

## OPTIMASI ALGORITMA SIMPLIFIED LESK DENGAN SPACY UNTUK WORD SENSE DISAMBIGUATION PADA KALIMAT BAHASA INGGRIS

Nahel Al Fath Ichida<sup>[1]</sup>; Andi Sunyoto<sup>[2]</sup>; Eko Pramono<sup>[3]</sup>

Universitas Amikom Yogyakarta Program Studi PJJ Magister Teknik Informatika  
Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta  
alfathichida96@students.amikom.ac.id

| INFO ARTIKEL   | INTISARI  |
|--|---|
| <b>Diajukan :</b><br>12-02-2024                                | <i>Word Sense Disambiguation (WSD)</i> merupakan salah satu permasalahan utama dalam pengolahan bahasa alami yang bertujuan untuk menentukan arti yang tepat dari kata-kata yang memiliki makna ganda dalam konteks kalimat. Dalam penelitian ini, kami memperkenalkan sebuah pendekatan baru dalam peningkatan akurasi WSD dalam bahasa Inggris dengan mengoptimalkan algoritma <i>Lesk</i> menggunakan <i>library Spacy</i> . Algoritma <i>Simplified Lesk</i> telah diketahui memiliki batasan dalam menangani berbagai pemrosesan kata dalam kalimat. <i>Spacy</i> adalah <i>library natural language processing (NLP)</i> yang sangat kuat, khususnya untuk pemrosesan bahasa Inggris. Selain fungsi dasar seperti tokenizer, perpustakaan ini juga mendukung fungsi NLP yang bergantung pada solusi berbasis pembelajaran mesin seperti <i>part-of-speech (POS) tagging</i> , <i>Named Entity Recognition (NER)</i> , dan <i>parsing dependency</i> . Penelitian sebelumnya menunjukkan nilai <i>recall</i> , <i>presisi</i> , dan <i>f-measure</i> sebesar 40% masing-masing. Dengan memanfaatkan kemampuan <i>Spacy</i> , kami berhasil meningkatkan nilai <i>recall</i> , <i>presisi</i> , dan <i>f-measure</i> menjadi 45% masing-masing pada sejumlah dataset uji. |
| <b>Diterima :</b><br>28-02-2024                                |   |
| <b>Diterbitkan:</b><br>30-06-2024                              |   |
| <b>Kata Kunci :</b><br><i>WSD, NLP, Simplified Lesk, Spacy</i> |   |

### I. PENDAHULUAN

Pemrosesan Bahasa Alami (*Natural Language Processing* atau *NLP*) telah menjadi salah satu bidang yang paling berkembang dalam ilmu komputer dan linguistik komputasional [1]. Sebuah aspek penting dalam NLP adalah *Word Sense Disambiguation (WSD)*, yang bertujuan untuk mengidentifikasi makna yang sesuai dari kata-kata yang memiliki lebih dari satu makna dalam konteks kalimat [2]. Kemampuan untuk memahami dan menginterpretasikan makna kata dalam konteks adalah langkah penting dalam memahami teks secara lebih mendalam, yang memiliki dampak positif dalam berbagai aplikasi NLP, termasuk terjemahan mesin, analisis sentimen, dan pencarian informasi.

Pada dasarnya, masalah WSD menghadirkan tantangan utama karena banyak kata dalam bahasa memiliki beberapa makna yang mungkin bergantung pada konteks kalimatnya [2]. Oleh karena itu, pengembangan algoritma yang dapat mengidentifikasi makna kata secara akurat dalam konteks yang beragam telah menjadi fokus penelitian yang signifikan dalam bidang NLP. Salah satu algoritma yang telah digunakan secara luas dalam WSD adalah algoritma *Lesk*, yang diperkenalkan oleh Michael Lesk pada tahun 1986. Algoritma ini berdasarkan pada ide bahwa makna sebuah kata dapat ditentukan melalui tumpang

tindih kata dalam definisi kata tersebut dengan kata-kata dalam kalimat yang berisi kata tersebut.

Meskipun algoritma *Lesk* telah terbukti bermanfaat dalam WSD, implementasi tradisionalnya (*Simplified Lesk*) memiliki keterbatasan dalam menangani kalimat yang kompleks dan dalam menghadapi variasi kata yang kaya dalam bahasa Inggris. Dalam konteks penelitian ini, kami memperkenalkan sebuah pendekatan baru yang memanfaatkan kekuatan *library spacy*. *Spacy* adalah *library python* yang menyediakan kemampuan tokenisasi yang canggih, analisis tanda baca, serta vektor kata yang terlatih secara mendalam, untuk dapat mengoptimalkan algoritma *Simplified Lesk* dalam memecahkan masalah WSD [3].

Pada dasarnya, algoritma *Simplified Lesk* bekerja dengan cara membandingkan nilai pada dua input utama yaitu kata yang ingin dicari disambiguasi dan konteks di mana kata tersebut digunakan. Algoritma *Simplified Lesk* memerlukan akses ke kamus atau sumber data leksikal yang berisi definisi kata-kata. Kamus yang umum digunakan adalah WordNet, yang menyediakan definisi lengkap untuk banyak kata dalam bahasa Inggris [4]. Dalam WordNet, setiap kata memiliki beberapa makna atau *sense* yang terkait dengan konteks tertentu. Misalnya, kata "*bank*" bisa

memiliki makna sebagai lembaga keuangan atau sebagai bagian dari sungai.

Dalam penelitian ini, yang menjadi objek penelitian yaitu pada kalimat Bahasa Inggris. Bahasa Inggris adalah bahasa yang cukup kompleks karena memiliki banyak kata yang memiliki arti ambigu [5]. Kalimat Bahasa Inggris yang digunakan berasal dari dataset FEWS (*Few-shot Examples of Word Senses*). FEWS adalah kumpulan data WSD baru dengan jumlah contoh rendah yang diekstraksi secara otomatis dari kalimat contoh di Wiktionary. FEWS memiliki cakupan makna yang tinggi di berbagai domain NLP yang menyediakan dataset *training* besar yang mencakup jauh lebih banyak makna daripada kumpulan data sebelumnya dan dataset testing komprehensif yang berisi contoh-contoh sedikit dan *few-shot* dari berbagai makna [6].

Penelitian Kumar Manish (2020), mengusulkan adaptasi Algoritma *Lesk* untuk *Word Sense Disambiguation (WSD)* dengan menggabungkan informasi konteks. Algoritma *Lesk* secara tradisional bergantung pada tumpang tindih kata antara definisi kata target dan konteksnya untuk disambiguasi. Namun, adaptasi yang diusulkan meningkatkan proses ini dengan mempertimbangkan tidak hanya definisi kata target tetapi juga konteks di mana kata tersebut muncul. Adaptasi ini melibatkan pengambilan informasi konteks yang relevan dari korpus atau dataset dan mengintegrasikannya ke dalam proses disambiguasi. Ini memungkinkan algoritma untuk menangkap hubungan semantik antara kata target dan kata-kata di sekitarnya dengan lebih efektif, sehingga meningkatkan akurasi WSD menjadi 65% [7].

Penelitian oleh Basuki (2019), merancang sebuah sistem untuk mengatasi ambiguitas pada homograf dalam bahasa Indonesia. Penelitian ini menggunakan algoritma *Lesk* untuk mencari dan mendefinisikan makna dari kata-kata yang ambigu. Pengujian sistem dilakukan untuk mengevaluasi fungsionalitasnya dengan membandingkan hasil pengujian dengan data dari KBBI. Hasil penelitian menunjukkan tingkat akurasi sebesar 78,6% untuk satu kata ambigu dan 62,5% untuk dua kata ambigu dalam menentukan makna yang tepat [8].

Penelitian Najlah (2021) menggunakan kumpulan kalimat bahasa Inggris dari situs web FEWS. Pengujian dilakukan pada 500 kalimat bahasa Inggris yang mengandung kata-kata ambigu dan maknanya, yang dibagi menjadi kelompok 100, 200, 300, 400, dan 500 kalimat. Hasil pengujian dengan algoritma *Simplified Lesk* menunjukkan bahwa algoritma ini hanya berhasil menentukan makna kata dalam kurang dari 50% kalimat pada dataset. Akurasi tertinggi diperoleh pada kelompok 100 kalimat, yaitu sebesar 0,40, sedangkan akurasi

terendah ditemukan pada kelompok 500 dan 400 kalimat, yaitu sebesar 0,34. Rata-rata akurasi yang dihasilkan adalah 0,36 [9].

Dalam penelitian ini, kami akan mengoptimasi algoritma *simplified Lesk* yang telah digunakan pada beberapa penelitian lain. Kami menggunakan *library Spacy* untuk dapat memaksimalkan performa algoritma *Simplified Lesk* dalam menangani WSD. Detas yang digunakan yaitu adalah 100 kalimat dari situs web FEWS yang mengandung kalimat ambigu, data kata ambigu dan definisi kata pada kalimat ambigu. Detas melalui proses *preprocessing (Tokenizing, POS Tagging, Removal Stop Word)*. Selanjutnya dengan data latih dari Wordnet, sistem menentukan makna kata pada kalimat ambigu. Hasil prediksi makna kata pada sistem akan dibandingkan dengan makna kata pada dataset 100 kalimat dari situs web FEWS. Setelah hal ini dilakukan, kami menggunakan hasilnya untuk menentukan akurasi performa algoritma *Lesk* dengan *library Spacy*.

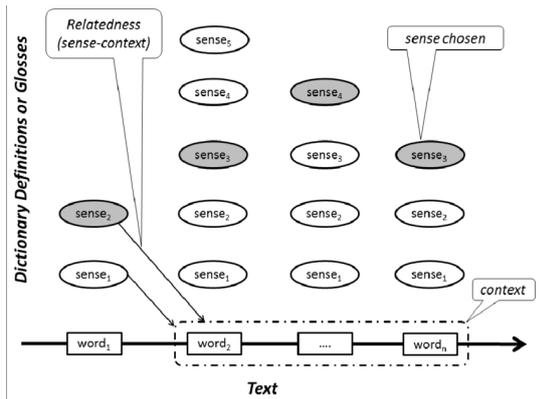
Penelitian ini bertujuan untuk meningkatkan performa algoritma *Lesk* dengan *library Spacy* pada permasalahan WSD pada kalimat bahasa Inggris. Selain itu, penelitian ini juga bertujuan untuk mengetahui berapa performa akurasi algoritma *Lesk* yang dioptimasi dengan *library Spacy*. Dengan tercapainya tujuan penelitian ini, diharapkan dapat memudahkan proses translasi bahasa Inggris agar dapat menentukan konteks pada kata ambigu pada sebuah kalimat.

## II. BAHAN DAN METODE

### 2.1 Rancangan Penelitian

Dalam penelitian ini, kami mengusulkan metode WSD dengan algoritma *Simplified Lesk* yang dioptimasi dengan *library Spacy* untuk dapat meningkatkan akurasi. Modifikasi yang kami lakukan meliputi:

- 1) Penggunaan definisi kamus yang lebih luas. Dalam penelitian ini, kami menggunakan definisi kamus yang lebih luas, yaitu definisi yang terdiri dari beberapa kalimat.
- 2) Penggunaan informasi konteks yang lebih luas. Algoritma *Lesk* yang asli hanya menggunakan konteks kalimat yang terdiri dari kata-kata yang berada di sekitar kata yang berambigu. Dalam penelitian ini, kami menggunakan konteks kalimat yang lebih luas, yaitu konteks yang terdiri dari kata-kata yang berada di seluruh kalimat.
- 3) Penggunaan *Library Spacy* untuk menangani dataset yang sangat besar dan memaksimalkan performa akurasi.
- 4) Algoritma *Simplified Lesk* diperkenalkan oleh Kilgarriff dan Rosenzweig pada tahun 2000. Berikut adalah gambar yang menunjukkan cara kerja Algoritma *Simplified Lesk*.



Sumber: Torres-Ramos, Román-Godínez, & Mendizabal-Ruiz (2017)

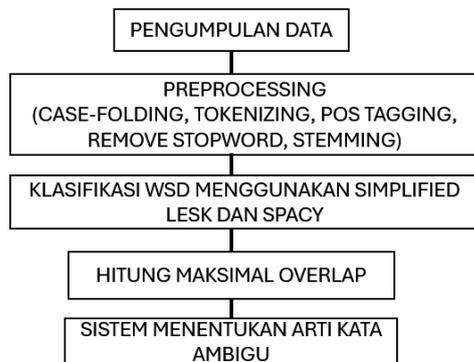
Gambar 1. Ilustrasi Algoritma *Simplified Lesk*

Fungsi pada algoritma *Simplified Lesk* akan membandingkan nilai pada dua kata yang dimasukkan dan mengembalikan nilai tertinggi. Gambar berikut menjelaskan *pseudocode* pada algoritma *Simplified Lesk* dengan *library Spacy*.

```
nlp = spacy.load("en_core_web_sm")
def simplified_lesk(word, context):
    best_sense = None
    max_overlap = 0
    # Tokenisasi dan proses konteks menggunakan spaCy
    context_tokens = nlp(context.lower())
    # Dapatkan definisi WordNet untuk kata tersebut
    senses = wordnet.synsets(word)
    for sense in senses:
        # Dapatkan definisi dan tokenisasikan menggunakan spaCy
        sense_tokens = nlp(sense.definition().lower())
        # Hitung Overlap antara definisi token dan konteks
        overlap = sum((sense_tokens.text.count(token.text) for token in context_tokens))
        # Perbarui definisi kata terbaik jika overlap lebih tinggi
        if overlap > max_overlap:
            max_overlap = overlap
            best_sense = sense
    return best_sense
```

Gambar 2. Pseudocode Algoritma *Simplified Lesk* dengan *Library Spacy*

Gambar 3 dibawah ini menjelaskan proses pada sistem yaitu pengumpulan data, proses *preprocessing*, klasifikasi menggunakan *Simplified Lesk* dan *Spacy*, mencari skor *overlap* tertinggi dan menentukan arti kata ambigu pada sistem.



Sumber: Dokumen Pribadi

Gambar 3. Diagram Alur Penelitian

## 2.2 Sumber Data

Sumber data dihimpun dengan cara mengunduh data dari website FEWS (<https://nlp.cs.washington.edu/fews/>). FEWS (*Few-shot Example of Word Sense Disambiguation*) bahasa Inggris yang dikumpulkan dari Wiktionary, kamus daring yang bersumber dari crowd-sourced dictionary. FEWS berisi lebih dari 121.000 contoh berlabel kata-kata ambigu, bersamaan dengan lebih dari 71.000 jenis makna.

## 2.3. Dataset

Data sampel terdiri dari 100 kalimat yang diambil dari sumber data, kemudian dari 100 data sampel tersebut, diambil satu kata untuk mengetahui maknanya. Berikut adalah contoh data sampel yang diambil dari dataset.

Tabel 1. Contoh sampel dataset

| No | Kata Ambigu                               | Kalimat  |
|----|---|--|
| 1  | demonstration<br>(demonstration .noun.02) | So he reads the demonstration of it, which directed him back to the proposition, and he read that proposition. |
| 2  | heavyweights<br>(heavyweight .noun.01)    | Ah, the two heavyweights finally meet. Sure you're tired of all the buzz.                                      |
| 3  | beguiled<br>(juggle.v.01)                 | I assure you, sir, I am no flatterer: the one who beguiled you, speaking plainly, was simply a knave.          |

Sumber: Dokumen Pribadi

Detas FEWS terdiri dari 3 jenis data, yaitu kata ambigu, *sense* pada ambigu dan contoh kalimat yang digunakan.

## 2.4 Pre-processing

*Pre-processing* adalah tahap yang diperlukan untuk mempersiapkan data masukan yang akan diproses pada tahap selanjutnya. Tujuan dari *pre-processing* adalah untuk mempermudah pemrosesan data dan mengubah data yang belum terstruktur menjadi bagian yang terstruktur sesuai dengan kebutuhan. *Pre-processing* memiliki beberapa tahap, yaitu :

- 1) *Case Folding*: menjadikan semua huruf dalam dokumen huruf kecil. Semua huruf dari "a" hingga "z" diterima. Karakter yang tidak termasuk huruf (?;!./) dihilangkan dan dianggap sebagai delimiter.
- 2) *Tokenizing*: proses memecah permintaan masukan menjadi kata-kata individual. Setiap kata dikenal sebagai token.

- 3) *POS Tagging*: proses memecah permintaan masukan menjadi kata-kata individual. Setiap kata dikenal sebagai token.

```

kalimat = "I loved the screen on this phone."
doc = nlp(kalimat)
for token in doc:
    print(token.text, token.pos_)

I PRON
loved VERB
the DET
screen NOUN
on ADP
this DET
phone NOUN
. PUNCT
    
```

Sumber: Dokumen Pribadi

Gambar 4. Sistem Setelah Melalui *Case Folding, Tokenizing, dan Pos-Tagging*

- 4) *Stop Word*: Tahap pemilihan kata-kata penting dari hasil *Pos Tagging* untuk menghilangkan kata yang tidak memiliki makna.
- 5) *Stemming*: proses mengurangi imbuhan (*affixes*) dari kata turunan sehingga menjadi kata dasar dengan menghapus awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*), dan *confixes* (gabungan awalan dan akhiran) dari kata turunan.

## 2.5 Implementasi WSD dengan Simplified Lesk dan Spacy

Implementasi algoritma *Simplified Lesk* dapat digambarkan pada penjelasan berikut:

Contoh kalimat : *will write letter*

Kata ambigu : *letter*

Tabel 2. *Sense* dari Kata *Letter*

| <i>Sense</i>                   | Makna  |
|--------------------------------|--|
| Makna 1 ( <i>letter.n.01</i> ) | A character that represents one or more sounds used in speech, specifically any symbol from an alphabet. |
| Makna 2 ( <i>letter.n.02</i> ) | A <b>written</b> , typed, or printed message sent in an envelope via postal service or messenger.        |
| Makna 3 ( <i>letter.n.03</i> ) | The exact terms of a statement or requirement; the strict verbal interpretation.                         |

Sumber: Dokumen Pribadi

Hasil perhitungan *max overlap Simplified Lesk* dapat dilihat pada tabel berikut.

Tabel 3. Hasil Perhitungan *Max Overlap*

| Makna 1<br>( <i>letter.n.01</i> ) | Makna 2<br>( <i>letter.n.02</i> ) | Makna 3<br>( <i>letter.n.03</i> ) |
|-----------------------------------|-----------------------------------|-----------------------------------|
| 0                                 | 1                                 | 0                                 |

Sumber: Dokumen Pribadi

Dalam kata tersebut, makna yang memiliki kesamaan maksimal dengan kata "*letter*" adalah makna 2, karena mencakup kata "*written*". Karena makna 2 memiliki kesamaan yang paling tinggi, sistem memilih makna tersebut sebagai definisi dari kata ambigu.

## III. HASIL DAN PEMBAHASAN

Pada fase ini, pengujian menggunakan dataset yang telah ada dilakukan untuk mengukur akurasi, ketepatan, recall, dan ukuran *f* dari algoritma *Simplified Lesk*. Dalam studi ini, algoritma *Simplified Lesk* dengan *Spacy* digunakan untuk menentukan makna kata dalam kalimat bahasa Inggris.

### 3.1 Skenario Pengujian 1

Pada skenario uji pertama, dataset yang ada digunakan untuk memeriksa makna kata. Hasilnya, yang berasal dari Wordnet, menunjukkan makna kata sesuai dengan bahasa alami manusia.

Tabel 4. Pengujian dengan Wordnet.

|              |   |
|--------------|---|
| Contoh 1     |   |
| Kalimat      | <i>So he reads the demonstration of it, which directed him back to the proposition, and he read that proposition.</i> |
| Kata Ambigu  | <i>demonstration</i>  |
| <i>Sense</i> | <i>demonstration.n.04</i>   |
| Definisi     | <i>Proofing through a sequence of arguments or propositions to prove an asserted conclusion.</i>                      |
| Contoh 2     |   |
| Kalimat      | <i>Ah, the two heavyweights finally meet. Sure you're tired of all the buzz.</i>                                      |
| Kata Ambigu  | <i>heavyweights</i>   |
| <i>Sense</i> | <i>heavyweight.n.03</i>   |
| Definisi     | <i>A professional boxer whose weight exceeds 190 pounds.</i>  |
| Contoh 3     |   |
| Kalimat      | <i>I assure you, sir, I am no flatterer: the one who beguiled you, speaking plainly, was simply a knave.</i>          |
| Kata Ambigu  | <i>beguiled</i>   |
| <i>Sense</i> | <i>juggle.v.01</i>  |
| Definisi     | <i>influence by slyness</i>   |

Sumber: Dokumen Pribadi

### 3.2 Skenario Pengujian 2

Pada skenario pengujian pertama, dataset yang ada digunakan untuk memeriksa makna kata. Hasilnya, yang berasal dari Wordnet, menunjukkan makna kata sesuai dengan bahasa alami manusia. Pada skenario pengujian kedua, pengujian dilakukan pada dataset yang telah ada menggunakan Wordnet untuk mengidentifikasi makna kata. Hasil penggunaan algoritma *Simplified Lesk* dari Wordnet ditampilkan dalam gambar di bawah ini.

```

sentence = "So he reads the demonstration of it, which referred him back to suc ↑ ↓
word_to_disambiguate = "demonstration"
sense = lesk(word_to_disambiguate, sentence)
if sense:
    print(f"Word: {word_to_disambiguate}")
    print(f"Disambiguated Sense: {sense.name()}")
    print(f"Sense Definition: {sense.definition()}")
else:
    print("Word not found in WordNet.")

Word: demonstration
Disambiguated Sense: demonstration.n.04
Sense Definition: proof by a process of argument or a series of proposition proving an asserted
    
```

Sumber: Dokumen Pribadi

Gambar 4. Pengujian dengan Algoritma *Simplified Lesk* dan *Spacy*.

Gambar 4 menunjukkan hasil disambiguasi dari data sebelumnya. Hasil *output* tampilan terdiri dari kata yang dicari makna ambiguannya, hasil *sense* dari sistem yang memiliki *overlap* terbaik dan definisi dari kata tersebut.

Tabel 5. Hasil Pengujian dengan *Simplified Lesk* dan *Spacy*.

| Hasil Evaluasi 1 |   |
|------------------|---|
| Kalimat          | <i>So he reads the demonstration of it, which directed him back to the proposition, and he read that proposition.</i> |
| Kata Ambigu      | <i>demonstration</i>  |
| Sense            | <i>demonstration.n.04</i>   |
| Definisi         | <i>Proofing through a sequence of arguments or propositions to prove an asserted conclusion.</i>                      |
| Keterangan       | <b>Sesuai</b>   |
| Hasil Evaluasi 2 |   |
| Kalimat          | <i>Ah, the two heavyweights finally meet. Sure you're tired of all the buzz.</i>                                      |
| Kata Ambigu      | <i>heavyweights</i>   |
| Sense            | <i>heavyweight.n.03</i>   |
| Definisi         | <i>A professional boxer whose weight exceeds 190 pounds.</i>  |
| Keterangan       | <b>Sesuai</b>   |
| Hasil Evaluasi 3 |   |
| Kalimat          | <i>I assure you, sir, I am no flatterer: the one who beguiled you, speaking plainly, was simply a knave.</i>          |

|             |                             |
|-------------|-----------------------------|
| Kata Ambigu | <i>beguiled</i>             |
| Sense       | <i>juggle.v.01</i>          |
| Definisi    | <i>influence by slyness</i> |
| Keterangan  | <b>Sesuai</b>               |

Sumber: Dokumen Pribadi

### 3.3 Menghitung *Precision*, *Recall* dan *F-Measure*

Dari 100 dataset yang dianalisis, sistem berhasil menjawab dengan benar pada 45 kasus, sementara 55 kasus lainnya kurang tepat. Oleh karena itu, *precision*, *recall*, dan *f-measure* dapat dihitung sebagai berikut:

$$Precision = \frac{\text{jumlah makna yang dijawab dengan benar oleh sistem (45)}}{\text{jumlah makna menurut bahasa manusia (100)}}$$

$$Precision = 45/100 = 0.45 = 45\%$$

$$Recall = \frac{\text{jumlah makna yang dijawab dengan benar oleh sistem (45)}}{\text{jumlah makna yang sebenarnya (100)}}$$

$$Recall = 45 / 100 = 0.45 = 45\%$$

$$F\text{-measure} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

$$F\text{-measure} = 2 \times \frac{(45 \times 45)}{(45 + 45)}$$

$$F\text{-measure} = 2 \times (2025 / 90)$$

$$F\text{-measure} = 2 \times 22.5$$

$$F\text{-measure} = 45$$

Hasil perhitungan menunjukkan bahwa algoritma *Simplified Lesk* dengan *Spacy* memiliki nilai *precision* sebesar 45%, *recall* sebesar 45%, dan nilai *F-measure* sebesar 45%..

### 3.4 Perbandingan dengan Penelitian Sebelumnya

Penelitian ini akan dibandingkan dengan penelitian sebelumnya untuk mengevaluasi perbedaan hasil, metodologi yang digunakan, dan potensi kontribusi baru terhadap pemahaman kita tentang topik tersebut.

Penelitian referensi ialah penelitian Najlah Afifah (2021) mengenai *Word Sense Disambiguation* dengan menggunakan algoritma *Simplified Lesk* untuk kalimat bahasa Inggris. Penelitiannya hanya menggunakan algoritma *Simplified Lesk*, sementara kami menggabungkan algoritma *Simplified Lesk* dengan *library Spacy*. Berikut adalah tabel perbandingan hasil *precision*, *recall* dan *F-Measure*.

Tabel 6. Tabel Perbandingan Penelitian

|                   | <i>Simplified Lesk</i> | <i>SL dengan Spacy</i> |
|-------------------|------------------------|------------------------|
| Jumlah Data       | 100                    | 100                    |
| Data Sesuai       | 40                     | 45                     |
| Data Tidak Sesuai | 60                     | 55                     |
| <i>Precision</i>  | 40%                    | 45%                    |
| <i>Recall</i>     | 40%                    | 45%                    |
| <i>F-Measure</i>  | 40%                    | 45%                    |

Sumber: Dokumen Pribadi

Dari tabel perbandingan di atas dapat disimpulkan bahwa penggunaan *library Spacy* dapat mengoptimalkan hasil *precision*, *recall* dan *f-measure* dari penelitian sebelumnya sebesar 5% (45%-40%).

#### IV. KESIMPULAN DAN SARAN

##### 4.1 Kesimpulan

Hasil pengujian menunjukkan bahwa algoritma *Simplified Lesk* dengan *Spacy* dapat digunakan untuk mengidentifikasi makna kata ambigu dalam kalimat bahasa Inggris. Pada dataset yang terdiri dari 100 kalimat, algoritma ini mencapai tingkat *precision*, *recall*, dan *f-measure* sebesar 45%. Meskipun ada peningkatan akurasi sebesar 5% dibandingkan penelitian sebelumnya, algoritma ini masih memiliki kekurangan karena 55% kata ambigu tidak dapat ditemukan maknanya.

##### 4.2 Saran

Beberapa saran untuk pengembangan penelitian ini ialah memperluas pustaka data Wordnet agar lebih maksimal dalam meninjau dataset FEWS dan untuk penelitian selanjutnya, diharapkan dapat menggunakan metode yang lebih andal untuk meningkatkan performa akurasi lebih dari 45%.

#### V. REFERENSI

- [1] Jurafsky, D., & Martin, J. H. (2019). *Speech and Language Processing (3rd ed.)*. Pearson.
- [2] Scarlini, B., Pasini, T., & Navigli, R. (2020). *Sense-Annotated Corpora for Word Sense Disambiguation in Multiple Languages and Domains*. Makalah disajikan dalam *International Conference on Language Resources and Evaluation*.
- [3] Honnibal, M., & Montani, I. (2017). *Spacy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing*. To Appear.
- [4] Feinerer, I., & Hornik, K. (2020). Wordnet: WordNet Interface The Comprehensive R Archive Network. Retrieved from

[https://cran.r-](https://cran.r-project.org/package=wordnet)

[project.org/package=wordnet](https://cran.r-project.org/package=wordnet)

- [5] Nuhui, M., & Ejupi, S., (2015). *Ambiguity in English and Albanian Sentences*. *Procedia - Social and Behavioral Sciences*, 192, 949-951. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1877042815039105?via%3Dihub>
- [6] Blevins, T., Joshi, M., & Zettlemoyer, L. (2021). *FEWS: Large-Scale, Low-Shot Word Sense Disambiguation with the Dictionary*. Makalah disajikan dalam *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*.
- [7] Kumar, M., Mukherjee, P., Hendre, M., Godse, M., & Chakraborty, B. (2020). *Adapted Lesk Algorithm based Word Sense Disambiguation using the Context Information*. *International Journal of Advanced Computer Science and Applications*, 11 (3), 259. Retrieved from [https://thesai.org/Downloads/Volume11N03/Paper\\_30-Adapted\\_Lesk\\_Algorithm.pdf](https://thesai.org/Downloads/Volume11N03/Paper_30-Adapted_Lesk_Algorithm.pdf)
- [8] Basuki, S., Kholimi, A.S., Minarno, A.E., Sumadi, F.D., & Effendy, M.R. (2019). *Word Sense Disambiguation (WSD) for Indonesian Homograph Word Meaning Determination by LESK Algorithm Application*. Makalah disajikan dalam *12th International Conference on Information & Communication Technology and System (ICTS)*, 8-15.
- [9] Afifah, N., Yusliani, N., & Osvari, A., (2021). *Word Sense Disambiguation Menggunakan Algoritma Simplified Lesk pada Kalimat Bahasa Inggris*. Retrieved from [https://repository.unsri.ac.id/54266/2/RAM\\_A\\_55201\\_09021181621035\\_0008118205\\_0028068806\\_01\\_front\\_ref.pdf](https://repository.unsri.ac.id/54266/2/RAM_A_55201_09021181621035_0008118205_0028068806_01_front_ref.pdf)
- [10] Torres-Ramos, S., Román-Godínez, I., & Mendizabal-Ruiz, E. G., (2017). *An Associative Method for Lesk-Based Word sense disambiguation*. *Revista Signos*, 50, 287-308. Retrieved from <https://www.scielo.cl/pdf/signos/v50n94/0718-0934-signos-50-94-00287.pdf>