

BAB 1

PENDAHULUAN

Bab ini membahas mengenai latar belakang penelitian, perumusan masalah, tujuan dan manfaat dilakukannya penelitian, ruang lingkup penelitian, dan terakhir metodologi yang digunakan.

1.1 Latar Belakang

Berbagai penelitian yang berkaitan dengan *Information Retrieval* dan *Natural Language Processing* mulai muncul menggunakan Bahasa Indonesia. Informasi mengenai kata dan artinya sangat dibutuhkan. Sayangnya, *resource* yang dimiliki berbasis Bahasa Indonesia masih sangat terbatas. Informasi tersebut umumnya disimpan dalam kamus digital seperti WordNet. Penelitian ini dilaksanakan untuk mengembangkan *resource* dalam Bahasa Indonesia.

WordNet adalah kamus digital yang dapat digunakan untuk menunjang penelitian di bidang *Information Retrieval* dan *Natural Language Processing*. WordNet yang paling sering digunakan dalam berbagai penelitian adalah WordNet Princeton berbahasa Inggris yang dibentuk secara manual oleh ahli linguistik. Setiap *entry* pada WordNet disimpan dalam bentuk set sinonim atau biasa disebut *synset* dan arti dari *synset* tersebut atau biasa disebut *sense*. Informasi lain yang disimpan dalam suatu *synset* adalah relasi antar *synset*. Relasi semantik yang tersimpan dalam WordNet adalah *synonym*, *antonym*, *hypernym*, *hyponym*, *holonym*, *meronym*, *troponym*, dan *entailment*.

Beberapa penelitian telah dilakukan untuk membangun WordNet Bahasa Indonesia. WordNet Bahasa Indonesia yang telah ada adalah Indonesian WordNet (IWN) yang dibuat oleh Fakultas Ilmu Komputer Universitas Indonesia (Fasilkom UI) dan WordNet Bahasa yang dibuat oleh Nanyang Technology University (NTU). Salah satu kelemahan kedua WordNet tersebut adalah ukurannya yang masih sangat terbatas. Selain itu, informasi mengenai relasi kata juga belum dapat tersimpan secara baik. Kedua WordNet memetakan *synset* Bahasa Indonesia ke WordNet Princeton dan memanfaatkan relasi yang di dalamnya. Mengetahui hal tersebut, dilakukan penelitian yang dapat mengekstrak relasi antar kata dengan hanya menggunakan korpus Bahasa Indonesia. Proses ekstraksi berjalan secara cepat dan data yang dihasilkan berjumlah besar. Korpus yang dihasilkan diharapkan dapat berguna

untuk penelitian selanjutnya.

Relasi kata adalah salah satu hal penting yang perlu diketahui jika ingin mengetahui hubungan antar kata secara semantik. Informasi yang berkaitan dengan semantik atau arti kata sulit diperoleh tanpa adanya pengetahuan sebelumnya. Kata-kata yang mirip secara leksikal belum tentu berelasi secara semantik. Sementara kata-kata yang tidak memiliki kesamaan secara leksikal bisa memiliki arti yang mirip atau berhubungan secara semantik. Korpus relasi yang dibuat diharapkan dapat membantu memperoleh informasi tersebut. Selain itu, pengetahuan mengenai relasi kata dapat dimanfaatkan dalam berbagai penelitian lain seperti *question answering* (Ravichandran dan Hovy, 2002), *information extraction*, dan *anaphora resolution*.

Melihat adanya kebutuhan akan korpus relasi kata, dilakukanlah penelitian *word relation extraction*. Penelitian ini berusaha mengekstrak kata berdasarkan relasi tertentu dari suatu dokumen sehingga dihasilkan korpus relasi kata. Penelitian kali ini akan fokus pada relasi kata *hypernym-hyponym*. Keduanya menyatakan relasi antara kata yang lebih umum (*hypernym*) dengan kata yang lebih khusus (*hyponym*). Metode yang digunakan adalah *pattern matching* dengan memanfaatkan korpus Wikipedia Bahasa Indonesia. Wikipedia memuat banyak kata dari berbagai domain sehingga dapat dimanfaatkan untuk membuat *pattern* yang general serta menghasilkan korpus relasi berukuran besar.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, pertanyaan yang menjadi rumusan penelitian adalah sebagai berikut.

1. Bagaimana cara membangun korpus relasi secara cepat dan berkualitas baik secara otomatis?
2. Apakah metode *pattern extraction* dan *matching* baik dilakukan untuk ekstraksi relasi kata Bahasa Indonesia?
3. Bagaimana cara mengevaluasi *pattern* dan korpus relasi kata dari hasil eksperimen?

1.3 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian *word relation extraction* ini adalah membangun korpus relasi kata Bahasa Indonesia berukuran besar dan berkualitas baik secara otomatis. Selain

itu, ingin diketahui pula apakah metode *pattern extraction* dan *matching* baik digunakan untuk mengekstrak relasi kata Bahasa Indonesia. Diharapkan korpus relasi kata yang dihasilkan dapat menunjang berbagai penelitian selanjutnya.

Penelitian ini juga diharapkan dapat memotivasi adanya penelitian selanjutnya di bidang *Language Resource Development*, terutama pembangunan WordNet Bahasa Indonesia. Penelitian mengenai ekstraksi relasi kata berikutnya dengan berbagai metode lain diharapkan terus dilaksanakan sehingga Bahasa Indonesia memiliki *resource* yang semakin baik.

1.4 Ruang Lingkup Penelitian

Penelitian ini hanya fokus pada pembuatan korpus pasangan kata dengan relasi *hypernym* dan *hyponym* Bahasa Indonesia. Kelas kata yang menjadi fokus penelitian adalah kata benda (*noun*). Pengembangan korpus dilakukan secara *semi-supervised* dengan metode *pattern matching*. *Pattern* yang dibuat masih terbatas hanya merupakan *pattern* leksikal. Evaluasi akan dilakukan pada *pattern* dan korpus pasangan kata yang dihasilkan. Proses evaluasi korpus pasangan kata dilakukan menggunakan teknik *random sampling*. Data yang digunakan untuk pembuatan *pattern* maupun untuk ekstraksi korpus pasangan kata baru adalah Wikipedia Bahasa Indonesia. Pemilihan relasi *hypernym-hyponym* dalam penelitian karena memiliki berbagai manfaat. Relasi tersebut dapat mengidentifikasi *Proper Noun* baru yang belum diketahui maknanya.

1.5 Tahapan Penelitian

Proses penelitian dilakukan dalam beberapa tahapan sebagai berikut.

1. Studi Literatur

Pada tahap ini, dilakukan pembelajaran mengenai penelitian-penelitian sebelumnya yang telah dilakukan di bidang *word relation extraction* sehingga diketahui langkah yang perlu diambil selanjutnya.

2. Perumusan Masalah

Perumusan masalah dilakukan untuk mendefinisikan masalah yang ingin diselesaikan, tujuan penelitian, dan hasil yang diharapkan sehingga proses penelitian dapat berjalan dengan baik.

3. Perancangan Penelitian

Setelah diketahui hasil yang ingin dicapai, dirancang tahap-tahap eksperimen

secara terstruktur. Hal-hal yang diperhatikan mulai dari pengumpulan korpus awal (*seed*), *pre-processing* dokumen, perancangan implementasi *pattern extraction matching*, hingga proses evaluasi.

4. Implementasi

Implementasi dilaksanakan sesuai dengan rancangan penelitian untuk menjawab rumusan masalah. Segala hasil yang ditemukan digunakan untuk terus memperbaiki metode dan teknik penelitian sehingga didapatkan hasil yang semakin baik.

5. Analisis dan Kesimpulan

Tahap terakhir dari penelitian ini adalah menganalisis korpus pasangan kata relasi yang dihasilkan. Pertanyaan dari perumusan masalah dijawab, kemudian ditarik kesimpulan.

BAB 2

TINJAUAN PUSTAKA

Pada bab ini, dijelaskan mengenai studi literatur yang dilakukan. Studi literatur yang dilakukan digunakan sebagai dasar konsep dan teknik penelitian. Dipaparkan pula berbagai istilah dan metode yang digunakan dalam penelitian.

2.1 WordNet

WordNet adalah kamus leksikal yang tersimpan secara digital dan digunakan untuk berbagai keperluan komputasi (Miller, 1995). Pembuatan WordNet dilatarbelakangi keperluan mendapatkan *sense* atau arti semantik suatu kata. Informasi tersebut perlu disimpan dan dapat dibaca oleh mesin. WordNet pertama dibuat oleh Miller (1995) berbasis Bahasa Inggris dan sekarang dikenal dengan nama Princeton WordNet (PWN). WordNet menyimpan informasi dalam bentuk database dimana setiap entry-nya adalah pasangan *synset* dan arti semantiknya (*sense*). Set sinonim (*synset*) adalah himpunan kata yang memiliki arti yang sama atau saling berelasi *synonym*.

WordNet mengandung beberapa kelas kata seperti kata benda (*noun*), kata kerja (*verb*), kata sifat (*adjective*), dan kata keterangan (*adverb*). WordNet juga menyimpan informasi mengenai relasi semantik antar *synset*. Relasi yang disimpan adalah *synonymy*, *antonymy*, *hyponymy*, *hypernym*, *meronymy*, *holonymy*, *troponymy*, dan *entailment*.

2.1.1 Indonesian WordNet

Penelitian mengenai WordNet Bahasa Indonesia pernah dilakukan sebelumnya oleh Desmond Darma Putra (2008) dan Margaretha dan Manurung (2008). Indonesian WordNet (IWN) dibangun menggunakan metode mapping antara WordNet yang sudah ada ke dalam Bahasa Indonesia (Desmond Darma Putra, 2008). WordNet yang digunakan sebagai dasarnya adalah Princeton WordNet. Synset dalam PWN akan dipetakan ke dalam entry Kamus Besar Bahasa Indonesia (KBBI), sehingga menghasilkan hasil yang berkualitas baik secara cepat dan mudah. Penelitian tersebut menghasilkan 1441 synset dan 3074 sense. Relasi semantik antar synset diperoleh dengan memetakan IWN synset dengan PWN synset, sehingga relasi yang dimiliki dalam PWN dapat diturunkan.

2.1.2 WordNet Bahasa

Pengembangan WordNet untuk Bahasa Indonesia juga dilakukan oleh Nanyang Technology University (NTU) sejak tahun 2011 dan diberi nama WordNet Bahasa (Nurhil Hirfana Mohamed Noor, 2011). WordNet ini telah diintegrasikan dengan salah satu *tools* NLP berbasis Python yaitu nltk sehingga dapat dengan mudah digunakan dalam komputasi. WordNet Bahasa juga memanfaatkan PWN untuk mendapatkan relasi semantik antar *synset*. Pada penelitian ini, dimanfaatkan *tools* tersebut untuk mendapatkan *seed* relasi semantik antar kata dalam Bahasa Indonesia.

2.1.3 Kelemahan WordNet Bahasa Indonesia

Hingga saat ini, relasi semantik antar kata yang dimiliki oleh WordNet Bahasa Indonesia merupakan hasil turunan dari relasi semantik WordNet Princeton. Sayangnya, pemanfaatan PWN untuk mendapatkan relasi semantik kata Bahasa Indonesia menemui beberapa hambatan. WordNet Bahasa Indonesia menjadi sangat tergantung dengan struktur PWN untuk mendapatkan relasi semantik suatu *synset*. Selain itu, beberapa *synset* Bahasa Indonesia juga tidak dapat dipetakan secara tepat ke *synset* PWN, beberapa kata kehilangan arti atau mendapat arti yang kurang tepat. Hal ini menyebabkan pasangan kata relasi semantik Bahasa Indonesia yang dihasilkan terlihat kurang baik. Untuk itu, dicetuskanlah penelitian untuk mengekstrak relasi semantik dalam Bahasa Indonesia secara mandiri.

2.2 Relasi

Relasi menggambarkan hubungan atau koneksi yang dimiliki oleh suatu hal dengan yang lain (KBBI). Dalam bidang matematika, relasi memetakan suatu anggota dari himpunan satu ke himpunan lain sesuai dengan hubungan yang didefinisikan. Dalam penelitian ini, relasi yang diperhatikan adalah relasi semantik antar kata. Domainnya adalah kata-kata yang tergabung dalam kelas kata benda (*noun*).

Satu relasi dapat terdiri dari beberapa entitas dan dituliskan dalam bentuk tuple $t = (e_1, e_2, \dots, e_n)$ dimana e_i adalah suatu entitas yang memiliki relasi r dalam dokumen D (Bach dan Badaskar, 2007). Relasi sinonim dapat ditulis dalam notasi tersebut. Banyak pula relasi yang hanya menghubungkan antar dua entitas (relasi biner), seperti *terletak-di*(Universitas Indonesia, Depok) atau *ditulis-oleh*(Habib Gelap Terbitlah Terang, RA Kartini).

2.2.1 Relasi Semantik

Semantik adalah arti (*sense*) dari suatu kata. Umumnya informasi tersebut disimpan dalam kamus bahasa. Relasi semantik adalah hubungan yang dimiliki antar kata berdasarkan arti atau makna dari kata tersebut. Beberapa relasi semantik adalah sebagai berikut (Miller, 1995).

- *Synonymy* adalah relasi antar kata dimana dua kata yang berbeda memiliki arti yang sama. Semua kelas kata dapat memiliki relasi *synonymy*. Dalam WordNet, relasi ini direpresentasikan dalam bentuk *synset* dan bersifat simetris. Sebagai contoh 'makan', 'melahap', dan 'menyantap' memiliki makna yang sama.
- *Antonymy* adalah yang menggambarkan arti yang saling berkebalikan antar kata. Umumnya relasi ini digunakan pada kelas kata sifat (*adverb*) dan kata keterangan (*adjective*). Sama seperti *synonymy*, relasi ini memiliki sifat simetris. Sebagai contoh kata 'tinggi' memiliki makna yang berkebalikan dengan kata 'pendek'.
- *Hyponymy* adalah relasi yang menyatakan hubungan kata yang lebih khusus. Sementara untuk kata yang lebih umum dikenal dengan relasi *hypernym*. Kedua relasi ini diperuntukan kelas kata benda (*noun*) dan umumnya satu kata memiliki hanya satu *hypernym*. Kedua relasi ini bersifat transitif, sehingga dapat digambarkan dalam bentuk hirarki. Sebagai contoh kucing, ikan, kelinci (*hyponymy*) adalah binatang (*hypernym*). Binatang adalah *hyponym* dari makhluk hidup. Sehingga dapat dikatakan pula bahwa kucing, ikan, kelinci (*hyponym*) adalah makhluk hidup (*hypernym*).
- *Meronymy* dan *holonym* adalah relasi yang menyatakan hubungan bagian satu dengan yang lain, dimana *meronym* menyatakan sub-bagian dan *holonym* menyatakan bagian yang lebih besar. Seperti relasi *hyponym-hypernym*, relasi *meronym-holonym* bersifat transitif dan dapat digambarkan dalam bentuk hirarki. Dalam WordNet, relasi ini dibagi ke dalam tiga bagian yaitu *part-meronym*, *member-meronym*, dan *substance-meronym*. Sebagai contoh sebuah sel (*holonym*) memiliki nukleus, ribosom, mitokondria (*meronym*).
- *Troponymy* adalah relasi seperti *hyponymy-hypernymy* yang khusus untuk kelas kata kerja (*verb*). Dalam Bahasa Inggris, contoh kata yang memiliki relasi ini adalah 'stroll' dan 'walk'.

2.3 Relation Extraction

Relation extraction adalah cabang dari *Information Extraction* (IE) yang berfokus pada proses ekstraksi relasi antar kata. Proses ini berusaha mengekstrak informasi terstruktur dengan definisi yang diinginkan dari teks dokumen atau resource yang tidak terstruktur. Beberapa contoh penelitian ini seperti *named entity recognition* (NER) yang mengetahui apakah suatu entitas adalah orang, organisasi, atau lokasi (Daniel M. Bikel, 1999).

2.3.1 Semantic Relation Extraction

Semantic relation extraction mengkhususkan pada proses ekstraksi relasi semantik antar kata. Penelitian dalam bidang ini sudah banyak dilakukan dengan berbagai metode. Salah satu metode populer untuk mendapatkan relasi semantik satu domain bahasa adalah menggunakan *pattern extraction* dan *pattern matching* seperti yang telah dilakukan pada penelitian (Hearst, 1992), (Ruiz-Casado dkk, 2005), dan (Arnold dan Rahm, 2014). Penelitian lain memanfaatkan distribusi kata untuk memperoleh *semantic distance* antar kata. *Semantic distance* adalah nilai yang merepresentasikan kedekatan antar kata berdasarkan semantiknya.

Penelitian yang dilakukan oleh Hearst merupakan salah satu penelitian awal ekstraksi relasi semantik yang menggunakan metode *pattern extraction* dan *matching*. Hearst menggunakan *lexico-syntactic pattern* untuk mendapatkan pasangan kata relasi tanpa membutuhkan pengetahuan sebelumnya dan dapat diaplikasikan dalam teks apapun. Pada awal, Hearst mendefinisikan dua *pattern* berdasarkan hasil observasi dari teks. Selanjutnya, *pattern* baru didapat menggunakan langkah berikut.

1. Tentukan relasi yang akan diamati dan kumpulkan entitas yang menggambarkan relasi tersebut.
2. Dalam teks dokumen, cari lokasi dimana entitas-entitas tersebut berada dan simpan kata-kata diantaranya (lingkungan).
3. Cari kesamaan dari teks yang terekstraksi dan bentuk suatu *pattern* baru.
4. Jika *pattern* baru telah terbukti benar, gunakan *pattern* tersebut untuk mendapatkan entitas baru.

Proses evaluasi dilakukan dengan membandingkan entitas yang dihasilkan dengan synset dalam WordNet.

Penelitian yang dilakukan oleh Ruiz-Casado dkk, memanfaatkan WordNet dan Wikipedia sebagai korpus untuk mendapatkan pasangan kata relasi. Pertama dilakukan *entry sense disambiguation* yang merupakan tahap *pre-processing* untuk memetakan setiap entri dalam Wikipedia dengan *synset* dalam WordNet. Tahap berikutnya adalah ekstraksi *pattern* antara dua konsep. Jika terdapat dua konsep yang saling berhubungan dan memiliki suatu relasi semantik dalam WordNet maka kalimat yang mengandung dua konsep tersebut akan disimpan. Proses tersebut menghasilkan daftar relasi semantik dengan masing-masing memiliki *pattern* di dalamnya. Dari banyak *pattern* yang dihasilkan, proses selanjutnya adalah *pattern generalisation* yang bertujuan membuat *pattern* yang lebih umum. Tahap ini memanfaatkan algoritma *edit-distance* dengan modifikasi. Setelah mendapatkan *pattern*, tahapan terakhir adalah menggunakannya ke dalam korpus untuk mendapatkan entitas baru.

Penelitian yang dilakukan Arnold dan Rahm juga memanfaatkan korpus Wikipedia dan menggunakan *pattern*. Selain *hypernym-hyponymy* dan *holonym-meronymy*, penelitian ini juga mengidentifikasi relasi *synonymy*. Kalimat definisi pada Wikipedia di-parse menggunakan *Finite State Machine* (FSM) dan konsep-konsep baru diekstrak menggunakan *pattern* yang telah didefinisikan. Penelitian lain yang dilakukan oleh Sumida dan Torisawa [9] memanfaatkan struktur internal Wikipedia dalam mengekstraksi *pattern* relasi untuk Bahasa Jepang.

2.4 Korpus

Korpus adalah data yang dengan format yang dapat dibaca oleh mesin dan memiliki fungsi spesifik (Atkins et al., 1991).

2.4.1 Korpus Pasangan Kata Relasi Semantik

Saat ini, belum ada korpus independen yang menggambarkan relasi semantik antar kata dalam Bahasa Indonesia. Relasi semantik Bahasa Indonesia yang ada saat ini didapatkan dari hasil relasi turunan pemetaan *synset* Bahasa Indonesia dengan PWN.

Korpus relasi semantik yang ingin dibuat menyimpan informasi dalam bentuk pasangan kata yang berelasi serta relasi yang menghubungkannya. Selanjutnya, pasangan kata berelasi akan disebut sebuah *pair*. Untuk relasi yang bersifat transitif seperti *hypernym-hyponym*, ada identifikasi kata mana yang merupakan *hypernym* dan mana yang merupakan *hyponym*. Relasi *hypernym* dan *hyponym* adalah relasi bersifat transitif yang dapat direpresentasikan dalam bentuk hirarki. Pada penelitian

ini, korpus *pair* yang diperoleh merepresentasikan hubungan yang berjarak tepat satu dari yang lain.

2.5 Wikipedia

Wikipedia adalah ensiklopedia terbuka yang memuat berbagai bahasa dan merupakan hasil kolaborasi berbagai penulis (Doneyer dan Gallinari, 2006). Wikipedia adalah salah satu korpus teks dokumen terbesar yang disimpan secara *online* yang dapat diakses dan diunduh secara bebas. Wikipedia dikelola oleh organisasi non-profit bernama Wikimedia Foundation. Pada tahun 2009, jumlah kontributor Wikipedia Bahasa Indonesia telah mencapai 2.502 pengguna aktif. Walau ditulis oleh berbagai narasumber, informasi yang dimuat dalam Wikipedia dibuat secara terstruktur dengan bahasa yang formal. Wikipedia juga memuat informasi umum terbaru (Arnold dan Rahm, 2014).

2.5.1 Korpus Wikipedia

Wikipedia Bahasa Indonesia saat ini telah memuat lebih dari 400.000 artikel dari berbagai domain. Artikel-artikel yang disimpan dalam Wikipedia dapat diunduh secara gratis dalam bentuk *dumps* dengan format XML. Beberapa tipe *dump* dapat dipilih, diantaranya adalah halaman seluruh artikel, halaman artikel beserta *revision history*, daftar judul artikel, dan lainnya. Secara berkala, Wikimedia membuat *dump* terhadap seluruh artikel terakhir yang disimpan untuk setiap bahasa. Pada situs yang menyediakan pengunduhan data Wikipedia, terdapat tanggal unik yang menyatakan tanggal terakhir data tersebut di-*update*.

2.5.2 Wikipedia sebagai Lexical Semantic Resource

Artikel dalam Wikipedia terdiri dari berbagai domain kategori dan memuat berbagai *noun* umum. Walau ditulis secara kolaboratif, Wikipedia memuat artikel secara terstruktur dimana pada paragraf pertama umumnya berisi kalimat-kalimat definisi mengenai topik yang sedang dibahas. Bahasa yang ditulis juga cukup formal dan terstruktur. Selain itu, sudah banyak penelitian ekstraksi relasi semantik dengan metode pattern yang menggunakan artikel Wikipedia seperti [8], [7].

2.6 Pattern

Pattern adalah suatu bentuk yang dapat merepresentasikan kumpulan data berukuran besar. Pada teks dokumen, *pattern* dapat terbentuk secara eksplisit ataupun implisit. Secara eksplisit seperti *lexico-syntactic pattern* yang terbentuk dari kata-kata di dalam dokumen tersebut. Sementara *pattern* terbentuk secara implisit jika dilihat dari kemiripan vektor yang merepresentasikan dokumen atau kata-kata di dalam dokumen tersebut.

2.6.1 Lexico-syntactic Pattern

Lexico-syntactic pattern adalah *pattern* yang hanya memanfaatkan kata-kata dalam korpus dokumen dan dimanfaatkan untuk proses *string matching*. Salah satu *pattern* leksikal yang terkenal adalah Hearst Pattern yang merepresentasikan pola-pola untuk mendapatkan relasi *hyponymy* dari dokumen (Hearst, 1992). Menurut Hearst, beberapa syarat yang harus dipenuhi suatu *lexico-syntactic pattern* yang baik adalah sebagai berikut.

- Kemunculannya sering pada teks dalam berbagai domain sehingga dapat mengekstrak banyak entitas.
- Merepresentasikan relasi yang diinginkan sehingga hasil ekstraksi juga benar.
- Dapat dikenali tanpa membutuhkan pengetahuan sebelumnya sehingga dapat dibentuk dalam situasi apapun.

2.7 Pattern Extraction dan Matching

Pattern extraction atau *pattern recognition* adalah salah satu cabang dalam *machine learning* yang berusaha mencari pola kemiripan tertentu dari kumpulan data yang diberikan. *Pattern matching* adalah proses untuk mencocokkan suatu *pattern* dengan kumpulan data yang belum dianotasi. Penelitian ekstraksi relasi semantik menggunakan metode *pattern matching* telah dilakukan oleh (Hearst, 1992), (Ruiz-Casado dkk, 2005), (Arnold dan Rahm, 2014), dan (Sumida dan Torisawa, [9]).

Banyaknya penelitian menggunakan metode *pattern matching* dan *extraction* menjadi salah satu alasan penggunaan metode tersebut untuk mengekstrak relasi kata dalam Bahasa Indonesia. Pada penelitian ini, relasi semantik yang diekstrak dibatasi hanya untuk relasi *hypernym-hyponym*.

2.8 Sequence Analysis

Dalam bidang bioinformatik, *sequence analysis* digunakan untuk mengetahui apakah suatu DNA atau RNA mengandung *sequence* tertentu. Pada penelitian ini, *pattern* yang dihasilkan adalah *pattern* leksikal yang merupakan deretan kata-kata. Algoritma seperti *standard trie* dan *suffix tree* dimanfaatkan secara berurutan untuk proses *pattern extraction* dan *matching*.

2.8.1 Standard Trie

Standard Trie untuk suatu himpunan *string* S adalah *ordered tree* dengan ketentuan berikut.

- Setiap *node*, selain *root*, diberi label sebuah *character*
- *Children* dari sebuah *node* terurut sesuai alfabet
- *Path* dari eksternal *node* hingga *root* membentuk suatu *string* dalam S .

Standard Trie membutuhkan memori sebesar $O(n)$ dimana n adalah total ukuran *string* dalam S . Operasi *insert* butuh waktu $O(dm)$ dimana m adalah ukuran *string* yang baru dan d adalah ukuran alfabet.

Sebagai contoh berikut adalah *standard trie* yang dibangun dari himpunan *string* apel, ayam, baju, baja, bara.

2.8.2 Suffix Tree

Suffix tree sering digunakan untuk pencarian *sequence* yang panjang seperti *genomes* untuk bidang bioinformatik. Pembentukan *suffix tree* mirip seperti *Standard Trie*, namun untuk seluruh *suffix* dalam *string*. Jika diberikan *string* dengan panjang n , dibentuk cabang dengan $n(n-1)/2$ *suffix*. Metode ini banyak dimanfaatkan untuk mempercepat proses pencarian jika diberikan sebuah masukan *query*. Jika terdapat sebuah *pattern* dengan panjang *string* m , maka waktu yang dibutuhkan untuk menjalankan proses *pattern matching* adalah $O(dm)$ dengan d adalah ukuran alfabet. Proses pencarian dilakukan dengan menelusuri *path* dari *root* sesuai dengan *sequence query*. Jika seluruh karakter dalam *query* selesai dijalankan, maka proses pencarian berhasil.

Sebagai contoh *string* 'babaa' menghasilkan *suffix tree* berikut. Jika diberi *query* 'ba' maka akan berhasil terhadap *path* 'babaa' dan 'baa'.

2.9 Semi Supervised

Dalam *machine learning* terdapat dua tipe pendekatan yang umum digunakan yaitu *supervised* dan *unsupervised learning*. *Supervised* menggunakan data berlabel sebagai data *training* maupun *testing*. Dari kedua data tersebut, dibentuk suatu *classifier* yang dapat memenuhi segala kasus yang mungkin terjadi. Data *testing* digunakan untuk menguji kebaikan *classifier* yang terbentuk. *Unsupervised* menggunakan data yang tidak diberi label sama sekali dan berusaha untuk menemukan pola yang sama untuk suatu kumpulan data tertentu (10). Pendekatan lain yang merupakan kombinasi antara *supervised* dan *unsupervised learning* adalah *semi supervised learning*.

Semi supervised adalah pendekatan *machine learning* dimana informasi *supervised* data diberikan tidak untuk seluruh data. Sebagian data merupakan data berlabel sementara sebagian lainnya belum memiliki label. Beberapa metode penerapan semi supervised adalah *bootstrapping (self training)*, *mixture models*, *graph based methods*, *co-training*, dan *multiview learning*. (10) A Survey on Semi-Supervised Learning Techniques. V. Jothi Prakash, Dr. L.M. Nithya. International Journal of Computer Trends and Technology. 2014 (<http://www.ijcttjournal.org/Volume8/number-1/IJCTT-V8P105.pdf>) (12) MITPress - Semi Supervised Learning

2.9.1 Bootstrapping

Model *bootstrapping* merupakan salah satu model *semi supervised learning* yang paling umum digunakan. *Bootstrapping* menggunakan data berlabel berukuran kecil dan data tidak berlabel berukuran jauh lebih besar. Proses anotasi data tidak berlabel dilakukan secara bertahap melalui sejumlah iterasi. Dari data *training* berlabel, dibentuk suatu *classifier* yang kemudian digunakan untuk menganotasi data tidak berlabel. Sejumlah k data baru yang merupakan hasil pelabelan, dimasukkan ke dalam kelompok data berlabel. Proses tersebut dilakukan secara berulang, sehingga semakin lama iterasi jumlah data berlabel akan bertambah.

Terdapat dua algoritma *bootstrapping* yang pernah digunakan untuk proses *pattern extraction* dan *matching* yaitu *Meta-Bootstrapping* dan *Basilisk* (Rillof dkk, 2003). Keduanya digunakan untuk mengelompokkan kata ke dalam suatu kategori semantik jika diberikan korpus teks yang belum dianotasi dan suatu *seed*. *Seed* didefinisikan sebagai korpus kata yang sudah diketahui kategori semantiknya. Secara umum, proses ini akan mencari *pattern* berdasarkan *seed* yang diberikan. Dari *pattern* yang dihasilkan dan teks yang belum dianotasi, diekstrak entitas baru dan dikelompokkan berdasarkan kategori semantiknya. Kata-kata tersebut akan digu-

bungkan ke dalam korpus pasangan kata berelasi.

2.9.2 Meta Bootstrapping

Berikut adalah beberapa proses (Riloff dan Jones, 1999) yang dijalankan algoritma *meta bootstrapping* jika diberikan *seed* berukuran kecil yang berasal dari suatu kategori semantik dan korpus yang belum dianotasi.

1. Mengekstraksi *pattern* secara otomatis dengan menerapkan syntactic template.
2. Untuk setiap *pattern* akan diberi bobot berdasarkan jumlah *seed* yang menghasilkan *pattern*.
3. Diambil *pattern* terbaik dan seluruh *seed* lama yang merepresentasikan *pattern* maupun *seed* baru yang berhasil diekstrak disimpan.
4. Dilakukan pembobotan ulang untuk setiap *pattern* menggunakan *seed* lama dan baru.

Proses diatas dinamakan *mutual bootstrapping* dan setelah proses tersebut selesai, semua entitas baru hasil ekstraksi dievaluasi. Pembobotan entitas baru diberikan berdasarkan jumlah *pattern* yang mengekstrak kata tersebut. Lima kata terbaik diterima dan dimasukkan ke kamus (korpus) kata berelasi untuk selanjutnya diproses ulang.

2.9.3 Basilisk

Algoritma *Basilisk* (Thelen dan Riloff, 2002) juga memanfaatkan *pattern* dan *seed* dalam membangun korpus untuk suatu kategori semantik tertentu. Beberapa tahapan yang dijalankan adalah sebagai berikut.

1. Secara otomatis membentuk *pattern* dan memberi bobot berdasarkan jumlah *seed* yang menghasilkan *pattern*. *Pattern* terbaik dimasukan ke dalam *Pattern Pool*.
2. Untuk setiap entitas baru yang terekstraksi dari *pattern*, dimasukan ke dalam *Candidate Word Pool*. Pemberian bobot dilakukan berdasarkan jumlah *pattern* yang mengekstraksi dan asosiasi kumulatif kata dengan *seed*.
3. Sepuluh kata terbaik diambil dan dimasukan ke dalam kamus (korpus) yang kemudian digunakan untuk iterasi selanjutnya.

Kategori semantik untuk proses ini bisa lebih dari satu. *Basilisk* memberi bobot berdasarkan informasi kolektif dari kumpulan *pattern* yang mengekstrak kata tersebut. Sementara *Meta-Bootstrapping* hanya mengambil satu *pattern* terbaik dan mengelompokkan seluruh kata yang terekstrak dari *pattern* ke dalam kategori semantik yang sama. Dari hasil penelitian komparatif yang pernah dilakukan (Riloff dkk, 2003), didapatkan *Basilisk* mengungguli performa *Meta-Bootstrapping*. (11) Learning Subjective Noun using Extraction Pattern Bootstrapping. Riloff, Ellen, etc. Seventh Conference on Natural Language Learning. (<https://www.cs.utah.edu/~riloff/pdfs/conll03.pdf>) [19] A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Context (<https://www.cs.utah.edu/~riloff/pdfs/emnlp02-thelen.pdf>)

2.10 Word Embedding

Word Embedding digunakan untuk menentukan similarity antar kata relasi yang dihasilkan dari proses *Pattern Matching*.

2.11 Pointwise Mutual Information

Pointwise Mutual Information (PMI) adalah pengukuran nilai asosiasi antar variabel. Dalam bidang *information theory*, PMI dapat dimanfaatkan untuk menghitung asosiasi kemunculan dua buah kata. Jika diberikan dua buah kata x dan y , maka nilai PMI kata tersebut dalam suatu dokumen dapat dihitung menggunakan rumus berikut.

$$pmi(x;y) = \log \frac{p(x,y)}{(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}$$

dengan $p(x) = \frac{f(x)}{N}$

$$pmi(x;y) = \log \frac{f(x)N}{f(x)f(y)}$$

- $p(x)$ adalah probabilitas kemunculan kata x dalam korpus
- $f(x)$ adalah frekuensi kemunculan kata x dalam korpus
- N adalah toatal seluruh kata dalam korpus

Pengukuran ini bersifat simetris, sehingga $p(x;y) = p(y;x)$. Nilai PMI dapat merupakan bilangan positif maupun negatif. Jika nilai PMI adalah nol (0), berarti kedua variabel saling *independent*.

2.11.1 Skip PMI

PMI umumnya hanya menggunakan model *bigram* atau *trigram*. Model ini hanya melihat hubungan kata yang berdampingan. Sebagai contoh ingin diketahui PMI untuk *bigram* 'hong kong', 'sepak bola', dan 'amerika serikat'. Pada penelitian ini dilakukan modifikasi yaitu membuat model *skip-gram* PMI. Kita menghitung nilai PMI antar dua kata yang dipisahkan dengan n diantaranya.

2.12 Evaluasi

Evaluasi dilakukan untuk mengetahui kebaikan hasil penelitian. Evaluasi dapat dilakukan dengan mengukur akurasi data yang dihasilkan. Akurasi adalah nilai perbandingan antara jumlah data yang benar dengan jumlah seluruh data (Manning).

$$akurasi = \frac{jumlah\ data\ benar}{jumlah\ seluruh\ data}$$

Selain menghitung akurasi, proses evaluasi juga menghitung nilai-nilai lainnya. Berikut ada beberapa metode dan teknik evaluasi lain yang digunakan dalam penelitian.

2.12.1 Sampling

Terdapat dua kategori utama dalam *sampling* yaitu *probability* dan *non-probability sampling*. Perbedaan utama keduanya adalah pada *probability sampling*, diambil data secara acak (*random*). Dalam *probability sampling*, terdapat beberapa metode yang dapat digunakan seperti *simple random sampling*, *systematic sampling*, *stratified random sampling*, dan *cluster sampling*.

- *Simple random sampling* perlu mengetahui seluruh data yang ada dan dari data tersebut dipilih secara acak. Hal ini membuat seluruh data memiliki nilai probabilitas terpilih yang sama.
- *Systematic sampling* memilih setiap data ke- n untuk dijadikan *sample*.
- *Stratified random sampling* akan mengelompokkan data ke dalam kategori berdasarkan karakteristik tertentu (*strata*), kemudian data diambil secara acak dari kategori yang ada. Hal ini menyebabkan hasil lebih representatif.
- *Cluster sampling* mirip seperti *stratified sampling* namun dilakukan jika data kelompok yang ingin di-*sampling* sulit berada di lokasi yang terpisah jauh.

Proses *sampling* bermanfaat untuk merepresentasikan data tanpa perlu mengevaluasi seluruh data yang ada. Jika jumlah data yang ingin dievaluasi berukuran besar, proses *sampling* mempercepat pengukuran. Jumlah data yang direpresentasikan oleh satu *sample* berdasarkan jumlah data asli. Sebagai contoh jika total data adalah 1000 dan jumlah data *sample* adalah 50, maka satu data *sample* merepresentasikan 20 data asli. (Sumber: <https://ecduganda.files.wordpress.com/2014/08/how-to-choose-sampling-techniques-for-evaluations.pdf>)

2.12.2 Precision dan Recall

Teknik yang umum digunakan untuk mengevaluasi suatu ekstraksi adalah *precision* dan *recall*. *Precision* adalah nilai yang menyatakan jumlah dokumen benar dan berhasil diambil dibandingkan dengan seluruh jumlah dokumen yang terambil. *Recall* adalah nilai yang menyatakan jumlah dokumen benar dan berhasil diambil dibandingkan dengan jumlah seluruh dokumen yang benar. Semakin banyak dokumen yang diambil maka nilai *recall* akan meningkat sementara nilai *precision* cenderung menurun.

2.12.3 Kappa

Nilai kappa (κ) merepresentasikan tingkat persetujuan antar anotator. Kappa digunakan pada penelitian yang menggunakan bantuan anotator untuk memberi penilaian secara manual. Penilaian didapatkan menggunakan rumus berikut.

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

- $P(A)$ adalah proporsi penilaian yang setuju (*agreement*)
- $P(E)$ adalah proporsi penilaian yang kebetulan

Landis dan Koch [18] mendefinisikan tingkat persetujuan berdasarkan nilai Kappa yang diperoleh.

Tabel 2.1: Skala pengukuran Kappa

| Statistik Kappa | Tingkat persetujuan |
|-----------------|-----------------------|
| < 0.00 | <i>Poor</i> |
| 0.00 - 0.20 | <i>Slight</i> |
| 0.21 - 0.40 | <i>Fair</i> |
| 0.41 - 0.60 | <i>Moderate</i> |
| 0.61 - 0.80 | <i>Substantial</i> |
| 0.81 - 1.00 | <i>Almost Perfect</i> |

Beberapa variasi perhitungan untuk Kappa adalah Cohen's Kappa dan Fleiss' Kappa. Cohen's Kappa digunakan untuk mengukur tingkat persetujuan antar dua anotator. Jika diberikan data dengan n label dan m_{ij} merepresentasikan jumlah data yang diberi label i oleh anotator pertama dan label j oleh anotator kedua, maka proses perhitungan $P(A)$ dan $P(E)$ untuk Cohen's Kappa adalah sebagai berikut.

$$P(A) = \frac{\sum_{k=1}^n m_k k}{\text{total data}}$$

$$P(E) = \frac{\sum_{k=1}^n (\sum_{j=1}^n m_{kj} \cdot \sum_{i=1}^n m_{ik})}{\text{total data}}$$

Fleiss' Kappa mengukur tingkat persetujuan antar sekelompok anotator berjumlah lebih dari dua. Jika diberikan N data dengan n anotator dimana setiap data diantosi ke dalam salah satu dari k kategori dan n_{ij} merepresentasikan total anotator yang memberi data i ke label j , proses perhitungan $P(A)$ dan $P(E)$ untuk Fleiss' Kappa adalah sebagai berikut.

$$P(A) = \frac{1}{N} \sum_{i=1}^N P_i \text{ dengan } P_i = \frac{1}{n(n-1)} [(\sum_{j=1}^k n_{ij}^2 j) - (n)]$$

2.12.4 Spearman's Rho

Spearman's rank correlation coefficient adalah nilai koefisien korelasi antar *ranking* dua parameter. Nilai *Spearman correlation* sama dengan nilai *Pearson correlation* antar dua parameter yang telah di-*ranking*. *Pearson correlation* menggambarkan nilai linear antara dua parameter. *Spearman correlation* berkisar antara -1 hingga $+1$.

Spearman's rho adalah nilai Pearson correlation coefficient antar dua variabel yang telah di-*ranking*. Untuk mendapatkan nilai koefisien (r_s), menggunakan rumus

berikut.

$$r_s = \rho_{rg_X, rg_Y} = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X} \sigma_{rg_Y}}$$

- ρ adalah *Pearson correlation coefficient* yang diaplikasikan pada variabel *ranking*
- $cov(rg_X, rg_Y)$ adalah nilai *covariance* antar variabel *ranking*
- σ_{rg_X} dan σ_{rg_Y} adalah nilai standard deviasi variabel *ranking*

Jika seluruh *ranking* berbeda, proses komputasi dapat dilakukan menggunakan rumus berikut.

$$r_s = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)}$$

- $d_i = rg(X_i) - rg(Y_i)$ adalah selisih antara dua *ranking*
- n adalah jumlah observasi

BAB 3

APLIKASI YANG DIGUNAKAN

Pada bab ini dipaparkan mengenai rancangan dan tahap-tahap proses ekstraksi relasi semantik, mulai dari pengumpulan seed, proses pattern extraction dan matching, dan evaluasi hingga dihasilkan korpus pasangan relasi kata hypernym-hyponym Bahasa Indonesia.

3.1 Spesifikasi Penelitian

Berikut adalah spesifikasi penelitian yang dilaksanakan dan batas-batasnya.

- Relasi yang diperhatikan adalah *hypernym-hyponym*.
- Kelas kata yang diperhatikan adalah *noun* dan *proper noun*.
- Kata yang diekstrak termasuk *multi word*. Contoh: sepak bola, Amerika Serikat, dan lainnya.

3.2 Rancangan Pengembangan Korpus

Pengembangan korpus pasangan kata berelasi yang digunakan pada penelitian ini menggunakan arsitektur yang dapat dilihat pada gambar 3.1. Terdapat dua sumber data utama yang digunakan yaitu WordNet Bahasa untuk pembentukan *seed* dan artikel Wikipedia Bahasa Indonesia sebagai korpus teks. Secara garis besar, terdapat enam tahap yang perlu dilakukan yaitu pembuatan *seed*, preprocessing data Wikipedia, *sentence tagging*, *pattern extraction*, *pattern matching*, dan terakhir adalah evaluasi. Untuk proses *sentence tagging*, *pattern extraction*, dan *pattern matching* dilakukan secara berulang, sesuai dengan metode *bootstrapping*. Berikut adalah penjelasan singkat setiap tahapan.

1. Pembentukan *seed* dilakukan untuk mendapatkan pasangan kata berelasi yang digunakan sebagai dasar penelitian. Proses ini memanfaatkan *resource* yang dimiliki WordNet Bahasa.
2. Artikel Wikipedia diperoleh dalam bentuk *Wikipedia dump* dan perlu diolah sehingga memiliki representasi sesuai dengan format yang diharapkan.

Informasi yang diperlukan hanya bagian isi artikel dan kemudian ditulis berdasarkan kalimat untuk setiap baris.

3. Menggunakan *seed* dan korpus Wikipedia, dilakukan *tagging* pasangan kata berelasi terhadap kalimat-kalimat yang ada. Kalimat yang mengandung pasangan kata berelasi akan ditandai dan disimpan sebagai dasar proses selanjutnya.
4. Kalimat-kalimat yang sudah di-tag dengan pasangan kata berelasi kemudian akan digunakan untuk proses *pattern extraction*. Hasil dari proses ini adalah sejumlah *pattern* leksikal terbaik dari banyak *pattern* yang dihasilkan.
5. Proses berikutnya adalah *pattern matching* dimana *pattern* hasil ekstraksi dan korpus Wikipedia kembali digunakan untuk membentuk pasangan kata relasi baru (*pair*).
6. Korpus pasangan relasi kata yang terbentuk digunakan untuk iterasi selanjutnya sesuai dengan metode *bootstrapping*. Proses iterasi dilakukan hingga pasangan kata relasi baru yang dihasilkan jenuh. Terakhir dilakukan evaluasi untuk mengetahui akurasi data yang dihasilkan.

Proses ini diharapkan dapat menghasilkan korpus pasangan kata relasi *hyponym-hypernym* yang berkualitas baik dan berukuran besar. *Pair* yang dihasilkan ditulis dalam bentuk *tuple* ($kata_{hyponym}; kata_{hypernym}$) dengan kedua kata berada dalam kelas kata benda.

3.3 Pre-processing Data

Proses inti dari penelitian ini, *pattern extraction* dan *matching*, memerlukan dua masukan utama yaitu sejumlah pasangan kata *hypernym-hyponym* dan teks dokumen yang digunakan sebagai korpus. Pasangan kata *hypernym-hyponym* digunakan untuk proses pembentukan *pattern* sementara teks dokumen digunakan untuk memperoleh pasangan kata baru. Dikarenakan belum ada korpus pasangan kata relasi *hypernym-hyponym* Bahasa Indonesia, perlu didefinisikan *seed* yang akan digunakan sebagai dasar pasangan kata *hypernym-hyponym*. Teks dokumen yang digunakan, yaitu Wikipedia, juga memerlukan pemrosesan sebelum digunakan.

3.3.1 Pre-processing Data Wikipedia

Data Wikipedia yang diperoleh dari Wikimedia *dumps* masih mengandung banyak *tag* yang tidak digunakan pada penelitian ini seperti *tag id* dan *revision*. Se-

lain itu simbol-simbol khusus (*markup*). Penelitian ini ingin mengekstrak *pattern* dari *free text*, sehingga format-format khusus tersebut perlu dibersihkan. Setelah data Wikipedia dibersihkan dari simbol-simbol tersebut, langkah selanjutnya adalah merepresentasikan korpus dalam bentuk kalimat.

Artikel-artikel Wikipedia dibentuk ke dalam format yang telah didefinisikan dengan satu kalimat dipisahkan untuk setiap barisnya. Data tersebut juga digunakan untuk proses *part-of-speech tagging*. Hasil dari *pre-processing* data Wikipedia adalah dua korpus besar yaitu korpus tanpa *pos tag* yang digunakan sebagai masukan *pattern extraction* dan korpus dengan *pos tag* yang digunakan sebagai masukan *pattern matching*.

3.3.2 Pembentukan Seed

Pasangan kata relasi *hypernym-hyponym* diambil dari data yang dimiliki oleh WordNet Bahasa yang dikembangkan oleh NTU. Pemanfaatan WordNet Bahasa dilatarbelakangi jumlahnya yang lebih besar dibanding Indonesian WordNet (IWN). Relasi semantik antar kata pada WordNet Bahasa memanfaatkan WordNet Princeton versi 3.0. Synset pada WordNet Bahasa dipetakan ke *synset* WordNet Princeton, sehingga relasi kata yang dimiliki oleh WordNet Princeton ikut diwarisi. Alasan lain penggunaan WordNet Bahasa karena telah terintegrasi dengan *tools nltk* sehingga dapat langsung digunakan untuk membentuk *seed* secara mudah.

Seluruh lema Bahasa Indonesia yang dimiliki oleh WordNet tersebut akan dipasangkan dengan lema *hypernym*-nya, sehingga terbentuk relasi biner antara kata yang merupakan *hyponym* dan kata yang merupakan *hypernym*. Format untuk korpus ini mengikut format korpus pasangan kata relasi yang akan digunakan.

BAB 4

PERANCANGAN IMPLEMENTASI DAN ANALISIS

Bab ini menjelaskan secara detail proses implementasi dari pengolahan data, proses *pattern extraction* dan *matching*, iterasi *semi-supervised*, dan proses evaluasi.

4.1 Pembentukan Korpus Kalimat Wikipedia

Data hasil Wikipedia Dump diformat secara otomatis menggunakan *tools* WikiExtractor. WikiExtractor adalah program berbasis Python yang dibuat oleh Giuseppe Attardi dan Antonio Fuschetto. Menggunakan tools ini, didapatkan data yang sudah tidak mengikuti format MediaWiki Markup Language. Program ini dapat diunduh dari Github dan dijalankan menggunakan perintah berikut.

```
$ WikiExtractor.py xml-dump-file -o output-file
```

Jika tidak memberi spesifikasi opsi apapun, artikel yang dihasilkan membersihkan seluruh markup language dan hanya menyimpan isi artikel tanpa disertakan informasi seperti kategori, riwayat, dan versi artikel.

4.1.1 Sentence Splitter

Korpus yang dihasilkan menghasilkan baris-baris yang merepresentasikan suatu paragraf dalam artikel Wikipedia. Pada penelitian kali ini, ingin dilihat relasi *hypernym-hyponym* antar dua kata pada kalimat yang sama. Untuk itu, perlu dilakukan proses *Sentence Splitter* yang dapat mengidentifikasi suatu kalimat dalam paragraf. Hasil dari proses tersebut adalah dokumen yang terdiri dari baris-baris yang merepresentasikan satu kalimat.

Proses ini dilakukan dengan menggunakan *script* yang telah dibuat sebelumnya oleh Ken Nabila Setya dari Fasilkom UI, Indonesia. Ditambah pula satu *script* yang dapat secara otomatis melakukan *splitting* untuk seluruh dokumen dalam folder. Sebuah paragraf yang dihasilkan menggunakan WikiExtractor adalah sebagai berikut.

Charles Anthony Johnson (3 Juni 1829 - 17 Mei 1917), kemudian dikenal sebagai Charles Brooke memerintah Sarawak sebagai Raja Putih kedua dari 3 Agustus 1868 hingga meninggal dunia. Dia menggantikan pamannya, James Brooke sebagai raja.

Setelah dihalankan *sentence splitter*, berikut adalah hasilnya.

| |
|--|
| Charles Anthony Johnson (3 Juni 1829 - 17 Mei 1917), kemudian dikenal sebagai Charles Brooke memerintah Sarawak sebagai Raja Putih kedua dari 3 Agustus 1868 hingga meninggal dunia. |
|--|

| |
|---|
| Dia menggantikan pamannya, James Brooke sebagai raja. |
|---|

4.1.2 Rule Based Formatter

Dari korpus yang berisi kalimat-kalimat, diberikan beberapa aturan tambahan untuk membuat korpus sesuai format yang diinginkan. Penambahan aturan juga untuk mengurangi ambiguitas dan bentuk usaha melakukan generalisasi *pattern*. Berikut adalah beberapa aturan tambahan untuk *pre-processing* korpus Wikipedia.

1. Menghilangkan frase yang berada di dalam tanda kurung.
Frase yang terletak di dalam tanda kurung dapat dianggap sebagai penjelas kata atau frasa sebelumnya. Proses ini merupakan salah satu upaya generalisasi *pattern*.
2. Memisahkan simbol-simbol yang berhimpit pada awal dan akhir kata.
Beberapa token yang dipisahkan oleh spasi dalam kalimat merupakan kata yang berhimpit dengan tanda baca. Untuk mempermudah proses selanjutnya yaitu *sentence tagging*, dilakukan *pre-processing* tambahan yaitu memisahkan simbol-simbol *non-alphanumeric*.
3. Memberi penanda awal kalimat dengan '<start>' dan akhir kalimat dengan '<end>'.
Pemberian simbol awal dan akhir kalimat memperjelas isi kalimat dan juga menunjang proses *pattern extraction* dan *pattern matching*.

Dari contoh kalimat diatas, setelah di proses menggunakan aturan yang didefinisikan member hasil berikut.

| |
|--|
| <start> Charles Anthony Johnson , kemudian dikenal sebagai Charles Brooke memerintah Sarawak sebagai Raja Putih kedua dari 3 Agustus 1868 hingga meninggal dunia . <end> |
|--|

| |
|---|
| <start> Dia menggantikan pamannya , James Brooke sebagai raja . <end> |
|---|

4.2 POS Tagging Kalimat Wikipedia

Proses POS Tagging dilakukan pada korpus Wikipedia yang telah berbentuk kalimat dengan format yang didefinisikan. Pada penelitian ini, kelas kata yang menjadi

pengamatan adalah *noun* (NN) dan *proper noun* (NNP), sehingga proses *pos tagging* dapat menunjang identifikasi kata-kata tersebut. Proses *pos tagging* menggunakan *tools* Stanford POS Tagger. Model yang digunakan juga merupakan hasil penelitian sebelumnya yang menggunakan Bahasa Indonesia. Setelah selesai melalui proses tagging, dilakukan penyesuaian agar format korpus lebih rapi dan terstruktur.

Berikut adalah contoh kalimat yang sudah melalui tahap POS Tagging. Korpus ini digunakan untuk proses *pattern matching*.

```
<start>_X Charles_NNP Anthony_NNP Johnson_NNP ,_Z kemudian_CC
dikenal_VB sebagai_IN Charles_NNP Brooke_NNP memerintah_VB
Sarawak_NNP sebagai_IN Raja_NNP Putih_NNP kedua_CD dari_IN 3_CD
Agustus_NNP 1868_CD hingga_IN meninggal_VB dunia_NN ._Z <end>_X
```

4.3 Seed Builder

Proses pengumpulan *seed* diawali dengan mengambil seluruh lema Bahasa Indonesia yang dimiliki oleh korpus nltk. Setelah itu, ambil seluruh *synset* yang mengandung lemma tersebut. Dari setiap *synset*, ambil relasi *hypernym*-nya. Dari setiap *synset* relasi, ambil lema Bahasa Indonesianya. Dilakukan pula filterisasi *synset* ataupun lema untuk mengurangi ambiguitas. Untuk setiap *synset* maupun lema yang diambil pada setiap tahapan, hanya boleh berasal dari kelas kata kerja (*noun*). Setelah didapatkan, bentuk ke dalam pasangan *tuple* 1-1 (*kata_hyponym,kata_hyponym*).

4.3.1 Filterisasi Seed

Filterisasi dilakukan dengan tujuan mengurangi ambiguitas, namun tetap berusaha mendapatkan *seed* yang cukup banyak. Filterisasi juga dilakukan untuk mendapatkan *seed* awal yang diyakini benar dan berkualitas. Salah satu bagian terpenting proses ini adalah memasang hanya lema yang merupakan *noun* ke lema yang juga adalah *noun*. Jika kemungkinan lema tersebut tergolong ke dalam kelas kata bukan *noun*, lema tidak diikutsertakan sebagai *seed* awal.

Salah satu tantangan dalam proses ini adalah banyak ditemukan kasus dimana satu lema berasal dari lebih dari satu *synset* atau satu *synset* memiliki lebih dari satu *synset hypernym*. Ambiguitas dalam kasus ini dapat mengurangi kualitas *seed* yang dihasilkan. Mengatahui hal tersebut, dibuatlah dua pendekatan berbeda untuk proses filterisasi *seed*.

Pendekatan pertama adalah tetap mengambil lemma yang sama pada *synset* *hypernym* yang berbeda. Hal ini dilatarbelakangi adanya lema yang berasal dari *synset* berbeda namun memiliki lema *hypernym* yang sama. Pada contoh (i), satu lema yang sama dimiliki oleh dua *synset* yang berbeda namun kedua *synset* tersebut memiliki *synset hypernym* yang sama. Pada contoh (ii), satu lema berasal dari dua *synset* yang berbeda dan dua *synset hypernym* berbeda, namun ada lema yang sama yaitu "lalu-laluan".

Pendekatan lainnya adalah dengan metode filterisasi yang paling ketat. Jika satu lema memiliki lebih dari satu *synset hypernym*, maka lema tersebut dianggap ambigu dan langsung tidak diikutsertakan ke dalam *seed* awal. Berdasarkan tabel contoh lema, *synset*, dan *hypernym*-nya, hanya contoh (i) yang diterima sebagai *seed* karena *synset hypernym* untuk lema tersebut sama. Sementara (ii) ditolak karena *synset hypernym* berbeda.

| | |
|-----|--|
| i. | ('paruh', Synset('beak.n.02')) => (['bibir', 'kuala', 'muara'], [Synset('mouth.n.02')]) ('paruh', Synset('beak.n.01')) => (['bibir', 'kuala', 'muara'], [Synset('mouth.n.02')]) ('paruh', Synset('beak.n.01')) => (['bibir', 'kuala', 'muara'], [Synset('mouth.n.02')]) |
| ii. | ('pintu_masuk', Synset('entrance.n.01')) => (['akses', 'capaian', 'lalu-laluan'], [Synset('access.n.03')]) ('pintu_masuk', Synset('orifice.n.01')) => (['koridor', 'lalu-laluan', 'lorong'], [Synset('passage.n.07')]) |

DAFTAR REFERENSI

- Bach, N. dan Badaskar, S. (2007). A review of relation extraction. *Literature review for Language and Statistics II*.
- Daniel M. Bikel, Richard Schwartz, R. M. W. (1999). An algorithm that learns what's in a name.
- Desmond Darma Putra, Abdul Arfan, R. M. (2008). Building an indonesian wordnet.
- Margaretha, E. dan Manurung, R. (2008). Comparing the value of latent semantic analysis on two english-to-indonesian lexical mapping tasks.
- Miller, G. A. (1995). Wordnet: A lexical database for english.
- Nurhil Hirfana Mohamed Noor, Suerya Sapuan, F. B. (2011). Creating the open wordnet bahasa.
- Ravichandran, D. dan Hovy, E. (2002). Learning surface text patterns for a question answering system.