
WORD SENSE DISAMBIGUATION DENGAN ALGORITMA

LESK (SIMPLIFIED LESK)

¹Muh. Wildan Suyuti (1210651248), ²Bagus Setya Rintyarna, M.kom, ³Triawan Adi Cahyanto, M.kom
Program Studi Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Jember Email :
em.wildan.es@gmail.com

Abstrak

Ambiguitas makna pada kalimat bahasa Inggris adalah gejala terjadinya tafsiran lebih dari satu makna. Hal ini dapat terjadi baik dalam ujaran lisan maupun tulisan. Tafsiran lebih dari satu ini dapat menimbulkan keraguan dan kebingungan dalam mengambil keputusan tentang makna yang dimaksud. Untuk menaggulangi permasalahan ini digunakan metode pencarian makna terhadap kata ambigu pada kalimat bahasa Inggris dengan menggunakan algoritma *simplified lesk*. Algoritma ini digunakan karena kesederhanaannya dalam menentukan makna ambigu, yaitu menghitung jumlah kesamaan kata antara konteks kalimat dengan gloss (definisi) makna kata ambigu beserta examples (contoh) kalimat yang diambil dari WordNet. Algoritma ini digunakan untuk menentukan makna kata ambigu pada masing-masing kalimat yang terdapat pada sample dataset yang telah disediakan sebanyak 100 kalimat bahasa Inggris. Kemudian evaluasi dilakukan dengan tujuan mencari precision, recall, f-measure dan accuracy dari algoritma *simplified lesk*. Dari hasil evaluasi didapatkan precision dari algoritma *simplified lesk* adalah 46,5%, recall 40%, f-measure 43% dan accuracy 47%.

Kata Kunci : Bahasa Inggris, Ambiguitas, Simplified Lesk, WordNet

1. PENDAHULUAN

1.1 Latar Belakang

Bahasa Inggris merupakan bahasa yang umum digunakan di seluruh dunia. Bahasa Inggris telah menjadi sarana komunikasi internasional dan dipakai oleh sekitar 350.000.000 orang sebagai bahasa pertama dan sekitar 400.000.000 orang sebagai bahasa kedua atau bahasa asing (Kitao, 1996) dan jumlah orang yang memakai bahasa Inggris akan terus bertambah sejalan dengan pertumbuhan penduduk dunia [1].

Oleh karena itu didalam ilmu pengetahuan, bahasa Inggris sering digunakan pada berbagai sumber informasi seperti artikel-artikel dari internet, jurnal, paper, maupun buku elektronik dari berbagai negara belahan dunia. Hal ini tentu sangat membantu bagi masyarakat terutama bagi kalangan pelajar maupun mahasiswa, untuk melengkapi referensi tentang ilmu yang mereka pelajari. Namun hal ini bukan tanpa kendala, terkadang beberapa kalimat dalam bahasa Inggris mempunyai makna ambigu / kata yang memiliki makna lebih dari satu.

Sehingga agak sedikit menyulitkan untuk dipahami.

Maka dari itu diperlukan sebuah solusi untuk menentukan makna pada kalimat tersebut agar dapat lebih dimengerti oleh pembaca. Dalam bidang computing *Word Sense Disambiguation* (WSD) yang merupakan salah satu topik yang cukup penting dalam *Natural*

Language Processing (NLP), dimana bertujuan untuk menentukan makna dari suatu kata, terutama pada kata ambigu. Seperti contoh dalam bahasa Inggris, kata *call* dapat berarti *command to come*, dapat juga berarti *a telephone connection*, bahkan dalam *wordnet* ditemukan ± 30 makna *call*, di sinilah peranan WSD untuk memilih dari 30 makna *call*, manakah makna yang paling sesuai, tentu saja saat kata tersebut dimasukkan dalam konteks kalimat [2].

Berbagai macam metode dikembangkan untuk WSD salah satu metode yang digunakan pada WSD adalah algoritma *Simplified Lesk* yang diperkenalkan oleh Kilgarriff dan Rosenzweig pada tahun 2000. Algoritma ini membandingkan definisi

kata ambigu (*gloss*) beserta *examples* (contoh) kalimat dari wordnet dengan konteks kalimat dari kata ambigu tersebut [3]. Hasil yang diharapkan pada penelitian ini adalah, sebuah sistem yang dapat menentukan makna kata yang tepat pada sebuah kalimat bahasa Inggris. Agar kalimat tersebut dapat lebih mudah dimengerti dan dipahami oleh pembaca.

2. TINJAUAN PUSTAKA

2.1 Ambiguitas (Ambiguity)

Menurut Kamus besar bahasa Indonesia ambiguitas adalah : /am-bi-gu-i-tas/n 1 sifat atau hal yang bermakna dua; kemungkinan yang mempunyai dua pengertian; 2 ketidaktentuan; ke-tidakjelasan; 3 kemungkinan adanya makna atau penafsiran yang lebih dari satu atas suatu karya sastra; 4 *Ling* kemungkinan adanya makna lebih dari satu dalam sebuah kata, gabungan kata, atau kalimat; ketaksaan. Ambiguitas makna terbagi menjadi ambiguitas fonetik, gramatikal, dan lexical.

Ambiguitas fonetik adalah keambiguan yang terjadi akibat dari kesamaan bunyi – bunyi yang diucapkan dan biasanya banyak terjadi dalam dialog atau percakapan sehari – hari seperti contoh berikut : Dia datang kemari memberi tahu. Kalimat yang bergaris bawah menimbulkan keambiguan karena memiliki banyak tafsir, yaitu : Apakah dia datang memberi tahu yang terbuat dari kacang kedelai, atau Apakah dia datang memberi suatu informasi. Untuk mengetahui arti atau makna kalimat tersebut secara keseluruhan, maka harus mendengarkan pembicaraan secara utuh [3].

Ambiguitas gramatikal adalah keambiguan yang terjadi melalui proses pembentukan suatu ketatabahasaan baik pembentukan kata, prasa, maupun kalimat. Kata-kata atau frasa yang memiliki keambiguitasan jenis ini akan hilang jika dimasukkan dalam konteks kalimat seperti contoh berikut : Orang tua, frasa orang tua jika tidak dimasukkan ke dalam kalimat akan memiliki dua makna yaitu orang yang sudah tua dan orang yang telah melahirkan kita [4].

Ambiguitas leksikal adalah keambiguitasan yang terjadi karena faktor dari kata itu sendiri seperti contoh kata lari dalam kalimat berikut :

- a. Kata lari memiliki arti yang berbeda bila berada dalam kalimat yang berbeda.
- b. Rani lari memutar lapangan karena terlambat.
- c. Rani berlari karena takut dengan anjing.
- d. Rani berlari mengejar bus yang menuju sekiolahnya.

- e. Rani ingin lari dari kenyataan yang pahit ini.

Penjelasan mengenai pengertian kalimat ambigu dan contoh kalimatnya dapat terjadi karena beberapa faktor, yaitu morfologi, sintaksis, dan structural [4].

2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah salah satu bidang ilmu komputer, kecerdasan buatan, dan bahasa (linguistik) yang berkaitan dengan interaksi antara komputer dan bahasa alami manusia, seperti bahasa Indonesia atau bahasa Inggris. Tujuan utama dari studi NLP adalah membuat mesin yang mampu mengerti dan memahami makna bahasa manusia lalu memberikan respon yang sesuai [5].

Sejarah NLP dimulai pada tahun 1950-an, meskipun telah ada penelitian NLP pada tahun-tahun sebelumnya. Pada tahun 1950, Alan Turing (bapak ilmu komputer) mempublikasikan artikel terkenalnya yang berjudul “*Computing Machinery and Intelligence*” yang di dalamnya Alan Turing mengusulkan tes yang sekarang disebut dengan *Turing Test*. Tes Turing adalah sebuah tes yang mengukur kemampuan mesin (dalam hal ini program komputer) untuk menunjukkan perilaku cerdas. Dalam ilustrasi contoh aslinya, seorang juri manusia akan terlibat dalam percakapan dengan manusia dan mesin yang akan dites. Semua peserta dipisahkan satu sama lain. Jika juri tidak bisa membedakan antara manusia dan mesin, maka mesin tersebut dikatakan lulus tes [5].

2.2.1 Istilah Di bidang NLP

Istilah *Natural Language Processing* sendiri mengacu pada sejumlah teknik untuk *automated generation*, manipulasi dan analisis terhadap bahasa natural (bahasa manusia). Meski sebagian besar teknik NLP diturunkan dari tema *Linguistics* dan *Artificial Intelligence*, teknik ini juga dipengaruhi oleh *Machine Learning*, *Computational Statistics* dan *Cognitive Science*.

Didalam bidang NLP ada beberapa istilah didalamnya beberapa diantaranya yaitu :

1. **Token** : Sebelum proses dijalankan, teks perlu dikelompokkan menjadi unit-unit linguistik seperti kata, tanda petik, angka atau alphanumerics. Unit-unit kecil ini dinamakan dengan token.
2. **Sentence** : Sekumpulan token-token yang berurutan, dalam bahasa Indonesia disebut sebagai kalimat.
3. **Tokenization** : Proses memecah kalimat (*sentence*) menjadi *token-token*.

4. **Corpus** : *Corpus* didenisikan sebagai kumpulan teks dalam jumlah yang sangat besar, bisa berbentuk mentah atau terkelompok, bisa fokus ke satu tema atau terdiri dari berbagai tema. Contoh: *Brown Corpus*, merupakan corpus pertama, terbesar dan sudah dikelompokkan berdasar genre. *Web text corpus*, misal *reviews*, *forums*, dsbnya. *Reuters*, merupakan corpus berita, *Inaugural corpus*, yang berisi pidato inaugural presiden Amerika, dan *udhr corpus*, *corpus multilingual*. Apa kegunaan dari corpus ?
- to create linguistic resources (lexicon, annotated corpora, etc
 - to train statistical tools (taggers, parsers, etc.
 - to evaluate existing tools (taggers, parsers, etc.

5. **Part-of-speech (POS) Tag** : Sebuah kata dapat dikelompokkan ke dalam satu atau lebih kumpulan lexical atau kategori *part-of-speech* seperti *Nouns*, *Verbs*, *Adjectives*, *Articles*. POS tag adalah simbol yang mewakili kategori leksikal NN (*Noun*), VB (*Verb*), JJ (*Adjective*), dan AT (*Article*). Salah satu POS tag yang paling tua dan sering dipakai adalah Brown Corpus. Kita akan membahas corpus ini di bagian lain.

6. **Parse Tree** : *Tree* menggambarkan syntactic structure sebuah kalimat yang tersusun dari struktur grammar formal (resmi) [6].

2.3 Word Sense Disambiguation

Word Sense Disambiguation (WSD) adalah kemampuan mengidentifikasi arti kata - kata dalam kalimat tertentu. Hal ini yang merepresentasikan riset fundamental pada *Natural Language Processing* (NLP) / pemrosesan bahasa alami [7]. Penelitian ini dimulai di era tahun 1940an, di tahun 1949 Zipf mengusulkan teori "*Law Of Meaning*". Terori ini menyatakan hukum hubungan diantara kata-kata yang sering digunakan dan kata-kata yang tidak terlalu sering digunakan. Kata-kata yang sering digunakan memiliki beberapa sense daripada kata-kata yang jarang digunakan. Kemudian hubungan ini di konfirmasi untuk *British National Corpus*. Pada tahun 1950, Kaplan mendefinisikan bahwa dalam konteks tertentu dua kata di kedua sisi dari sebuah kata yang ambigu setara dengan seluruh konteks kalimat.

Pada tahun 1957 Masterman mengusulkan teori

untuk menemukan arti sebenarnya pada sebuah kata menggunakan judul pada sebuah kategori yang dipresentasikan di *Roget's International Thesaurus*. Pada tahun 1975, Wilks mengembangkan model pada "preference semantics" dimana pembatasan selectional dan semantik leksikal berbasis bingkai yang digunakan untuk menemukan arti kata yang ambigu. Riger dan Small di tahun 1979 mempunyai revolusi ide individu "*word expert*". Pada tahun 1980 ada perkembangan yang luar biasa di bidang penelitian WSD penelitian berskala besar sumber *lexical* dan *corpora* tersedia selama kurun waktu tersebut. Akibatnya para peneliti mulai menggunakan prosedur ekstraksi pengetahuan otomatis yang berbeda (Wilks et al., 1990) serupa dengan methodology *handcrafting* [8]. Pada tahun 1986 algoritma lesk mulai diusulkan berdasarkan definisi kamus (*gloss*) untuk mendisambiguasi kata polysemous dalam konteks kalimat. Tujuan utama dari Idenya adalah untuk menghitung jumlah kata-kata yang dibagi antara dua *gloss* [9].

Dengan demikian hal ini banyak digunakan di dunia nyata seperti *machine translation* (MT), *semantic mapping* (SM), *semantic annotation* (SA), dan *ontology learning* (OL). Dan juga banyak digunakan untuk membantu kinerja aplikasi seperti *information retrieval* (IR), *information extraction* (IE), dan *speech recognition* (SR) [10].

2.4 Wordnet

WordNet adalah sebuah database network semantik untuk bahasa Inggris yang dikembangkan di Princeton University [11]. Perbedaan antara WordNet dengan kamus bahasa pada umumnya adalah kamus bahasa memfokuskan pada kata sedangkan WordNet memfokuskan diri kepada makna kata. Satu makna dalam WordNet dapat dinyatakan dengan *synset*(*synonym set*), yaitu kumpulan kata yang merepresentasikan suatu makna. Selain dari representasi makna, di dalam WordNet juga terdapat relasi/hubungan antar makna seperti hipernim, hiponim, holonim, meronim, dll [12].

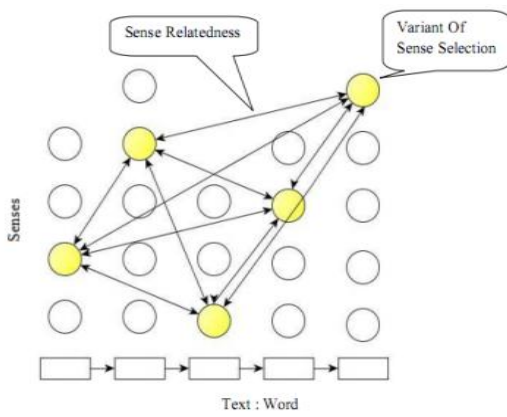
Hal ini tentu memudahkan untuk pengembangan lebih lanjut di bidang NLP. Wordnet menyimpan informasi tentang kata-kata yang berasal dari empat *part-of-speech*. Yaitu *nouns*, *verbs*, *adjective*, dan *adverbs*. Wordnet 1.7 mempunyai 107, 930 *nouns* di atur dalam 74, 448 *synsets*, 10, 680 *verbs* di dalam 12, 754 *synsets*, 12, 365 *adjectives* didalam 18, 523 *synsets*, dan 4, 583 *adverbs* di dalam 3, 612 *synsets* [9]. Dengan demikian hasilnya telah memunculkan aplikasi dalam berbagai *fields* penelitian seperti tool untuk

disambiguasi makna / WSD, *semantic tagging* dan *information interval* [13].

2.5 Algoritma Lesk (Original Lesk)

Algoritma Lesk diperkenalkan oleh Michael E. Lesk pada tahun 1986. Algoritma Lesk asli mendisambiguasi kata dalam frasa dekat. Dengan adanya kata disambiguasi, definisi kamus atau gloss dari masing-masing sense dibandingkan dengan beberapa gloss dari setiap kata lain dalam frasa. Sebuah kata diberikan sense yang berbagi dengan jumlah gloss terbesar dari kata-kata yang sama dengan beberapa gloss dari kata lain [9].

Gambar 2.1 Menunjukkan representasi grafis dari algoritma *lesk* [14].



Misalnya dalam mendisambiguasi frasa “*pine cone*” menurut kamus Oxford Advanced Learner’s

Terdapat dua *sense* untuk kata *pine*

Sense 1: kind of evergreen tree with needle-shaped leaves

Sense 2: waste away through sorrow or illness.

Terdapat tiga *sense* untuk *cone*

Sense 1: solid body which narrows to a point

Sense 2: something of this shape whether solid or hollow

Sense 3: fruit of certain evergreen tree

Masing-masing dari dua pengertian sense dari kata *pine* dibandingkan dengan masing-masing pengertian sense dari kata *Cone* dan ditemukan bahwa kata *evergreen tree* terdapat pada kedua sense tersebut, yaitu sense dari kata *pine* dan sense dari kato *cone*. Kedua sense kemudian dinyatakan sebagai sense yang paling tepat ketika kata-kata *pine* dan *cone* digunakan bersama-sama [9].

Menurut Padwardhan et al. ada dua hipotesis yang mendasari pendekatan ini. Yang pertama adalah tentang kedekatan kata-kata, yaitu, kata-kata yang muncul bersama-sama dalam sebuah kalimat

dapat tidak ambigu dengan menetapkan sense yang paling berkaitan erat dengan kata-kata di sekitarnya. Ide di balik hipotesis ini, kata-kata yang muncul bersama-sama dalam sebuah kalimat umumnya yang terkait dalam beberapa cara, karena untuk mengekspresikan ide beberapa manusia membutuhkan serangkaian kata-kata yang bekerja bersama-sama [14].

Hipotesis kedua adalah bahwa sense terkait dapat diidentifikasi dengan menemukan kata-kata yang tumpang tindih dalam definisinya. Ide di balik ini sama-sama masuk akal, karena kata-kata yang terkait akan sering dapat didefinisikan dengan menggunakan kata-kata yang sama, dan bahkan mungkin merujuk kepada satu sama lain dalam definisinya. Keterbatasan utama algoritma ini adalah bahwa Kamus glosses cukup singkat, dan mungkin tidak mencukupi Kosakata untuk mengidentifikasi sense terkait[14].

2.6 Algoritma Simplified Lesk

Algoritma *simplified lesk* diperkenalkan oleh Kilgarriff dan Rosenzweig pada tahun 2000 [15]. Algoritma ini mengukur overlap (tumpang tindih) antara definisi sense kata pada sebuah konteks kalimat. Untuk pseudocode dari algoritma ini dapat dilihat dari pseudocode berikut :

```
function SIMPLIFIED LESK(word, sentence)
returns best sense of word
best-sense := most frequent sense for word
max-overlap := 0
context := set of words in sentence
for each sense in senses of word do
signature := set of words in gloss and
examples of sense
overlap := COMPUTE_OVERLAP(signature,
context)
if overlap > max-overlap then
max-overlap := overlap
best-sense := sense
end
return(best-sense)
```

COMPUTE_OVERLAP : returns the number of words in common between two sets, ignoring function words or other words on a stop list [15].

3. METODOLOGI PENELITIAN

3.1 Langkah - Langkah Penelitian

1. Identifikasi masalah

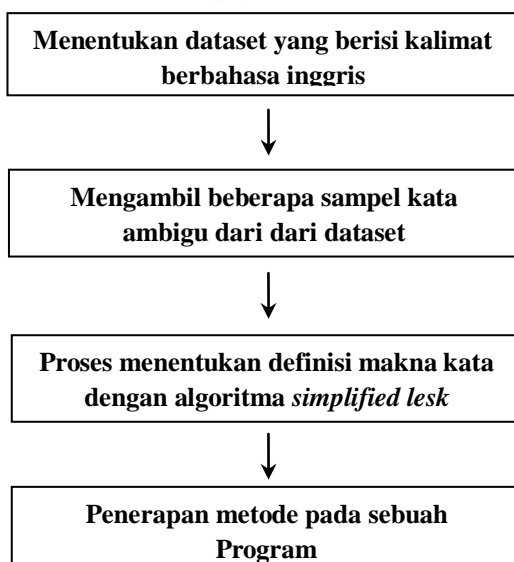
Merupakan tahapan bagaimana menerapkan metode penentuan makna ambigu pada

- kalimat berbahasa inggris dengan menggunakan algoritma *simplified lesk*.
2. Studi literatur
Pengumpulan data dengan cara mengumpulkan literatur, jurnal, paper dan bacaan-bacaan yang terkait dengan judul penelitian, untuk melengkapi pengetahuan dasar, memahami dan mempelajari teori tentang algoritma *simplified lesk*.
 3. Menentukan Dataset
Membuat dataset yang digunakan untuk menerapkan algoritma *simplified lesk* yang berisi kumpulan kalimat berbahasa inggris. Menerapkan algoritma *simplified lesk* berdasarkan dataset yang telah dibuat.
 4. Penerapan Kedalam Sebuah Sistem
Penerapan kedalam sistem disesuaikan dengan sampel kalimat yang ada dalam dataset untuk menentukan makna kata yang tepat pada kata ambigu yang terdapat pada sebuah kalimat berbahasa inggris.
 5. Pengujian Sistem
Merupakan tahap pengujian terhadap sistem yang telah dibuat.
 6. Hasil
Hasil yang diharapkan adalah sebuah sistem aplikasi yang dibuat untuk menentukan makna kata yang ambigu pada kalimat berbahasa inggris.

3.2 Blog Diagram

Sistem dirancang untuk dapat menentukan makna kata ambigu pada sebuah kalimat berbahasa inggris dengan algoritma *simplified lesk*. Dengan perencanaan yang akan digambarkan seperti berikut :

Gambar 3.2 Blog diagram implementasi algoritma *simplified Lesk*



3.3 Implementasi Algoritma

Implementasi algoritma bertujuan untuk mengetahui proses disambiguasi makna pada algoritma yang akan diterapkan.

3.3.1 Implementasi Algoritma Lesk (Original Lesk)

Untuk penerapan algoritma *lesk*(original lesk) dapat diambarkan pada contoh berikut : terdapat dua kata *pine* dan *cone* untuk menghitung jumlah kesamaan kata dari masing-masing definisi berikut contohnya :

Pine

Pine#n#1 kinds of *evergreen tree* with needle-shaped leaves

Pine#n#2 waste away through sorrow or illness

Cone

Cone#n#1 solid body which narrows to a point

Cone#n#2 something of this shape whether solid or hollow

Cone#n#3 fruit of certain *evergreen trees*

$$\text{Pine}\#n\#1 \cap \text{Cone}\#n\#1 = 0$$

$$\text{Pine}\#n\#1 \cap \text{Cone}\#n\#2 = 0$$

$$\text{Pine}\#n\#1 \cap \text{Cone}\#n\#3 = 2 \text{ (Evergreen, tree)}$$

$$\text{Pine}\#n\#2 \cap \text{Cone}\#n\#1 = 0$$

$$\text{Pine}\#n\#2 \cap \text{Cone}\#n\#2 = 0$$

$$\text{Pine}\#n\#2 \cap \text{Cone}\#n\#3 = 0$$

Dari hasil perhitungan diatas nilai kesamaan kata terbanyak terdapat pada *pine*#n#1 – *cone*#n#3 maka definisi pine menurut algoritma lesk adalah : *kinds of evergreen tree with needle-shaped leaves* sedangkan definisi cone menurut algoritma lesk adalah : *fruit of certain evergreen trees*

3.3.2 Implementasi Algoritma Lesk (Simplified Lesk)

Untuk penerapan algoritma *lesk* (*simplified lesk*) dapat dilihat dari penjelasan berikut :

Terdapat Sebuah Kalimat :

The bankcan guarantee *deposits* will eventually coverfuture tuition costs because it invests in adjustable-rate *mortgage* securities.

Kata ambigu : Bank.

Tabel 3.1 Pencarian makna dengan algoritma *simplified lesk*

Bank / 1	Deffinition (gloss) :
----------	-----------------------

	a financial institution that accepts <i>deposits</i> and channels themoney into lending activities Examples : “he cashed a check at the bank”, “that bank holds the <i>mortgage</i> on my home”
Bank / 2	Definition (gloss) : sloping land (especially the slope beside a body of water)Examples : “they pulled the canoe up on the bank”, “he sat on the bank of the river and watched the currents”

- Sense bank/1 memiliki dua kata yang sama / tumpang tindih (overlap) yaitu (*deposits, mortgage*).
- Sense bank/2 tidak memiliki kata yang overlap / kosong.

Dari hasil diatas sense bank/1 yang terpilih karena memiliki dua kata yang sama (overlap) [3].

4. Hasil Dan Pembahasan

Pada bab ini akan dilakukan pengujian menggunakan dataset yang sudah ada, guna untuk mengetahui precision, recall, beserta f measure dari algoritma *Simplified lesk*. Algoritma *Simplified lesk* dalam penelitian ini digunakan untuk proses disambiguasi makna kata dalam kalimat bahasa inggris.

4.1 Dataset

Sample data berupa 100 kalimat berbahasa inggris, kemudian dari 100 sample data tersebut masing-masing diambil satu kata untuk diketahui maknanya.

Contoh Sampel Data yang diambil dari dataset

No	Kata Ambigu	Kalimat
	Bank	The bank can guarantee deposits will eventually coverfuture tuition costs because it invests in adjustable-rate mortgage

		securities.
2	Bank	The river bank was full of dead fishes
3	Save	We save a lot of money by pooling our rent.

4.3 Skenario Pengujian 1

Pada skenario pengujian 1 dilakukan pengujian pada dataset yang sudah ada, dengan memanfaatkan wordnet untuk mengambil makna kata. Berikut hasil makna kata menurut bahasa alami manusia yang diambil dari wordnet.

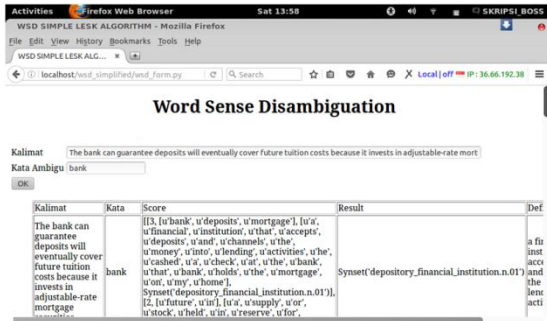
Tabel 4.2 Pengujian dengan bahasa alami manusia

Pengujian Dataset No 1 Dengan Definisi Diambil dari kamus Wordnet	
Kalimat	The bank can guarantee deposits will eventually coverfuture tuition costs because it invests in adjustable-rate mortgage securities.
Kata	Bank
Hasil	Synset('depository_financial_institution.n.01')
Definisi	a financial institution that accepts deposits and channels themoney into lending activities
Pengujian Dataset No 2 Dengan Definisi Diambil dari kamus Wordnet	
Kalimat	The river bank was full of dead fishes
Kata	Bank
Hasil	Synset('bank.n.01')
Definisi	sloping land (especially the slope beside a body of water)
Pengujian Dataset No 3 Dengan Definisi Diambil dari kamus Wordnet	
Kalimat	We save a lot of money by pooling our rent.
Kata	Save
Hasil	Synset('save.v.06')
Definisi	make unnecessary an expenditure or effort

4.4 Skenario Pengujian 2

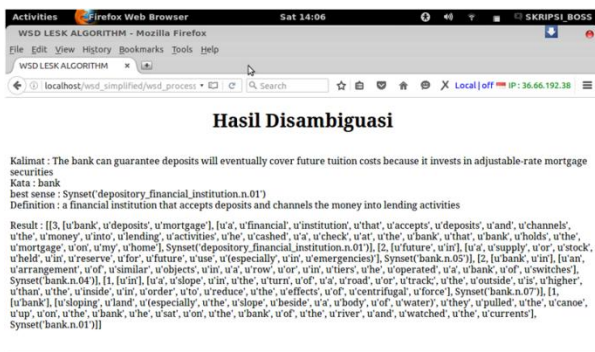
Pada skenario pengujian 2 dilakukan pengujian pada dataset yang sudah ada, dengan memanfaatkan wordnet untuk mengambil makna kata. Berikut hasil makna kata menurut algoritma *Simplified lesk* yang diambil dari wordnet.

Gambar 4.2 Pengujian dengan algoritma *simplified lesk* tampilan utama



Gambar 4.2 merupakan sebuah tampilan sistem yang dibuat dimana user dapat menginputkan isi teks kalimat bahasa Inggris beserta kata ambigu untuk kemudian diketahui hasil disambiguasi dari kata yang telah diinputkan.

Gambar 4.3 Hasil disambiguasi dengan algoritma *simplified lesk*



Gambar 4.3 merupakan hasil disambiguasi dari data yang telah diinputkan sebelumnya hasil tampilan berupa definisi makna kata yang telah diinputkan yang diambil dari definisi makna menurut wordnet dan hasil score overlap oleh algoritma *simplified lesk*.

Tabel 4.3 Hasil pengujian dengan algoritma *simplified lesk*

Pengujian Dataset No 1 Dengan Definisi Diambil dari kamus Wordnet	
Kalimat	The bank can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate

	mortgage securities.
Kata	Bank
Hasil	Synset('depository_financial_institution.n.01')
Definisi	a financial institution that accepts deposits and channels the money into lending activities
Keterangan	√
Pengujian Dataset No 2 Dengan Definisi Diambil dari kamus Wordnet	
Kalimat	The river bank was full of dead fishes
Kata	Bank
Hasil	Synset('bank.n.01')
Definisi	sloping land (especially the slope beside a body of water)
Keterangan	√
Pengujian Dataset No 3 Dengan Definisi Diambil dari kamus Wordnet	
Kalimat	We save a lot of money by pooling our rent.
Kata	Save
Hasil	Synset('save.n.01')
Definisi	(sports) the act of preventing the opposition from scoring
Keterangan	X

4.4 Menghitung Precision, Recall, dan F Measure

Dari 100 dataset yang telah disediakan Algoritma *Simplified lesk* dapat menjawab makna dengan benar sebanyak 48, sedangkan 54 lainnya kurang tepat. Dengan demikian dapat dihitung precision, recall, dan f measure sebagai berikut :

$Precision = \frac{\text{Jumlah makna yang dapat dijawab oleh sistem dengan benar}}{\text{Jumlah makna menurut bahasa alami manusia}} = \frac{48}{100}$

$$Precision = 48 / 100 = 0.48 (48\%)$$

$Recall = \frac{\text{Jumlah makna yang dapat dijawab oleh sistem dengan benar}}{\text{Jumlah makna sebenarnya}} = \frac{48}{100}$

$$Recall = 48 / 100 = 0.48 (48\%)$$

$$F\ measure = \frac{2 \times ((precision \times recall) / (precision + recall))}{1}$$

$$F\ measure = 2 \times ((48 \times 48) / (48 + 48))$$

$$F\ measure = 2 \times (2304 / 96)$$

$$F\ measure = 2 \times 24$$

$$F\ measure = 48$$

Dari hasil perhitungan di atas dapat diketahui tingkat precision dari algoritma *simplified lesk* adalah 48%, recall 48%, dan f measure sebesar 48%.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari hasil pengujian dan pembahasan tentang implementasi Algoritma *simplified lesk* pada bab sebelumnya, didapatkan beberapa kesimpulan tentang implementasi algoritma *simplified lesk* pada dataset berupa 100 sample kalimat bahasa inggris, yang masing-masing pada kalimat tersebut, diambil satu kata yang mempunyai makna ganda (ambigu).

1. *Word sense disambiguation* dengan algoritma *Simplified lesk* dapat menentukan makna kata ambigu pada kalimat bahasa inggris. Tingkat *precision*, *recall*, beserta *f measure* adalah sama sebesar 48% dari sample dataset berupa 100 kalimat berbahasa inggris. Dari penelitian tersebut algoritma *Simplified lesk* masih mengalami banyak kekurangan sekitar 58% untuk pencarian makna kata ambigu (word sense disambiguation). Untuk itu dibutuhkan penambahan metode lain guna untuk menambah tingkat *precision*, *recall* dan *f measure* dari *word sense disambiguation*.
2. Dari pengujian 100 sample data algoritma *Simplified lesk* dapat mengetahui makna dari beberapa kata saja yaitu 48 makna katadari 100, sedangkan 58 kata lainnya masih kurang tepat

5.2 Saran

Beberapa saran yang dapat dijadikan pertimbangan dari penelitian ini adalah :

1. Diharapkan penambahan beberapa metode untuk menambah tingkat *precision* dan *recall* untuk meningkatkan *f measure* dari algoritma *simplified lesk* untuk pencarian makna kata ambigu (word sense disambiguation).

Agar makna yang dipilih dapat lebih dekat dengan hasil makna bahasa alami manusia.

DAFTAR PUSTAKA

- [1] Astika Gusti, “*Globalisasi Bahasa Inggris: So What ?*”, *Lingua*, vol. 12, no. 1, pp. 90, Maret 2015.
- [2] Harmoejanto Jeany, “*Pembentukan Graf Untuk Pelabelan Makna Kata Synset, Relai Synset Dan Gloss*”, in Seminar Sistem Informasi Indonesia (SESINDO2010), ITS Surabaya, 2010, pp. 1-2.
- [3] Kelas Indonesia. 2015. *Pengertian dan Contoh*

Kalimat Ambigu Lengkap [Online], Available :

<http://www.kelasindonesia.com/2015/04/pengertian-dan-contoh-kalimat-ambigu-lengkap.html> diakses tanggal 17 Mei 2016

[4] Bahasa Indonesiaku. 2015. *Pengertian Kalimat Ambigu dan Contoh Kalimatnya Yang Terbaik* [Online], Available :

<http://www.bahasaindonesiaku.net/2015/09/pengertian-kalimat-ambigu-dan-contoh-kalimatnya-terbaik.html> diakses tanggal 20 Mei 2016

[5] M.K Entin. 2011. *Kecerdasan Buatan* [Online]. Available

FTP: <http://entin.lecturer.pens.ac.id/Kecerdasan%20Buatan/Buku/Bab%205%20Natural%20Language%20Processing.pdf> diakses tanggal 20 Mei 2016

[6] Fuadi Kholid, “*Pengenalan NLP (Natural Language Processing dengan Python)*”, Jogja, 2013, pp 2-4

[7] Barbagallo Donato et all, 2010, “*Exploiting Wordnet glosses to disambiguate nouns through verbs*”, Milano Italy : Department of Electronics and Information Politecnico di Milano

[8] R.P Alok and Saha Diganta, “*Word Sense Disambiguation: A Survey*”, *IJCTM*, vol. 5, no. 3, pp. 2, July 2015

[9] Banerjee, Satanjeev, 2002, “*Adapting the Lesk Algorithm for Word Sense Disambiguation to WordNet*”, Duluth U.S.A : Department of Computer Science University of Minnesota.

[10] Zhou Xiaohua, Han Hyoul, “*Survey of Word Sense Disambiguation Approaches*”, in The 18th *Flairs Conference*, Florida, 2005. pp. 1

[11] Princeton University. (2015). *Wordnet A Lexical Database For English* [Online]. Available: <http://wordnet.princeton.edu/> diakses tanggal 30Mei 2016

[12] Fakultas Ilmu Komputer UI. (2008). *Wordnet Bahasa Indonesia* [Online]. Available: bahasa.cs.ui.ac.id/iwn/ diakses tanggal 30Mei 2016

[13] Morato Jorge et all, 2004, “*Wordnet Aplications*”, Madrid Spain : Department Computer Science Universidad Carlos III

[14] S. Torres, and A. Gelbukh, “*Comparing similarity measures for original WSD lesk algorithm*”, *Research in Computing Science*, Citeseer, No. 43, pp. 155-166, 2009.

[15] Teufel Simone, “*Word Sense Disambiguation Algorithms*” in *Word Meaning and Discourse Understanding*, Computer Laboratory Natural Language and Information Processing (NLIP) University Of Cambridge England, 2011/2012