Insinyur pengetahuan sendiri adalah seseorang yang akrab dengan teknik-teknik untuk merepresentasikan pengetahuan dalam komputer, dan dapat menerjemahkan informasi yang ia terima ke dalam bentuk yang tepat (Potter, 2003). Sumber pengetahuan bisa berupa:

1. Wawancara terhadap seorang pakar/ahli pada bidang minat yang diteliti.
2. Analisis protokol, yaitu mempelajari pola pikir pakar dengan cara meminta pakar untuk melakukan suatu pekerjaan sambil menjelaskan proses penalaran dan segala detail dari pekerjaan tersebut.

b. Metode Otomatis

Pada metode otomatis, akuisisi pengetahuan dilakukan oleh komputer secara otomatis. Menurut penelitian yang dilakukan oleh (Potter, 2003) yang meneliti tentang survey metode akuisisi pengetahuan dari bahasa alami, terdapat beberapa pendekatan untuk melakukan akuisisi pengetahuan secara otomatis, di antaranya yaitu:

1. Mengotomaskan metode konvensional yang ada, yaitu wawancara dan analisis protokol, sehingga proses pengumpulan informasinya dilakukan oleh komputer.
2. Pendekatan analisis teks, di mana suatu teks atau dokumen yang berisi pengetahuan dianalisis secara otomatis menggunakan proses pengolahan bahasa alami.

3. Akuisisi berbasis ontologi, di mana pengetahuan dasar didapatkan dari proses penalaran terhadap suatu ontologi dengan bidang minat yang sama, kemudian dijadikan dasar untuk proses akuisisi pengetahuan lebih lanjut sehingga dapat memperkaya pengetahuan dasar yang sudah didapat di awal.
4. Pencocokan pola dan *template*, di mana pola teks tertentu dirumuskan dan dicocokkan berdasarkan pola-pola kata dan tatabahasa yang dinilai sering muncul dalam bidang minat yang diteliti.
5. Akuisisi berbasis memori, yaitu mengintegrasikan “bottom-up” yaitu pemrosesan sintaks teks dan “top-down” yaitu penggunaan ulang dari konsep-konsep yang mirip yang tersimpan dalam memori sistem.
6. Pembelajaran mesin, yaitu memanfaatkan algoritma-algoritma pembelajaran mesin (*machine learning*) untuk mengumpulkan pengetahuan.
7. Pendekatan interaktif, yaitu pendekatan semi-otomatis di mana model konseptual dari bidang minat yang diteliti dibangun secara bertahap.
8. Pendekatan terintegrasi, yaitu pendekatan pembelajaran di mana beberapa pendekatan yang berbeda digunakan dalam suatu kerangka umum.
9. *Data mining* teks, yaitu pendekatan teknik *data mining* pada suatu data teks.

2.2. Web Semantik

Di zaman sekarang ini, terdapat banyak database yang tersedia di Web yang dapat diakses dengan cara tertentu oleh *user* dan program aplikasi yang terbuka dan dinamis. Dalam keadaan demikian, data yang tersedia juga harus dapat dipahami secara semantik. Bagi manusia, hal ini dilakukan dengan cara menggunakan format presentasi data yang tepat. Sedangkan untuk program aplikasi, semantik dari data harus disediakan dalam bentuk dalam bentuk yang formal dan dapat diproses dengan mudah oleh mesin (*machine-processable*). Inilah yang disebut Web Semantik, yaitu usaha untuk merevolusi Web sehingga data yang tersedia tidak hanya dapat dipahami oleh manusia saja, tapi juga dapat dipahami dengan mudah oleh mesin/komputer (Edition dkk., 2008).

Jaringan Internet memiliki data dan pengetahuan dalam jumlah yang sangat besar, namun sebagian besar data tersebut tidak dapat diproses dengan mudah oleh komputer karena berbentuk HTML yang tidak memiliki arti secara semantik. Hal ini disebabkan karena HTML dikembangkan hanya untuk menampilkan data secara estetika saja. Web Semantik merupakan salah satu pendekatan yang dilakukan sebagai bagian dari usaha untuk memajukan teknologi

di bidang kecerdasan buatan. Tidak seperti metode lain dalam kecerdasan buatan di mana algoritma tertentu dikembangkan untuk membuat komputer bisa belajar dari masukan yang diberikan (data), Web Semantik bertujuan untuk memudahkan proses pembelajaran tersebut dengan membuat data dalam Web menjadi lebih mudah dipahami oleh komputer.

2.3. Ontologi

2.3.1. Definisi Ontologi

Berikut merupakan definisi ontologi menurut beberapa ahli:

- a. Menurut Gruber (1995), ontologi adalah “spesifikasi eksplisit dari suatu konseptualisasi.”
- b. Menurut Borst, Akkermans, & Top (1997), ontologi adalah “spesifikasi baku dari suatu konseptualisasi yang diakui bersama (*shared*).” Definisi ini menambahkan bahwa konseptualisasi tersebut harus diterima oleh beberapa pihak, tidak hanya oleh satu pihak saja.
- c. Staab & Studer (1998) menggabungkan definisi dari Gruber (1995) dan Borst dkk. (1997) menjadi “ontologi adalah spesifikasi eksplisit dan baku dari suatu konseptualisasi yang diakui bersama (*shared*).”

Terdapat tiga aspek utama dari definisi ontologi yang dikemukakan oleh Staab & Studer (1998), yaitu:

1. Apa yang dimaksud dengan “konseptualisasi” ?

Gruber (1995) mengutip istilah konseptualisasi dari Genesereth & Nilsson (1987), yang mengatakan “suatu kumpulan dari pengetahuan yang dinyatakan secara formal tersusun berdasarkan suatu konseptualisasi: objek, konsep, dan entitas lain yang diasumsikan berada dalam bidang minat tertentu, dan hubungan yang ada di antaranya. Suatu konseptualisasi adalah suatu persepsi yang abstrak dan disederhanakan terhadap fenomena tertentu yang dinyatakan untuk tujuan tertentu. Setiap basis pengetahuan, sistem berbasis pengetahuan, atau agen dengan tingkat pengetahuan tertentu terikat oleh konseptualisasi tertentu, baik secara eksplisit maupun implisit.”

2. Apa bentuk “spesifikasi eksplisit dan baku” yang tepat?
3. Mengapa “pengakuan bersama” patut diperhitungkan?

2.3.2. Komponen Ontologi

Wicaksana (2004) menjelaskan bahwa komponen dari ontologi dijelaskan dengan menggunakan notasi sebagai berikut:

a. Konsep (*concept*)

Digunakan dalam pemahaman yang luas. Sebuah konsep dapat sesuatu yang dikatakan, sehingga dapat pula merupakan penjelasan dari tugas, fungsi, aksi, strategi, dan sebagainya.

b. Relasi (*relations*)

Merupakan representasi sebuah tipe dari interaksi antara konsep dari sebuah domain. Secara formal dapat didefinisikan sebagai subset dari sebuah produk dari n set, $R: C_1 \times C_2 \times \dots \times C_n$. Sebagai contoh dari relasi binary termasuk *subclass-of* dan *connected-to*.

c. Fungsi (*functions*)

Adalah sebuah relasi khusus di mana elemen ke- n dari relasi adalah unik untuk elemen ke- $n-1$. $F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$, contohnya adalah *Mother-of*.

d. Aksioma (*axioms*)

Digunakan untuk memodelkan sebuah *sentence* yang selalu benar.

e. *Instances*

Digunakan untuk merepresentasikan elemen.

2.3.3. Relasi Taksonomi dan Non-Taksonomi

Hubungan antar konsep/kelas dalam suatu ontologi dinyatakan dalam bentuk relasi. Terdapat dua jenis relasi yang menghubungkan antara satu konsep dengan konsep lainnya, yaitu relasi taksonomi dan relasi non-taksonomi.

a. Relasi Taksonomi

Relasi taksonomi menghubungkan dua konsep yang memiliki hubungan layaknya suatu taksonomi, yaitu hubungan dari konsep umum ke khusus atau hubungan yang menunjukkan hirarki tertentu. Dalam ontologi, relasi taksonomi biasanya dinyatakan dalam istilah yang menunjukkan hirarki, misalnya *subclassOf*.

b. Relasi Non-Taksonomi

Relasi non-taksonomi menghubungkan dua konsep yang tidak berbentuk hirarki. Oleh karena itu, relasi non-taksonomi dapat menyatakan relasi yang kompleks, misalnya *UII-terletakDi-Yogyakarta*.

2.4. Metode Pembelajaran Relasi Non-Taksonomi

Pembelajaran relasi non-taksonomi dalam bidang ontologi bukan hal baru. Sudah terdapat beberapa penelitian yang membahas pendekatan-pendekatan tertentu untuk dapat menentukan nama relasi non-taksonomi dari suatu pasangan data kelas ontologi. Pendekatan pembelajaran menggunakan dokumen web juga bukan hal baru, karena dokumen web merupakan sumber data dengan jumlah yang sangat besar.

Penelitian yang dilakukan oleh Serra dkk. (2012) membahas empat teknik *state-of-the-art* LNTRO. Solusi yang ditawarkan oleh keempat teknik tersebut dibahas dalam penelitian tersebut beserta kelebihan, kekurangan, dan batasan penggunaan teknik tersebut. Uraian ringkas mengenai pembahasan keempat teknik/metode LNTRO oleh Serra dkk. (2012) dijabarkan dalam bagian-bagian berikut.

2.4.1. *LNTRO based on the Extraction of Association Rules*

LNTRO based on the Extraction of Association Rules (LNTRO berdasarkan Ekstraksi Aturan Asosiasi) diambil dari penelitian yang dilakukan oleh (Villaverde dkk., 2009). Metode ini memiliki dua fase, yaitu “Identifikasi kemunculan relasi” dan “*Mining* asosiasi”. Fase “Identifikasi kemunculan relasi” menerima korpus dan himpunan konsep/kelas ontologi, yang kemudian menghasilkan suatu *tuple* dalam bentuk $\langle c_1, v, c_2 \rangle$ di mana c_1 dan c_2 adalah kelas ontologi dan v adalah kata kerja. *POS Tagging* dilakukan untuk mengenali kata kerja. Hanya kalimat yang memenuhi kondisi berikut yang menghasilkan *tuple*: (1) kalimat yang memiliki tepat dua konsep/kelas dan satu kata kerja di antaranya, dan (2) kedua konsep tersebut berjarak maksimal N kata di mana N adalah parameter yang ditentukan oleh ahli. Setelah himpunan *tuple* terbentuk, selanjutnya fase “*Mining* asosiasi” dilakukan dengan menerapkan algoritma yang diberikan untuk memperbaiki hasil pada fase sebelumnya. Ringkasan alur kerja dari solusi yang ditawarkan oleh metode *LNTRO based on the Extraction of Association Rules* dapat diamati pada Tabel 2.1.

Tabel 2.1 Solusi *LNTRO based on the Extraction of Association Rules*

Phase	Adopted solution
Corpus construction	Ad-hoc. A corpus already available in the medical field (Genia) was used in its experiment.
Corpus annotation	POS-tagging
Extraction of relationships	Uses the algorithm already described to extract candidate relationships in the form of tuples $\langle c_1, c_2, v \rangle$
Refinement	Uses a technique known as the "Extraction of Association Rules" to suggested non-taxonomic relationships in the form of rules $\langle c_1 \wedge c_2 \rightarrow v \rangle$

Sumber: Serra, Girardi, & Novais (2012)

2.4.2. *LNTRO based on the Extraction of Generalized Association Rules*

LNTRO based on the Extraction of Generalized Association Rules (LNTRO berdasarkan Ekstraksi Aturan Asosiasi yang Digeneralisasi) diambil dari penelitian yang dilakukan oleh (Maedche & Staab, t.t.). Metode ini hampir sama dengan metode *LNTRO based on the Extraction of Association Rules* yang diusulkan oleh (Villaverde dkk., 2009), hanya saja perbedaannya terletak pada penggunaan algoritma *generalized association rules* untuk mengajukan kemungkinan tingkat hirarki relasi yang paling tepat dan dapat digunakan pada teks berbahasa Jerman. Metode ini memiliki dua fase, yaitu “Pemrosesan teks” dan “Mining asosiasi”. Fase “Pemrosesan teks” bertujuan untuk mengekstrak pasangan konsep yang berhubungan dengan kandidat relasi. Untuk mencapai hal ini, digunakan teknik *title heuristic* dan *sentence heuristic*. Teknik pertama mendefinisikan bahwa sepasang konsep yang berhubungan perlu dibuat untuk setiap konsep dalam teks dengan setiap konsep pada judul teks. Teknik kedua menghasilkan suatu *tuple* untuk setiap pasangan konsep yang muncul dalam kalimat yang sama. Pada fase “Mining asosiasi”, relasi yang berbentuk pasangan konsep dari fase sebelumnya diolah menggunakan algoritma *Mining Generalized Association Rules* untuk mengekstrak relasi non-taksonomi dalam bentuk aturan asosiasi. Ringkasan alur kerja dari solusi yang ditawarkan oleh metode *LNTRO based on the Generalized Extraction of Association Rules* dapat diamati pada Tabel 2.2.

Tabel 2.2 Solusi *LNTRO based on the Generalized Extraction of Association Rules*

Phase	Adopted solution
Corpus construction	Ad-hoc. A corpus already available in the touristic domain (Lonely Planet) was used in its experiment.
Corpus annotation	Uses chunking, stemming and REN.
Extraction of relationships	Uses sentence and title heuristics to extract candidate relationships as concept pairs ($CP = \{(a_{i,1}, a_{i,2}) \mid a_{i,j} \in C\}$).
Refinement	Uses a technique known as mining generalized association rules [ref] to recommend relationships as association rules in the form $c_1 \rightarrow c_2$.

Sumber: Serra, Girardi, & Novais (2012)

2.4.3. *LNTRO based on Queries on Web Search Engines*

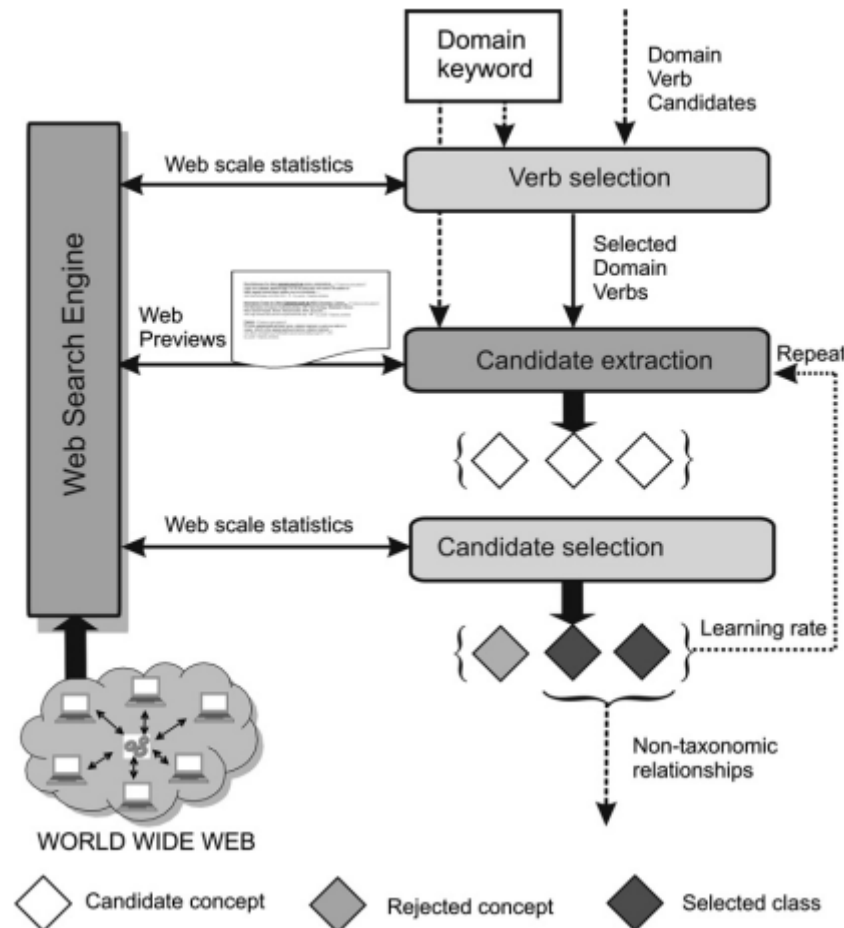
LNTRO based on Queries on Web Search Engines (LNTRO berdasarkan *query* pada mesin pencari Web) diambil dari penelitian yang dilakukan oleh (Sánchez & Moreno, 2008). Dalam penelitian tersebut, pembelajaran relasi non-taksonomi mengandalkan identifikasi kata kerja (*verb*) sebagai kandidat nama relasi yang akan dipilih. Selain itu, karena data yang digunakan adalah data berbahasa Inggris, maka kata kerja yang akan dipilih juga memiliki batasan-batasan tertentu, yaitu:

- Hanya memilih kata kerja yang masuk dalam kategori kalimat *present tense* saja.
- Frasa kata kerja yang melibatkan *modifier* berupa kata keterangan (*adverb*) atau lainnya juga tidak dipilih.
- Frasa kata kerja yang terdiri dari beberapa kata kerja tidak dipilih karena dianggap sulit untuk mengenali kata kerja utama, dengan pengecualian “*to be*” yang sangat umum dalam bahasa Inggris.
- Kata kerja yang merepresentasikan relasi taksonomi (*is/are*, *include*, dan lain-lain) juga tidak dipilih, karena relasi taksonomi dianggap sebagai kasus yang berbeda.

Penelitian tersebut juga mengklasifikasikan kata kerja yang terpilih berdasarkan posisi katanya menjadi dua, yaitu kata kerja aktif dan kata kerja pasif. Kemudian dilakukan penyaringan lebih lanjut dengan hanya memilih kata kerja yang sesuai dan digunakan spesifik dalam *domain* ontologi yang diberikan. Hal ini dilakukan dengan menggunakan analisis statistik secara otomatis untuk menentukan tingkat relevansi antara *domain* dengan kata kerja yang dipilih. Penelitian tersebut juga memanfaatkan *wordnet* untuk menganalisis kata kerja yang dipilih lebih jauh, yang tidak diterapkan dalam penelitian ini karena kurangnya ketersediaan *wordnet* bahasa Indonesia. Metodologi pembelajaran relasi non-taksonomi oleh

Sánchez & Moreno (2008) dapat diamati pada Gambar 2.1 dan ringkasan alur kerja *LNTRO based on Queries on Web Search Engines* pada Tabel 2.3.

D. Sánchez, A. Moreno / Data & Knowledge Engineering 64 (2008) 600–623



Gambar 2.1 Metodologi pembelajaran relasi non-taksonomi berbahasa Inggris

Sumber : Sánchez & Moreno (2008)

Tabel 2.3 Solusi *LNTRO based on Queries on Web Search Engines*

Phase	Adopted solution
Corpus construction	Based on documents returned by a Web search engine.
Corpus annotation	Chunking.
Extraction of relationships	Extracts verb phrases and noun phrases as labels and concepts of relationships respectively.
Refinement	Statistical processing based on the result of queries in a web search engine.

Sumber: Sánchez & Moreno (2008)

2.4.4. *LNTRO based on logistic regression*

LNTRO based on logistic regression (LNTRO berdasarkan regresi logistik) diambil dari penelitian yang dilakukan oleh (Fader dkk., 2011). Metode ini tidak terikat oleh *domain* manapun dan mengekstrak relasi non-taksonomi dari korpus berbahasa Inggris. Metode ini menggunakan pembatas sintaksis dan pembatas leksikal. Pembatas sintaksis mengharuskan setiap frasa kata kerja untuk memenuhi pola berikut: (1) merupakan kata kerja, (2) merupakan kata kerja yang langsung diikuti oleh kata depan/preposisi, atau (3) merupakan kata kerja yang diikuti oleh kata benda, kata sifat, atau kata keterangan yang diakhiri dengan kata depan/preposisi. Pembatas sintaksis digunakan untuk mengurangi hasil ekstraksi yang tidak informatif atau tidak relevan. Namun karena pembatas sintaksis memungkinkan terpilihnya frasa yang terlalu spesifik, maka pembatas leksikal digunakan untuk membedakan antara kalimat yang memiliki relasi nyata dengan kalimat yang frasa relasinya terlalu spesifik. Metode ini memiliki tiga fase, yaitu “ekstraksi relasi”, “ekstraksi argumen”, dan “perbaikan/refinement”. Fase “ekstraksi relasi” bertujuan untuk mengekstrak relasi dengan melibatkan pembatas sintaksis dan leksikal. Fase “ekstraksi argumen” bertujuan untuk memilih relasi argumen-argumen yang berbeda (sehingga relasi dianggap cukup umum digunakan). Kemudian fase “perbaikan/refinement” menggunakan *logistic regression classifier* dan bertujuan untuk mengurutkan relasi berdasarkan probabilitas untuk merepresentasikan relasi valid. Ringkasan alur kerja dari solusi yang ditawarkan oleh metode *LNTRO based on logistic regression* dapat diamati pada Tabel 2.4.

Tabel 2.4 Solusi *LNTRO based on logistic regression*

Phase	Adopted solution
Corpus construction	Ad-hoc construction.
Corpus annotation	Chunking.
Extraction of relationships	Uses the algorithms described in the phases "Relations extraction" and "Extraction of arguments".
Refinement	Uses a logistic regression classifier to assign a probability to each relationship.

Sumber: Sánchez & Moreno (2008)

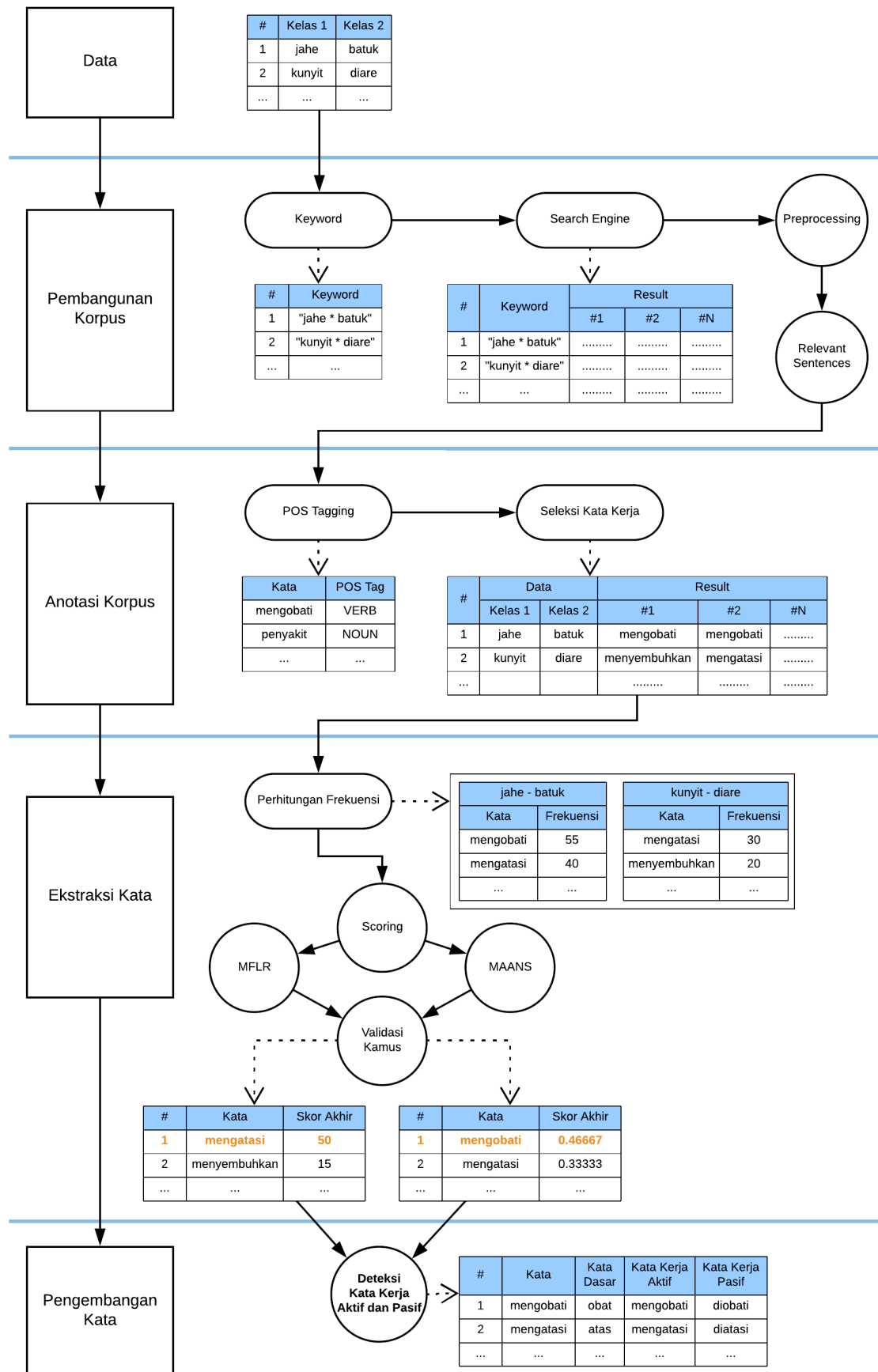
BAB III

RANCANGAN ALUR KERJA

3.1. Alur Kerja Pembelajaran Relasi Non-Taksonomi

Pembelajaran relasi non-taksonomi ditujukan untuk menemukan nama relasi non-taksonomi yang tepat antara suatu kelas dengan kelas lainnya. Dengan demikian, maka hasil akhir dari pembelajaran relasi non-taksonomi ini adalah sebuah kata yang dapat merepresentasikan relasi tersebut dengan tepat. Masukan (*input*) dari sistem pembelajaran ini adalah sejumlah pasangan data yang merupakan *instance* dari kedua kelas yang ingin ditemukan relasinya. Melalui jaringan Internet dan mesin pencari, sistem akan mencoba untuk menemukan calon kata yang cocok dijadikan sebagai relasi dari kedua kelas tersebut. Hasil pencarian dari mesin pencari kemudian diolah melalui beberapa proses untuk menyaring kata yang memiliki kemungkinan terbesar merupakan relasi yang dicari. Setelah seluruh proses penentuan nama relasi telah dilakukan, beberapa kata yang memiliki skor tertinggi dilakukan evaluasi dengan cara melakukan pencarian ulang untuk setiap kata tersebut.

Secara garis besar, tahapan alur kerja pembelajaran relasi non-taksonomi yang dirancang ada empat tahapan utama dan satu tahapan tambahan. Empat tahapan utama meliputi: pengembangan korpus (*corpus construction*), anotasi korpus (*corpus annotation*), ekstraksi kata (*term extraction*), dan pengembangan kata (*term refinement*). Sedangkan tahapan tambahannya adalah evaluasi kata (*term evaluation*). Tahap pengembangan korpus bertujuan untuk membangun suatu korpus yang akan dijadikan sebagai sumber data utama dalam menentukan kandidat nama relasi non-taksonomi. Tahap anotasi korpus bertujuan untuk mengolah dan menyaring korpus sehingga didapatkan kandidat nama relasi non-taksonomi. Tahap ekstraksi korpus bertujuan untuk melakukan penilaian (*scoring*) pada setiap kandidat nama relasi non-taksonomi untuk menentukan kandidat mana yang paling mungkin dipilih sebagai nama relasi non-taksonomi. Tahap pengembangan kata bertujuan untuk mengolah nama relasi non-taksonomi yang sudah dipilih sehingga lebih informatif (dalam kasus penelitian ini, nama relasi dicari bentuk kata kerja aktif dan pasifnya). Yang terakhir, tahap tambahan evaluasi kata bertujuan untuk membantu peneliti melakukan pengujian alur kerja yang diusulkan. Pemetaan alur kerja pembelajaran relasi non-taksonomi dapat diamati pada Gambar 3.1.

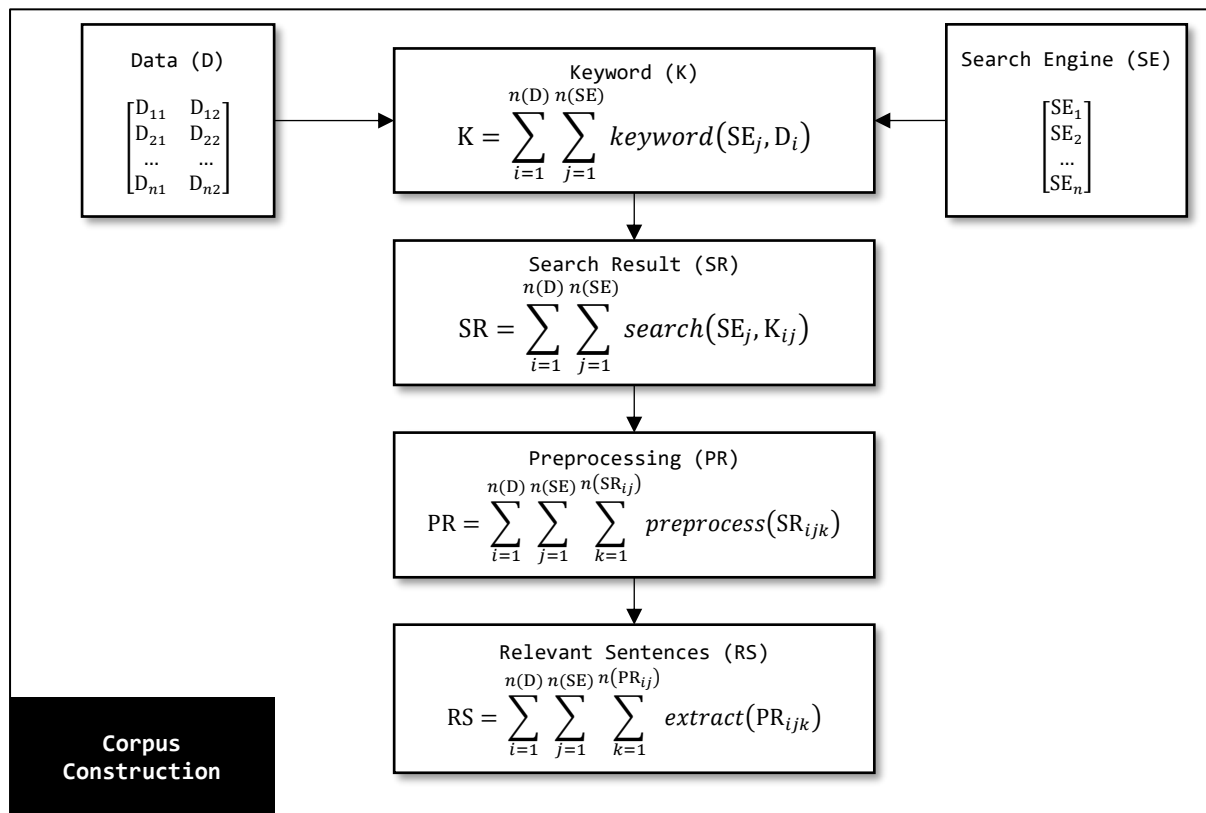


Gambar 3.1 Alur kerja pembelajaran relasi non-taksonomi

3.2. Pembangunan Korpus (*Corpus Construction*)

Dalam konteks pengolahan bahasa alami, korpus adalah suatu kumpulan teks dalam jumlah besar. Dalam penelitian ini, korpus yang dimaksud adalah kumpulan teks yang berbentuk kalimat-kalimat yang dijadikan referensi untuk mencari nama relasi.

Tahap pertama pembelajaran relasi non-taksonomi dimulai dari pembangunan korpus. Secara garis besar, pembangunan korpus dilakukan dengan cara melakukan pencarian untuk setiap pasangan data melalui mesin pencari Internet (*search engine*). Setiap hasil pencarian yang ditemukan akan mengandung judul dokumen dan ringkasan isi dokumen (*snippet*) yang diasumsikan mengandung kalimat yang relevan dengan pasangan data yang dicari. Oleh karena itu, sejumlah hasil pencarian tersebut disimpan dalam koleksi korpus sebagai referensi untuk mencari nama relasi non-taksonomi. Hasil pencarian dari setiap pasangan data membentuk korpus masing-masing untuk kemudian diproses pada tahap selanjutnya. Hasil dari tahap pembangunan korpus adalah himpunan yang berisi korpus untuk setiap pasangan data, di mana setiap korpus berisi kumpulan hasil pencarian dalam bentuk kalimat teks. Secara umum, proses pembangunan korpus dapat diamati pada Gambar 3.2.

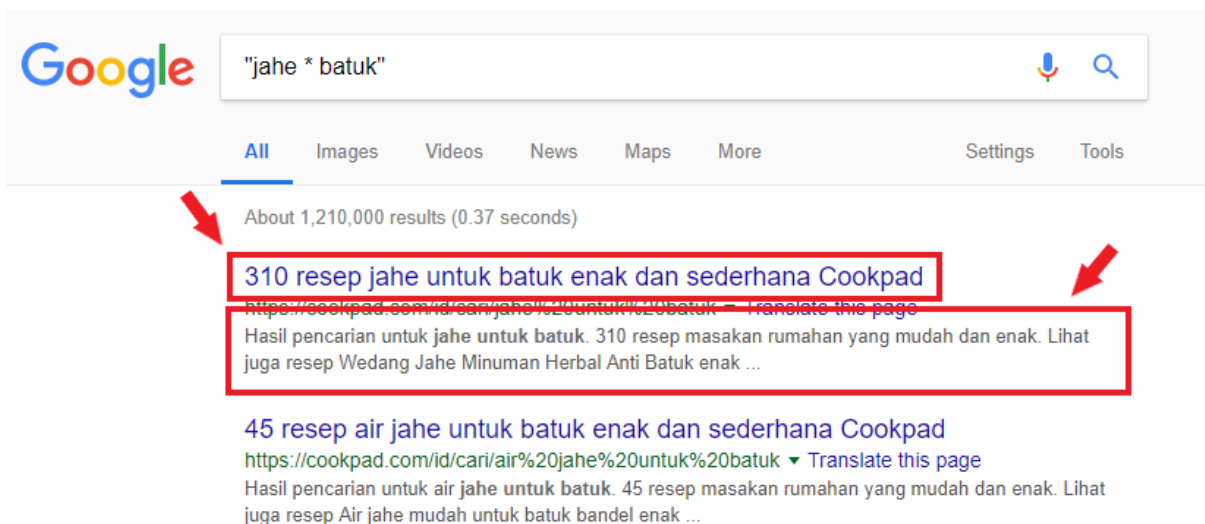


Gambar 3.2 Formula matematis sederhana pembangunan korpus

3.2.1. Mesin Pencari Web (Web Search Engine)

Mesin pencari adalah suatu layanan Internet yang digunakan untuk mencari dokumen di Internet. Dalam penelitian ini, mesin pencari digunakan sebagai alat utama untuk membangun korpus. Cara kerja mesin pencari cukup sederhana, yaitu memasukkan *query*/pertanyaan berupa kata kunci (*keyword*) yang ingin ditemukan. Mesin pencari akan memberikan hasil pencarian berupa daftar dokumen, yang terdiri dari judul, pranala (*link*) atau alamat dokumen, dan ringkasan isi dokumen yang dianggap relevan dengan kata kunci yang diberikan.

Dalam penelitian ini, bagian dari hasil pencarian yang digunakan sebagai sumber korpus adalah judul dokumen dan ringkasan dokumen yang ada dalam setiap butir hasil pencarian. Untuk itu dilakukan ekstraksi judul halaman dan ringkasan yang diberikan oleh mesin pencari, seperti terlihat pada Gambar 3.3.



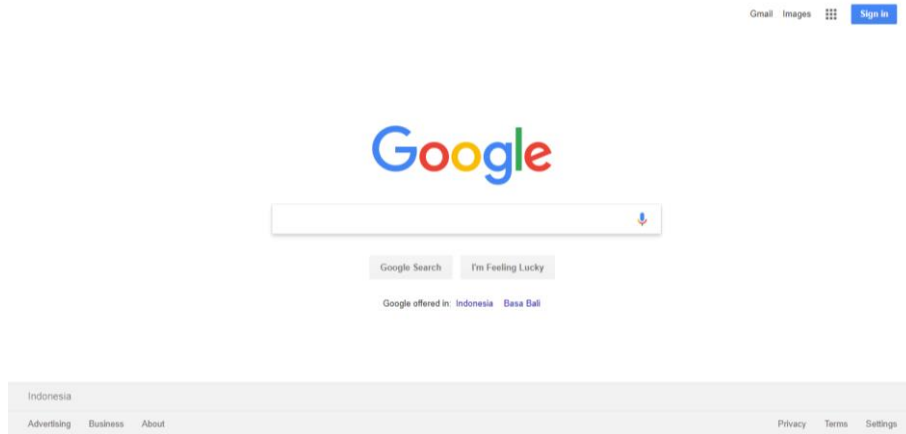
Gambar 3.3 Ekstraksi kalimat dari setiap butir hasil pencarian

Dalam penelitian ini, terdapat enam mesin pencari yang digunakan untuk membangun korpus. Dengan menggunakan beberapa mesin pencari, diharapkan hasil pencarian yang didapatkan lebih bervariasi dan mengurangi kemungkinan terjadinya *break down* sistem apabila proses pencarian pada mesin pencari tertentu tidak sesuai harapan. Berikut adalah mesin pencari yang digunakan dalam penelitian ini:

- Google (<https://www.google.com>)

Google merupakan mesin pencari terbesar di dunia dan memiliki *market share* kurang lebih 90% dari seluruh penggunaan mesin pencari di dunia. Hasil pencarian yang didapatkan dari mesin pencari Google dianggap dapat diandalkan karena Google memiliki

algoritma yang canggih dan dapat memahami *query* yang diberikan dengan cukup baik. Halaman awal Google seperti terlihat pada Gambar 3.4.

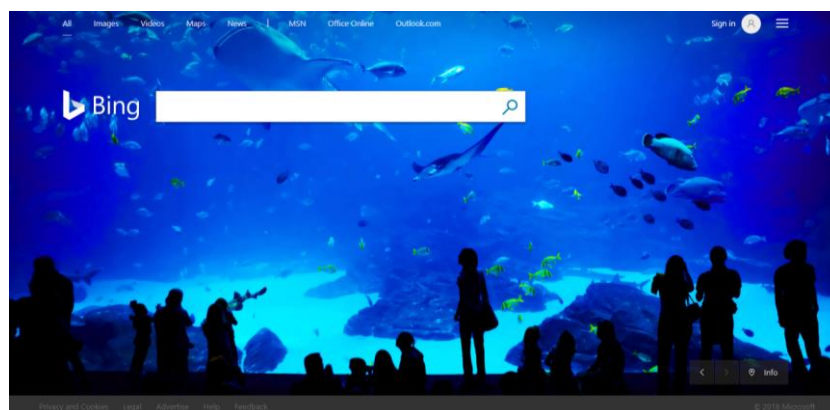


Gambar 3.4 Halaman awal mesin pencari Google

Sumber: <https://www.google.com>

b. Bing (<https://www.bing.com>)

Bing merupakan mesin pencari yang ditawarkan oleh Microsoft dan memiliki *market share* kurang lebih 2% dari seluruh penggunaan mesin pencari di dunia. Karena Bing merupakan mesin pencari yang ditawarkan oleh Microsoft, salah satu perusahaan raksasa di bidang teknologi, maka peneliti menganggap mesin pencari ini memiliki prospek yang cukup bagus untuk digunakan sebagai salah satu mesin pencari korpus. Halaman awal Bing seperti terlihat pada Gambar 3.5.

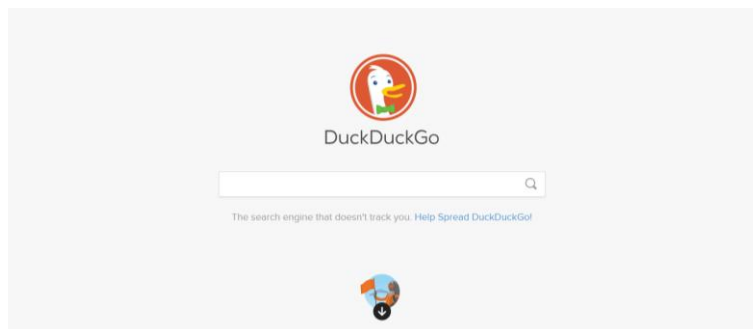


Gambar 3.5 Halaman awal mesin pencari Bing

Sumber: <https://www.bing.com>

c. DuckDuckGo (<https://duckduckgo.com>)

DuckDuckGo merupakan mesin pencari yang mengutamakan privasi pengguna sebagai fitur utama yang ditawarkan. Meskipun *market share* dari DuckDuckGo tidak besar, namun peneliti menganggap hasil pencarian yang diberikan oleh mesin pencari ini cukup relevan untuk digunakan sebagai salah satu mesin pencari korpus. Halaman awal DuckDuckGo seperti terlihat pada Gambar 3.6.

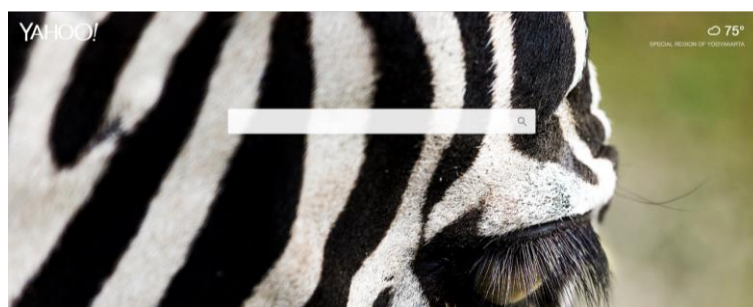


Gambar 3.6 Halaman awal mesin pencari DuckDuckGo

Sumber: <https://duckduckgo.com>

d. Yahoo (<https://search.yahoo.com>)

Yahoo merupakan mesin pencari yang sudah ada cukup lama dan memiliki *market share* kurang lebih 2% dari seluruh penggunaan mesin pencari di dunia. Yahoo merupakan salah satu mesin pencari yang paling banyak digunakan sebagai alternatif dari Google. Maka dari itu peneliti menganggap mesin pencari ini cukup relevan untuk digunakan sebagai salah satu mesin pencari korpus. Halaman awal Yahoo seperti terlihat pada Gambar 3.7.

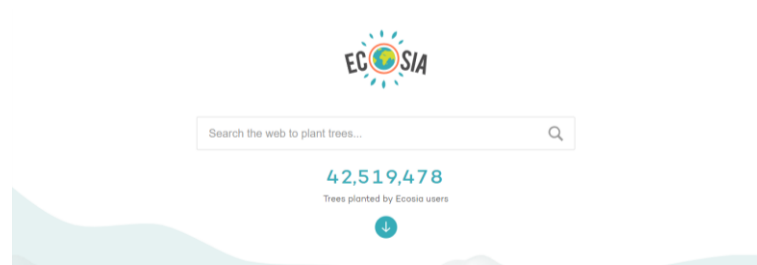


Gambar 3.7 Halaman awal mesin pencari Yahoo

Sumber: <https://search.yahoo.com>

e. Ecosia (<https://www.ecosia.org>)

Ecosia merupakan mesin pencari yang unik. Tim Ecosia menyatakan bahwa untuk setiap pencarian yang dilakukan melalui Ecosia, hampir seluruh keuntungan yang didapat oleh tim Ecosia dari iklan-iklan pada pencarian tersebut akan digunakan untuk menanam pohon di seluruh penjuru dunia. Data pencarian yang digunakan oleh Ecosia didapatkan dari Bing. Meskipun sumber datanya didapatkan dari Bing, namun mesin pencarinya berbeda dan menghasilkan hasil pencarian yang berbeda pula. Untuk itu peneliti menganggap mesin pencari ini cukup relevan untuk digunakan sebagai salah satu mesin pencari korpus. Halaman awal Ecosia seperti terlihat pada Gambar 3.8.

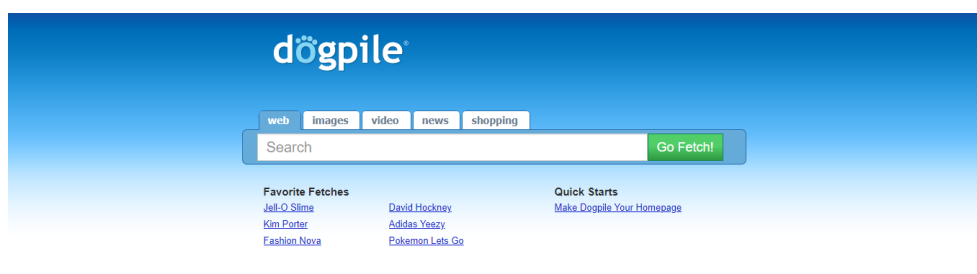


Gambar 3.8 Halaman awal mesin pencari Ecosia

Sumber: <https://www.ecosia.org>

f. Dogpile (<https://www.dogpile.com>)

Dogpile merupakan mesin pencari yang tergolong dalam kategori *meta search engine*. Sebagai salah satu mesin pencari alternatif yang cara kerjanya cukup berbeda dengan mesin pencari lain yang digunakan dalam penelitian ini, peneliti menganggap mesin pencari ini cukup relevan digunakan sebagai salah satu mesin pencari korpus dan juga sebagai sampel penggunaan *meta search engine* dalam penelitian. Halaman awal Dogpile seperti terlihat pada Gambar 3.9.



Gambar 3.9 Halaman awal mesin pencari Dogpile

Sumber: <https://www.dogpile.com>

3.2.2. Pembangunan Kata Kunci (*Keyword*)

Mesin pencari umumnya melakukan pencarian dengan masukan suatu kata kunci (*keyword*) yang berbentuk kalimat dengan sintaks tertentu. Oleh karena itu, pencarian dimulai dengan mengubah setiap pasangan data ke dalam bentuk kata kunci (*keyword*) yang dapat dipahami oleh mesin pencari yang digunakan seperti terlihat pada Tabel 3.1. Perlu dipahami bahwa bentuk kata kunci dari pasangan data yang sama mungkin berbeda untuk setiap mesin pencari yang berbeda.

Tabel 3.1 Pengubahan data ke dalam bentuk kata kunci pada mesin pencari

Pasangan Data		Kata Kunci (Google)
jahe	batuk	"jahe * batuk"
temulawak	osteorthritis	"temulawak * osteorthritis"
kunyit	diare	"kunyit * diare"
...	...	...

3.2.3. Proses Pencarian pada Mesin Pencari

Untuk setiap pasangan data, kata kunci yang dihasilkan kemudian diberikan pada mesin pencari untuk mendapatkan hasil pencarian. Untuk setiap butir hasil pencarian, dilakukan ekstraksi judul halaman dan ringkasan yang diberikan oleh mesin pencari, seperti terlihat pada Gambar 3.3. Dari proses ekstraksi ini, setiap pasangan data menghasilkan sekumpulan kalimat yang bersama-sama membentuk korpus untuk pasangan data tersebut. Contoh hasil pencarian setiap pasangan data dapat diamati pada Tabel 3.2.

Tabel 3.2 Korpus awal hasil pencarian melalui mesin pencari

Pasangan Data		Korpus (Google)			
		Kalimat 1	Kalimat 2	Kalimat 3	...
jahe	batuk	Berdasarkan penelitian yang dilakukan oleh...	Menurut Dr. Ani, jahe dapat mengobati...	Jahe merupakan salah satu tanaman herbal...	...
temulawak	osteorthritis	Banyak orang yang telah membuktikan...	Berikut adalah 5 obat tradisional yang digunakan...	Berikut adalah manfaat temulawak...	...
kunyit	diare	Apakah anda tahu bahwa kunyit dapat anda...	Kunyit telah terbukti dapat mengobati...	Manfaat kunyit dapat meredakan diare...	...

Supaya nantinya setiap pencarian dengan *keyword* yang sama dapat dilakukan dengan lebih efisien, maka setiap korpus yang dimiliki disimpan dalam suatu basis data khusus. Basis

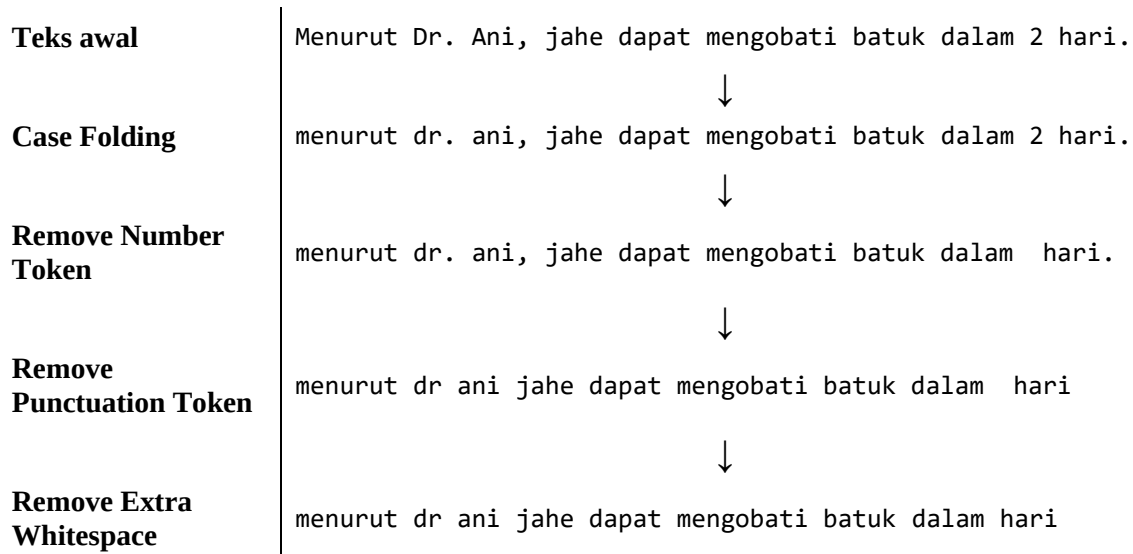
data khusus ini ditujukan sebagai *cache* untuk mengurangi *overhead* dari proses pencarian berulang-ulang dengan *keyword* yang sama.

3.2.4. *Preprocessing*

Setelah setiap pasangan data menghasilkan korpus, selanjutnya dilakukan *preprocessing*, yaitu proses pengolahan awal terhadap suatu teks untuk menghilangkan elemen-elemen yang dianggap tidak penting dan tidak sesuai konteks. Dalam tahap pembangunan korpus ini, *preprocessing* yang dilakukan yaitu:

- a. *Case Folding*, yaitu menyeragamkan huruf dari setiap kata dalam korpus dengan mengubah setiap huruf menjadi bentuk huruf kecilnya (*lowercase*). Hal ini dilakukan untuk menyeragamkan bentuk dari kata dengan makna yang sama, karena komputer menganggap kata yang sama dengan bentuk huruf yang berbeda sebagai kata yang berbeda (misal: batuk dengan BATUK adalah dua kata yang berbeda dari sudut pandang komputer, meskipun seharusnya maknanya sama).
- b. *Remove Number Token*, yaitu menghilangkan angka dari korpus. Hal ini dilakukan karena angka dianggap tidak relevan dalam proses pencarian nama relasi di mana nama relasi berbentuk kata dalam huruf dan tidak mengandung angka.
- c. *Remove Punctuation Token*, yaitu menghilangkan tanda baca dari korpus. Sama halnya dengan menghilangkan angka, tanda baca dianggap tidak penting dan relevan dalam proses pencarian nama relasi.
- d. *Remove Extra Whitespace*, yaitu menghilangkan spasi berlebih dalam korpus. Hal ini dilakukan karena setiap kata dipisahkan dengan karakter spasi, dan spasi yang berlebih berpotensi menimbulkan efek samping yang tidak perlu, seperti misalnya menandai kata yang tidak ada karakternya.

Bila digambarkan, alur *preprocessing* akan berbentuk seperti *pipeline* sebagaimana terlihat pada Gambar 3.10.



Gambar 3.10 Alur *preprocessing* (berbentuk *pipeline*)

3.2.5. Ekstraksi Kalimat Relevan dalam Korpus

Setelah proses *preprocessing* selesai, maka proses selanjutnya adalah melakukan ekstraksi kalimat yang dianggap relevan dalam korpus, dengan cara hanya memilih kata-kata yang letaknya berada di antara kedua pasangan data yang bersangkutan (data pertama dianggap sebagai subjek kalimat dan data kedua dianggap sebagai objek kalimat). Aturan ekstraksi kalimat relevan adalah sebagai berikut:

- a. Pilih rangkaian kata yang letak/posisinya berada di antara kedua pasangan data yang bersangkutan.

Misal untuk pasangan data [jahe] dengan [batuk], maka kalimat:

“sejak jaman dahulu, jahe dapat mengobati penyakit batuk dengan efektif”

Diekstrak menjadi:

“**jahe dapat mengobati penyakit batuk**”

Karena sejak awal asumsinya adalah bahwa hanya kata yang menghubungkan kedua pasangan data saja yang relevan dan memiliki kemungkinan untuk menjadi nama relasi. Letak kata yang potensial harus berada di antara kemunculan kedua pasangan data, dan kata-kata pada posisi lainnya tidak diperlukan.

- b. Apabila dalam satu kalimat terdapat lebih dari satu rangkaian kata relevan yang terdeteksi, maka akan dihasilkan kalimat baru untuk setiap rangkaian kata relevan tersebut.

Misalnya untuk pasangan data [jahe] dengan [batuk], maka kalimat:

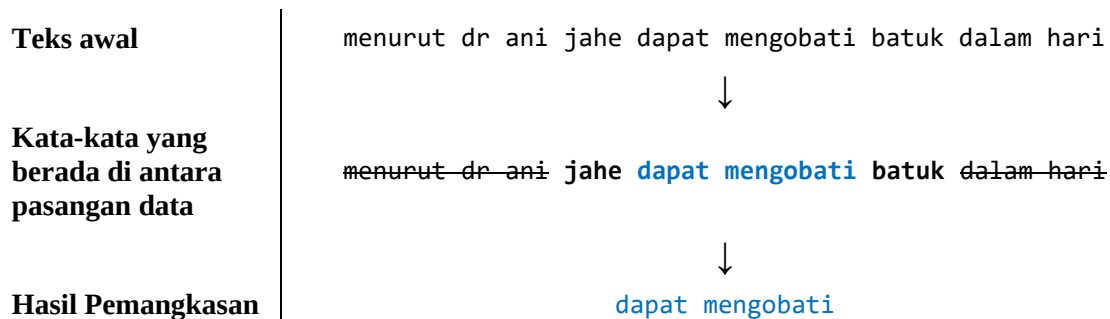
“meskipun mengkonsumsi jahe belum tentu bisa mengobati batuk, namun setidaknya mengkonsumsi jahe dapat mengurangi gejala batuk”

Diekstrak menjadi:

“jahe belum tentu bisa mengobati batuk”

“jahe dapat mengurangi gejala batuk”

Melalui ekstraksi ini, suatu kalimat yang memiliki lebih dari satu rangkaian frasa yang relevan akan menghasilkan kalimat masing-masing untuk setiap rangkaian frasa. Dengan demikian, maka korpus dapat menjadi lebih ringkas tanpa kehilangan esensi yang diperlukan untuk menemukan nama relasi yang tepat. Selain itu, untuk memudahkan proses *POS Tagging* pada tahap selanjutnya, maka teks pasangan data pada korpus juga dihilangkan. Proses pemangkasan kalimat dapat diamati pada Gambar 3.11.



Gambar 3.11 Contoh pemangkasan setiap kalimat dalam korpus

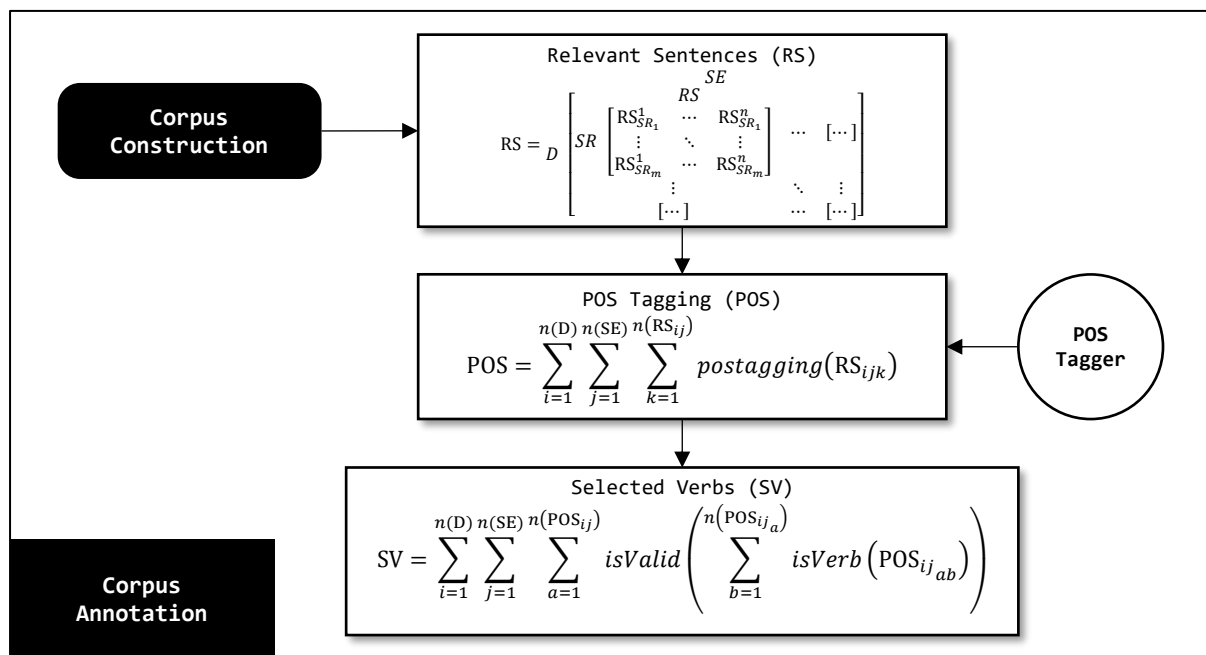
Hasil keseluruhan dari tahap pembangunan korpus ini adalah suatu korpus yang sudah diolah sehingga tidak menimbulkan ambiguitas dan hanya berisi kalimat-kalimat yang ringkas dan tepat sasaran, seperti terlihat pada Tabel 3.3.

Tabel 3.3 Korpus yang sudah diolah melalui tahapan *preprocessing* dan pemangkasan

Pasangan Data		Korpus (Google)			
		Kalimat 1	Kalimat 2	Kalimat 3	...
jahe	batuk	jahe banyak digunakan untuk mengobati batuk	jahe dapat mengobati penyakit batuk	...	...
temulawak	osteorthritis	temulawak bisa mengatasi osteorthritis	temulawak terbukti ampuh menyembuhkan osteorthritis	...	...
kunyit	diare	kunyit dapat anda gunakan untuk meredakan diare	kunyit telah terbukti dapat mengobati diare	...	...
...	...	...	...	...	...

3.3. Anotasi Korpus (*Corpus Annotation*)

Tahap kedua pembelajaran relasi non-taksonomi yaitu anotasi korpus. Pada penelitian ini, kata yang memiliki kemungkinan merupakan nama relasi yang dicari adalah kata-kata yang menghubungkan pasangan data kelas dalam sebuah kalimat. Dalam suatu ontologi, nama kelas biasanya berbentuk kata benda (*noun*) dan nama relasi biasanya berbentuk kata kerja (*verb*). Oleh karena itu, dalam penelitian ini diasumsikan bahwa nama relasi yang dicari adalah kata yang merupakan kata kerja. Penyaringan kata dilakukan dengan melihat bentuk dari setiap kata yang ada dalam korpus dan menyeleksi hanya kata-kata yang berbentuk kata kerja dan letak katanya berada di antara kedua pasangan data. Sistem harus dapat menentukan bentuk dari setiap kata dalam korpus, untuk itulah dilakukan anotasi korpus. Anotasi korpus adalah proses untuk menandai setiap kata dalam korpus dengan bentuk atau kelas kata yang tepat. Hasil dari tahap anotasi korpus adalah korpus yang sama dengan tahap sebelumnya, tetapi hanya berisi kata kerja saja. Alur kerja anotasi korpus dapat diamati pada Gambar 3.12.



Gambar 3.12 Formula matematis sederhana anotasi korpus

3.3.1. *Part-of-Speech Tagging*

Dalam bidang pengolahan bahasa alami, menandai setiap kata dengan kelas kata yang tepat disebut sebagai *Part-of-Speech (POS) Tagging*. *POS Tagging* adalah proses dalam pengolahan bahasa alami yang digunakan untuk mengenali jenis kata seperti kata benda, kata kerja, kata sifat, kata depan, kata keterangan, dan lain-lain.

Dalam penelitian ini, *POS Tagging* dilakukan dengan menggunakan *library POS Tagger* dari Stanford (<https://nlp.stanford.edu/software/tagger.shtml>) yang merupakan implementasi dari konsep *log-linear POS Tagger* oleh Toutanova dkk. (2003). Dengan model yang tepat, *POS Tagger* tersebut dapat mengenali jenis kata dalam bahasa yang diinginkan. Model yang baik dapat dihasilkan dengan cara melakukan pelatihan (*model training*) *POS Tagger* menggunakan korpus di mana setiap kata telah ditandai dengan jenis kata yang sesuai (*POS Tag*) secara manual.

Model *POS Tagger* untuk bahasa Indonesia tidak disediakan oleh Stanford, oleh karena itu dalam penelitian ini dibuat model tersendiri yang dilatih khusus untuk mengenali jenis kata dalam bahasa Indonesia. Model dilatih menggunakan korpus berbahasa Indonesia yang telah ditandai secara manual oleh Dinakaramani dkk. (2014). Himpunan simbol jenis kata (*tagset*) yang dikenali juga menyesuaikan dengan *tagset* yang digunakan pada korpus tersebut. Himpunan *tagset* terlampir pada Lampiran C. *Tagset* tersebut mendefinisikan jenis kata yang dikenali dalam 23 kategori *tag* yang berbeda. Penjelasan lebih lanjut mengenai proses pelatihan *POS Tagger* dipaparkan pada bagian 4.1.2.

Proses *POS Tagging* dapat diamati pada Gambar 3.13. Setiap kata akan ditandai dengan *tag* yang sesuai dan kemudian dikembalikan dalam bentuk himpunan kalimat.

Teks awal	menurut dr ani jahe dapat mengobati batuk dalam hari		
POS Tagging	↓		
	Kata	POS Tag	Keterangan
	menurut	VB	Verba (Kata Kerja)
	dr	X	Tidak Diketahui
	ani	NN	Noun (Kata Benda)
	jahe	NN	Noun (Kata Benda)
	dapat	RB	Adverb
	mengobati	VB	Verba (Kata Kerja)
	batuk	NN	Noun (Kata Benda)
Representasi Hasil Secara Tekstual	dalam	JJ	Adjektiva (Kata Sifat)
	hari	NN	Noun (Kata Benda)
	↓		
	menurut/VB dr/X ani/NN jahe/NN dapat/RB mengobati/VB		
	batuk/NN dalam/JJ hari/NN		

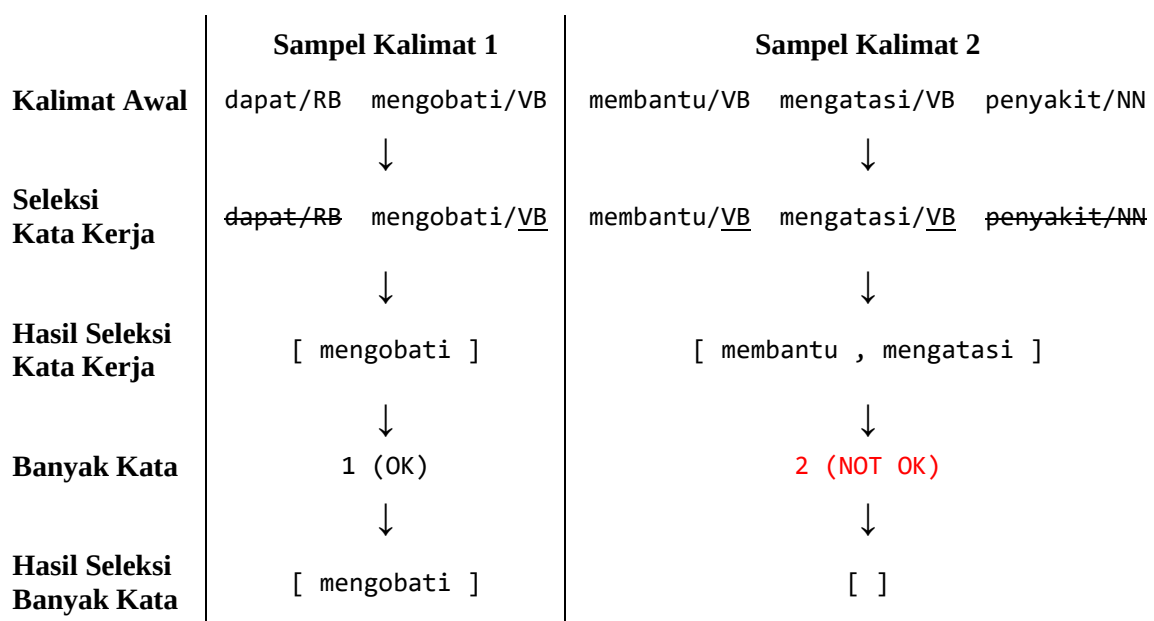
Gambar 3.13 Contoh proses *POS Tagging* oleh *POS Tagger*

3.3.2. Seleksi Kata Kerja

Setelah proses *POS Tagging* selesai, selanjutnya dilakukan penyaringan kata sehingga hanya kata kerja saja yang diseleksi. Aturan penyaringan kata untuk setiap kalimat yang ada dalam korpus adalah sebagai berikut:

- Dalam satu kalimat, hanya diambil kata-kata yang merupakan kata kerja saja, karena asumsinya adalah bahwa nama relasi selalu berbentuk kata kerja.
- Apabila dalam satu kalimat menghasilkan lebih dari satu kata kerja, maka kalimat tersebut dianggap tidak memenuhi kriteria dan semua kata kerja yang ada dalam kalimat tersebut tidak diseleksi (dibuang). Hal ini dilakukan karena kalimat-kalimat yang mengandung lebih dari satu kata kerja akan menimbulkan ambiguitas konteks, karena secara komputasi sulit menentukan kata kerja mana yang lebih tepat sebagai nama relasi.

Hasil keseluruhan dari tahap anotasi korpus ini adalah korpus yang terdiri dari kumpulan kata kerja yang merupakan kandidat potensial nama relasi, seperti terlihat pada Tabel 3.4.



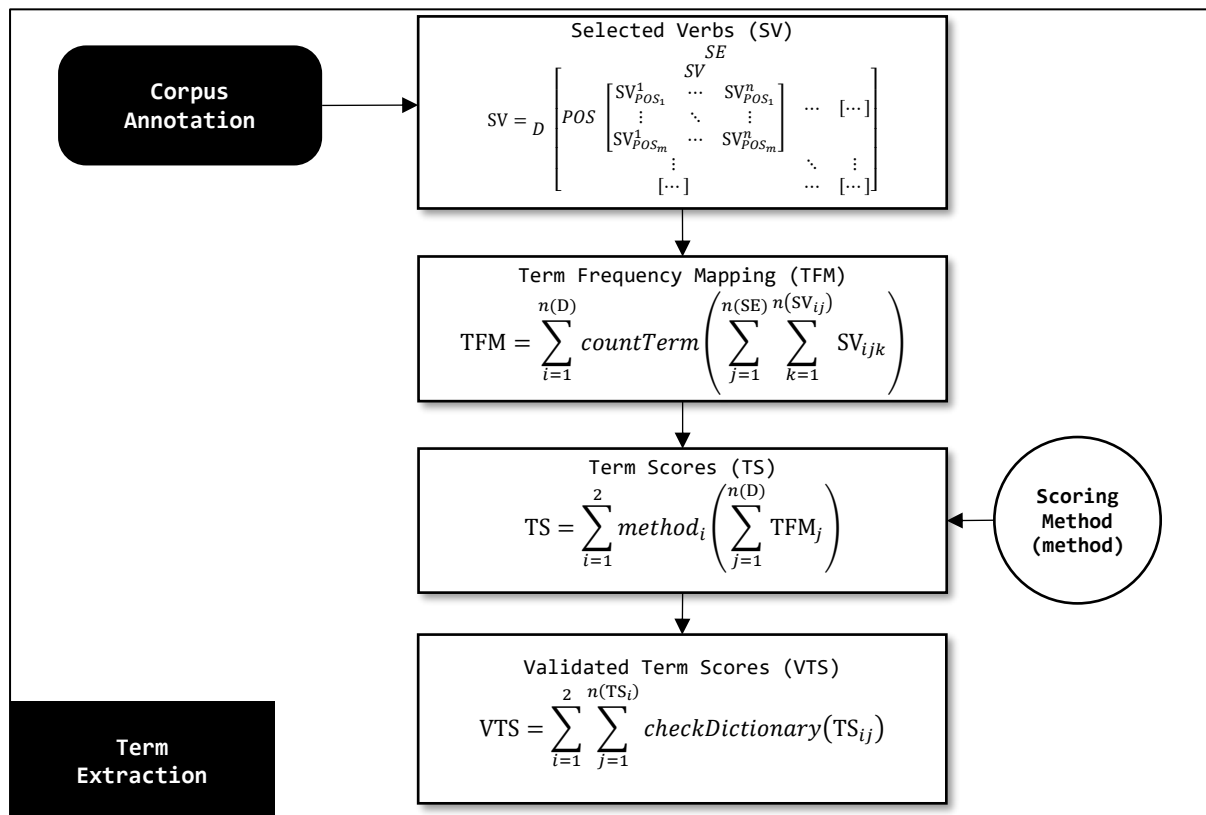
Gambar 3.14 Alur seleksi kata

Tabel 3.4 Korpus yang sudah diolah melalui tahapan *POS Tagging* dan seleksi kata

Pasangan Data		Korpus (Google)			
		Kalimat 1	Kalimat 2	Kalimat 3	...
jahe	batuk	mengobati	mengobati	...	...
temulawak	osteorthritis	mengatasi	-	...	...
kunyit	diare	-	-	...	...
...	...	...	...	...	...

3.4. Ekstraksi Kata (*Term Extraction*)

Tahap ketiga pembelajaran relasi non-taksonomi yaitu ekstraksi kata. Pada tahap ini, setiap kata yang telah ditandai telah disaring sehingga hanya kata kerja saja yang diambil. Kemudian setiap kata kerja yang muncul akan dihitung jumlah kemunculannya secara terpisah untuk tiap korpus, sehingga setiap korpus menghasilkan himpunan kata kerja masing-masing beserta jumlah kemunculan kata tersebut dalam korpus yang bersangkutan. Dari titik ini, diusulkan dua metode terpisah yang dilakukan untuk memilih kata yang tepat dijadikan sebagai nama relasi, yaitu metode Maksimum dari Rata-rata Absolut Skor Ternormalisasi (Maximum of Absolute Average of Normalized Score - MAANS) dan metode Frekuensi Terbesar dari Rekomendasi Lokal (Most Frequent Local Recommendation - MFLR). Kedua metode tersebut akan menghasilkan himpunan kata kerja keseluruhan dari seluruh korpus di mana setiap kata kerja akan memiliki nilai skor masing-masing. Meskipun kedua metode menerapkan pendekatan yang berbeda untuk memberikan nilai skor untuk masing-masing kata kerja, namun pada kedua metode tersebut kata kerja yang memiliki nilai skor paling tinggi adalah kata yang dipilih sebagai nama relasi. Alur kerja ekstraksi kata dapat diamati pada Gambar 3.15.



Gambar 3.15 Formula matematis sederhana ekstraksi kata

3.4.1. Perhitungan Frekuensi Kemunculan Kata

Sebelum dilakukan proses pemilihan nama relasi, frekuensi kemunculan tiap kata kerja untuk setiap korpus dari setiap pasangan data harus dihitung terlebih dahulu. Setiap pasangan data dengan korpus masing-masing akan menghasilkan suatu daftar kata kerja di mana setiap kata kerja memiliki nilai masing-masing, yaitu jumlah frekuensi kemunculan kata kerja tersebut pada korpus yang bersangkutan. Hasil perhitungan frekuensi kemunculan kata kerja dapat diamati pada Tabel 3.5.

Tabel 3.5 Korpus yang sudah diubah menjadi daftar kata kerja beserta nilainya

Pasangan Data		Daftar Kata Kerja			
		mengobati	mengatasi	menyembuhkan	...
jahe	batuk	55	40	5	...
temulawak	osteorthritis	50	20	30	...
kunyit	diare	35	40	-	...
...	...	...	...	...	...

Setelah dilakukan proses perhitungan frekuensi kemunculan kata, selanjutnya adalah perhitungan skor untuk memilih nama relasi.

3.4.2. *Maximum of Absolute Average of Normalized Score (MAANS)*

Metode *Maximum of Absolute Average of Normalized Score (MAANS)* menerapkan pendekatan secara statistik di mana setiap kata dalam setiap pasangan data dihitung bobotnya relatif terhadap banyak kata dalam pasangan data tersebut sebagai bentuk relevansi terhadap pasangan data. Setiap skor dalam satu himpunan yang sama kemudian dinormalisasi dengan metode normalisasi min-max. Hal ini dilakukan karena normalisasi membantu mencegah atribut dengan rentang data yang besar untuk mendominasi atribut dengan rentang data yang lebih kecil (Han dkk., 2011). Kemudian setiap kata dihitung rata-rata nilai skornya dengan diagregasikan dengan kata yang sama pada pasangan data yang berbeda. Metode ini diusulkan untuk meningkatkan probabilitas kata yang lebih sering muncul dalam pasangan data yang berbeda, meskipun frekuensi kemunculan katanya tidak selalu tinggi pada pasangan data tertentu. Berikut adalah langkah-langkah penyelesaian untuk metode MAANS:

- Buat suatu himpunan yang berisi semua kata kerja unik yang muncul dari hasil pencarian semua pasangan data. Contoh himpunan kata kerja untuk setiap pasangan data dapat diamati pada Tabel 3.6 dan himpunan kata kerja unik pada Tabel 3.7.

Tabel 3.6 Himpunan frekuensi kata kerja dari beberapa sampel data (MAANS)

[jahe] – [batuk]		[temulawak] – [osteorthritis]		[kunyit] – [diare]	
Kata Kerja	Frek.	Kata Kerja	Frekuensi	Kata Kerja	Frek.
mengobati	55	mengobati	50	mengatasi	40
mengatasi	40	menyembuhkan	30	mengobati	35
menyembuhkan	5	mengatasi	20	meredakan	25
Total	100	Total	100	Total	100

Tabel 3.7 Himpunan kata kerja unik dari semua sampel data (MAANS)

Kata Kerja	Skor
mengobati	-
mengatasi	-
menyembuhkan	-
meredakan	-

- b. Untuk setiap kata kerja yang ada dalam setiap pasangan data, beri nilai skor ternormalisasi berdasarkan frekuensi kemunculan kata kerja pada pasangan data tersebut. Contoh perhitungan skor ternormalisasi dapat diamati pada Tabel 3.8 dan persamaan (3.1):

Tabel 3.8 Himpunan frekuensi kata kerja untuk salah satu sampel data

[jahe] – [batuk]		
Kata Kerja	Frekuensi Kemunculan	Skor Ternormalisasi
mengobati	55	0.55
mengatasi	40	0.40
menyembuhkan	5	0.05
Total	100	1.00

$$\text{Skor ternormalisasi} = \frac{\text{frekuensi kata kerja}}{\text{frekuensi total}} \quad (3.1)$$

- c. Untuk setiap kata kerja yang ada dalam himpunan kata kerja unik dari semua data (seperti terlihat pada Tabel 3.7), jumlahkan skor dari kata kerja yang sama untuk setiap pasangan data. Apabila suatu kata kerja tidak muncul pada suatu pasangan data, maka skor dianggap 0. Penjumlahan skor dapat diamati seperti pada Tabel 3.9.

Tabel 3.9 Penjumlahan skor pada himpunan kata kerja unik dari semua data

Kata Kerja	Pasangan Data			Penjumlahan Skor
	[jahe] – [batuk]	[temulawak] – [osteorthritis]	[kunyit] – [diare]	
mengobati	0.55	0.50	0.35	1.40
mengatasi	0.40	0.20	0.40	1.10
menyembuhkan	0.05	0.30	-	0.35
meredakan	-	-	0.25	0.25

- d. Skor akhir setiap kata kerja merupakan rata-rata skor, maka dari itu setiap skor dari hasil penjumlahan dibagi dengan banyaknya pasangan data yang ada seperti terlihat pada persamaan (3.2). Perhitungan skor akhir dapat diamati pada Tabel 3.10.

Tabel 3.10 Perhitungan skor total setiap kata kerja metode MAANS

Kata Kerja	Penjumlahan Skor	Skor Akhir
mengobati	1.45	$1.40 / 3 = 0.46667$
mengatasi	1.05	$1.10 / 3 = 0.36667$
menyembuhkan	0.35	$0.35 / 3 = 0.11667$
meredakan	0.25	$0.25 / 3 = 0.08333$

$$\text{Skor akhir} = \frac{\text{jumlah skor total}}{\text{jumlah pasangan data}} \quad (3.2)$$

Kata kerja dengan rata-rata nilai skor tertinggi adalah kata yang dipilih sebagai nama relasi. Untuk contoh pada Tabel 3.10, kata kerja yang dipilih sebagai relasi non-taksonomi adalah kata “**mengobati**” dengan rata-rata skor **0.46667**.

3.4.3. *Most Frequent Local Recommendation (MFLR)*

Metode *Most Frequent Local Recommendation* (MFLR) menerapkan pendekatan sederhana di mana setiap pasangan data diharuskan menghasilkan keluaran berupa satu kata yang paling banyak muncul dalam pasangan data tersebut. Kemudian setelah semua pasangan data memiliki satu kata pilihan, pilih kata yang paling banyak dipilih oleh pasangan-pasangan data. Berikut adalah langkah-langkah penyelesaian untuk metode MFLR:

- a. Untuk setiap pasangan data, pilih salah satu kata kerja yang frekuensi kemunculannya tertinggi dan masukkan ke dalam suatu himpunan pasangan baru di mana setiap pasangan adalah pasangan data dengan kata kerja terpilih. Pemilihan kata kerja untuk setiap data dapat diamati pada Tabel 3.11 dan himpunan kata kerja terpilih pada Tabel 3.12.

Tabel 3.11 Himpunan frekuensi kata kerja diurutkan berdasarkan frekuensi

[jahe] – [batuk]		[temulawak] – [osteorthritis]		[kunyit] – [diare]	
Kata Kerja	Frekuensi	Kata Kerja	Frekuensi	Kata Kerja	Frekuensi
mengobati	55	mengobati	50	mengatasi	40
mengatasi	40	menyembuhkan	30	mengobati	35
menyembuhkan	5	mengatasi	20	meredakan	25
Total	100	Total	100	Total	100

Tabel 3.12 Himpunan pasangan data dengan kata kerja terpilih

Pasangan Data	Kata Kerja Terpilih
[jahe] – [batuk]	mengobati
[temulawak] – [osteorthritis]	mengobati
[kunyit] – [diare]	mengatasi

- b. Untuk setiap kata kerja yang muncul sebagai kata kerja terpilih suatu pasangan data, hitung banyak data yang memilih kata kerja tersebut sebagai kata kerja terpilih. Hasil pencacahan pemilihan tersebut menjadi skor akhir untuk setiap kata kerja. Hasil skor akhir dapat diamati pada Tabel 3.13.

Tabel 3.13 Perhitungan skor total setiap kata kerja metode MFLR

Kata Kerja	Hasil Pencacahan (Skor Akhir)
mengobati	2
mengatasi	1

Kata kerja dengan nilai skor tertinggi adalah kata yang dipilih sebagai nama relasi. Untuk contoh pada Tabel 3.13, kata kerja yang dipilih sebagai relasi non-taksonomi adalah kata “**mengobati**” dengan skor 2 (terdapat 2 data yang memilih kata “**mengobati**” sebagai kata kerja dengan frekuensi tertinggi).

3.4.4. Penyaringan Nama Relasi Taksonomis

Karena nama relasi yang dicari adalah nama relasi non-taksonomi, maka nama-nama relasi yang digunakan untuk merepresentasikan nama relasi taksonomis harus disaring sehingga tidak mempengaruhi hasil akhir ekstraksi nama relasi. Nama-nama relasi taksonomis disimpan sebagai suatu himpunan *stopwords*. Dalam perhitungan skor total ekstraksi kata, apabila suatu kata terdeteksi sebagai anggota dari himpunan nama relasi taksonomis maka kata tersebut akan dibuang terlepas dari berapapun skor yang dihasilkan.

Tabel 3.14 Contoh daftar *stopwords* nama relasi taksonomis

#	Kata Nama Relasi Taksonomis
1	adalah
2	ialah
3	merupakan
4	termasuk

3.4.5. Validasi Kata Kerja Menggunakan Kamus

Sebelum dilakukan pemilihan kata kerja sebagai nama relasi non-taksonomi dari hasil perhitungan skor tahap sebelumnya, ada satu validasi yang harus dilakukan, yaitu memeriksa apakah kata kerja yang ada dalam tabel skor adalah kata kerja yang valid. Pada tahap anotasi korpus, pendeteksian kata kerja dilakukan dengan menggunakan POS Tagger yang menggunakan model terlatih. Karena sifat pembelajaran/pelatihannya yang *unsupervised learning*, maka akurasi deteksi dari POS Tagger tidak selalu tepat sasaran. Dalam klasifikasi, ada empat kategori yang menentukan keakuratan dan presisi dari suatu proses klasifikasi yang dapat diamati pada Tabel 3.15.

Tabel 3.15 Kategori penentuan akurasi hasil prediksi/klasifikasi

		Hasil Prediksi	
		POS Tagging	
Nilai Sebenarnya	Verb	True Positive (TP)	False Negative (FN)
	Not Verb	False Positive (FP)	True Negative (TN)

Dalam penelitian ini, proses klasifikasi yang diterapkan adalah mencari kata kerja yang ada dalam korpus, yang didapat dari penentuan jenis kata yang dilakukan menggunakan POS Tagger. Pemeriksaan kebenaran penentuan jenis kata dilakukan dengan cara pemeriksaan ulang pada kamus, yaitu Kamus Besar Bahasa Indonesia. Namun karena kamus yang dipakai merupakan layanan eksternal (diluar program), maka pemeriksaan semua kata satu per satu tidak mungkin dilakukan. Untuk mengurangi beban, maka dari itu dalam penelitian ini diputuskan bahwa hanya kata-kata yang terdeteksi sebagai kata kerja saja yang akan diperiksa menggunakan kamus. Pada tahap anotasi korpus, *True Negative* dan *False Negative* sudah dieliminasi dengan menyingkirkan kata-kata yang dideteksi oleh POS Tagger sebagai bukan kata kerja. Artinya tahap ini hanya diterapkan untuk menyingkirkan *False Positives* (FP), yaitu kata yang sebenarnya bukan kata kerja tetapi terdeteksi oleh POS Tagger sebagai kata kerja. Sehingga pada tahap akhir kata kerja yang terdaftar adalah kata kerja yang masuk dalam kategori *True Positive* (TP), yaitu kata kerja yang terdeteksi oleh POS Tagger sebagai kata kerja dan benar-benar merupakan kata kerja.

3.5. Pengembangan Kata (*Term Refinement*)

Tahap terakhir pembelajaran relasi non-taksonomi yaitu pengembangan kata. Pada tahap ini, daftar skor akhir yang sudah diolah kemudian diurutkan berdasarkan nilai skornya dari nilai terbesar ke terkecil (*descending*). Nama relasi non-taksonomi yang dipilih adalah kata kerja dengan skor tertinggi. Namun untuk memberikan pilihan tambahan bagi pengguna, maka diambil lima kata kerja dengan skor tertinggi yang kemudian dijabarkan bentuk katanya menjadi tiga bagian, yaitu kata dasarnya, bentuk aktifnya, dan bentuk pasifnya.

3.5.1. Deteksi Kata Kerja Aktif dan Pasif

Pendeteksian kata kerja aktif dan pasif dilakukan terhadap lima kata kerja dengan skor tertinggi. Program pendeteksi kata kerja aktif dan pasif dikembangkan sendiri oleh peneliti dengan berbasis aturan (*rule-based*). Aturan-aturan dalam program tersebut dibuat berdasarkan aturan linguistik bahasa Indonesia yang merujuk ke buku oleh (Sugihastuti & Saudah, 2016). Aturan-aturan linguistik yang digunakan meliputi pemenggalan suku kata dan aturan bentuk kata (imbuhan). Aturan-aturan tersebut kemudian diimplementasikan dalam bentuk *regular expression* dan percabangan program. Dalam pendeteksian juga dilakukan *stemming* untuk mendapatkan kata dasar. Hasil inferensi kata dasar dan perubahan imbuhan kata kerja aktif dan pasif kemudian diperiksa ulang ke kamus untuk menentukan apakah hasil inferensi sudah benar. Meski demikian, program pendeteksi yang dibuat belum sempurna dan masih banyak ruang untuk pengembangan lebih lanjut. Contoh *regular expression* yang digunakan untuk melakukan *stemming* dapat diamati pada Gambar 3.16.

```

^{all}
  ({prefiks_1}
    ({pre1_se} (? :se))
    | ({pre1_di} (? :di))
    | ({pre1_ke} (? :ke))
    | ({pre1_te} (? :te(? :r|)))
    | ({pre1_be} (? :be(? :r|)))
    | ({pre1_me}
      (? :me(? >m(? :kh|ng|ny|sy|[bcdfghjklmnpqrstvwxyz]))|ng(? =[gh]|(? :kh|ng|ny|sy|[bcdfghjklmnpqrstvwxyz]))|n(? !(? :[aiueo])|[gy]))))
    | ({pre1_pe}
      (? :pe(? >m(? :kh|ng|ny|sy|[bcdfghjklmnpqrstvwxyz]))|ng(? =[gh]|(? :kh|ng|ny|sy|[bcdfghjklmnpqrstvwxyz]))|n(? !(? :[aiueo])|[gy])|r|)))
    |
  )
  ({prefiks_2}
    (? (pre1_ke)
      (? :(? :te(? :r|))|(? :be(? :r|)))(? :(? :kh|ng|ny|sy|[bcdfghjklmnpqrstvwxyz]))|)
    | (? (pre1_be)
      (? :({pre2_ke_1} (? :ke))|(? :pe(? >m(? :kh|ng|ny|sy|[bcdfghjklmnpqrstvwxyz]))|ng(? =[gh]|(? :kh|ng|ny|sy|[bcdfghjklmnpqrstvwxyz]))|n(? !(? :[aiueo])|[gy])|r|)))|)
  )

```

```

    | (? (pre1_pe)
    (? : ({pre2_ke_2} (? : ke)) | (? : be (? : r)) | (? = (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz])) | )
    | (? (pre1_me)
    (? : ({pre2_ke_3} (? : ke)) | (? : pe (? > m (? = (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz])) | ng (? = [gh] | (?
    : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz])) | n (? ! (? : [aiueo]) | [gy]) | r)) | (? : be (? : r)) | (? = (? : kh | n
    g | ny | sy | [bcdfghjklmnpqrstvwxyz])) | )
    | (? (pre1_di)
    (? : ({pre2_ke_4} (? : ke)) | (? : pe (? > m (? = (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz])) | ng (? = [gh] | (?
    : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz])) | n (? ! (? : [aiueo]) | [gy]) | r)) | (? : be (? : r)) | (? = (? : kh | n
    g | ny | sy | [bcdfghjklmnpqrstvwxyz])) | )
    | (? (pre1_te)
    (? : ({pre2_ke_5} (? : ke)) | (? : pe (? > m (? = (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz])) | ng (? = [gh] | (?
    : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz])) | n (? ! (? : [aiueo]) | [gy]) | r)) | (? : be (? : r)) | (? = (? : kh | n
    g | ny | sy | [bcdfghjklmnpqrstvwxyz])) | )
    |
    ))))
  )
  ({kata_dasar}

  (? : (? : [aiueo]) (? : kh | ng | ny | sy | \w) {2,} ? | (? <= nge) (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz]) (?
  : [aiueo]) (? : kh | ng | ny | sy | \w) | (? <= \w) (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz]) (? : [aiueo]) (
  ? : kh | ng | ny | sy | \w) | (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz]) (? : kh | ng | ny | sy | \w) {3,} ?)
  (? : (? : \w-
  (? : (? : [aiueo]) (? : kh | ng | ny | sy | \w) {2,} ? | (? <= nge) (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz]) (?
  : [aiueo]) (? : kh | ng | ny | sy | \w) | (? <= \w) (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz]) (? : [aiueo]) (
  ? : kh | ng | ny | sy | \w) | (? : kh | ng | ny | sy | [bcdfghjklmnpqrstvwxyz]) (? : kh | ng | ny | sy | \w) {3,} ?) *)
  )
  ({suffiks_1}
  isasi
  | isme
  | ({suf1_an}
  (? (pre1_ke) an
  | (? (pre2_ke_1) an
  | (? (pre2_ke_2) an
  | (? (pre2_ke_3) an
  | (? (pre2_ke_4) an
  | (? (pre2_ke_5) an
  |
  (? : (? (pre1_me) | (? (pre1_di) | (? (pre1_te) | an))) | (? (pre1_se) | (? (pre1_pe) | (? (pre1_ke) | kan))) |
  )
  ))))
  )
  )
  ({suffiks_2}
  (? (pre1_se) | (? (pre1_pe) | (? (pre1_ke) | kan)))
  | (? (pre1_be) | (? (pre1_ke) | (? (pre1_se) | (? (pre1_pe) | i))))
  | (? (suf1_an) | (? (pre1_me) | (? (pre1_di) | (? (pre1_te) | an))))
  |
  )
  ({posesif}
  ku | mu | nya |
  )
  ({partikel}
  lah | kah | tah | pun |
  )
  )
  )$

```

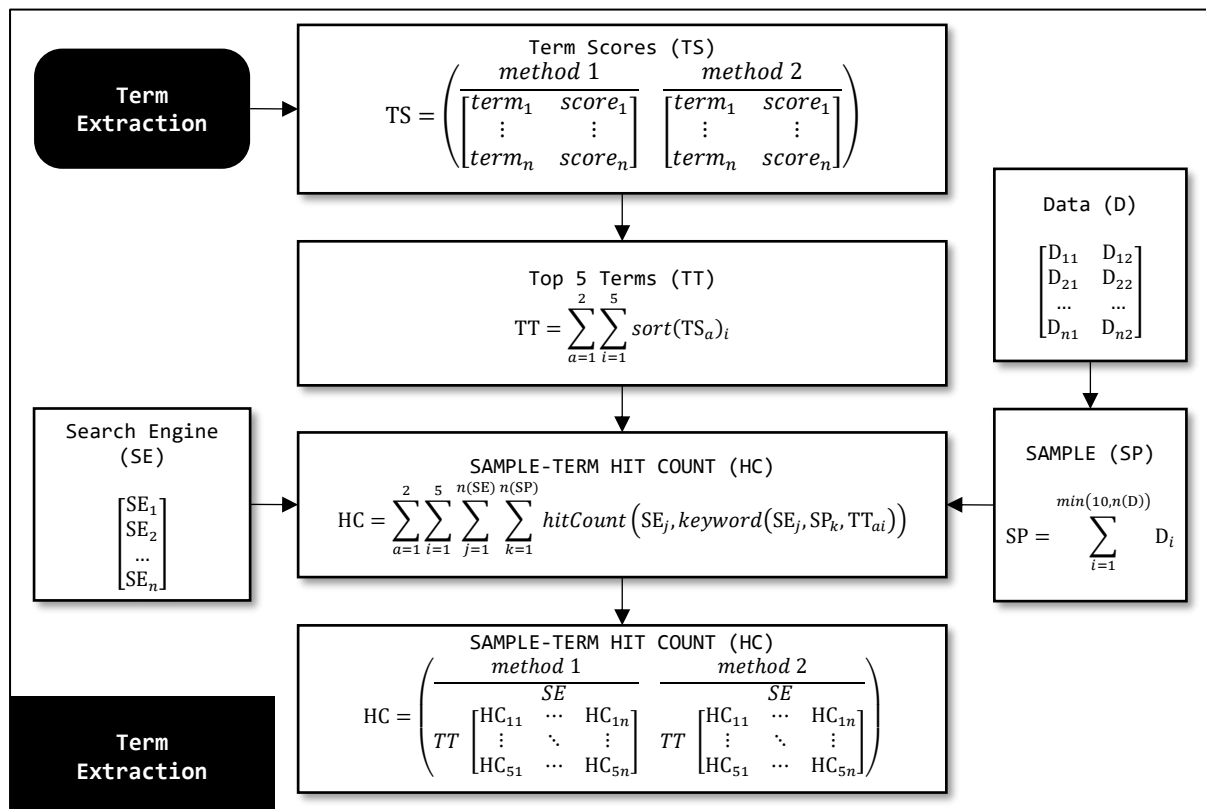
Gambar 3.16 *Regular expression* untuk *stemming* (sintaks menyesuaikan *library*)

3.5.2. Pemilihan Nama Relasi Non-Taksonomi

Setelah semua proses telah dilalui, pemilihan nama relasi non-taksonomi cukup mudah, yaitu memilih kata kerja dengan skor tertinggi sebagai nama relasi non-taksonomi. Karena ada dua metode penilaian skor, maka program akan memilih salah satu metode secara acak, kemudian mengambil kata kerja dengan skor tertinggi pada metode tersebut.

3.6. Evaluasi Kata (*Term Evaluation*)

Tambahan proses untuk pembelajaran relasi non-taksonomi yaitu evaluasi kata. Pada proses ini, beberapa kata yang memiliki nilai skor tertinggi akan diambil untuk dilakukan pencarian ulang melalui mesin pencari dengan pasangan data masukan. Tahap ini dilakukan dengan tujuan untuk menilai apakah kata dengan nilai skor tertinggi merupakan nama relasi yang tepat apabila dikombinasikan dengan pasangan-pasangan data yang dimiliki. Penilaian dilakukan dengan cara sederhana, yaitu melihat jumlah *hit* sampel pasangan data dan kata kerja yang dievaluasi pada hasil pencarian mesin pencari. Dari jumlah hasil pencarian untuk masing-masing kata, pengguna bisa menilai apakah kata yang dipilih oleh sistem merupakan kata yang tepat. Alur kerja proses evaluasi kata dapat diamati pada Gambar 3.17.



Gambar 3.17 Formula matematis sederhana evaluasi kata

3.6.1. Sampling Data untuk Evaluasi

Untuk mengurangi beban evaluasi dalam kasus di mana pasangan data kelas yang digunakan berjumlah banyak, maka untuk tahap evaluasi dilakukan *sampling data* untuk mengambil beberapa pasangan data kelas secara acak. Dalam penelitian ini, jumlah sampel yang digunakan ada sebanyak 10 data sampel (atau sesuai jumlah pasangan data jika jumlahnya kurang dari 10). Contoh hasil sampling dapat diamati pada Tabel 3.16.

Tabel 3.16 Contoh himpunan pasangan data sampel dari hasil *random sampling*

No.	Pasangan Data	
1	jahe	batuk
2	temulawak	osteorthritis
3	kunyit	diare
...	...	...
10	daun dewa	wasir

3.6.2. Seleksi Kata Berdasarkan Skor Tertinggi

Setiap pasangan data sampel akan diujikan dengan setiap kata kerja yang akan dievaluasi. Kata-kata kerja yang dievaluasi adalah beberapa kata kerja yang skor akhirnya paling tinggi, yaitu lima (5) kata kerja dengan skor tertinggi. Karena setiap metode (MAANS dan MFLR) mempunyai kemungkinan untuk menghasilkan lima (5) kata kerja yang berbeda, maka masing-masing metode dievaluasi secara independen. Contoh himpunan kata kerja terpilih yang akan dievaluasi dapat diamati pada Tabel 3.17.

Tabel 3.17 Himpunan kata kerja terpilih untuk evaluasi masing-masing metode

Metode MAANS		Metode MFLR	
Kata Kerja	Skor Akhir	Kata Kerja	Skor Akhir
mengobati	0.46667	mengobati	10
mengatasi	0.36667	mengatasi	6
menyembuhkan	0.08333	menyembuhkan	4
meredakan	0.06667	meredakan	2
menghilangkan	0.03333	mengurangi	2

3.6.3. Proses Pencarian Jumlah Hit pada Mesin Pencari

Kemudian untuk setiap pasangan data sampel dan kata kerja dilakukan pencarian melalui mesin pencari untuk menemukan jumlah *hit* total pencarian. Artinya untuk sampel sejumlah 10 sampel data dan himpunan kata kerja sejumlah 5 kata kerja, maka akan dilakukan proses pencarian untuk 50 (10 x 5) kata kunci pada setiap mesin pencari. Hasil pencarian untuk setiap sampel data dapat diamati pada Tabel 3.18.

Tabel 3.18 Hasil pencarian setiap data sampel untuk setiap kata kerja

[jahe] – [batuk]			[kunyit] – [diare]	...
Kata Kerja	Kata Kunci (Google)	Jumlah HIT	...	...
mengobati	"jahe * mengobati * batuk"	1.500	...	...
mengatasi	"jahe * mengatasi * batuk"	1.200	...	...
menyembuhkan	"jahe * menyembuhkan * batuk"	1.000	...	...
meredakan	"jahe * meredakan * batuk"	400	...	...
menghilangkan	"jahe * menghilangkan * batuk"	500	...	...

Setelah setiap data sampel menghasilkan jumlah hit masing-masing berdasarkan mesin pencari, selanjutnya dilakukan pengelompokan jumlah *hit* dari setiap data sampel yang dievaluasi berdasarkan mesin pencari. Hasil pengelompokan untuk setiap pasangan data dapat diamati pada Tabel 3.19.

Tabel 3.19 Hasil penjumlahan jumlah *hit* setiap mesin pencari

Pasangan data [jahe] – [batuk]			
Kata Kerja	Jumlah HIT		
	Google	Ecosia	Yahoo
mengobati	1.500	1.300	1.400
mengatasi	1.200	900	600
menyembuhkan	1.000	900	900
meredakan	400	900	1.500
menghilangkan	500	900	400

Langkah terakhir yaitu dilakukan penjumlahan jumlah *hit* dari semua pasangan data untuk masing-masing kata kerja yang dikelompokkan berdasarkan mesin pencari. Hasil penjumlahan untuk setiap mesin pencari ditambah dengan hasil penjumlahan total dari semua mesin pencari merupakan hasil akhir proses evaluasi dan dapat diamati pada Tabel 3.20. Tujuan dari proses evaluasi ini adalah untuk memberikan gambaran sederhana pada pengguna tentang kesesuaian pencarian nama relasi non-taksonomi dengan asumsi kenyataan dengan mengandalkan kemampuan mesin pencari dalam mencacah dokumen yang dianggap relevan.

Tabel 3.20 Hasil akhir penjumlahan jumlah *hit* total setiap kata kerja

Kata Kunci	Search Engine			Total
	Google	Yahoo	Ecosia	
mengobati	10.700	9.900	8.600	29.200
mengatasi	9.400	7.800	7.500	24.700
menyembuhkan	5.700	5.400	7.800	18.900
meredakan	5.500	6.000	4.900	16.400
menghilangkan	4.400	5.200	4.000	13.600

BAB IV

IMPLEMENTASI PROGRAM

4.1. Implementasi Program Utama

Implementasi program dilakukan untuk menerapkan dan menguji alur kerja pembelajaran relasi non-taksonomi yang diusulkan. Ada beberapa hal yang perlu diperhatikan dalam proses implementasi program, yaitu kebutuhan perangkat lunak seperti bahasa pemrograman yang digunakan serta *library* yang dibutuhkan untuk melakukan proses tertentu, layanan eksternal yang digunakan, dan URL yang digunakan untuk mengakses API (*application programming interface*) program utama.

4.1.1. Kebutuhan Perangkat Lunak Program Utama

Berikut adalah bahasa pemrograman yang digunakan, *library* yang dibutuhkan, dan perangkat lunak eksternal yang digunakan oleh program utama:

a. Bahasa pemrograman: Java

Bahasa pemrograman Java dipilih karena terdapat cukup banyak *library* yang dibutuhkan, yang ditulis dalam bahasa pemrograman Java. Selain itu, Java juga merupakan bahasa pemrograman yang *cross-platform*, artinya output dari kode program Java (*Java bytecode*) dapat digunakan pada sistem operasi apapun selama sistem operasi tersebut memiliki *Java Virtual Machine* (JVM). Selain itu Java juga mendukung komputasi paralel menggunakan konsep *threading*, sehingga memungkinkan beberapa proses untuk dilakukan dalam waktu bersamaan. Dalam implementasi program, komputasi paralel digunakan untuk melakukan pencarian melalui beberapa mesin pencari yang berbeda dalam waktu bersamaan.

b. Package manager: Maven

Maven adalah salah satu *package manager* yang digunakan oleh pengguna Java untuk memudahkan pengunduhan dan penggunaan *library* Java.

c. Framework Web: Java Spring 5

Framework Java Spring digunakan untuk memudahkan pengembangan program yang dapat diakses dengan mudah melalui web (HTTP). Dalam implementasi program, Java Spring digunakan untuk menyediakan API kepada antarmuka pengguna sehingga pengguna dapat mengakses layanan yang disediakan oleh program utama dengan mudah.

d. *Library* NLP dan POS Tagger: Stanford CoreNLP

Stanford CoreNLP memiliki beragam fitur dalam konteks operasi NLP (*natural language processing*) seperti *tokenization*, *sentence splitting*, dan juga *POS tagging*. Beberapa operasi NLP dalam program yang dilakukan menggunakan *library* ini meliputi:

1. Tokenisasi kata: pemisahan per kata menjadi bentuk *array*
2. *Sentence split*: pemisahan kalimat menjadi bentuk *array*
3. *POS Tagger*: mendeteksi jenis kata

Model POS Tagger dilatih menggunakan korpus di mana setiap katanya sudah ditandai jenis katanya secara manual, yang dibuat oleh Dinakaramani dkk. (2014). Halaman situs NLP Stanford seperti terlihat pada Gambar 4.1.



Gambar 4.1 Halaman situs Stanford NLP Group (bagian POS Tagger)

Sumber: <https://nlp.stanford.edu/>

e. *Library* Komunikasi Web: Jsoup

Jsoup digunakan untuk mengakses dokumen dari web mesin pencari dan web Kamus Besar Bahasa Indonesia. Jsoup dapat melakukan *request* dokumen dalam bentuk HTML, yang kemudian dapat dilakukan *parsing* untuk mengekstrak data dalam dokumen HTML yang dianggap penting dan relevan.

f. *Library Regex: JRegex*

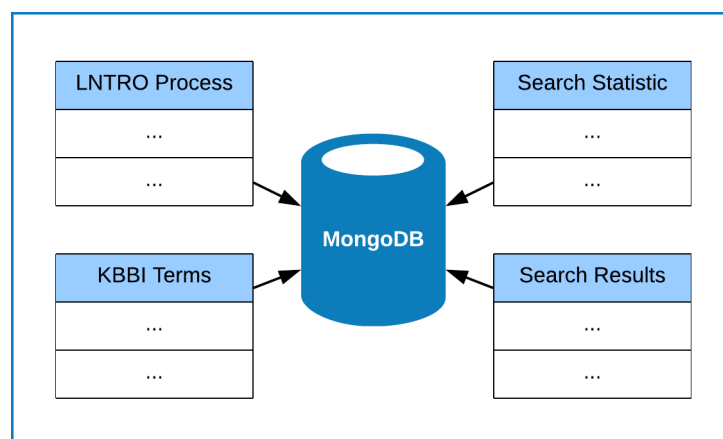
JRegex digunakan untuk melakukan operasi *regular expression (regex)* yang kompleks. *Library regex* yang disediakan oleh Java tidak memiliki kemampuan kompleks layaknya JRegex, seperti kemampuan *backreference* dan *named grouping*. Dalam implementasi program, JRegex digunakan untuk melakukan operasi *stemming* sederhana, yang kemudian digunakan untuk melakukan deteksi kata kerja aktif dan pasif.

g. *Library Ontologi: Apache Jena*

Apache Jena digunakan untuk melakukan operasi yang berhubungan dengan ontologi. Dalam implementasi program, Apache Jena digunakan untuk memodelkan data kelas dan relasi yang sudah ditemukan ke dalam bentuk dokumen ontologi (.owl).

h. *Basisdata: MongoDB*

MongoDB adalah salah satu basisdata yang termasuk dalam kategori NoSQL dan berorientasi dokumen (*document-oriented*), di mana dokumen disimpan dalam sintaks yang mirip JSON. Dalam implementasi program, basisdata digunakan sebagai *cache* untuk menyimpan beberapa hasil pemrosesan yang membutuhkan layanan eksternal. Hal ini dilakukan untuk mengurangi beban terhadap layanan eksternal yang kadang sulit diprediksi dan membutuhkan waktu yang lama. Selain itu basisdata juga digunakan untuk menyimpan hasil pembelajaran relasi non-taksonomi yang sudah dilakukan. Data yang disimpan dalam basisdata dapat diamati pada Gambar 4.2.



Gambar 4.2 Data yang disimpan pada basisdata MongoDB

4.1.2. Pengembangan Model POS Tagger

Seperti yang telah dipaparkan sebelumnya, pada penelitian ini POS Tagger yang digunakan adalah POS Tagger dari *library* Stanford CoreNLP yang merupakan implementasi dari konsep *log-linear POS Tagger* oleh Toutanova dkk. (2003). Dengan model yang tepat, *POS Tagger* tersebut dapat mengenali jenis kata dalam bahasa yang diinginkan. Namun Stanford tidak menyediakan model berbahasa Indonesia, untuk itu dalam penelitian ini dilatih model khusus untuk mengenali jenis kata dalam bahasa Indonesia.

Pelatihan Model POS Tagger

Pelatihan model POS Tagger membutuhkan dua hal utama, yaitu:

a. Data Latih

Data latih adalah data yang digunakan untuk mempelajari jenis kata yang akan dikenali. Data latih berupa korpus yang berisi kalimat-kalimat, di mana setiap kata dalam setiap kalimat telah ditandai dengan jenis kata yang tepat. Untuk memastikan kebenaran jenis kata yang telah ditandai, maka korpus harus dibangun dan ditandai secara manual oleh manusia.

Dalam penelitian ini, data latih yang digunakan adalah korpus yang telah ditandai secara manual oleh Dinakaramani dkk. (2014). Korpus tersebut berisi 10.030 kalimat dengan jumlah total kata sebanyak 256.622 kata. Korpus tersebut juga memiliki dokumen *tagset* yang menjelaskan kode setiap *tag* beserta maknanya. Dokumen *tagset* tersebut mendeskripsikan 23 jenis kata yang berbeda, yang dapat diamati pada Lampiran C.

b. Parameter Pelatihan

Parameter pelatihan digunakan untuk menentukan karakteristik dan sifat dari model yang ingin dilatih. POS Tagger dari Stanford menyediakan banyak jenis parameter untuk mengatur pelatihan model sehingga pembelajaran yang dilakukan relevan dengan data latih dan juga relevan dengan karakteristik bahasa yang dipilih. Misalnya, dalam bahasa Indonesia salah satu karakteristik utama dalam penentuan jenis kata adalah penggunaan imbuhan, seperti prefiks, suffiks, dan konfiks.

Dalam penelitian ini, pelatihan model dicoba dilakukan beberapa kali dengan menggunakan beberapa parameter yang berbeda. Hal ini dilakukan untuk mencari tahu parameter seperti apa yang menghasilkan model yang paling akurat. Salah satu dokumen parameter pelatihan (disebut sebagai *file properties*) dapat diamati pada Gambar 4.3.

```

## tagger training invoked at Sun Mar 25 17:35:04 WIB 2018 with arguments:
    model = postagger/bahasa-basic.tagger
    arch = generic,prefix(4),suffix(3), prefixsuffix(4)
    wordFunction =
    trainFile = format=TSV,wordColumn=0,tagColumn=1,./data/tagged/Corpus.tsv
    closedClassTags =
closedClassTagThreshold = 40
curWordMinFeatureThresh = 2
    debug = true
    debugPrefix = [DEBUG][BASIC]
    tagSeparator = /
    encoding = UTF-8
    iterations = 100
    lang =
    learnClosedClassTags = true
    minFeatureThresh = 5
    openClassTags =
rareWordMinFeatureThresh = 10
    rareWordThresh = 5
    search = qn
    sgml = false
    sigmaSquared = 0.5
    regL1 = 1.0
    tagInside =
    tokenize = false
    tokenizerFactory =
    tokenizerOptions =
    verbose = false
    verboseResults = true
veryCommonWordThresh = 250
    xmlInput =
    outputFile =
    outputFormat = slashTags
    outputFormatOptions =
    nthreads = 1

```

Gambar 4.3 Parameter pelatihan model POS Tagger

Rincian setiap parameter beserta efeknya dijelaskan dalam dokumentasi java (*javadoc*) dari kelas `MaxentTagger` pada *library* Stanford CoreNLP, yang tersedia secara *online* di <https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/tagger/maxent/MaxentTagger.html>. Berikut adalah beberapa parameter yang mempengaruhi hasil pelatihan model:

- a. `[arch]`: arsitektur pembelajaran model, menentukan jenis fitur/karakteristik setiap kata. Jenis-jenis arsitektur yang tersedia dipaparkan dalam referensi *javadoc* di atas.
- b. `[learnClosedClassTags]`: bernilai `true` supaya model mempelajari jenis kata yang masuk dalam kategori *closed class tag*, yaitu kategori jenis kata yang hanya terdiri dari beberapa kata baku saja, seperti kata hubung (dan, atau), kata depan (di, ke, dari), dan lain-lain.
- c. `[closedClassTagThreshold]`: banyak kata yang masuk dalam kategori tertentu sehingga dikategorikan sebagai *closed class tag*.
- d. `[search]`: metode optimasi model yang menentukan bagaimana model diperbaiki dalam setiap iterasi pembelajarannya.

- e. [sigmaSquared]: parameter peubah untuk metode [search] dengan nilai *cg* atau *qn*
- f. [iterations]: parameter peubah untuk metode [search] dengan nilai *iis*
- g. [rareWordThresh]: ambang batas frekuensi kemunculan suatu kata sehingga digolongkan sebagai kata yang jarang muncul (*rare word*)
- h. [minFeatureThresh]: ambang batas minimal frekuensi kemunculan dari suatu fitur atau karakteristik tertentu, apabila kurang dari ambang batas maka fitur tersebut diabaikan

Pelatihan model dilakukan secara langsung dengan menggunakan file *executable library* berbentuk .jar melalui *command line interface* (CLI) seperti pada Gambar 4.4.

```
$ java -classpath target/dependency/stanford-corenlp-3.1.9.jar
edu.stanford.nlp.tagger.maxent.MaxentTagger
-prop postagger/bahasa-train-basic.props
```

Gambar 4.4 Perintah pelatihan model POS Tagger dengan *library* Stanford CoreNLP

Proses eksekusi perintah pelatihan model POS Tagger di atas terlihat pada layar CLI, yang dapat diamati pada Gambar 4.5.

```
TaggerExperiments: adding word/tags
Loading tagged words from ./data/tagged/Indonesian_Manually_Tagged_Corpus.tsv
Read 256622 words from ./data/tagged/Indonesian_Manually_Tagged_Corpus.tsv [done].
Read 10030 sentences, min 2 words, max 82 words.
Featurizing tagged data tokens...
Featurized 266652 data tokens [done].
xSize [num Phi templates] = 212133; ySize [num classes] = 24
Hashing histories ...
Hashed 212133 histories.
Hashing populated histories ...
Hashed populated histories.
TaggerExperiments.getFeaturesNew: initializing fnumArr.
  length of sTemplates keys: 135167
getFeaturesNew adding features ...
  total feats: 135167, populated: 62822
  Max features per x,y pair: 11
  Max non-zero y values for an x: 24
  Number of non-zero feature x,y pairs: 4928685
  Number of zero feature x,y pairs: 162507
end getFeaturesNew.
Samples from format=TSV,wordColumn=0,tagColumn=1,./data/tagged/Indonesian_Manually_Tagged_Corpus.tsv
Number of features: 62822
Tag set: [NN, JJ, PR, NND, PRP, SC, DT, NNP, FW, WH, UH, .$. , OD, MD, X, Z, CC, CD, IN, SYM, VB, RB, NEG, RP]
pcond initialized
zlambd initialized
ftildeArr initialized
QNMinimizer called on double function of 62822 variables, using M = 10.
class edu.stanford.nlp.maxent.CGRunner
Iter: 0: neg. log cond. likelihood = 847434.4099741676 [1 calls to valueAt]
An explanation of the output:
Iter      The number of iterations
evals     The number of function evaluations
SCALING   <D> Diagonal scaling was used; <I> Scaled Identity
LINESEARCH [## M steplength] Minpack linesearch
          1-Function value was too high
          2-Value ok, gradient positive, positive curvature
          3-Value ok, gradient negative, positive curvature
          4-Value ok, gradient negative, negative curvature
          [... B] Backtracking
VALUE     The current function value
TIME      Total elapsed time
IGNORM1   The current norm of the gradient
```

Gambar 4.5 Keluaran CLI dari pelatihan model POS Tagger

Pengujian Model POS Tagger

Pengujian model POS Tagger membutuhkan tiga hal utama, yaitu:

a. Data Uji

Data uji adalah data yang digunakan untuk menguji model yang telah dilatih. Sifat data uji sama seperti data latih. Dalam penelitian ini, data uji yang digunakan juga sama seperti data latih, yaitu korpus yang telah ditandai secara manual oleh Dinakaramani dkk. (2014).

b. Model POS Tagger

Pengujian membutuhkan model POS Tagger yang telah dilatih. Proses pelatihan model menghasilkan dua *file*, yaitu *file properties* yang telah diringkas dan *file model*. *File model* POS Tagger yang dihasilkan dari tahap sebelumnya berbentuk *binary file*.

Pengujian model dilakukan secara langsung dengan menggunakan *file executable library* berbentuk *.jar* melalui *command line interface* (CLI), sama seperti halnya dengan pelatihan model. Eksekusi perintah pengujian model seperti pada Gambar 4.6.

```
$ java -classpath target/dependency/stanford-corenlp-3.9.1.jar
edu.stanford.nlp.tagger.maxent.MaxentTagger
-model postagger/bahasa-basic.tagger
-testFile format=TSV,wordColumn=0,tagColumn=1,./data/tagged/Corpus.tsv
```

Gambar 4.6 Perintah pengujian model POS Tagger dengan *library* Stanford CoreNLP

Hasil eksekusi perintah pengujian model POS Tagger di atas terlihat pada layar CLI, yang dapat diamati pada Gambar 4.7.

```
Total tags correct: 251543; wrong: 4970; unknown wrong: 0
pejabat-pejabat/NN rumah sakit/NN mengkonfirmasi/VB bahwa/SC aktivis/NN tersebut/PR tewas/VB dalam/IN serangan/NN di/IN jalur/NNP Gaza/NNP utara/NNP ./Z
Sentence number: 10024; length 14; correct: 14; wrong: 0; unknown wrong: 0
Total tags correct: 251543; wrong: 4970; unknown wrong: 0
fatah/NNP menuduh/VB pasukan/NN keamanan/NN pimpinan/NN hamas/NNP yang/SC '/Z membunuh/VB '/Z aktivis/NN tersebut/PR ./Z
Sentence number: 10025; length 13; correct: 13; wrong: 0; unknown wrong: 0
Total tags correct: 251556; wrong: 4970; unknown wrong: 0
seorang/NN juru bicara/NN bagi/IN pasukan/NN pimpinan/NN hamas/NNP tak/NEG bisa/MD diperoleh/VB komentar/NN -nya/PRP mengenai/IN hal/NN itu/PR ./Z
Sentence number: 10026; length 15; correct: 15; wrong: 0; unknown wrong: 0
Total tags correct: 251571; wrong: 4970; unknown wrong: 0
para/DT penduduk/NN setempat/NN mengatakan/VB ./Z para/DT anggota/NN pasukan/NN pimpinan/NN hamas/NNP telah/MD mengepung/VB rumah/NN aktivis/NN tersebut/PR dan/CC kemudia
n/CC menyerang/VB -nya/PRP ./Z
Sentence number: 10027; length 20; correct: 20; wrong: 0; unknown wrong: 0
Total tags correct: 251591; wrong: 4970; unknown wrong: 0
Word: dengan; correct: SC; guessed: IN
mereka/PRP mengajukan/VB pertanyaan/NN kepada/IN -nya/PRP berkaitan/VB dengan/IN peledakan/NN bom/NN pinggir/NN jalan/NN ./Z
Sentence number: 10028; length 12; correct: 11; wrong: 1; unknown wrong: 0
Total tags correct: 251602; wrong: 4971; unknown wrong: 0
ledakan/NN bom/NN meninggalkan/VB kedua/CD anggota/NN pasukan/NN pimpinan/NN hamas/NNP itu/PR dalam/IN keadaan/NN kondisi/NN kritis/JJ dan/CC tujuh/CD lain/JJ -nya/PRP ya
ng/SC menderita/VB luka-luka/NN ./Z kata/VB para/DT pejabat/NN rumah sakit/NN ./Z
Sentence number: 10029; length 26; correct: 26; wrong: 0; unknown wrong: 0
Total tags correct: 251628; wrong: 4971; unknown wrong: 0
Word: orang; correct: NND; guessed: NN
tiga/CD orang/NN yang/SC berdiri/VB di/IN dekat/JJ kejadian/NN ./Z termasuk/VB dua/CD orang/NN anak/NN ./Z di/IN antara/NN yang/SC menderita/VB cedera/JJ ./Z ujar/VB -nya
/PRP menambahkan/VB ./Z
Sentence number: 10030; length 23; correct: 22; wrong: 1; unknown wrong: 0
Total tags correct: 251650; wrong: 4972; unknown wrong: 0
Tagged 256622 words at 48657.94 words per second.
Model postagger/bahasa-basic.tagger has xSize=212133, ySize=24, and numFeatures=62822.
Results on 10030 sentences and 256622 words, of which 0 were unknown.
Total sentences right: 6956 (69.351944%); wrong: 3074 (30.648056%).
Total tags right: 251650 (90.062520%); wrong: 4972 (1.927480%).
```

Gambar 4.7 Keluaran CLI dari pengujian model POS Tagger

4.1.3. Layanan Eksternal

Program yang dikembangkan dalam penelitian ini membutuhkan beberapa layanan eksternal. Berikut adalah layanan-layanan eksternal yang digunakan oleh program:

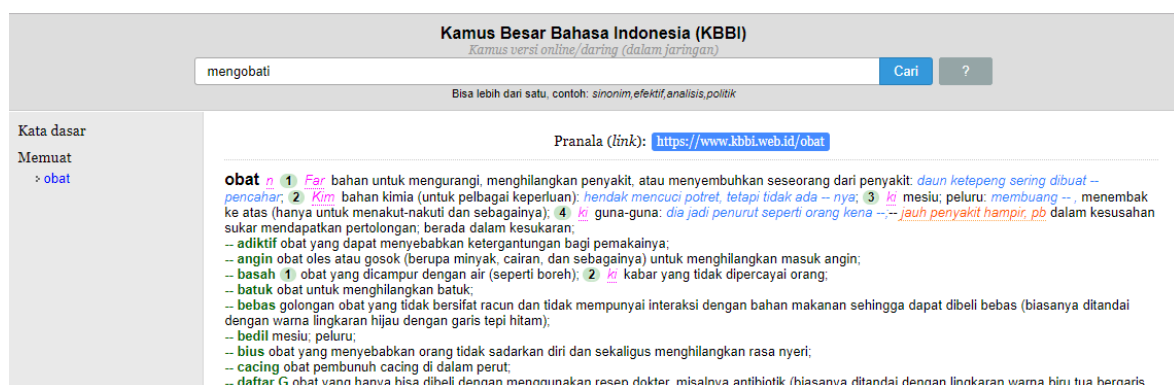
a. Mesin Pencari

Dalam penelitian ini, mesin pencari digunakan sebagai sumber korpus utama. Hasil pencarian dari mesin pencari diolah menjadi bentuk korpus yang berisi kumpulan kalimat. Mesin pencari yang digunakan dalam penelitian ini adalah seperti yang tertera dalam bagian 3.2.1, yaitu:

1. Google (<https://www.google.com>)
2. Bing (<https://www.bing.com>)
3. DuckDuckGo (<https://duckduckgo.com>)
4. Yahoo (<https://search.yahoo.com>)
5. Ecosia (<https://www.ecosia.org>)
6. Dogpile (<https://www.dogpile.com>)

b. Kamus Besar Bahasa Indonesia (<https://www.kbbi.web.id>)

Dalam penelitian ini, kamus digunakan untuk melakukan validasi terhadap kata kerja yang terdeteksi oleh *POS Tagger*. Program tidak memiliki kamus lokal yang dapat digunakan untuk menganalisis suatu kata, oleh karena itu kamus eksternal seperti situs Kamus Besar Bahasa Indonesia digunakan dalam program. Halaman situs Kamus Besar Bahasa Indonesia seperti terlihat pada Gambar 4.8.



Gambar 4.8 Halaman situs Kamus Besar Bahasa Indonesia

Sumber: <https://www.kbbi.web.id/>

4.1.4. Spesifikasi Alamat URL API

Program utama hanya didesain untuk melakukan komunikasi dengan pengguna melalui API. Antarmuka pengguna (*user interface*) diimplementasi secara terpisah dan akan menjadi jembatan antara pengguna dengan API program utama. Hal ini dilakukan untuk memisahkan antara *logic* (bagaimana data diproses) dengan *presentation* (bagaimana data disajikan). Berikut adalah daftar alamat URL API yang disediakan oleh program:

- a. Submit *request* pembelajaran: [POST http://\[HOST\]:9090/Intro/submit](http://[HOST]:9090/Intro/submit)

Digunakan untuk menerima *request* pembelajaran relasi non-taksonomi. Spesifikasi data yang harus dikirim bersama dengan *request* adalah sebagai berikut:

1. [title]: Judul *request* sebagai identifikasi bagi pengguna
2. [description]: Deskripsi mengenai *request*
3. [data]: Data berupa pasangan data kelas yang akan diproses (.csv atau .tsv)
4. [separator]: Karakter pemisah dalam file data (tab, koma, dan lain-lain)
5. [corpusCount]: Jumlah ukuran maksimal korpus (kalimat) per data yang diinginkan
6. [corpusMethod]: Menggunakan korpus hanya dari cache, mesin pencari, atau keduanya
7. [constructionSearchEngineIDs]: Mesin pencari yang ingin digunakan dalam tahap pembangunan korpus
8. [posTaggerID]: Model POS Tagger yang ingin digunakan dalam tahap anotasi korpus
9. [extractionMethods]: Metode penilaian yang ingin digunakan (MAANS, MFLR) dalam tahap ekstraksi kata
10. [evaluationSearchEngineIDs]: Mesin pencari yang ingin digunakan dalam tahap evaluasi kata

- b. Daftar hasil pembelajaran terdahulu: [GET http://\[HOST\]:9090/Intro/processes](http://[HOST]:9090/Intro/processes)

Digunakan untuk meminta daftar semua hasil proses pembelajaran relasi non-taksonomi yang pernah dilakukan.

- c. Rincian hasil pembelajaran tertentu: [GET http://\[HOST\]:9090/Intro/process/{ID}](http://[HOST]:9090/Intro/process/{ID})

Digunakan untuk meminta rincian hasil proses pembelajaran relasi non-taksonomi tertentu berdasarkan ID prosesnya. ID proses diberikan pada bagian URL.

- d. Download ontologi: [GET http://\[HOST\]:9090/download/ontology/{FILE}](http://[HOST]:9090/download/ontology/{FILE})

Digunakan untuk mengunduh dokumen ontologi (.owl) yang dihasilkan dari hasil proses pembelajaran relasi non-taksonomi tertentu berdasarkan nama *file*-nya. Nama *file* diberikan pada bagian URL.

4.2. Implementasi Antarmuka Pengguna

Antarmuka pengguna (*user interface*) diimplementasikan untuk menyajikan data kepada pengguna dalam bentuk yang mudah dibaca dan dipahami. Selain itu, antarmuka juga memudahkan pengguna untuk mengirimkan data ke program utama. Yang perlu diperhatikan dalam implementasi antarmuka pengguna yaitu kebutuhan perangkat lunak seperti bahasa pemrograman yang digunakan serta *library* yang dibutuhkan.

4.2.1. Kebutuhan Perangkat Lunak Antarmuka Pengguna

Berikut adalah bahasa pemrograman yang digunakan, *library* yang dibutuhkan, dan perangkat lunak eksternal yang digunakan oleh antarmuka pengguna:

a. Bahasa pemrograman: TypeScript

TypeScript merupakan bahasa pemrograman turunan JavaScript, atau lebih tepatnya bahasa pemrograman yang keluarannya adalah kompilasi ke dalam bentuk kode program JavaScript (*transpile*). Bahasa pemrograman ini digunakan karena *framework* yang digunakan untuk mengembangkan antarmuka adalah Angular 6 yang menggunakan TypeScript sebagai bahasa pemrogramannya. Program node.js dibutuhkan untuk mendukung penggunaan program berbasis JavaScript di lingkungan server.

b. Package manager: Yarn

Yarn adalah salah satu *package manager* yang digunakan oleh pengguna node.js untuk memudahkan pengunduhan dan penggunaan *library* Javascript.

c. Framework: Angular 6

Framework Angular 6 digunakan untuk memudahkan pengembangan antarmuka yang interaktif. Angular memberikan layanan antarmukanya melalui web (HTTP). Dalam implementasi program, Angular digunakan untuk membuat program antarmuka yang dapat berkomunikasi dengan program utama.

4.2.2. Tampilan Antarmuka Pengguna

Antarmuka pengguna diimplementasikan dengan mengikuti kebutuhan dari spesifikasi alamat URL API seperti dijelaskan pada bagian 4.1.4. Halaman yang dibuat adalah halaman Form LNTRO untuk melakukan *request* pembelajaran relasi non-taksonomi, halaman Home yang menampilkan daftar hasil pembelajaran terdahulu, dan halaman Detail yang menampilkan hasil pembelajaran tertentu secara rinci.

Halaman Home

Halaman Home adalah halaman awal yang ditampilkan kepada pengguna ketika awal program dibuka. Pada halaman ini, terdapat tautan (*link*) untuk menuju ke Form LNTRO untuk melakukan permintaan pembelajaran relasi non-taksonomi seperti terlihat pada Gambar 4.9. Selain itu, pada halaman ini pula semua hasil pembelajaran relasi non-taksonomi terdahulu ditampilkan dalam bentuk tabel seperti terlihat pada Gambar 4.10. Terdapat tombol detail untuk membuka rincian hasil pembelajaran dan tombol *download* untuk mengunduh hasil pembelajaran dalam bentuk ontologi (.owl) pada setiap baris hasil pembelajaran pada tabel.



Gambar 4.9 Halaman Home yang merupakan halaman awal program

Daftar Hasil Pembelajaran Terdahulu [Refresh](#)

Show 10 entries Search:

#	Fetch Date	Title	Action
9	13 December 2018 12:38:34	Pengujian 1.a : [herbal_penyakit] => Google	Detail Download
8	13 December 2018 05:25:06	Pengujian 4.b : [tempat_wisata] => All	Detail Download
7	13 December 2018 05:23:21	Pengujian 4.a : [herbal_penyakit] => All	Detail Download
6	13 December 2018 05:21:17	Pengujian 3.b : [tempat_wisata] => Ecosia	Detail Download
5	13 December 2018 05:20:35	Pengujian 3.a : [herbal_penyakit] => Ecosia	Detail Download
4	12 December 2018 17:35:25	Pengujian 2.b : [tempat_wisata] => Yahoo	Detail Download
3	12 December 2018 17:34:52	Pengujian 2.a : [herbal_penyakit] => Yahoo	Detail Download
2	12 December 2018 17:33:57	Pengujian 1.b : [tempat_wisata] => Google	Detail Download
1	12 December 2018 17:29:44	Pengujian 1.a : [herbal_penyakit] => Google	Detail Download

Showing 1 to 9 of 9 entries Previous 1 Next

Elang Cergas Pembrani (2018)

Gambar 4.10 Daftar hasil pembelajaran relasi non-taksonomi terdahulu

Halaman Form LNTRO

Halaman Form LNTRO adalah halaman utama yang digunakan untuk melakukan permintaan (*request*) pembelajaran relasi non-taksonomi ke program utama. Dari halaman ini, pengguna dapat memasukkan masukan (*input*) yang dibutuhkan oleh program serta menentukan parameter pembelajaran. Masukan dan parameter yang ada dalam form sama seperti masukan yang dibutuhkan oleh API program utama, yaitu:

- a. Input Data
 - Data: Data berupa pasangan data kelas yang akan diproses (.csv atau .tsv)
 - Separator: Karakter pemisah dalam file data (tab, koma, dan lain-lain)
 - Data Title: Judul *request* sebagai identifikasi bagi pengguna
 - Data Description: Deskripsi mengenai *request*
- b. Corpus Construction
 - Corpus Count: Jumlah ukuran maksimal korpus (kalimat) per data yang diinginkan
 - Corpus Method: Menggunakan korpus hanya dari *cache*, mesin pencari, atau keduanya
 - Search Engine Sources: Mesin pencari yang ingin digunakan dalam tahap pembangunan korpus
- c. Corpus Annotation
 - POS Tagger Model: Model POS Tagger yang ingin digunakan dalam tahap anotasi korpus (ada beberapa model dengan sedikit perbedaan parameter pelatihan model)
- d. Relationship Extraction
 - Extraction Methods: Metode penilaian yang ingin digunakan (MAANS, MFLR) dalam tahap ekstraksi kata
- e. Relationship Evaluation
 - Evaluation Search Engines: Mesin pencari yang ingin digunakan dalam tahap evaluasi kata

Halaman Form LNTRO bagian Input Data dapat diamati pada Gambar 4.11 dan bagian parameter pembelajaran pada Gambar 4.12.

LNTRO Bahasa
Home
Form LNTRO

LNTRO Form

Input Data

Data (TSV / CSV) tanaman_herbal_penyakit_7.tsv

Separator

Data Title

Data Description

Corpus Construction

Corpus Count Per Search Engine x 1 Search Engine : 300 Total

Construction Method

Gambar 4.11 Halaman Form LNTRO (a)

Corpus Construction

Corpus Count Per Search Engine x 1 Search Engine : 300 Total

Construction Method

Search Engine Sources ☒ Google ☐ DuckDuckGo ☐ Bing ☐ Yahoo ☐ Dogpile ☐ Ecosia

Corpus Annotation

POS Tagger Model

Relationship Extraction

Extraction Methods ☒ [MAANS] Maximum of Absolute Average of Normalized Scores ☒ [MFLR] Most Frequent Local Recommendation

Relationship Evaluation

Evaluation Search Engines ☒ Google ☐ Yahoo ☐ Ecosia

Elang Cergas Pembrani (2018)

Gambar 4.12 Halaman Form LNTRO (b)

Halaman Detail

Halaman Detail berisi rincian proses pembelajaran yang menampilkan hasil dari langkah-langkah besar dalam proses pembelajaran relasi non-taksonomi, yaitu parameter pembelajaran, data yang dimasukkan, tahap pembangunan korpus (*corpus construction*), tahap anotasi korpus (*corpus annotation*), tahap ekstraksi kata (tabel frekuensi kata dan tabel skor), tahap pengembangan kata (hasil pemilihan nama relasi), dan tambahan evaluasi kata (data sampel dan hasil evaluasi).

Parameter pembelajaran dan deskripsi yang dimasukkan oleh pengguna saat awal *request* pembelajaran relasi non-taksonomi dapat diamati pada Gambar 4.13.

[LNTRO Bahasa](#) [Home](#) [Form LNTRO](#)

LNTRO Process Detail - 16 December 2018 10:48:00

Pengujian 1.a : [herbal_penyakit] => Google

Pengujian 1.a pada data [herbal_penyakit] dengan fokus mesin pencari Google.

Params

Corpus Count	300
Construction Search Engines	Google
POS Tagger Model	Basic
Extraction Methods	[MAANS] Maximum of Absolute Average of Normalized Scores, [MFLR] Most Frequent Local Recommendation
Evaluation Search Engines	Google

Output

Data
Search Results
Corpus

Gambar 4.13 Deskripsi dan parameter hasil pembelajaran relasi non-taksonomi

Daftar keluaran yang dihasilkan oleh proses pembelajaran relasi non-taksonomi dapat diamati pada Gambar 4.14.

Output

Data
Search Results
Corpus
Annotated Corpus
Term Frequency Maps
Extraction Scores
Top Extraction Terms
Evaluation Data Sample
Evaluation Scores

[Download Ontology \(OWL\)](#)

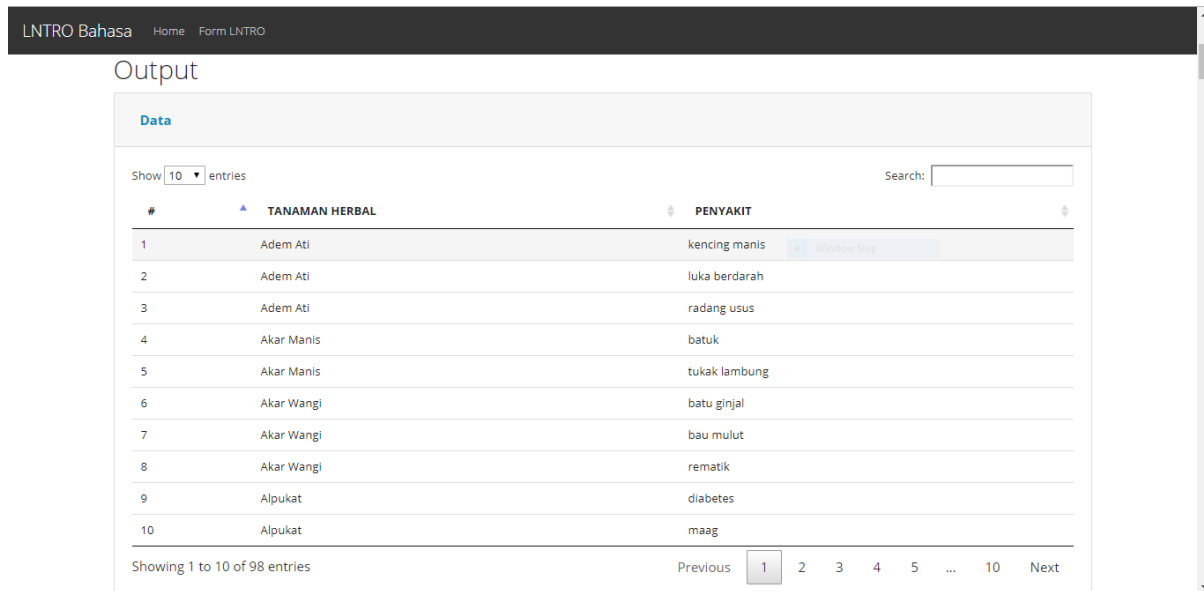
[Download Ontology \(OWL\) using other term](#)

Benchmark

Input Data	00:00:00.003
Corpus Construction	00:51:33.614
Corpus Annotation	00:00:08.228
LNTRO Extraction	00:00:29.903
LNTRO Evaluation	00:20:46.722
Full Process	01:12:58.470

Gambar 4.14 Rangkuman keluaran dari hasil pembelajaran relasi non-taksonomi

Data yang dimasukkan oleh pengguna dapat diamati dalam bentuk tabel seperti terlihat pada Gambar 4.15.

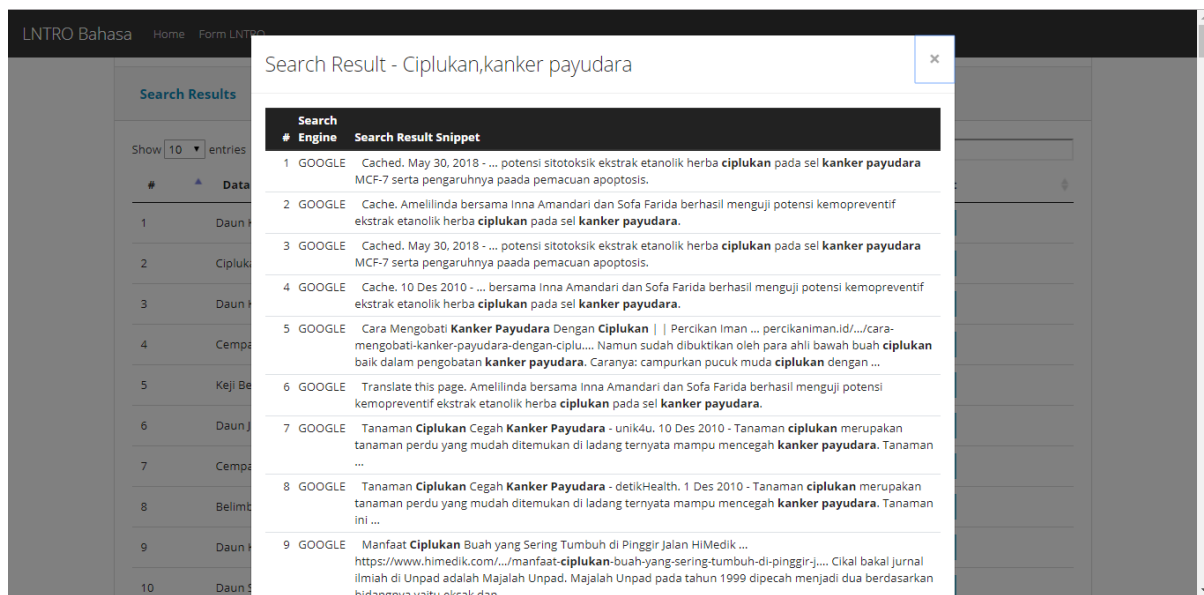


The screenshot shows the 'Output' section of the LNTRO Bahasa application. It features a table with two columns: 'TANAMAN HERBAL' and 'PENYAKIT'. The table contains 10 entries, numbered 1 to 10. The first entry is 'Adem Ati' for 'kencing manis'. The second entry is 'Adem Ati' for 'luka berdarah'. The third entry is 'Adem Ati' for 'radang usus'. The fourth entry is 'Akar Manis' for 'batuk'. The fifth entry is 'Akar Manis' for 'tukak lambung'. The sixth entry is 'Akar Wangi' for 'batu ginjal'. The seventh entry is 'Akar Wangi' for 'bau mulut'. The eighth entry is 'Akar Wangi' for 'rematik'. The ninth entry is 'Alpukat' for 'diabetes'. The tenth entry is 'Alpukat' for 'maag'. The table is displayed with 10 entries per page, and there are search and pagination controls.

#	TANAMAN HERBAL	PENYAKIT
1	Adem Ati	kencing manis
2	Adem Ati	luka berdarah
3	Adem Ati	radang usus
4	Akar Manis	batuk
5	Akar Manis	tukak lambung
6	Akar Wangi	batu ginjal
7	Akar Wangi	bau mulut
8	Akar Wangi	rematik
9	Alpukat	diabetes
10	Alpukat	maag

Gambar 4.15 Data masukan pembelajaran yang disajikan dalam bentuk tabel

Hasil proses pencarian pada mesin pencari dapat diamati pada Gambar 4.16. Setiap baris merepresentasikan setiap butir hasil pencarian yang dihasilkan oleh data tersebut.

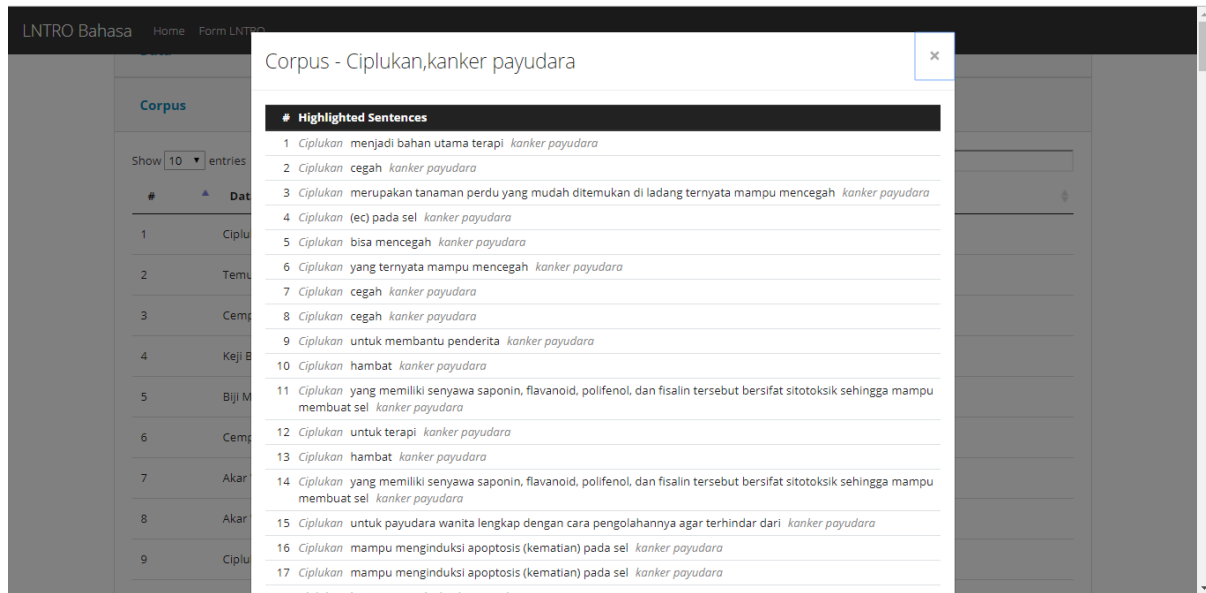


The screenshot shows the 'Search Results' section of the LNTRO Bahasa application. It displays a list of search results for the query 'Ciplukan, kanker payudara'. The results are numbered 1 to 9, each showing the search engine (GOOGLE), the date (May 30, 2018), and a snippet of the search result. The snippets describe the potential of 'ciplukan' (a type of herb) in treating breast cancer (kanker payudara).

#	Engine	Search Result Snippet
1	GOOGLE	Cached, May 30, 2018 - ... potensi sitotoksik ekstrak etanolik herba ciplukan pada sel kanker payudara MCF-7 serta pengaruhnya paada pemacuan apoptosis.
2	GOOGLE	Cache, Amellinda bersama Inna Amandari dan Sofa Farida berhasil menguji potensi kemopreventif ekstrak etanolik herba ciplukan pada sel kanker payudara .
3	GOOGLE	Cached, May 30, 2018 - ... potensi sitotoksik ekstrak etanolik herba ciplukan pada sel kanker payudara MCF-7 serta pengaruhnya paada pemacuan apoptosis.
4	GOOGLE	Cache, 10 Des 2010 - ... bersama Inna Amandari dan Sofa Farida berhasil menguji potensi kemopreventif ekstrak etanolik herba ciplukan pada sel kanker payudara .
5	GOOGLE	Cara Mengobati Kanker Payudara Dengan Ciplukan Percikan Iman ... percikaniman.id/.../cara-mengobati-kanker-payudara-dengan-ciplu.... Namun sudah dibuktikan oleh para ahli bawah buah ciplukan baik dalam pengobatan kanker payudara . Caranya: campurkan pucuk muda ciplukan dengan ...
6	GOOGLE	Translate this page. Amellinda bersama Inna Amandari dan Sofa Farida berhasil menguji potensi kemopreventif ekstrak etanolik herba ciplukan pada sel kanker payudara .
7	GOOGLE	Tanaman Ciplukan Cegah Kanker Payudara - unik4u. 10 Des 2010 - Tanaman ciplukan merupakan tanaman perdu yang mudah ditemukan di ladang ternyata mampu mencegah kanker payudara . Tanaman ini ...
8	GOOGLE	Tanaman Ciplukan Cegah Kanker Payudara - detikHealth. 1 Des 2010 - Tanaman ciplukan merupakan tanaman perdu yang mudah ditemukan di ladang ternyata mampu mencegah kanker payudara . Tanaman ini ...
9	GOOGLE	Manfaat Ciplukan Buah yang Sering Tumbuh di Pinggir Jalan HiMedik ... https://www.himedik.com/.../manfaat-ciplukan-buah-yang-sering-tumbuh-di-pinggir-j... Cikal bakal jurnal ilmiah di Unpad adalah Majalah Unpad. Majalah Unpad pada tahun 1999 dipecah menjadi dua berdasarkan hiranannya yaitu eksek dan ...

Gambar 4.16 Hasil pencarian data masukan pada mesin pencari

Hasil proses pembangunan korpus dapat diamati pada Gambar 4.17. Setiap baris merepresentasikan setiap data beserta tabel korpus yang dihasilkan oleh data tersebut.

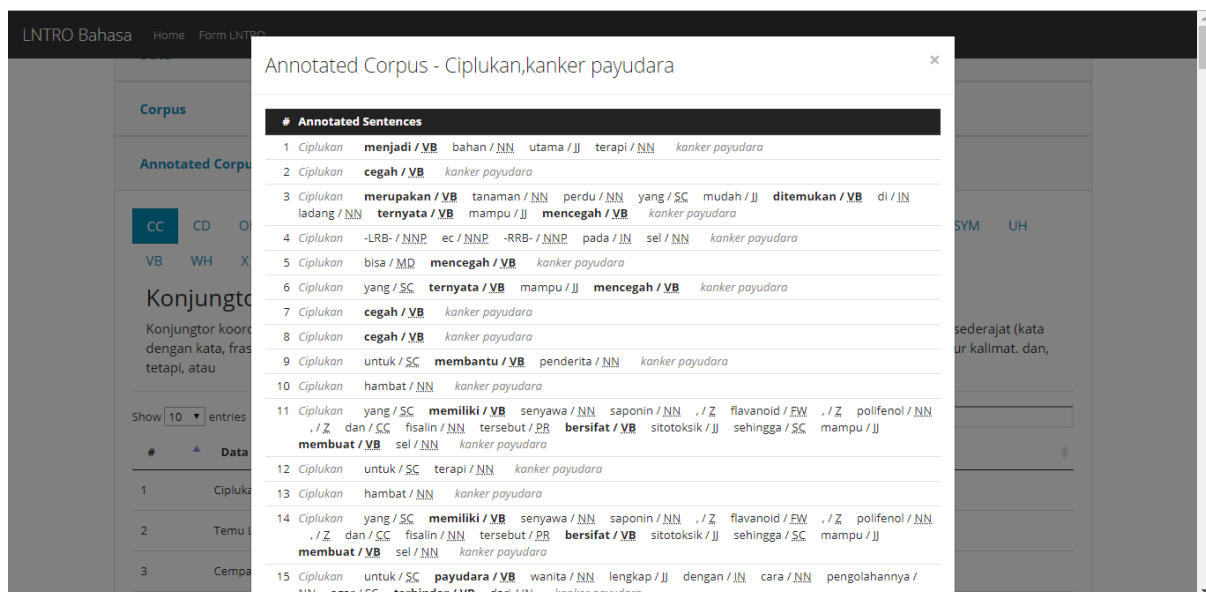


The screenshot shows a web application titled "LNTRO Bahasa" with a sidebar menu containing "Corpus" and "Annotated Corpus". The main content area displays a table of sentences under the heading "Corpus - Ciplukan, kanker payudara". The table has a "Highlighted Sentences" section and a list of 17 sentences, each with a number and a description of the sentence's content.

#	Highlighted Sentences
1	Ciplukan menjadi bahan utama terapi kanker payudara
2	Ciplukan cegah kanker payudara
3	Ciplukan merupakan tanaman perdu yang mudah ditemukan di ladang ternyata mampu mencegah kanker payudara
4	Ciplukan (ec) pada sel kanker payudara
5	Ciplukan bisa mencegah kanker payudara
6	Ciplukan yang ternyata mampu mencegah kanker payudara
7	Ciplukan cegah kanker payudara
8	Ciplukan cegah kanker payudara
9	Ciplukan untuk membantu penderita kanker payudara
10	Ciplukan hambat kanker payudara
11	Ciplukan yang memiliki senyawa saponin, flavanoid, polifenol, dan fisalin tersebut bersifat sitotoksik sehingga mampu membuat sel kanker payudara
12	Ciplukan untuk terapi kanker payudara
13	Ciplukan hambat kanker payudara
14	Ciplukan yang memiliki senyawa saponin, flavanoid, polifenol, dan fisalin tersebut bersifat sitotoksik sehingga mampu membuat sel kanker payudara
15	Ciplukan untuk payudara wanita lengkap dengan cara pengolahannya agar terhindar dari kanker payudara
16	Ciplukan mampu menginduksi apoptosis (kematian) pada sel kanker payudara
17	Ciplukan mampu menginduksi apoptosis (kematian) pada sel kanker payudara

Gambar 4.17 Hasil pembangunan korpus melalui mesin pencari setelah diolah

Hasil proses anotasi korpus dapat diamati pada Gambar 4.18. Setiap baris merepresentasikan setiap data beserta hasil *postagging* dari korpus. Setiap kemunculan kata kerja dicetak tebal.

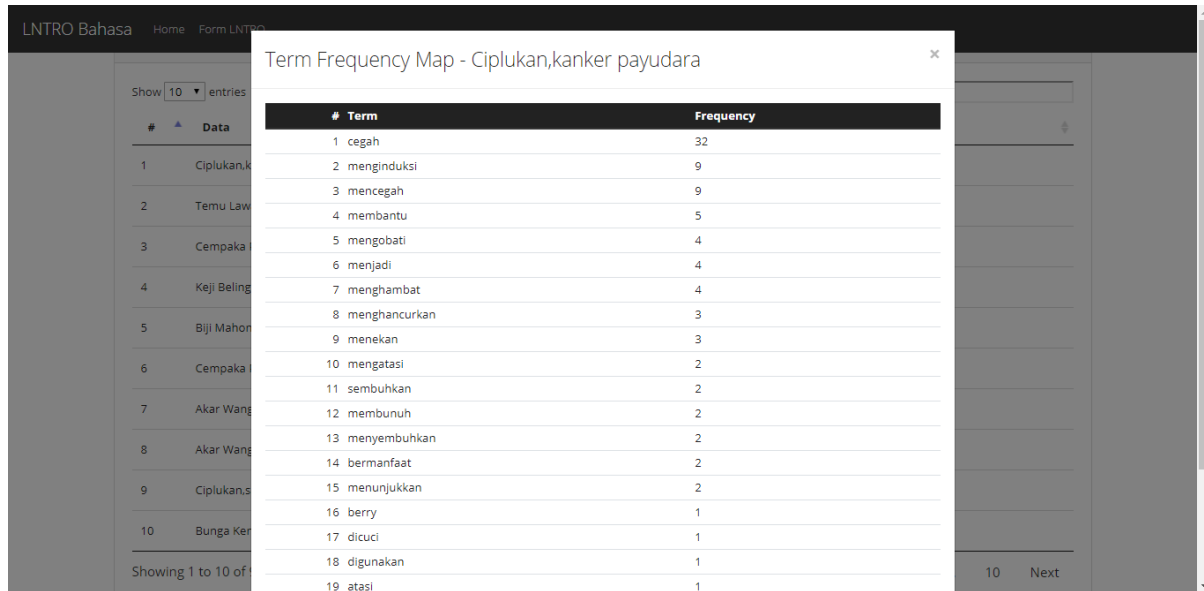


The screenshot shows the "Annotated Corpus" view of the same data. Each sentence is now annotated with Part-of-Speech (POS) tags and verbs are bolded. The table structure is similar to the one in Gambar 4.17, but the content of the sentences is annotated.

#	Annotated Sentences
1	Ciplukan menjadi / VB bahan / NN utama / JJ terapi / NN kanker payudara
2	Ciplukan cegah / VB kanker payudara
3	Ciplukan merupakan / VB tanaman / NN perdu / NN yang / SC mudah / JJ ditemukan / VB di / IN ladang / NN ternyata / VB mampu / JJ mencegah / VB kanker payudara
4	Ciplukan -LRB- / NNP ec / NNP -RRB- / NNP pada / IN sel / NN kanker payudara
5	Ciplukan bisa / MD mencegah / VB kanker payudara
6	Ciplukan yang / SC ternyata / VB mampu / JJ mencegah / VB kanker payudara
7	Ciplukan cegah / VB kanker payudara
8	Ciplukan cegah / VB kanker payudara
9	Ciplukan untuk / SC membantu / VB penderita / NN kanker payudara
10	Ciplukan hambat / NN kanker payudara
11	Ciplukan yang / SC memiliki / VB senyawa / NN saponin / NN , / Z flavanoid / EW , / Z polifenol / NN , / Z dan / CC fisalin / NN tersebut / PB bersifat / VB sitotoksik / JJ sehingga / SC mampu / JJ membuat / VB sel / NN kanker payudara
12	Ciplukan untuk / SC terapi / NN kanker payudara
13	Ciplukan hambat / NN kanker payudara
14	Ciplukan yang / SC memiliki / VB senyawa / NN saponin / NN , / Z flavanoid / EW , / Z polifenol / NN , / Z dan / CC fisalin / NN tersebut / PB bersifat / VB sitotoksik / JJ sehingga / SC mampu / JJ membuat / VB sel / NN kanker payudara
15	Ciplukan untuk / SC payudara / VB wanita / NN lengkap / JJ dengan / IN cara / NN pengolahannya / NN agar / SC terhindar / VB dari / IN kanker payudara

Gambar 4.18 Hasil anotasi korpus oleh POS Tagger beserta informasi tag

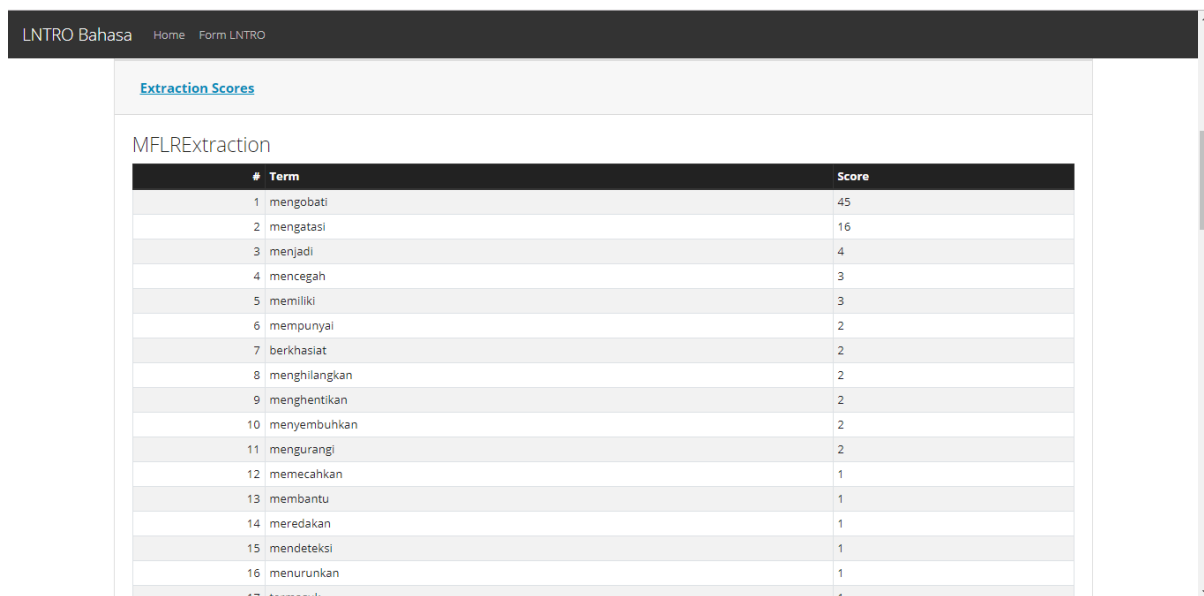
Hasil perhitungan frekuensi kemunculan kata dapat diamati pada Gambar 4.19. Setiap baris merepresentasikan setiap data beserta daftar jumlah kemunculan setiap kata kerja yang dihasilkan oleh korpus dari data tersebut.



#	Term	Frequency
1	cegah	32
2	menginduksi	9
3	mencegah	9
4	membantu	5
5	mengobati	4
6	menjadi	4
7	menghambat	4
8	menghancurkan	3
9	menekan	3
10	mengatasi	2
11	sembuhkan	2
12	membunuh	2
13	menyembuhkan	2
14	bermanfaat	2
15	menunjukkan	2
16	berry	1
17	dicuci	1
18	digunakan	1
19	atasi	1

Gambar 4.19 Hasil perhitungan frekuensi kemunculan kata

Hasil perhitungan skor akhir menggunakan metode MFLR (*Most Frequent Local Recommendation*) dapat diamati pada Gambar 4.20. Kata kerja diurutkan dari skor tertinggi ke skor terendah.



#	Term	Score
1	mengobati	45
2	mengatasi	16
3	menjadi	4
4	mencegah	3
5	memiliki	3
6	mempunyai	2
7	berkhasiat	2
8	menghilangkan	2
9	menghentikan	2
10	menyembuhkan	2
11	mengurangi	2
12	memecahkan	1
13	membantu	1
14	meredakan	1
15	mendeteksi	1
16	menurunkan	1
17	termasuk	1

Gambar 4.20 Hasil perhitungan skor akhir menggunakan metode MFLR

Hasil perhitungan skor akhir menggunakan metode MAANS (*Maximum of Absolute Average of Normalized Score*) dapat diamati pada Gambar 4.21. Kata kerja diurutkan dari skor tertinggi ke skor terendah.

LNTRO Bahasa Home Form LNTRO

MAANSExtraction

#	Term	Score
1	mengobati	0.5033527696793003
2	mengatasi	0.20488816738816737
3	menjadi	0.04846938775510204
4	menyembuhkan	0.04076479076479076
5	mencegah	0.03401360544217687
6	menghilangkan	0.030612244897959183
7	mengurangi	0.02806122448979592
8	berkhasiat	0.025510204081632654
9	mempunyai	0.022448979591836737
10	menghentikan	0.021865889212827987
11	meredakan	0.02074829931972789
12	memiliki	0.019742219232015152
13	bermanfaat	0.012244897959183673
14	memecahkan	0.01020408163265306
15	membantu	0.01020408163265306
16	mendeteksi	0.01020408163265306
17	menderita	0.01020408163265306
18	punya	0.01020408163265306
19	muntah	0.01020408163265306

Gambar 4.21 Hasil perhitungan skor akhir menggunakan metode MAANS

Hasil pemilihan nama relasi non-taksonomi dapat diamati pada Gambar 4.22. Baris paling atas dicetak dengan warna yang berbeda dan merupakan nama relasi yang dipilih oleh program.

LNTRO Bahasa Home Form LNTRO

[Top Extraction Terms](#)

MFLRExtraction

Hasil Ekstraksi	Kata Dasar	Kata Kerja Aktif	Kata Kerja Pasif
mengobati	obat	mengobati	diobati
mengatasi	atas	mengatasi	diatasi
mencegah	cegah	mencegah	dicegah
menjadi	jadi	menjadi	dijadi
menghilangkan	hilang	menghilangkan	dihilangkan

MAANSExtraction

Hasil Ekstraksi	Kata Dasar	Kata Kerja Aktif	Kata Kerja Pasif
mengobati	obat	mengobati	diobati
mengatasi	atas	mengatasi	diatasi
menyembuhkan	sembuh	menyembuhkan	disembuhkan
menjadi	jadi	menjadi	dijadi
menghilangkan	hilang	menghilangkan	dihilangkan

Blank Diagram (1).png Blank Diagram.png Blank Diagram.pdf LNTRO Bahasa (8).png LNTRO Bahasa (7).png Show all

Gambar 4.22 Hasil pemilihan nama relasi non-taksonomi

Data sampel yang digunakan pada tahap tambahan evaluasi kata dapat diamati pada Gambar 4.23. Jumlah data sampel yang digunakan maksimal berjumlah 10 data.

#	TANAMAN HERBAL	PENYAKIT
1	Cempaka Putih	sinusitis
2	Cempaka Putih	batuk rejan
3	Biji Mahoni	masuk angin
4	Daun Jinten	perut kembung
5	Daun Beluntas	perut kembung
6	Daun Cincau	demam
7	Daun Cincau	diabetes milltus
8	Kumis Kucing	diabetes
9	Bunga Kenop	batuk
10	Daun Dewa	pendarahan

Gambar 4.23 Data sampel yang dipakai pada saat evaluasi

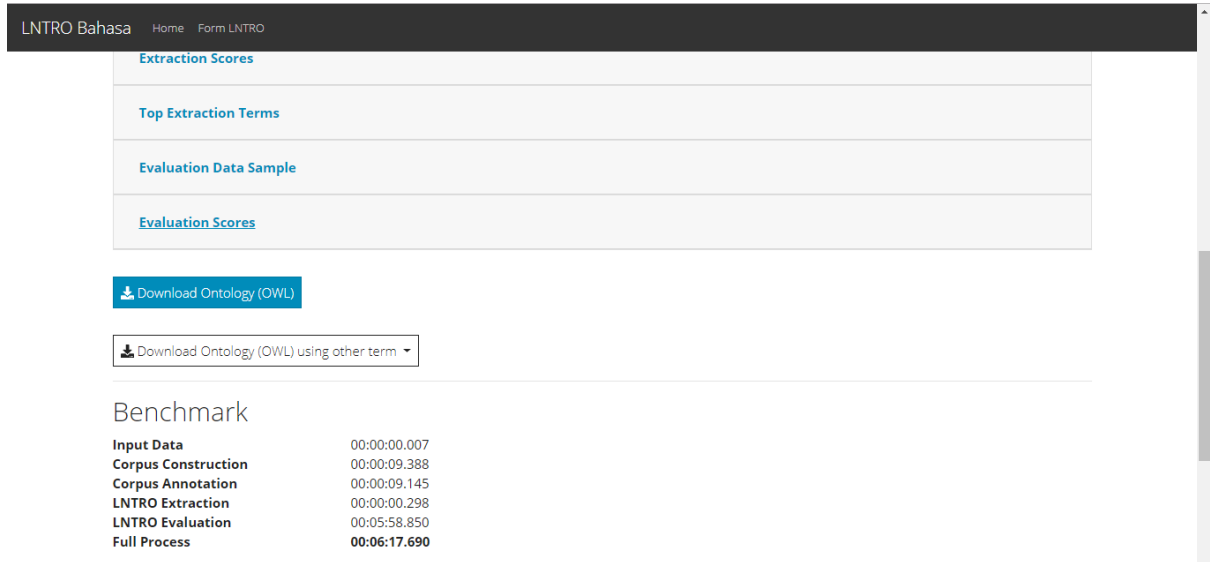
Hasil evaluasi kata yang dilakukan dengan menggunakan mesin pencari dapat diamati pada Gambar 4.24. Jumlah *hit* dari setiap mesin pencari akan ditampilkan dan keakuratan hasil pemilihan nama relasi diserahkan sepenuhnya kepada kebijaksanaan pengguna.

Evaluated Terms	Google	Yahoo	Ecosia	Total
mengobati	35.089	22.350	9.553	66.992
mengatasi	232	30	3.822	4.084
mencegah	18	20	3.912	3.950
mengurangi	1	0	42	43
berkhasiat	7	0	1	8

Evaluated Terms	Google	Yahoo	Ecosia	Total
mengobati	35.089	22.350	9.553	66.992
mengatasi	232	30	3.822	4.084
mencegah	18	20	3.912	3.950
menyembuhkan	140	0	3.238	3.378
menjadi	38	0	211	249

Gambar 4.24 Hasil evaluasi kata dengan pencarian melalui mesin pencari

Untuk mengunduh hasil pembelajaran dalam bentuk ontologi (.owl), pengguna dapat menggunakan tombol *download* seperti terlihat pada Gambar 4.25. Pada bagian paling akhir halaman juga ditampilkan informasi *benchmark* proses pembelajaran.



The screenshot shows the LNTRO Bahasa web interface. At the top, there is a dark navigation bar with 'LNTRO Bahasa', 'Home', and 'Form LNTRO'. Below this, a sidebar on the left contains links: 'Extraction Scores', 'Top Extraction Terms', 'Evaluation Data Sample', and 'Evaluation Scores'. The main content area features a blue button labeled 'Download Ontology (OWL)' with a download icon. Below it is a dropdown menu labeled 'Download Ontology (OWL) using other term'. The bottom section, titled 'Benchmark', displays a table of performance metrics.

Benchmark	
Input Data	00:00:00.007
Corpus Construction	00:00:09.388
Corpus Annotation	00:00:09.145
LNTRO Extraction	00:00:00.298
LNTRO Evaluation	00:05:58.850
Full Process	00:06:17.690

Gambar 4.25 Tombol *download* ontologi dan informasi *benchmark*

BAB V

HASIL DAN PEMBAHASAN

5.1. Metodologi Pengujian

Pada bagian metodologi pengujian ini akan dipaparkan bagaimana peneliti melakukan pengujian terhadap program dan usulan solusi yang sudah dibuat. Metodologi pengujian terdiri dari tiga bagian, yaitu langkah pengujian, data yang diujikan, dan parameter pembelajaran.

5.1.1. Langkah Pengujian

Pengujian diintegrasikan pada program dengan adanya tahap tambahan, yaitu tahap evaluasi kata. Tujuan diintegrasikannya proses pengujian dengan kerangka alur kerja pembelajaran relasi non-taksonomi adalah supaya proses pengujian dapat dilakukan secara otomatis. Hasil dari tahap evaluasi ini tidak mempengaruhi hasil pembelajaran relasi non-taksonomi yang didapatkan, namun keluaran yang dihasilkan akan diamati oleh peneliti untuk mengukur tingkat keberhasilan usulan solusi yang dirancang.

Tahap evaluasi kata secara otomatis telah dipaparkan pada bagian 3.6 dalam laporan penelitian ini. Tahap tersebut menguji lima (5) kata kerja dengan skor tertinggi, yang kemudian dikombinasikan dengan 10 data sampel untuk dilakukan pencarian ulang melalui mesin pencari. Hasilnya adalah jumlah *hit* setiap kata kunci kombinasi antara data dengan kata kerja. Hasil tersebut kemudian ditampilkan melalui antarmuka pengguna. Tahap selanjutnya diserahkan kepada kebijaksanaan peneliti. Secara sederhana keakuratan hasil pembelajaran ditentukan dengan cara melihat urutan jumlah *hit* dari kelima kata kerja, dengan memeriksa apakah kata kerja yang dihasilkan oleh program memiliki jumlah *hit* tertinggi atau tidak.

Program dan usulan solusi dianggap lolos pengujian apabila kata kerja nama relasi non-taksonomi yang dihasilkan dan dipilih dianggap relevan dan cocok dengan pasangan data yang dimasukkan. Apabila nama relasi terpilih juga memiliki jumlah *hit* tertinggi, maka hal tersebut akan menjadi nilai tambah bagi program. Namun apabila nama relasi terpilih tidak memiliki jumlah *hit* tertinggi (atau dalam kasus terburuk, jumlah *hit*-nya paling rendah dibanding kandidat lain) maka disimpulkan bahwa solusi yang ditawarkan belum sempurna. Dalam kasus tersebut, kemungkinan ada beberapa faktor yang berpengaruh sehingga hasilnya kurang tepat, atau ada kesalahan asumsi dari peneliti sendiri sehingga solusi yang ditawarkan belum akurat.

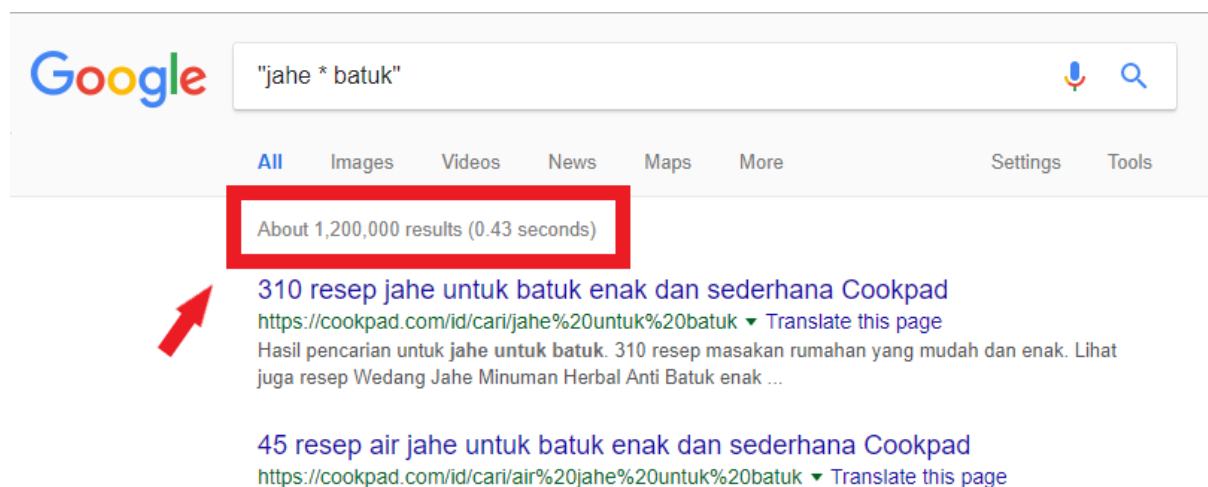
5.1.2. Data Uji

Untuk menguji hasil pembelajaran relasi non-taksonomi, peneliti menggunakan dua jenis pasangan data kelas yang berbeda. Kedua data tersebut kemudian dimasukkan ke dalam program sebagai masukan pembelajaran. Berikut adalah deskripsi dari kedua data yang diujikan:

- Data [herbal_penyakit]: Data pertama merupakan pasangan data tanaman herbal dengan penyakit tertentu dan terlampir pada Lampiran A. Data ini masuk ke dalam *domain* kesehatan dan farmasi. Menurut penalaran peneliti, seharusnya nama relasi yang dihasilkan oleh data ini adalah salah satu atau merupakan sinonim dari kata berikut: **mengobati, menyembuhkan, mengatasi**
- Data [tempat_wisata]: Data kedua merupakan pasangan data tempat wisata dengan provinsi dan terlampir pada Lampiran B. Data ini masuk ke dalam *domain* pariwisata. Menurut penalaran peneliti, seharusnya nama relasi yang dihasilkan oleh data ini adalah salah satu atau merupakan sinonim dari kata berikut: **terletak, berada**

5.1.3. Parameter Pembelajaran

Program yang dikembangkan memberikan beberapa pilihan sebagai parameter untuk mengatur pembelajaran relasi non-taksonomi. Pada tahap evaluasi kata otomatis oleh program, tidak semua mesin pencari yang disediakan dapat digunakan dalam proses evaluasi. Hanya mesin pencari yang menampilkan jumlah hasil pencarian saja yang dapat digunakan untuk melakukan evaluasi otomatis. Contoh jumlah hasil pencarian seperti terlihat pada Gambar 5.1.



Gambar 5.1 Jumlah *hit* hasil pencarian yang dihasilkan oleh mesin pencari

Mesin pencari yang mendukung proses evaluasi kata secara otomatis yang digunakan dalam penelitian ini hanya tiga, yaitu Google, Yahoo, dan Ecosia. Untuk ketiga mesin pencari tersebut, masing-masing dibuatkan parameter pembelajaran untuk menguji keakuratan penggunaan mesin pencari tersebut. Kemudian ada satu parameter tambahan yang menguji pembelajaran yang menggunakan semua jenis mesin pencari. Untuk itu, ada empat jenis parameter pembelajaran yang diterapkan untuk setiap data, yaitu tiga parameter pertama untuk menguji berdasarkan mesin pencari dan parameter terakhir yang menguji pembelajaran yang mengerahkan semua mesin pencari secara penuh. Dengan dua jenis data, maka total pengujian dilakukan sebanyak delapan kali pembelajaran. Kedua metode ekstraksi data digunakan dalam semua parameter pembelajaran, karena kedua metode tidak mempengaruhi satu sama lain.

Untuk POS Tagger sendiri tidak diujikan dalam penelitian ini. Meskipun dalam proses pengembangan program peneliti melatih empat model POS Tagger dengan parameter yang berbeda, namun hasil dari keempat model tersebut tidak begitu signifikan dan rata-rata mampu mengenali kata kerja dengan keakuratan yang hampir sama, sehingga peneliti merasa tidak perlu memasukkan POS Tagger sebagai variabel dalam pengujian. Keempat model tersebut hanya disediakan sebagai pilihan tambahan bagi pengguna. Dalam pengujian ini sendiri, model POS Tagger yang digunakan adalah model yang ditandai oleh peneliti sebagai model “Basic”.

Berikut adalah daftar parameter pembelajaran yang diujikan dalam pengujian program:

Tabel 5.1 Parameter pembelajaran yang diujikan

#	Corpus Count	Corpus Source	Construction Search Engine(s)	POS Tagger Model	Extraction	Evaluation Search Engine(s)	Data	ID
1	300	From Cache Then Search Engine	Google	Basic	MFLR MAANS	Google	[herbal_penyakit]	1.a
							[tempat_wisata]	1.b
2	300	From Cache Then Search Engine	Yahoo	Basic	MFLR MAANS	Yahoo	[herbal_penyakit]	2.a
							[tempat_wisata]	2.b
3	300	From Cache Then Search Engine	Ecosia	Basic	MFLR MAANS	Ecosia	[herbal_penyakit]	3.a
							[tempat_wisata]	3.b
4	600 (100 per Search Engine)	From Cache Then Search Engine	Google Bing Yahoo DuckDuckGo Ecosia Dogpile	Basic	MFLR MAANS	Google Yahoo Ecosia	[herbal_penyakit]	4.a
							[tempat_wisata]	4.b

5.2. Hasil Pengujian

Pengujian dilakukan dengan menerapkan empat parameter pembelajaran yang berbeda terhadap setiap data uji. Kedua metode ekstraksi juga digunakan dalam semua pembelajaran. Setiap hasil pembelajaran disajikan dalam empat aspek, yaitu statistik jumlah korpus yang dihasilkan setiap pasangan data, hasil ekstraksi kata metode MAANS (lima teratas), hasil ekstraksi kata metode MFLR (lima teratas), dan hasil evaluasi kata.

5.2.1. Pengujian Parameter Pembelajaran Pertama (Google)

Parameter pembelajaran pertama diujikan pada data [herbal_penyakit] dengan ID pengujian (1.a) yang dapat diamati pada Tabel 5.2 dan juga diujikan pada data [tempat_wisata] dengan ID pengujian (1.b) yang dapat diamati pada Tabel 5.3.

Tabel 5.2 Pengujian 1.a - Data [herbal_penyakit] pada Google

Statistik Jumlah Korpus	Min	11	kalimat	Median	100.5	kalimat
	Max	283	kalimat	Standar Deviasi	63.34	kalimat
	Mean	108.82	kalimat	Total	10664	kalimat
Ekstraksi Kata	MAANS			MFLR		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	mengobati	0.6713146809545524	1	mengobati	57
	2	mengatasi	0.2634664485373797	2	mengatasi	17
	3	menyembuhkan	0.0726435016678024	3	mencegah	2
	4	menurunkan	0.06270217941814842	4	meredakan	2
	5	meredakan	0.046988215156083805	5	menjadi	2
Evaluasi Kata	#	Kata Kerja	Pencarian Google			
	1	mengobati	611			
	2	mengatasi	200			
	3	mencegah	178			
	4	menyembuhkan	86			
	5	menjadi	34			
	6	menurunkan	1			
	7	meredakan	0			

Tabel 5.3 Pengujian 1.b - Data [tempat_wisata] pada Google

Statistik Jumlah Korpus	Min	39	kalimat	Median	129	kalimat
	Max	248	kalimat	Standar Deviasi	43.79	kalimat
	Mean	129.8	kalimat	Total	12331	kalimat
Ekstraksi Kata	MAANS			MFLR		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	terletak	0.22223882073605067	1	terletak	16
	2	berada	0.2031769382254147	2	berada	14
	3	ada	0.049629996042738427	3	menikmati	3
	4	menjadi	0.04145989974937344	4	menjadi	2
	5	menikmati	0.03644736842105263	5	dijual	2
Evaluasi Kata	#	Kata Kerja	Pencarian Google			
	1	terletak	683			
	2	ada	111			
	3	berada	94			
	4	menjadi	44			
	5	menikmati	15			
	6	dijual	0			

5.2.2. Pengujian Parameter Pembelajaran Kedua (Yahoo)

Parameter pembelajaran kedua diujikan pada data [herbal_penyakit] dengan ID pengujian (2.a) yang dapat diamati pada Tabel 5.4 dan juga diujikan pada data [tempat_wisata] dengan ID pengujian (2.b) yang dapat diamati pada Tabel 5.5.

Tabel 5.4 Pengujian 2.a - Data [herbal_penyakit] pada Yahoo

Statistik Jumlah Korpus	Min	0	kalimat	Median	94	kalimat
	Max	309	kalimat	Standar Deviasi	64.28	kalimat
	Mean	113.98	kalimat	Total	11170	kalimat
Ekstraksi Kata	MAANS			MFLR		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	mengobati	0.5649678420936551	1	mengobati	47
	2	mengatasi	0.2502591299701432	2	mengatasi	13
	3	menyembuhkan	0.0922895001512547	3	menyembuhkan	6
	4	menghilangkan	0.0565510490624464	4	menghilangkan	3
	5	berkhasiat	0.0472930839002268	5	berkhasiat	3
Evaluasi Kata	#	Kata Kerja	Pencarian Yahoo			
	1	berkhasiat	14.653.840			
	2	mengobati	540.572			
	3	mengatasi	252.970			
	4	menyembuhkan	177.999			
	5	menghilangkan	30.401			

Tabel 5.5 Pengujian 2.b - Data [tempat_wisata] pada Yahoo

Statistik Jumlah Korpus	Min	28	kalimat	Median	115	kalimat
	Max	296	kalimat	Standar Deviasi	56.00	kalimat
	Mean	121.99	kalimat	Total	11589	kalimat
Ekstraksi Kata	MAANS			MFLR		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	terletak	0.4644703510492983	1	terletak	34
	2	berada	0.27763302717596355	2	berada	13
	3	menjadi	0.06093807794683498	3	menjadi	3
	4	ada	0.05295815295815296	4	ada	3
	5	mengunjungi	0.03508771929824561	5	mengunjungi	2
Evaluasi Kata	#	Kata Kerja	Pencarian Yahoo			
	1	terletak	1.095.120			
	2	berada	1.079.360			
	3	mengunjungi	999.950			
	4	ada	250.945			
	5	menjadi	234.900			

5.2.3. Pengujian Parameter Pembelajaran Ketiga (Ecosia)

Parameter pembelajaran ketiga diujikan pada data [herbal_penyakit] dengan ID pengujian (3.a) yang dapat diamati pada Tabel 5.6 dan juga diujikan pada data [tempat_wisata] dengan ID pengujian (3.b) yang dapat diamati pada Tabel 5.7.

Tabel 5.6 Pengujian 3.a - Data [herbal_penyakit] pada Ecosia

Statistik Jumlah Korpus	Min	0	kalimat	Median	26	kalimat
	Max	106	kalimat	Standar Deviasi	22.12	kalimat
	Mean	25.78	kalimat	Total	2526	kalimat
Ekstraksi Kata	MAANS			MFLR		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	mengobati	0.39472789115646256	1	mengobati	38
	2	mengatasi	0.09059766763848397	2	mengatasi	7
	3	berkhasiat	0.02551020408163265	3	menjadi	2
	4	menyembuhkan	0.02414965986394558	4	berkhasiat	2
	5	menjadi	0.02040816326530612	5	memiliki	1
Evaluasi Kata	#	Kata Kerja	Pencarian Ecosia			
	1	mengobati	13.543			
	2	menyembuhkan	6.430			
	3	mengatasi	5.944			
	4	menjadi	1.122			
	5	memiliki	503			
	6	berkhasiat	93			

Tabel 5.7 Pengujian 3.b - Data [tempat_wisata] pada Ecosia

Statistik Jumlah Korpus	Min	1	kalimat	Median	31	kalimat
	Max	165	kalimat	Standar Deviasi	37.63	kalimat
	Mean	43.45	kalimat	Total	4128	kalimat
Ekstraksi Kata	MAANS			MFLR		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	terletak	0.34899749373433575	1	terletak	39
	2	berada	0.135421888053467	2	berada	15
	3	ada	0.03266081871345029	3	memiliki	4
	4	menjadi	0.03120300751879699	4	menjadi	2
	5	terkenal	0.01473684210526316	5	ada	2
Evaluasi Kata	#	Kata Kerja	Pencarian Ecosia			
	1	memiliki	1.633			
	2	terletak	1.520			
	3	menjadi	337			
	4	ada	334			
	5	berada	293			
	6	terkenal	21			

5.2.4. Pengujian Parameter Pembelajaran Keempat (Semua)

Parameter pembelajaran keempat diujikan pada data [herbal_penyakit] dengan ID pengujian (3.a) yang dapat diamati pada Tabel 5.8 dan juga diujikan pada data [tempat_wisata] dengan ID pengujian (3.b) yang dapat diamati pada Tabel 5.9.

Tabel 5.8 Pengujian 4.a - Data [herbal_penyakit] pada semua mesin pencari

Statistik Jumlah Korpus	Min	83	kalimat	Median	202	kalimat
	Max	579	kalimat	Standar Deviasi	100.72	kalimat
	Mean	228.19	kalimat	Total	22363	kalimat
Ekstraksi Kata	MAANS			MFLR		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	mengobati	0.7619753353637657	1	mengobati	59
	2	mengatasi	0.31195837212488686	2	mengatasi	17
	3	menyembuhkan	0.1050126946556854	3	menurunkan	3
	4	menurunkan	0.07704546864227126	4	berkhasiat	3
	5	menjadi	0.05387216993678981	5	mengurangi	2
Evaluasi Kata	#	Kata Kerja	Google	Yahoo	Ecosia	Total
	1	menurunkan	81	84.611	2.080	86.772
	2	mengobati	1.034	47.313	4.976	53.323
	3	mengatasi	490	123	2.590	3.203
	4	berkhasiat	24	96	98	218
	5	mengurangi	0	85	124	209
	6	menyembuhkan	228	15	181	424
	7	menjadi	16	20	153	189

Tabel 5.9 Pengujian 4.b - Data [tempat_wisata] pada semua mesin pencari

Statistik Jumlah Korpus	Min	95	kalimat	Median	203	kalimat
	Max	414	kalimat	Standar Deviasi	74.91	kalimat
	Mean	213.03	kalimat	Total	20238	kalimat
Ekstraksi Kata	MAANS			MFLR		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	terletak	0.5424013603005358	1	terletak	35
	2	berada	0.3431752447853141	2	berada	15
	3	menjadi	0.08747823156877113	3	ada	4
	4	ada	0.08484136872294767	4	menjadi	4
	5	terkenal	0.0332270894627874	5	memiliki	2
Evaluasi Kata	#	Kata Kerja	Google	Yahoo	Ecosia	Total
	1	berada	117	105.000	14	105.131
	2	terletak	158	83.990	18	84.166
	3	ada	90	47.491	52	47.633
	4	menjadi	0	46.040	96	46.136
	5	memiliki	0	0	18	18
	6	terkenal	2	6.880	2.514	9.396

5.3. Pembahasan Pengujian

Dari delapan hasil pengujian yang telah dipaparkan, ringkasan hasil ekstraksi kata untuk setiap data uji beserta hasil yang diharapkan dapat diamati pada Tabel 5.10. Berdasarkan tabel tersebut, hasil ekstraksi kata yang dipilih oleh setiap metode ekstraksi kata (MAANS dan MFLR) dengan beberapa parameter pembelajaran yang berbeda menghasilkan kata yang seragam, yaitu kata “**mengobati**” untuk data [herbal_penyakit] dan kata “**terletak**” untuk data [tempat_wisata].

Tabel 5.10 Ringkasan Hasil Pengujian LNTRO

#	Data	ID	Ekstraksi Kata MAANS	Ekstraksi Kata MFLR	Hasil yang Diharapkan
1	[herbal_penyakit]	1.a	mengobati	mengobati	<u>mengobati</u> mengatasi menyembuhkan
2	[herbal_penyakit]	2.a	mengobati	mengobati	
3	[herbal_penyakit]	3.a	mengobati	mengobati	
4	[herbal_penyakit]	4.a	mengobati	mengobati	
5	[tempat_wisata]	1.b	terletak	terletak	<u>terletak</u> berada
6	[tempat_wisata]	2.b	terletak	terletak	
7	[tempat_wisata]	3.b	terletak	terletak	
8	[tempat_wisata]	4.b	terletak	terletak	

5.3.1. Analisis Akurasi Hasil Pembelajaran

Dalam pengembangan program utama, tahap tambahan evaluasi kata ditambahkan untuk membantu peneliti dalam mengukur keakuratan hasil pembelajaran relasi non-taksonomi. Evaluasi dilakukan dengan melakukan *query* ulang ke mesin pencari dengan kata kunci berupa kombinasi antara pasangan data sampel dengan kata nama relasi non-taksonomi yang sudah diekstrak. Jumlah pencarian yang dihasilkan oleh mesin pencari kemudian diambil dan disajikan pada peneliti sebagai acuan keakuratan hasil pembelajaran. Oleh karena itu, hasil proses *sampling* pasangan data dan kemampuan internal dari setiap mesin pencari dalam menentukan jumlah pencarian sangat mempengaruhi hasil evaluasi. Dalam mengamati hasil evaluasi kata, penulis berasumsi bahwa kata terpilih dari suatu data sudah sesuai dan cukup akurat apabila hasil evaluasi dari kata tersebut menempati urutan pertama atau urutan kedua.

Analisis Akurasi pada Data [herbal_penyakit]

Untuk data [herbal_penyakit], hasil evaluasi untuk kata terpilih “**mengobati**” menempati urutan pertama pada mesin pencari Google seperti terlihat pada Tabel 5.2. Google adalah mesin pencari raksasa yang sudah tidak diragukan lagi kemampuannya dalam mencari dokumen, sehingga jumlah hasil pencarian tersebut dianggap sesuai dan relevan oleh peneliti.

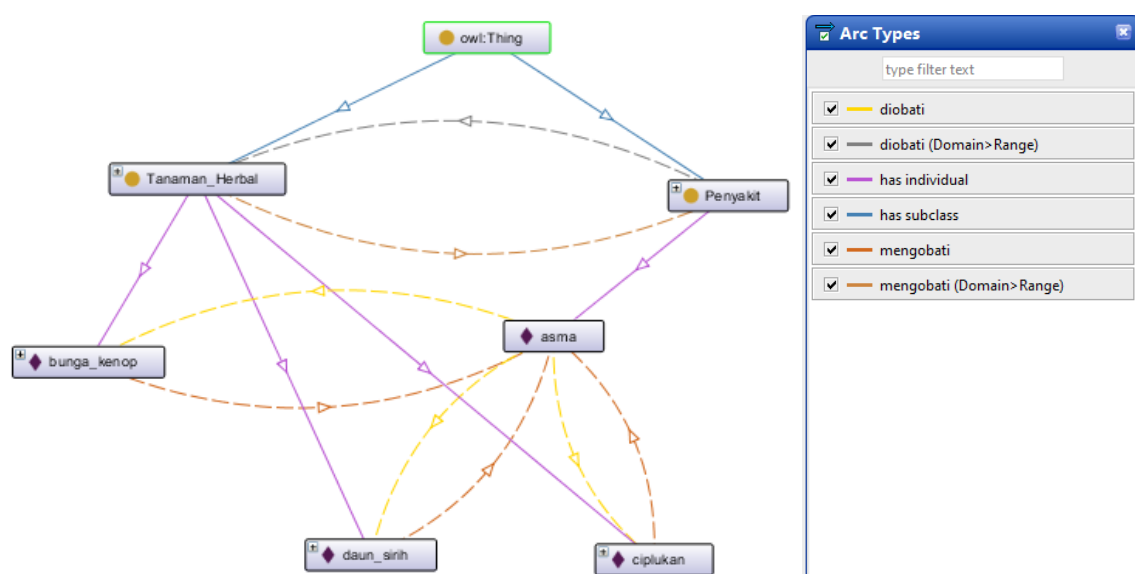
Pada mesin pencari Yahoo, kata terpilih “**mengobati**” menempati urutan kedua setelah kata “**berkhasiat**” seperti terlihat pada Tabel 5.4. Menurut analisis peneliti, hal tersebut terjadi karena kata “**berkhasiat**” relatif cukup banyak digunakan bersama dengan kata kerja lain dalam satu kalimat, misalnya “*jahe **berkhasiat** untuk **mengatasi** batuk*”. Berbeda halnya dengan mesin pencari yang menghitung kemunculan kata tersebut, rangkaian proses ekstraksi program tidak memperhitungkan kalimat tersebut karena kalimat tersebut akan tersaring pada tahap anotasi korpus bagian seleksi kata kerja, di mana suatu kalimat yang memiliki lebih dari satu kata kerja tidak akan digunakan karena adanya ambiguitas. Dalam kasus ini, peneliti berasumsi bahwa hasil evaluasi Yahoo masih cukup relevan karena kata terpilih “**mengobati**” menempati urutan kedua, mengingat adanya pembatasan seleksi kata kerja.

Pada mesin pencari Ecosia, kata terpilih “**mengobati**” menempati urutan pertama seperti terlihat pada Tabel 5.6. Dalam pengujian ini, Ecosia secara mengejutkan menghasilkan hasil evaluasi yang sesuai dan relevan dengan kata terpilih “**mengobati**”, meski secara pasar mesin pencari Ecosia adalah pemain baru. Dalam kasus ini, peneliti berasumsi bahwa hasil evaluasi Ecosia cukup baik karena didukung oleh data dari Bing (Ecosia menggunakan data pencarian dari Bing Microsoft).

Pada pengujian di mana ketiga mesin pencari (Google, Yahoo, Ecosia) digunakan, kata terpilih “**mengobati**” menempati urutan kedua setelah kata “**menurunkan**” seperti terlihat pada Tabel 5.8. Meskipun demikian, apabila sudut pandang dipersempit pada satu mesin pencari saja, maka kata terpilih “**mengobati**” menempati urutan pertama pada mesin pencari Google dan Ecosia. Jumlah hasil pencarian kata “**menurunkan**” pada mesin pencari Yahoo sangat besar dan mendominasi keseluruhan total penjumlahan hasil pencarian. Dalam kasus ini, penulis berasumsi bahwa hal tersebut terjadi karena adanya sebagian kecil sampel pasangan data yang lebih relevan dengan kata “**menurunkan**” dibandingkan kata “**mengobati**” dan jumlah hasil pencariannya cukup besar sehingga mendominasi hasil pencarian Yahoo.

Dari beberapa hasil analisis di atas, peneliti menyimpulkan bahwa kata terpilih “**mengobati**” adalah kata yang tepat sebagai hasil ekstraksi data [herbal_penyakit], karena kata “**mengobati**” merupakan salah satu kata yang diasumsikan sebagai kandidat nama relasi oleh penalaran peneliti dan hasil evaluasi kata juga menunjukkan bahwa kata terpilih “**mengobati**” konsisten muncul sebagai kata kerja yang menghasilkan hasil pencarian dengan urutan teratas. Meski demikian, dengan adanya kasus di mana salah satu sampel pasangan data memiliki jumlah hasil pencarian jauh lebih tinggi dibandingkan dengan sampel data lain, maka hasil evaluasi relatif kurang bisa diandalkan. Peneliti berasumsi ukuran sampel relatif kurang besar sehingga masih memungkinkan terjadinya dominasi sampel tertentu.

Ontologi yang di-*generate* oleh program untuk data [herbal_penyakit] dan nama relasi non-taksonomi terpilih “**mengobati**” dapat diamati pada Gambar 5.2.



Gambar 5.2 Ontologi data [herbal_penyakit] dengan nama relasi “mengobati”

Analisis Akurasi pada Data [tempat_wisata]

Untuk data [tempat_wisata], hasil evaluasi untuk kata terpilih **“terletak”** menempati urutan pertama pada mesin pencari Google seperti terlihat pada Tabel 5.3. Google adalah mesin pencari raksasa yang sudah tidak diragukan lagi kemampuannya dalam mencari dokumen, sehingga jumlah hasil pencarian tersebut dianggap sesuai dan relevan oleh peneliti.

Pada mesin pencari Yahoo, kata terpilih **“terletak”** menempati urutan pertama seperti terlihat pada Tabel 5.5. Namun jumlah hasil pencarian kata **“terletak”** memiliki selisih yang kecil dibandingkan dengan urutan kedua kata **“berada”** yang secara ekstraksi juga memiliki skor akhir kedua tertinggi. Dalam kasus ini, peneliti berasumsi bahwa jumlah hasil pencarian Yahoo sudah sesuai dan relevan dengan data.

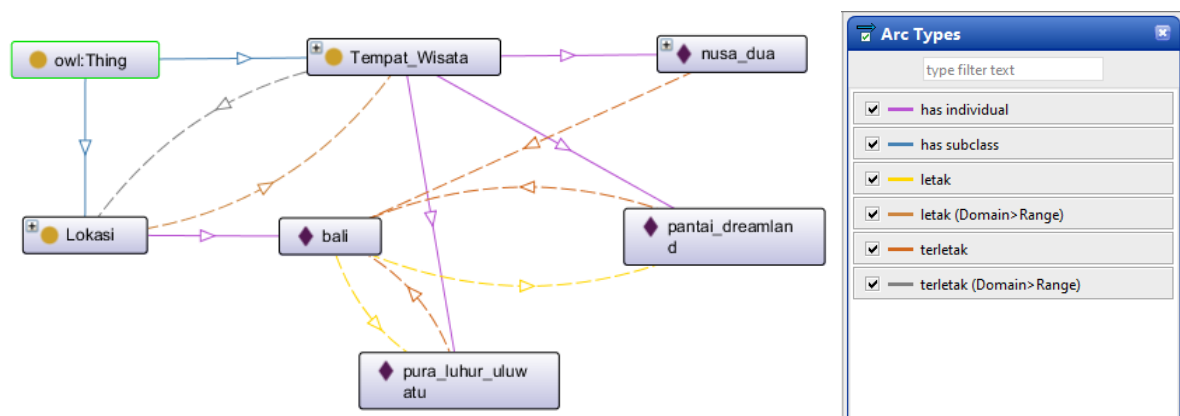
Pada mesin pencari Ecosia, kata terpilih **“terletak”** menempati urutan kedua setelah kata **“memiliki”** seperti terlihat pada Tabel 5.7. Menurut analisis peneliti, hal tersebut terjadi karena kata **“memiliki”** adalah kata yang cukup universal dan dapat dipasangkan dengan berbagai macam kata benda untuk menghasilkan makna baru. Oleh karena itu, kata **“memiliki”** mempunyai masalah yang sama dengan kata **“berkehasiat”** pada pengujian 2.a, yaitu kata tersebut banyak tersaring pada tahap anotasi korpus bagian seleksi kata kerja, namun pada mesin pencari tetap dihitung dalam pencacahan hasil pencarian. Dalam kasus ini, peneliti berasumsi bahwa hasil evaluasi Ecosia masih cukup relevan karena kata terpilih **“terletak”** menempati urutan kedua, mengingat adanya pembatasan seleksi kata kerja.

Pada pengujian di mana ketiga mesin pencari (Google, Yahoo, Ecosia) digunakan, kata terpilih **“terletak”** menempati urutan kedua setelah kata **“berada”** seperti terlihat pada Tabel 5.9. Pada mesin pencari Google, kata terpilih **“terletak”** menempati urutan pertama dan kata **“berada”** menempati urutan kedua. Pada mesin pencari Yahoo, kata terpilih **“terletak”** menempati urutan kedua setelah kata **“berada”** yang menempati urutan pertama. Pada mesin pencari Ecosia, kedua kata tersebut menghasilkan jumlah pencarian yang sangat sedikit bila dibandingkan dengan kata lainnya. Dalam kasus ini, peneliti berasumsi bahwa seperti halnya pada pengujian 4.a, hasil *sampling* pasangan data yang digunakan dalam tahap evaluasi sangat mempengaruhi hasil evaluasi kata. Meski demikian, untuk mesin pencari Google dan Yahoo hasilnya cukup relevan karena urutan kata terpilih **“terletak”** masih menempati urutan kedua.

Dari beberapa hasil analisis di atas, peneliti menyimpulkan bahwa kata terpilih **“terletak”** adalah kata yang tepat sebagai hasil ekstraksi data [tempat_wisata], karena kata **“terletak”** merupakan salah satu kata yang diasumsikan sebagai kandidat nama relasi oleh penalaran peneliti dan hasil evaluasi kata juga menunjukkan bahwa kata terpilih **“terletak”**

konsisten muncul sebagai kata kerja yang menghasilkan hasil pencarian dengan urutan teratas. Meski demikian, akurasi dan kemampuan setiap mesin pencari berbeda-beda sehingga hasil evaluasi secara keseluruhan relatif kurang bisa diandalkan. Peneliti berasumsi masih ada ruang untuk peningkatan dalam proses ekstraksi untuk memilih kata kerja yang tepat sehingga dapat memilih kata-kata kerja yang lebih relevan.

Ontologi yang di-generate oleh program untuk data [tempat_wisata] dan nama relasi non-taksonomi terpilih “**terletak**” dapat diamati pada Gambar 5.3.



Gambar 5.3 Ontologi data [tempat_wisata] dengan nama relasi “terletak”

5.3.2. Analisis Metode Ekstraksi

Dalam penelitian ini, diusulkan dua jenis metode ekstraksi kata untuk menentukan skor akhir dari setiap kata kerja yang telah didapatkan, yaitu metode MAANS (*Maximum of Absolute Average of Normalized Score*) dan MFLR (*Most Frequent Local Recommendation*). Dari delapan hasil pengujian, metode MAANS dan metode MFLR menghasilkan dua kata teratas yang selalu sama dan konsisten, yaitu kata “**mengobati**” dan “**mengatasi**” untuk data [herbal_penyakit], dan kata “**terletak**” dan “**berada**” untuk data [tempat_wisata]. Untuk masalah pengurutan rangking kata kerja berdasarkan skor akhirnya, metode MAANS tidak memiliki masalah karena perhitungannya yang lebih kompleks apabila dibandingkan dengan metode MFLR dan hasilnya berbentuk desimal, sehingga sangat kecil kemungkinan adanya beberapa kata kerja yang memiliki skor akhir yang sama (kecuali skor nol). Pada metode MFLR, masalah adanya skor akhir yang sama muncul karena skor akhir berbentuk pencacahan kata kerja untuk setiap pasangan data. Hasil pengujian menunjukkan ada banyak kata kerja dengan skor akhir yang sama yang sama-sama berhak untuk menempati posisi tertentu. Meskipun skor akhir yang sama pada metode MFLR tidak ditemui pada urutan pertama dan

kedua, namun untuk urutan ketiga dan seterusnya (skor akhir rendah) seringkali terdapat kesamaan skor akhir sehingga sulit untuk menentukan kata kerja mana yang harus dimasukkan dalam daftar lima kata kerja teratas. Untuk skor akhir yang sama, urutan ditentukan secara acak oleh program. Sehingga kekurangan dari metode MFLR adalah pengambilan daftar lima kata kerja teratas dengan korpus yang sama menjadi kurang konsisten.

Dari analisis di atas, peneliti menyimpulkan bahwa metode MAANS maupun metode MFLR memiliki akurasi yang hampir sama dalam menentukan kata kerja terpilih, karena urutan pertama dan keduanya hampir selalu konsisten. Namun tentu saja urutan kata kerja juga dipengaruhi oleh mesin pencari yang digunakan (korpus), tidak hanya metode ekstraksi semata. Secara keseluruhan, peneliti lebih memilih metode MAANS dibandingkan metode MFLR karena metode MAANS lebih konsisten dalam menentukan urutan rangking kata kerja.

5.3.3. Temuan Tambahan yang Berpotensi Mempengaruhi Hasil Pembelajaran

Dalam proses implementasi dan pengujian program, peneliti menemukan beberapa perubahan alur proses yang berpotensi mempengaruhi hasil pembelajaran relasi non-taksonomi. Temuan-temuan ini tidak termasuk dalam rancangan alur kerja yang diusulkan, namun menurut peneliti hasil percobaan implementasi dari temuan-temuan ini cukup penting untuk diperhatikan dan dipaparkan. Berikut adalah beberapa temuan yang ditemukan oleh peneliti dalam proses implementasi dan pengujian program:

Penggunaan titik sebagai pemisah antar kata atau kalimat dalam *preprocessing*

Penggunaan titik sebagai pemisah antar kata atau kalimat pada tahap *preprocessing* berpotensi berpengaruh dalam proses pembelajaran relasi non-taksonomi, karena penggunaan titik ini mempengaruhi jumlah hasil korpus yang dihasilkan pada tahap Pembangunan Korpus.

Dalam rancangan alur kerja yang diusulkan, titik tidak digunakan sebagai pemisah antar kata atau kalimat. Hal ini dilakukan karena dalam proses pengujian, peneliti menemukan bahwa penggunaan titik sebagai pemisah antar kata atau kalimat menghasilkan jumlah korpus yang lebih sedikit dibandingkan dengan ketika titik digunakan sebagai pemisah antar kata atau kalimat. Namun karena secara bahasa seharusnya titik digunakan sebagai pemisah kata, kalimat dan makna, maka penggunaan titik ini perlu diperhatikan dan dibandingkan dengan usulan rancangan alur kerja yang prosesnya tidak menggunakan titik.

Perbandingan alur kerja antara menggunakan titik dengan tidak menggunakan titik dapat diamati pada Tabel 5.11.

Tabel 5.11 Perbandingan alur kerja antara menggunakan dengan tidak menggunakan titik

#	Subproses	Detail Implementasi Alur Kerja	
		Tidak Menggunakan Titik	Menggunakan Titik
1	Preprocessing	Pada bagian <i>Remove Punctuation</i> , tanda baca titik dihilangkan	Pada bagian <i>Remove Punctuation</i> , tanda baca titik tidak dihilangkan
2	Ekstraksi Kalimat Relevan dalam Korpus	Titik tidak dideteksi sebagai pemisah kalimat maupun sebagai pemisah antara teks pasangan data	Titik dideteksi sebagai pemisah kalimat dan sebagai pemisah antara pasangan data

Setelah dilakukan pembelajaran dengan menggunakan parameter pembelajaran ID pengujian (4.a) dan (4.b), perbandingan statistik jumlah korpus yang dihasilkan dapat diamati pada Tabel 5.12 untuk ID pengujian (4.a) dan pada Tabel 5.14 untuk ID pengujian (4.b). Yang dimaksud dengan statistik selisih per pasangan data adalah statistik dari hasil perhitungan selisih jumlah korpus yang dihasilkan oleh setiap pasangan data.

Tabel 5.12 Perbandingan statistik penggunaan titik pada ID pengujian (4.a)

#	Statistik	Jumlah Korpus (satuan kalimat)		Statistik Selisih Per Pasangan Data
		Tidak Menggunakan Titik	Menggunakan Titik	
1	Max	579	485	200
2	Min	83	5	11
3	Mean	228,19	145,99	82,20
4	Median	202	133	72,5
5	Standar Deviasi	100,72	91,34	43,98
6	Total	22.363	14.307	8.056

Dari Tabel 5.12, dapat dilihat bahwa proses alur kerja awal yang tidak menggunakan titik menghasilkan korpus rata-rata lebih besar dibandingkan dengan alur kerja yang menggunakan titik dengan selisih sekitar 82 kalimat. Selain itu, terdapat pasangan data yang memiliki selisih paling besar antara menggunakan dengan tidak menggunakan titik, yaitu selisih 200 kalimat. Total selisih kalimat yang ada dalam korpus antara menggunakan dengan tidak menggunakan titik sebesar 8.056 kalimat. Sedangkan untuk perbandingan hasil ekstraksi kata dan hasil akhir nama relasi non-taksonominya dapat diamati pada Tabel 5.13.

Tabel 5.13 Perbandingan ekstraksi kata penggunaan titik pada ID pengujian (4.a)

	Tidak Menggunakan Titik			Menggunakan Titik		
Ekstraksi Kata MAANS	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	mengobati	0.7608226521	1	mengobati	0.6722766885
	2	mengatasi	0.3136171232	2	mengatasi	0.2830033792
	3	menyembuhkan	0.1046275936	3	menyembuhkan	0.1077757826
	4	menurunkan	0.0771528800	4	menurunkan	0.0703108356
	5	menghilangkan	0.0543112226	5	menjadi	0.0697800361
Ekstraksi Kata MFLR	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	mengobati	60	1	mengobati	52
	2	mengatasi	17	2	mengatasi	14
	3	berkhasiat	3	3	menjadi	5
	4	menurunkan	3	4	menghilangkan	4
	5	mengurangi	2	5	menurunkan	4

Dari perbandingan hasil ekstraksi kata dengan skor lima teratas pada Tabel 5.13, dapat dilihat bahwa hasil akhir nama relasi non-taksonomi yang dihasilkan masih sama, yaitu kata **“mengobati”**. Namun hasil urutan ranking kata menunjukkan bahwa ada perubahan urutan ranking yang terjadi pada kata-kata yang urutannya di bawah dua kata teratas. Alur kerja yang menggunakan titik menghasilkan jumlah kalimat yang lebih sedikit, dan dapat dilihat bahwa hal ini juga mempengaruhi skor akhir yang dihasilkan oleh kedua metode ekstraksi.

Tabel 5.14 Perbandingan statistik penggunaan titik pada ID pengujian (4.b)

#	Statistik	Jumlah Korpus (satuan kalimat)		Statistik Selisih Per Pasangan Data
		Tidak Menggunakan Titik	Menggunakan Titik	
1	Max	414	372	113
2	Min	95	39	7
3	Mean	213,03	163,79	49,24
4	Median	203	161	43
5	Standar Deviasi	74,91	63,46	23,53
6	Total	20.238	15.560	4.678

Dari Tabel 5.14, dapat dilihat bahwa proses alur kerja awal yang tidak menggunakan titik menghasilkan korpus rata-rata lebih besar dibandingkan dengan alur kerja yang menggunakan titik dengan selisih sekitar 49 kalimat. Selain itu, terdapat pasangan data yang memiliki selisih paling besar antara menggunakan dengan tidak menggunakan titik, yaitu selisih 113 kalimat. Total selisih kalimat yang ada dalam korpus antara menggunakan dengan tidak menggunakan

titik sebesar 4.678 kalimat. Sedangkan untuk perbandingan hasil ekstraksi kata dan hasil akhir nama relasi non-taksonominya dapat diamati pada Tabel 5.15.

Tabel 5.15 Perbandingan ekstraksi kata penggunaan titik pada ID pengujian (4.b)

	Tidak Menggunakan Titik			Menggunakan Titik		
Ekstraksi Kata MAANS	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	terletak	0.5424013603	1	terletak	0.4826341494
	2	berada	0.3431752448	2	berada	0.3582950237
	3	menjadi	0.0874782316	3	menjadi	0.0757380631
	4	ada	0.0848413687	4	ada	0.0746401260
	5	terkenal	0.0332270895	5	terdapat	0.0200853485
Ekstraksi Kata MFLR	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	terletak	35	1	terletak	35
	2	berada	15	2	berada	23
	3	ada	4	3	ada	5
	4	menjadi	4	4	menjadi	5
	5	memiliki	2	5	menghubungkan	1

Dari perbandingan hasil ekstraksi kata dengan skor lima teratas pada Tabel 5.15, dapat dilihat bahwa hasil akhir nama relasi non-taksonomi yang dihasilkan masih sama, yaitu kata “**terletak**”. Namun hasil urutan rangking kata menunjukkan bahwa ada perubahan urutan rangking yang terjadi pada kata-kata yang urutannya di bawah empat kata teratas. Alur kerja yang menggunakan titik menghasilkan jumlah kalimat yang lebih sedikit, dan dapat dilihat bahwa hal ini juga mempengaruhi skor akhir yang dihasilkan oleh kedua metode ekstraksi.

Akan tetapi pada kedua pengujian yang dilakukan, apabila diperhatikan pada individu setiap kalimat yang dihasilkan, korpus yang dihasilkan oleh alur kerja awal yang tidak menggunakan titik memiliki lebih banyak kalimat yang panjang (jumlah katanya banyak). Hal ini terjadi karena titik dihilangkan, sehingga pencarian kalimat relevan akan dilakukan terus hingga teks pasangan data ditemukan. Berbeda halnya dengan alur kerja yang menggunakan titik, korpus yang dihasilkan tidak memiliki banyak kalimat panjang. Dampak dari panjang kalimat adalah kalimat yang panjang akan cenderung memiliki jumlah kata kerja yang lebih banyak, yang justru akan tersaring pada tahap Anotasi Kata bagian seleksi kata kerja, di mana setiap kalimat dengan lebih dari satu kata kerja akan dibuang. Dalam kasus ini, ekstraksi kalimat yang panjang menjadi tidak berarti karena pada tahap selanjutnya pun akan tersaring.

Dari kedua statistik tersebut, dapat disimpulkan bahwa meskipun alur kerja yang menggunakan titik menghasilkan jumlah korpus yang lebih sedikit, namun penggunaan titik mempengaruhi hasil ekstraksi kata dan berpotensi mengubah hasil akhir nama relasi.

Penggunaan *stopwords* berupa daftar kata kerja bantu dalam proses seleksi kata kerja

Penggunaan *stopwords* berupa daftar kata kerja bantu dalam proses seleksi kata kerja berpotensi berpengaruh dalam proses pembelajaran relasi non-taksonomi, karena penggunaan *stopwords* ini mempengaruhi daftar kata kerja yang dihasilkan pada tahap Anotasi Korpus.

Dalam rancangan alur kerja yang diusulkan, *stopword* hanya digunakan dalam tahap Ekstraksi Kata untuk menyaring nama relasi yang bersifat taksonomis saja. Pada proses seleksi kata kerja, setiap kalimat yang jumlah kata kerjanya lebih dari satu tidak diperhitungkan, sehingga setiap kemungkinan nama relasi yang didapatkan dari kalimat-kalimat tersebut tidak dapat dimanfaatkan. Rancangan alur kerja yang demikian dibuat untuk mengurangi ambiguitas, karena peneliti menemukan bahwa ada banyak kata kerja yang termasuk kata kerja bantu yang cukup sering muncul. Hal ini tentu mempengaruhi daftar kandidat nama relasi, padahal kata kerja bantu tidak memiliki makna khusus dan tidak relevan.

Solusi alternatif untuk permasalahan ini adalah dengan membuat daftar kata kerja bantu sebagai *stopwords* untuk mengeliminasi kata kerja bantu yang muncul, dengan harapan kata kerja lain dalam kalimat dengan lebih dari satu kata kerja dapat tetap diperhitungkan. Karena temuan ini berpotensi mempengaruhi daftar kata kerja yang dihasilkan, maka penggunaan *stopwords* daftar kata kerja bantu ini perlu diperhatikan dan dibandingkan dengan usulan rancangan alur kerja awal, yaitu membuang kalimat yang memiliki lebih dari satu kata kerja.

Perbandingan alur kerja antara menggunakan *stopword* kata kerja bantu dengan menyaring kalimat dengan lebih dari satu kata kerja dapat diamati pada Tabel 5.16.

Tabel 5.16 Perbandingan alur kerja menggunakan *stopword* kata kerja bantu

#	Subproses	Detail Implementasi Alur Kerja	
		Filter Jumlah Kata Kerja	<i>Stopword</i> Kata Kerja Bantu
1	Seleksi Kata Kerja	Menggunakan filter jumlah kata kerja untuk setiap kalimat Membuang kalimat yang jumlah kata kerjanya lebih dari satu	Menghilangkan filter jumlah kata kerja Untuk setiap kalimat, hilangkan setiap kata kerja yang ada dalam daftar <i>stopword</i> kata kerja bantu

Setelah dilakukan pembelajaran dengan menggunakan parameter pembelajaran ID pengujian (4.a) dan (4.b), perbandingan statistik jumlah kata kerja yang dihasilkan dapat diamati pada Tabel 5.17 untuk ID pengujian (4.a) dan pada Tabel 5.19 untuk ID pengujian (4.b). Yang dimaksud dengan statistik selisih per pasangan data adalah statistik dari hasil perhitungan selisih jumlah kata kerja yang dihasilkan oleh setiap pasangan data.

Tabel 5.17 Perbandingan statistik penggunaan *stopword* pada ID pengujian (4.a)

#	Statistik	Jumlah Kata Kerja Keseluruhan (satuan kata)		Statistik Selisih Per Pasangan Data
		Filter Jumlah K. Kerja	Stopword K. Kerja Bantu	
1	Max	226	659	599
2	Min	12	80	22
3	Mean	74,78	271,63	196,85
4	Median	67	259,5	168
5	Standar Deviasi	40,39	126,76	110,61
6	Total	7.329	26.620	19.291
7	Total Unique	423	1.313	890

Dari Tabel 5.17, dapat dilihat bahwa proses alur kerja awal yang menggunakan filter jumlah kata kerja menghasilkan jumlah kata kerja rata-rata lebih sedikit dibandingkan dengan menggunakan *stopword* kata kerja bantu dengan selisih sekitar 196 kata. Total selisih jumlah kata kerja antara menggunakan filter jumlah dengan *stopword* sebesar 19.291 kata dengan selisih kata kerja unik sebesar 890 kata. Sedangkan untuk perbandingan hasil ekstraksi kata dan hasil akhir nama relasi non-taksonominya dapat diamati pada Tabel 5.18.

Tabel 5.18 Perbandingan ekstraksi kata penggunaan *stopword* pada ID pengujian (4.a)

	Filter Jumlah Kata Kerja			Stopword Kata Kerja Bantu		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
Ekstraksi Kata MAANS	1	mengobati	0.7608226521	1	mengobati	0.7941041612
	2	mengatasi	0.3136171232	2	mengatasi	0.4279540849
	3	menyembuhkan	0.1046275936	3	menyembuhkan	0.1704861799
	4	menurunkan	0.0771528800	4	menurunkan	0.1261656462
	5	menghilangkan	0.0543112226	5	berkhasiat	0.0978170806
Ekstraksi Kata MFLR	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	mengobati	60	1	mengobati	56
	2	mengatasi	17	2	mengatasi	19
	3	berkhasiat	3	3	mencegah	4
	4	menurunkan	3	4	menurunkan	4
	5	mengurangi	2	5	berkhasiat	2

Dari perbandingan hasil ekstraksi kata dengan skor lima teratas pada Tabel 5.18, dapat dilihat bahwa hasil akhir nama relasi non-taksonomi yang dihasilkan masih sama, yaitu kata “**mengobati**”. Namun hasil urutan rangking kata menunjukkan bahwa ada perubahan urutan rangking yang terjadi pada kata-kata yang urutannya di bawah dua kata teratas. Alur kerja yang

menggunakan *stopword* menghasilkan jumlah kata yang jauh lebih banyak, dan dapat dilihat bahwa hal ini juga mempengaruhi skor akhir yang dihasilkan oleh kedua metode ekstraksi.

Tabel 5.19 Perbandingan statistik penggunaan *stopword* pada ID pengujian (4.b)

#	Statistik	Jumlah Kata Kerja Keseluruhan (satuan kata)		Statistik Selisih Per Pasangan Data
		Filter Jumlah K. Kerja	Stopword K. Kerja Bantu	
1	Max	120	490	427
2	Min	8	15	6
3	Mean	42,69	140,24	97,55
4	Median	38	120	79
5	Standar Deviasi	22,22	78,48	67,95
6	Total	4.056	13.323	9.267
7	Total Unique	597	1.428	831

Dari Tabel 5.19, dapat dilihat bahwa proses alur kerja awal yang menggunakan filter jumlah kata kerja menghasilkan jumlah kata kerja rata-rata lebih sedikit dibandingkan dengan menggunakan *stopword* kata kerja bantu dengan selisih sekitar 97 kata. Total selisih jumlah kata kerja antara menggunakan filter jumlah dengan *stopword* sebesar 9.267 kata dengan selisih kata kerja unik sebesar 831 kata. Sedangkan untuk perbandingan hasil ekstraksi kata dan hasil akhir nama relasi non-taksonominya dapat diamati pada Tabel 5.20.

Tabel 5.20 Perbandingan ekstraksi kata penggunaan *stopword* pada ID pengujian (4.b)

	Filter Jumlah Kata Kerja			Stopword Kata Kerja Bantu		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
Ekstraksi Kata MAANS	1	terletak	0.5424013603	1	terletak	0.6336092900
	2	berada	0.3431752448	2	berada	0.4789208410
	3	menjadi	0.0874782316	3	ada	0.2242030601
	4	ada	0.0848413687	4	terkenal	0.1103362469
	5	terkenal	0.0332270895	5	terdapat	0.0649177459
Ekstraksi Kata MFLR	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	terletak	35	1	terletak	37
	2	berada	15	2	berada	20
	3	ada	4	3	ada	4
	4	menjadi	4	4	terkenal	3
	5	memiliki	2	5	menarik	2

Dari perbandingan hasil ekstraksi kata dengan skor lima teratas pada Tabel 5.20, dapat dilihat bahwa hasil akhir nama relasi non-taksonomi yang dihasilkan masih sama, yaitu kata “**terletak**”. Namun hasil urutan rangking kata menunjukkan bahwa ada perubahan urutan rangking yang terjadi pada kata-kata yang urutannya di bawah dua kata teratas. Alur kerja yang menggunakan *stopword* menghasilkan jumlah kata yang jauh lebih banyak, dan dapat dilihat bahwa hal ini juga mempengaruhi skor akhir yang dihasilkan oleh kedua metode ekstraksi.

Akan tetapi pada kedua pengujian yang dilakukan, apabila diperhatikan pada kata kerja yang dihasilkan, pada alur kerja yang menggunakan *stopword* kata kerja bantu ada banyak kata kerja yang bukan merupakan kata kerja bantu namun tidak relevan dengan domain pengetahuan dari data yang dimasukkan. Hal ini dianggap wajar, karena semakin banyak kata kerja yang terdeteksi oleh sistem maka kemungkinan munculnya kata kerja yang tidak relevan juga semakin tinggi. Di sisi lain, dengan tidak membuang kalimat yang memiliki lebih dari satu kata kerja, maka kata kerja yang memang relevan dan cocok dijadikan sebagai kandidat nama relasi non-taksonomi juga lebih sering muncul dan frekuensinya kemunculannya pun tetap relatif tinggi dibandingkan dengan kata kerja yang kurang relevan.

Dari kedua statistik tersebut, dapat disimpulkan bahwa alur kerja awal yang menggunakan filter jumlah kata kerja menghasilkan jumlah kata kerja yang jauh lebih sedikit dibandingkan dengan menggunakan *stopword* kata kerja bantu. Namun *stopword* daftar kata kerja bantu dibuat oleh manusia, sehingga akurasinya tergantung kelengkapan daftar yang dibuat dan juga tergantung pada konteks dan domain pengetahuan dari data yang digunakan. Untuk itu, daftar *stopword* yang digunakan harus independen terhadap domain pengetahuan dan juga harus berjumlah cukup besar.

Penggunaan *stemming* untuk menggabungkan kemunculan frekuensi beberapa kata yang berbeda dengan kata dasar yang sama

Penggunaan *stemming* untuk menggabungkan kemunculan frekuensi beberapa kata yang berbeda dengan kata dasar yang sama berpotensi berpengaruh dalam proses pembelajaran relasi non-taksonomi, karena penggunaan *stemming* ini mempengaruhi frekuensi kata kerja yang dihasilkan pada tahap Ekstraksi Kata.

Dalam rancangan alur kerja yang diusulkan, *stemming* hanya digunakan untuk mencari bentuk kata kerja aktif dan pasif. Namun ketika hasil implementasi program diujikan, ditemukan bahwa terdapat dua atau lebih kata yang berbeda dengan kata dasar yang sama yang muncul sebagai kandidat nama relasi yang berbeda. Misalnya kata “cegah” dengan

“mencegah”, kedua kata tersebut memiliki kata dasar yang sama, yaitu “cegah”. Namun pada saat pengujian ditemukan bahwa dua kata tersebut masuk sebagai dua kandidat nama relasi yang berbeda dengan nilai skor yang berbeda pula. Padahal dari segi makna kedua kata tersebut adalah kata kerja aktif yang memiliki makna yang sama, yaitu “mencegah”. Itu artinya, seharusnya frekuensi kemunculan untuk beberapa kata yang berbeda dengan kata dasar yang sama digabungkan menjadi satu.

Untuk itu, penggabungan frekuensi kemunculan beberapa kata dengan kata dasar yang sama perlu diperhatikan dan dibandingkan dengan ketika tidak digabungkan seperti pada rancangan alur kerja yang diusulkan. Namun karena dari segi semantik kata kerja aktif dan pasif memiliki makna yang berbeda, maka penggabungan frekuensi kemunculan kata dipisahkan antara kata kerja aktif dengan pasif. Adapun imbuhan yang digunakan sebagai bentuk kata kerja adalah imbuhan me-, di-, be-, dan te-. Untuk kata kerja aktif bentuk imbuhan yang digunakan adalah imbuhan me- dan be-. Dan untuk kata kerja pasif bentuk imbuhan yang digunakan adalah imbuhan di- dan te-. Untuk kata dasar yang juga termasuk dalam kata kerja (dalam bahasa Indonesia disebut sebagai *verba asal*) maka kata dasar tersebut dianggap sebagai kata kerja aktif. Hal ini dikarenakan karena meskipun verba asal dapat digunakan dalam kalimat pasif (Alwi dkk., 2003), namun letak verba dasarnya tidak lagi berada di antara subjek dan objek, sehingga tidak relevan dengan tahap ekstraksi kalimat relevan dalam korpus di mana kedua pasangan data dianggap sebagai komponen subjek dan objek.

Penjelasan proses penggabungan frekuensi untuk kata dasar yang sama dapat dimodelkan dalam kasus seperti berikut. Misalkan suatu korpus memiliki frekuensi kata “menurunkan” sebanyak 100 kata, frekuensi kata “turunkan” sebanyak 25 kata, dan frekuensi kata “diturunkan” sebanyak 50 kata. Maka untuk penggabungan kata kerja aktif dipilih kata yang memiliki imbuhan me-, yaitu “menurunkan” dengan frekuensi gabungan sebanyak $100+25 = 125$ kata, dan penggabungan kata kerja pasif dipilih kata yang memiliki imbuhan di-, yaitu “diturunkan” dengan frekuensi gabungan tetap yaitu 50 kata. Kata kerja “turunkan” yang sudah diserap kemudian dihapus dari himpunan kata kerja.

Perbandingan alur kerja antara penggabungan frekuensi untuk kata dasar yang sama dengan tidak digabungkan dapat diamati pada Tabel 5.21.

Tabel 5.21 Perbandingan alur kerja menggabungkan frekuensi berdasarkan kata dasar

#	Subproses	Detail Implementasi Alur Kerja	
		Tidak Menggabungkan Frekuensi Kata Dasar	Menggabungkan Frekuensi Kata Dasar
1	Stemming/ Lemmatization	Menggunakan <i>library</i> yang dikembangkan sendiri menggunakan <i>regular expression</i> berdasarkan aturan algoritma Nazief & Adriani (2007)	Menggunakan <i>library</i> JSastrawi yang menggunakan algoritma Nazief & Adriani (2007)
2	Metode Ekstraksi MAANS	Tidak ada penggabungan frekuensi untuk kata mana pun	Setiap kata kerja dicari kata dasarnya dan kemudian dilakukan penjumlahan frekuensi untuk kata dasar yang sama
3	Metode Ekstraksi MFLR	Tidak ada penggabungan frekuensi untuk kata mana pun	Setiap kata kerja dicari kata dasarnya dan kemudian dilakukan penjumlahan frekuensi untuk kata dasar yang sama

Setelah dilakukan pembelajaran dengan menggunakan parameter pembelajaran ID pengujian (4.a) dan (4.b), perbandingan statistik jumlah kata kerja yang dihasilkan dapat diamati pada Tabel 5.22 untuk ID pengujian (4.a) dan pada Tabel 5.24 untuk ID pengujian (4.b). Yang dimaksud dengan statistik selisih per pasangan data adalah statistik dari hasil perhitungan selisih jumlah kata kerja yang dihasilkan oleh setiap pasangan data.

Tabel 5.22 Perbandingan statistik penggabungan frekuensi pada ID pengujian (4.a)

#	Statistik	Jumlah Kata Kerja Keseluruhan (satuan kata)		Statistik Selisih Per Pasangan Data
		Tidak Menggabungkan	Menggabungkan Frekuensi	
1	Max	191	191	0
2	Min	5	5	0
3	Mean	58,19	58,19	0
4	Median	51	51	0
5	Standar Deviasi	37,77	37,77	0
6	Total	5.703	5.703	0
7	Total Unique	160	135	25

Dari Tabel 5.22, dapat dilihat bahwa proses alur kerja awal yang tidak menggabungkan frekuensi menghasilkan jumlah kata kerja yang sama dengan menggabungkan frekuensi berdasarkan kata dasarnya. Terdapat 25 kata yang jumlah frekuensinya digabungkan ke kata

lain yang memiliki kata dasar yang sama. Sedangkan untuk perbandingan hasil ekstraksi kata dan hasil akhir nama relasi non-taksonominya dapat diamati pada Tabel 5.23.

Tabel 5.23 Perbandingan ekstraksi kata penggabungan frekuensi pada ID pengujian (4.a)

	Tidak Menggabungkan Frekuensi			Menggabungkan Frekuensi Kata Dasar		
	#	Kata Kerja	Skor	#	Kata Kerja	Skor
Ekstraksi Kata MAANS	1	mengobati	0.7608226521	1	mengobati	0.7747937716
	2	mengatasi	0.3136171232	2	mengatasi	0.3132295880
	3	menyembuhkan	0.1046275936	3	menyembuhkan	0.1135986365
	4	menurunkan	0.0771528800	4	menurunkan	0.0771473111
	5	menghilangkan	0.0543112226	5	menjadi	0.0548566824
Ekstraksi Kata MFLR	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	mengobati	60	1	mengobati	64
	2	mengatasi	17	2	mengatasi	17
	3	berkhasiat	3	3	menurunkan	4
	4	menurunkan	3	4	berkhasiat	3
	5	mengurangi	2	5	menjadi	3

Dari perbandingan hasil ekstraksi kata dengan skor lima teratas pada Tabel 5.23, dapat dilihat bahwa hasil akhir nama relasi non-taksonomi yang dihasilkan masih sama, yaitu kata “**mengobati**”. Namun hasil urutan rangking kata menunjukkan bahwa ada perubahan urutan rangking yang terjadi pada kata-kata yang urutannya di bawah dua kata teratas. Penggabungan frekuensi cukup mempengaruhi skor akhir yang dihasilkan oleh kedua metode ekstraksi.

Tabel 5.24 Perbandingan statistik penggabungan frekuensi pada ID pengujian (4.b)

#	Statistik	Jumlah Kata Kerja Keseluruhan (satuan kata)		Statistik Selisih Per Pasangan Data
		Tidak Menggabungkan	Menggabungkan Frekuensi	
1	Max	111	111	0
2	Min	1	1	0
3	Mean	21,18	21,18	0
4	Median	19	19	0
5	Standar Deviasi	15,86	15,86	0
6	Total	2.012	2.012	0
7	Total Unique	195	172	23

Dari Tabel 5.24, dapat dilihat bahwa proses alur kerja awal yang tidak menggabungkan frekuensi menghasilkan jumlah kata kerja yang sama dengan menggabungkan frekuensi

berdasarkan kata dasarnya. Terdapat 23 kata yang jumlah frekuensinya digabungkan ke kata lain yang memiliki kata dasar yang sama. Sedangkan untuk perbandingan hasil ekstraksi kata dan hasil akhir nama relasi non-taksonominya dapat diamati pada Tabel 5.25.

Tabel 5.25 Perbandingan ekstraksi kata penggabungan frekuensi pada ID pengujian (4.b)

	Tidak Menggabungkan Frekuensi			Menggabungkan Frekuensi Kata Dasar		
Ekstraksi Kata MAANS	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	terletak	0.5424013603	1	terletak	0.6260322009
	2	berada	0.3431752448	2	berada	0.4124679168
	3	menjadi	0.0874782316	3	menjadikan	0.1423787039
	4	ada	0.0848413687	4	ada	0.1004480939
	5	terkenal	0.0332270895	5	terkenal	0.0455077912
Ekstraksi Kata MFLR	#	Kata Kerja	Skor	#	Kata Kerja	Skor
	1	terletak	35	1	terletak	49
	2	berada	15	2	berada	25
	3	ada	4	3	menjadi	8
	4	menjadi	4	4	ada	6
	5	memiliki	2	5	memiliki	2

Dari perbandingan hasil ekstraksi kata dengan skor lima teratas pada Tabel 5.25, dapat dilihat bahwa hasil akhir nama relasi non-taksonomi yang dihasilkan masih sama, yaitu kata **“terletak”**. Namun hasil urutan rangking kata menunjukkan bahwa ada perubahan urutan rangking yang terjadi pada kata-kata yang urutannya di bawah dua kata teratas. Penggabungan frekuensi cukup mempengaruhi skor akhir yang dihasilkan oleh kedua metode ekstraksi.

Pada kedua pengujian yang dilakukan, dengan sumber data yang sama, terlihat bahwa jumlah kata kerja unik yang dihasilkan oleh penggabungan frekuensi dengan kata dasar lebih sedikit dibandingkan dengan jumlah kata kerja unik yang dihasilkan tanpa adanya penggabungan frekuensi. Dengan adanya penggabungan frekuensi untuk setiap kata yang memiliki kata dasar yang sama, maka jumlah kata kerja uniknya menjadi berkurang, karena sebagai akibat dari penggabungan frekuensi maka salah satu kata harus dihilangkan untuk menghindari duplikasi kata. Meski demikian, karena jumlah frekuensinya digabungkan, maka meskipun jumlah kata kerja uniknya berkurang namun dalam esensinya banyak kata yang diperhitungkan dalam proses penilaian skor masih sama.

Dari kedua statistik tersebut, dapat disimpulkan bahwa alur kerja awal yang tidak menggabungkan frekuensi berdasarkan kata dasarnya menghasilkan jumlah kata kerja unik yang lebih banyak dibandingkan dengan menggabungkan frekuensi berdasarkan kata dasarnya.

5.3.4. Kendala Pengujian

Dalam pelaksanaan pengujian, peneliti menghadapi beberapa kendala, di antaranya yaitu:

- a. Proses *request* data ke mesin pencari (terutama Google) sering mengalami kendala, yang menyebabkan proses pembelajaran dan evaluasi menjadi terhambat.
- b. Proses *request* memakan waktu relatif cukup lama untuk satu kali *request*, sehingga akumulasi untuk *request* dengan jumlah besar memakan waktu sangat lama.
- c. Setiap proses pembelajaran dan pengujian tidak dapat dilakukan secara paralel dan harus dilakukan satu per satu (sekuensial). Untuk itu setiap permintaan pembelajaran relasi non-taksonomi disimpan dalam suatu daftar tunggu (*queue*) di mana setiap permintaan akan diproses dan diselesaikan terlebih dahulu, baru kemudian memproses permintaan selanjutnya.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dari hasil pengujian yang dilakukan, sudah terbukti bahwa rancangan alur kerja yang diusulkan sudah berhasil menentukan nama relasi non-taksonomis dengan sumber data dari dokumen Web. Sehingga dapat disimpulkan bahwa:

- a. Penentuan nama relasi non-taksonomis yang tepat untuk dua data kelas ontologi dari dokumen dan data dari Web dilakukan dengan cara melakukan ekstraksi kata kerja dari korpus yang diambil dari dokumen Web melalui mesin pencari.
- b. Ekstraksi kata kerja posisinya berada di antara dua teks pasangan data kelas ontologi.
- c. Kemudian pemilihan nama relasi dari kandidat-kandidat kata kerja yang didapatkan dilakukan dengan pendekatan statistik untuk menentukan skor berdasarkan frekuensi kemunculan kata.
- d. Dalam data uji pertama yang digunakan, pasangan kelas ontologi yang digunakan adalah tanaman herbal dan penyakit. Maka dalam kasus tersebut nama relasi non-taksonomis adalah kata kerja yang muncul di antara teks data tanaman herbal dan teks data penyakit.

6.2. Saran

Untuk pengembangan penelitian lebih lanjut, terdapat beberapa hal yang dapat dipertimbangkan untuk meningkatkan kualitas proses pembelajaran relasi non-taksonomi sehingga hasil pembelajaran bisa lebih akurat dan relevan sesuai dengan konteks. Berikut adalah beberapa perbaikan dan peningkatan yang dapat menunjang pembelajaran lebih lanjut:

- a. Memperbaiki metode MFLR sehingga kasus adanya skor akhir yang sama dapat ditangani dengan lebih efektif.
- b. Penambahan pendeteksian sinonim kata untuk mengelompokkan kata yang mirip atau bermakna sama sehingga dapat menangkap konteks dengan lebih baik lagi.
- c. Dengan adanya temuan tambahan berupa penggunaan *stemming* untuk menggabungkan kemunculan frekuensi beberapa kata yang berbeda dengan kata dasar yang sama, maka sebaiknya ditambahkan algoritma khusus untuk menentukan imbuhan kata kerja aktif maupun pasif yang tepat untuk mewakili kata kerja yang diserap.

DAFTAR PUSTAKA

- Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M., & Williams, H. E. (2007). Stemming Indonesian: A confix-stripping approach. *ACM Transactions on Asian Language Information Processing (TALIP)*, 6(4), 1–33.
- Alexiev, V., Breu, M., & de Bruijn, J. (2005). *Information integration with ontologies : experiences from an industrial showcase*. John Wiley & Sons.
- Alwi, H., Dardjowidjojo, S., Lapoliwa, H., & Moeliono, A. (2003). Tata Bahasa Baku Bahasa Indonesia (Indonesian Grammar). *Balai Pustaka, Jakarta, Indonesia*.
- Borst, P., Akkermans, H., & Top, J. (1997). Engineering ontologies. *International Journal of Human-Computer Studies*, 46(2–3), 365–406. <https://doi.org/10.1006/ijhc.1996.0096>
- Dinakaramani, A., Rashel, F., Luthfi, A., & Manurung, R. (2014). Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus (hlm. 66–69). IEEE. <https://doi.org/10.1109/IALP.2014.6973519>
- Edition, S., Antoniou, G., & Harmelen, F. V. (2008). *A semantic Web primer* (Vol. 46). <https://doi.org/10.5860/CHOICE.46-1523>
- Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. Dalam *Proceedings of the conference on empirical methods in natural language processing* (hlm. 1535–1545). Association for Computational Linguistics.
- Genesereth, M. R., & Nilsson, N. J. (1987). Logical foundations of artificial. *Intelligence*. Morgan Kaufmann, 2.
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5–6), 907–928. <https://doi.org/10.1006/IJHC.1995.1081>
- Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- Maedche, A., & Staab, S. (t.t.). Mining Non-Taxonomic Conceptual Relations from Text, 13.
- Potter, S. (2003). A Survey of Knowledge Acquisition from Natural Language, 41.
- Sánchez, D., & Moreno, A. (2008). Learning non-taxonomic relationships from web documents for domain ontology construction. *Data & Knowledge Engineering*, 64(3), 600–623. <https://doi.org/10.1016/J.DATAK.2007.10.001>
- Serra, I., Girardi, R., & Novais, P. (2012). The Problem of Learning Non-taxonomic Relationships of Ontologies from Text. Dalam S. Omatu, J. F. De Paz Santana, S. R.

- González, J. M. Molina, A. M. Bernardos, & J. M. C. Rodríguez (Ed.), *Distributed Computing and Artificial Intelligence* (Vol. 151, hlm. 485–492). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-28765-7_58
- Staab, S., & Studer, R. (1998). *Handbook on Ontologies*, Second edition, 832.
- Sugihastuti, & Saudah, S. (2016). *Buku Ajar Bahasa Indonesia Akademik*. Yogyakarta: Pustaka Pelajar.
- Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. Dalam *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03* (Vol. 1, hlm. 173–180). Edmonton, Canada: Association for Computational Linguistics. <https://doi.org/10.3115/1073445.1073478>
- Villaverde, J., Persson, A., Godoy, D., & Amandi, A. (2009). Supporting the discovery and labeling of non-taxonomic relationships in ontology learning. *Expert Systems with Applications*, 36(7), 10288–10294. <https://doi.org/10.1016/j.eswa.2009.01.048>
- Wicaksana, I. (2004). Survei dan evaluasi metode pengembangan ontology. *Proceedings, Komputer dan Sistem Intelejen (KOMMIT2004)*.

LAMPIRAN A

#	Tanaman Herbal	Penyakit
1	Adem Ati	kencing manis
2	Adem Ati	luka berdarah
3	Adem Ati	radang usus
4	Akar Manis	batuk
5	Akar Manis	tukak lambung
6	Akar Wangi	batu ginjal
7	Akar Wangi	bau mulut
8	Akar Wangi	rematik
9	Alpukat	diabetes
10	Alpukat	maag
11	Belimbing	hipertensi
12	Belimbing Wuluh	diabetes
13	Belimbing Wuluh	gondongan
14	Belimbing Wuluh	hipertensi
15	Belimbing Wuluh	jerawat
16	Belimbing Wuluh	rematik
17	Belimbing Wuluh	sakit gigi
18	Belimbing Wuluh	sariawan
19	Beluntas	keputihan
20	Biji Mahoni	demam
21	Biji Mahoni	masuk angin
22	Biji Mahoni	rematik
23	Brotowali	demam
24	Brotowali	demam kuning
25	Brotowali	diabetes
26	Brotowali	kencing manis
27	Brotowali	rheumatik sendi
28	Bunga Kenop	asma
29	Bunga Kenop	batuk
30	Bunga Kenop	disentri
31	Bunga Kenop	radang mata
32	Cempaka Putih	batuk berdahak
33	Cempaka Putih	batuk rejan
34	Cempaka Putih	bau badan
35	Cempaka Putih	keputihan
36	Cempaka Putih	sinusitis
37	Cempaka Putih	vertigo
38	Cincau	demam
39	Cincau	radang lambung
40	Cincau	tekanan darah tinggi

41	Ciplukan	asma
42	Ciplukan	bisul
43	Ciplukan	gondongan
44	Ciplukan	kanker payudara
45	Ciplukan	kencing manis
46	Daun Beluntas	keputihan
47	Daun Cincau	demam
48	Daun Cincau	radang lambung
49	Daun Dewa	kanker
50	Daun Dewa	luka memar
51	Daun Dewa	pendarahan
52	Daun Dewa	tekanan darah tinggi
53	Daun Jarak	hernia
54	Daun Jarak	rematik
55	Daun Jinten	ayan
56	Daun Jinten	sakit kepala
57	Daun Katuk	demam
58	Daun Keji Beling	gatal
59	Daun Keji Beling	kencing manis
60	Daun Kumis Kucing	keputihan
61	Daun Meniran	epilepsi
62	Daun Sirih	asma
63	Daun Sirih	keputihan
64	Daun Sirih	mimisan
65	Daun Sirih	radang tenggorokan
66	Daun Sirih	sakit gigi
67	Daun Sirih	sakit mata
68	Daun Tempuyung	asam urat
69	Daun Tempuyung	bisul
70	Daun Tempuyung	darah tinggi
71	Daun Tempuyung	kencing batu
72	Daun Tempuyung	wasir
73	Keji Beling	batu ginjal
74	Keji Beling	kencing manis
75	Kembang Telang	mata merah
76	Kumis Kucing	asam urat
77	Kumis Kucing	batuk
78	Kumis Kucing	diabetes
79	Kumis Kucing	hipertensi
80	Kumis Kucing	kencing batu
81	Kumis Kucing	masuk angin
82	Lidah Buaya	diabetes
83	Lidah Buaya	kanker
84	Lidah Buaya	luka

85	Lidah Buaya	radang tenggorokan
86	Lidah Buaya	sembelit
87	Lidah Buaya	serangan jantung
88	Mengkudu	asam urat
89	Mengkudu	jantung koroner
90	Mengkudu	sakit kepala
91	Mengkudu	sakit kuning
92	Sambiloto	sakit perut
93	Seledri	asam urat
94	Seledri	hipertensi
95	Seledri	mata kering
96	Temulawak	hepatitis B
97	Temulawak	kanker hati
98	Temulawak	radang

LAMPIRAN B

#	Tanaman Herbal	Penyakit
1	Air Terjun Kedung Pedut	Kulon Progo
2	Air Terjun Sri Gethuk	Yogyakarta
3	Anambas	Kepulauan Riau
4	Bukit Cendana	Yogyakarta
5	Bukit Kelam	Kalimantan Barat
6	Bukit Klangon	Yogyakarta
7	Bukit Kosakora	Yogyakarta
8	Bukit Lintang Sewu	Yogyakarta
9	Bukit Mojo Gumelem	Yogyakarta
10	Bukit Panguk Kediwung	Yogyakarta
11	Bukit Pengilon	Yogyakarta
12	Bukit Teletubbies	Yogyakarta
13	Bunker Kaliadem	Yogyakarta
14	Candi Abang	Yogyakarta
15	Candi Muaro Jambi	Jambi
16	Candi Prambanan	Yogyakarta
17	Candi Ratu Boko	Yogyakarta
18	Candi Sari	Yogyakarta
19	Danau Sentani	Papua
20	Danau Toba	Sumatera Utara
21	Fort Marlborough	Bengkulu
22	Gereja Ayam	Magelang
23	Gili Trawangan	Nusa Tenggara Barat
24	Goa Jomblang	Yogyakarta
25	Goa Pindul	Yogyakarta
26	Green Village Gedangsari	Yogyakarta
27	Gumuk Pasir	Yogyakarta
28	Gunung Bromo	Jawa Timur
29	Gunung Kerinci	Jambi
30	Gunung Rinjani	Lombok
31	Gunung Tangkuban Perahu	Jawa Barat
32	Halmahera	Maluku Utara
33	Jalan Malioboro	Yogyakarta
34	Jembatan Barito	Kalimantan Selatan
35	Jurang Tembelan	Yogyakarta
36	Kawah Ijen	Banyuwangi
37	Kawah Putih	Bandung
38	Kebun Buah Mangunan	Yogyakarta
39	Kepulauan Karimun Jawa	Jepara
40	Kepulauan Kei	Maluku

41	Kepulauan Seribu	DKI Jakarta
42	Kota Lama Semarang	Jawa Tengah
43	Museum Benteng Vredeburg	Yogyakarta
44	Museum Sultan Mahmud Badaruddin	Palembang
45	Ngarai Sianok	Bukittinggi
46	Nusa Dua	Bali
47	Pantai Baron	Yogyakarta
48	Pantai Baru	Yogyakarta
49	Pantai Batu Lamampu	Nunukan
50	Pantai Dato Majene	Sulawesi Barat
51	Pantai Depok	Yogyakarta
52	Pantai Dreamland	Bali
53	Pantai Drini	Yogyakarta
54	Pantai Greweng	Yogyakarta
55	Pantai Indrayanti	Yogyakarta
56	Pantai Jungwok	Yogyakarta
57	Pantai Losari	Makassar
58	Pantai Ngobaran	Yogyakarta
59	Pantai Ngrumput	Yogyakarta
60	Pantai Ora	Maluku
61	Pantai Pandansimo	Yogyakarta
62	Pantai Parai Tenggara	Bangka Belitung
63	Pantai Parangtritis	Yogyakarta
64	Pantai Sadranan	Yogyakarta
65	Pantai Sepanjang	Yogyakarta
66	Pantai Seruni	Yogyakarta
67	Pantai Siung	Gunungkidul
68	Pantai Siung	Yogyakarta
69	Pantai Slili	Yogyakarta
70	Pantai Sumurtiga	Sabang
71	Pantai Sundak	Yogyakarta
72	Pantai Wediombo	Yogyakarta
73	Pelabuhan Ratu	Sukabumi
74	Pinus Pengger	Yogyakarta
75	Pulau Derawan	Kalimantan Timur
76	Pulau Misool	Papua Barat
77	Pulau Peucang	Banten
78	Puncak Becici	Yogyakarta
79	Pura Luhur Uluwatu	Bali
80	Raja Ampat	Papua Barat
81	Seribu Batu Songgo Langit	Yogyakarta
82	Sindu Kusuma Edupark	Yogyakarta
83	Spot Riyadi	Yogyakarta
84	Sungai Kampar	Riau

85	Taman Laut Bunaken	Manado
86	Taman Nasional Bromo Tengger Semeru	Jawa Timur
87	Taman Nasional Way Kambas	Lampung
88	Taman Pintar	Yogyakarta
89	Tebing Watu Mabur	Yogyakarta
90	Tugu Khatulistiwa	Pontianak
91	Tugu Yogyakarta	Yogyakarta
92	Ujung Kulon	Banten
93	Wakatobi	Sulawesi Tenggara
94	Watu Amben	Yogyakarta
95	Watu Goyang	Yogyakarta

LAMPIRAN C

No	Tag	Jenis Kata	Deskripsi
1	CC	Konjungtor koordinatif	Konjungtor koordinatif, atau disebut juga koordinator. Konjungtor koordinatif menghubungkan dua satuan bahasa atau lebih yang sederajat (kata dengan kata, frasa dengan frasa, atau klausa dengan klausa) yang masing-masing mempunyai kedudukan yang setara dalam struktur kalimat. dan, tetapi, atau
2	CD	Numeralia kardinal	Numeralia kardinal. Numeralia kardinal, yaitu numeralia yang menjadi jawaban atas pertanyaan "Berapa?", meliputi: a. bilangan pokok, misalnya dua, b. bilangan gugus, misalnya juta, c. bilangan penuh, misalnya enam dan 7916, d. bilangan pecahan, misalnya sepertiga, e. bilangan desimal, misalnya 0,025 dan 0,525, f. numeralia pokok tak tentu, misalnya banyak, g. numeralia pokok kolektif, misalnya kedua, berpuluh-puluh, dan ribuan, h. tanggal, dan i. tahun. dua, juta, enam, 7916, sepertiga, 0,025, 0,525, banyak, kedua, ribuan, 2007, 25
3	OD	Numeralia ordinal	Numeralia ordinal. Numeralia ordinal menyatakan urutan dan menjadi jawaban atas pertanyaan "Yang keberapa?" ketiga, ke-4, pertama
4	DT	Artikel	Artikel, atau disebut juga artikula. Artikel bertugas membatasi makna nomina. para, sang, si
5	FW	Kata bahasa asing	Kata bahasa asing. Kata bahasa asing adalah kata yang berasal dari bahasa asing yang belum diserap ke dalam bahasa Indonesia. Pada dasarnya, kata bahasa asing adalah kata yang tidak terdapat di dalam kamus bahasa Indonesia. Jika sebuah kata bahasa asing menjadi bagian dari proper noun atau nama, kata bahasa asing tersebut diberi POS tag NNP. climate change, terms and conditions
6	IN	Preposisi (kata depan)	Preposisi, atau disebut juga kata depan. Preposisi menghubungkan kata atau frasa dengan konstituen di depan preposisi tersebut sehingga terbentuk frasa preposisional. dalam, dengan, di, ke, oleh, pada, untuk
7	JJ	Adjektiva (kata sifat)	Adjektiva, atau disebut juga kata sifat. Adjektiva adalah kata yang memberikan keterangan yang lebih khusus tentang sesuatu yang

No	Tag	Jenis Kata	Deskripsi
			dinyatakan oleh nomina dalam kalimat. Adjektiva meliputi: a. sifat, misalnya bersih, b. ukuran, misalnya panjang dan kecil, c. warna, misalnya hitam, d. waktu atau durasi, misalnya lama, e. jarak, misalnya jauh, f. sikap batin atau perasaan, misalnya marah, g. cerapan, yaitu adjektiva yang bertalian dengan pancaindra, misalnya manis, h. keanggotaan dalam suatu golongan, misalnya nasional, dan i. bentuk, misalnya bulat. bersih, panjang, hitam, lama, jauh, marah, suram, nasional, bulat
8	MD	Verba modal dan verba bantu	Verba modal dan verba bantu (modal and auxiliary verb) boleh, harus, sudah, mesti, perlu
9	NEG	Kata ingkar	Kata ingkar. tidak, belum, jangan
10	NN	Nomina (kata benda)	Nomina, atau disebut juga kata benda. Nomina, yaitu kata yang mengacu pada manusia, binatang, benda, konsep, atau pengertian, meliputi: a. flora dan fauna, misalnya monyet, b. nomina lokatif dan nomina yang menunjukkan tempat atau arah, misalnya bawah, c. nomina penunjuk waktu, misalnya sekarang, dan d. mata uang yang tidak ditulis dalam bentuk simbol, misalnya rupiah. monyet, bawah, sekarang, rupiah
11	NNP	Proper noun	Proper noun. Proper noun adalah nama spesifik dari seseorang, sesuatu, atau sebuah tempat. Proper noun meliputi: a. nama diri atau individu, misalnya Boediono, b. nama tempat geografis, misalnya Laut Jawa, c. nama negara, wilayah, atau daerah pemerintahan, misalnya Indonesia, d. nama organisasi, institusi, atau perusahaan, misalnya Bank Mandiri, e. kode emiten, misalnya BBKP, f. nama bulan dalam setahun, misalnya Januari, g. nama hari dalam sepekan, misalnya Senin, h. nama hari raya, misalnya Idul Fitri, i. nama kompetisi, kejuaraan, ajang j. judul karya, acara televisi, atau film, misalnya Lord of the Rings: The Return of the King. Proper noun yang ditulis dalam bahasa asing diberi POS tag NNP. Proper noun yang ditulis dalam bentuk singkatan atau inisial diberi POS tag NNP. Jika proper noun terdiri atas lebih dari satu kata atau bagian, setiap kata atau bagian dari proper noun tersebut diberi POS tag NNP. penghargaan, atau peristiwa bersejarah, misalnya Piala Dunia, dan

No	Tag	Jenis Kata	Deskripsi
			Boediono, Laut Jawa, Indonesia, India, Malaysia, Bank Mandiri, BBKP, Januari, Senin, Idul Fitri, Piala Dunia, Liga Primer, Lord of the Rings: The Return of the King
12	NND	Penggolong atau nomina ukuran	Penggolong atau nomina ukuran. Penggolong menempatkan nomina ke dalam sebuah kelompok tertentu dalam jumlah tertentu, misalnya orang dalam dua orang prajurit. Nomina ukuran merujuk pada ukuran, jarak, volume, kecepatan, berat, atau temperatur, misalnya ton. orang, ton, helai, lembar
13	PR	Demonstrativa atau pronomina penunjuk	Demonstrativa atau pronomina penunjuk. ini, itu, sini, situ
14	PRP	Pronomina persona	Pronomina persona. Pronomina persona, yaitu pronomina yang dipakai untuk mengacu pada orang, meliputi: a. pronomina persona pertama tunggal, misalnya saya, b. pronomina persona pertama jamak eksklusif, misalnya kami, c. pronomina persona pertama jamak inklusif, misalnya kita, d. pronomina persona kedua tunggal, misalnya kamu, e. pronomina persona kedua jamak, misalnya kalian, f. pronomina persona ketiga tunggal, misalnya dia, dan g. pronomina persona ketiga jamak, misalnya mereka. saya, kami, kita, kamu, kalian, dia, mereka
15	RB	Adverbia (kata keterangan)	Adverbia, atau disebut juga kata keterangan. sangat, hanya, justru, niscaya, segera
16	RP	Partikel	Partikel. Dalam penelitian ini, POS tag RP menandai partikel penegas, yaitu partikel yang digunakan untuk menegaskan kalimat interogatif, imperatif, atau deklaratif. pun, -lah, -kah
17	SC	Konjungtor subordinatif	Konjungtor subordinatif, atau disebut juga subordinator. Konjungtor subordinatif menghubungkan dua klausa atau lebih dan salah satu dari klausa-klausa tersebut merupakan klausa subordinatif. sejak, jika, seandainya, supaya, meski, seolah-olah, sebab, maka, tanpa, dengan, bahwa, yang, lebih ... daripada ..., semoga

No	Tag	Jenis Kata	Deskripsi
18	SYM	Simbol	Simbol. Simbol, yang diberi POS tag SYM, meliputi simbol matematis, misalnya +, dan simbol mata uang, misalnya IDR. IDR, +, %, @
19	UH	Interjeksi	Interjeksi. Interjeksi mengungkapkan rasa hati atau perasaan pembicara dan secara sintaktis tidak berhubungan dengan kata-kata lain di dalam kalimat atau ujaran. brengsek, oh, ooh, aduh, ayo, mari, hai
20	VB	Verba (kata kerja)	Verba, atau disebut juga kata kerja. Verba, yang diberi POS tag VB, dapat berupa verba transitif, verba intransitif, verba aktif, verba pasif, dan kopula. Jika kata dasar dari verba berimbuhan berupa kata bahasa asing, verba berimbuhan tersebut tetap diberi POS tag VB, misalnya di-arrange. merancang, mengatur, pergi, bekerja, tertidur
21	WH	Pronomina penanya (kata tanya)	Pronomina penanya, atau disebut juga kata tanya. Pronomina penanya digunakan dalam kalimat interogatif sebagai pemarah pertanyaan. Klausa tanya yang ada di dalam kalimat deklaratif merupakan klausa subordinatif sehingga pronomina penanya yang menghubungkan klausa tanya dengan klausa lain di dalam kalimat deklaratif menjadi konjungtor subordinatif dan diberi POS tag SC. siapa, apa, mana, kenapa, kapan, di mana, bagaimana, berapa
22	X	tidak diketahui	Kata atau bagian dari kalimat yang tidak diketahui atau belum diketahui secara pasti kategorinya. POS tag X juga digunakan untuk menandai kesalahan penulisan (typo). statemen
23	Z	Tanda baca	Tanda baca. "...", ?, .