Implementasi Algoritma Naïve Bayes untuk *Word Sense Disambiguation* dalam Bahasa Indonesia

Rifki Mifathur Sutomo[1], Arie Ardianti Suryani[2]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]rifkimis@students.telkomuniversity.ac.id, [2]ardiyanti@telkomuniversity.ac.id

Abstrak

Kata atau kalimat ambigu merupakan suatu permasalahan yang ditemukan dalam penggunaan bahasa baik dalam kontek verbal atau tekstual. Suatu kata dikatakan ambigu jika memiliki lebih dari satu makna. Dalam kalimat, makna kata dapat ditentukan oleh struktur dari kalimatnya. Dalam *Natural Language Processing* terdapat metode yang dapat mencari suatu makna kata dengan proses perhitungan yang tepat yaitu dinamakan *word sense disambiguation*. Penelitian ini berfokus dalam penerepan algoritma naïve bayes dalam *word sense disambiguation* bahasa Indonesia. Pemilihan algoritma *naïve bayes* dikarenakan naïve bayes dapat menghasilkan akurasi yang bagus sebagai salah satu bagian dari *supervised method*. Penelitian ini mengumpulkan data dari korpus bahasa Indonesia. Korpus diambil dari jejaring sosial, surat kabar *online* dan membuat kalimat ambigu sendiri. Dalam mempermudah klasifikasi penelitian ini menggunakan bahasa pemrograman *python*. Dalam *python* terdapat *module* yang digunakan untuk menemukan pola *string* tertentu yang disebut *regex*. Tujuan penelitian ini mengklasifikasi, serta menganalisis penerapan algoritma *naïve bayes* kedalam *word sense disambiguation* bahasa Indonesia.

Kata kunci : kata ambigu, Bahasa Indonesia, *natural language processing*, *word sense disambiguation*, *supervised*, klasifikasi naïve bayes

Abstract

Ambiguous words or sentences are a problem found in the use of language both in verbal and textual contexts. A word is ambiguous if it has more than one meaning. In a sentences, the meaning of words can be determined by the structure of the sentence. In Natural Language Processing there is a method that can search for a word's meaning with an exact calculation process that is called word sense disambiguation. This research focuses on the advance of naïve Bayes algorithm in Indonesian sense of word disambiguation. The choices of naïve bayes algorithm is because naïve bayes can produce good accuracy as part of the supervised method. This research collected data from the Indonesian corpus. The corpus was taken from social networks, online newspapers and made own ambiguous sentences. In simplifying the classification of this research using the python programming language. In python there is a module that is used to find certain string patterns called regex. The purpose of this study is to classify, also to analyze the application of the Naïve Bayes algorithm into word sense disambiguation in Indonesian language.

Keywords: ambigious words, Indonesia language, natural language processing, word sense disambiguation, supervised, naïve bayes classification

1. Pendahuluan

1.1 Latar Belakang

Bahasa Indonesia memiliki jenis kata - kata yang mempunyai makna lebih dari satu. Namun penggunaannya tergantung dari konteks kalimat yang menyertainya. Penggunaan kata yang memiliki makna lebih dari satu ini dapat meyebabkan ambiguitas kalimat. Ambiguitas kalimat ini dapat menyebabkan keraguan dalam penafsiran kalimat itu sendiri atau disebut dengan redundansi kalimat. Untuk mengatasi masalah ambiguitas tersebut diperlukan cara untuk memilih kata yang tepat sesuai dengan konteks kalimat. Cara atau teknik untuk menyelesaikan ambiguitas kata dalam kaliat dikenal sebagai *Word Sense Disambiguation*.

Word Sense Disambiguation (WSD) adalah proses untuk mengindentifikasi makna kata yang digunakan dalam kalimat tertentu ketika kata memiliki sejumlah makna yang berbeda [1]. Dalam *natural language processing*, penggunaan WSD ini bermanfaat untuk perbaikan kata dalam mesin penerjemah. Perbaikan kata ini dapat dilihat dalam *google translate*, dimana pengguna dapat membantu memperbaiki konteks kalimat yang telah diubah ke bahasa target dengan memilih kata yang dianggap ambigu. Selain itu dapat digunakan pula untuk *proof read*, dimana *proof read* adalah metode membaca ulang untuk memeriksa sebuah penulisan serta memastikan tidak ada penulisan yang salah, tidak konsisten dan tidak mengandung makna yang ambigu.

Saat ini, masih sedikit publikasi penelitian tentang *word sense disambugiuation* untuk bahasa Indonesia. Salah satu penelitian *word sense disambiguation* bahasa Indonesia yang telah dipublikasikan oleh Edi Faisal, Farza Nurifan, dan Riyanarto Sarno dengan menggunakan algoritma *Support Vector Machine* (SVM) sebagai penyelesaianya [9]. Dimana dalam penelitian ini menggunakan teks bahasa Indonesia dalam situs Wikipedia, dengan nilai akurasi yang didapat sebesar 87%. Akurasi ini dianggap bagus mengingat sedikitnya penelitian – penelitian yang membahas *word sense disambiguation* Bahasa Indonesia.

Dalam penyelesaian *word sense disambiguatio*n, secara umum penyelesaiannya dibagi menjadi tiga pendekatan yaitu pendekatan *supervised*, pendekatan *unsupervised* dan pendekatan *knowledge-based*. Dalam penelitian ini penulis menggunakan pendekatan *supervised* dengan menggunakan algortima naïve bayes yang diterapkan kedalam teks Bahasa Indonesia.

1.2 Perumusan dan Batasan Masalah

Dalam permasalahan *word sense disambiguation* Bahasa Indonesia, penelitianya masih sedikit dilakukan. Karena itu penulis akan membahas *word sense disambiguation* dalam Bahasa Indonesia dengan menggunakan algoritma *naïve bayes*. Pemilihan algoritma *naïve bayes* didasarkan penelitian oleh Thwet Aung, N. T., Soe, N. K., & Thein, N. L dengan menggunakan Bahasa Myanmar [12]. Penelitian tersebut menghasilkan akurasi sebesar 89%. Oleh karena itu, penulis akan menerapkan algortima *naïve bayes* ke dalam *word sense disambiguation* Bahasa Indonesia.

Adapun rumusan masalah yang disusun:

1. Bagaimana mencari kata yang memiliki makna ambigu dan mengetahui makna yang tepat?
2. Bagaimana nilai akurasi yang didapat menggunakan algoritma *Naïve Bayes* bila diterapkan dalam *word sense disambiguation* Bahasa Indonesia?

Serta batasan masalah dalam penelitian ini:

1. Data yang digunakan dalam penelitian ini menggunakan teks Bahasa Indonesia menggunakan bahasa formal.
2. Menggunakan bahasa pemrograman *python.*

1.3 Tujuan

Tujuan penelitian ini adalah:

1. Menambah penelitian tentang *word sense disambiguation* dalam Bahasa Indonesia dengan menggunakan algoritma *naïve bayes*.
2. Menentukan *sense* dari suatu kata dengan menggunakan algoritma *naïve bayes* yang diterapkan dalam *word sense disambiguation.*
3. Menemukan nilai akurasi dari algoritma *naïve bayes* yang diterapakan pada *word sense disambiguation* dalam Bahasa Indonesia.

1.4 Organisasi Tulisan

Bagian selanjutnya, pada bab kedua akan menjabarkan tentang studi – studi terkait tentang penelitian *word sense disambiguation* Bahasa Indonesia dengan menggunakan algoritma *naïve bayes.* Setelah menjabarkan studi terkait, pada bab ketiga dijelaskan tentang sistem yang dirancang dari penelitian ini. Selanjutnya pada bab keempat dijabarkan tentang hasil penelitian serta analisis dari penelitian *word sense disambiguation* Bahasa Indonesia dengan menggunakan algoritma *naïve bayes* ini. Bab kelima menampilkan kesimpulan dari penelitian ini. Setelah kesimpulan ditampilkan daftar Pustaka yang digunakan dalam penelitian.

2. Studi Terkait

2.1 Kata Ambigu

Dalam Kamus Besar Bahasa Indonesia (KBBI) kata ambigu merupakan kata yang memiliki makna lebih dari satu. Kata yang ambigu ini dapat menyebabkan kerancuan atau ketidakjelasan dalam satuan kalimat. Keambiguan ini dapat terjadi dalam tingkat kata, frasa atau kalimat. Untuk menghindari adanya kata atau kalimat yang ambigu maka diperlukan usaha untuk memilih kata dengan semestinya agar sesuai dengan konteks kalimat.

Berdasarkan KBBI kalimat ambigu terbagi menjadi tiga jenis yaitu ambiguitas fonetik, ambiguitas gramatikal, dan ambiguitas leksikal. Dimana ambiguitas fonetik merupakan jenis kalimat ambigu yang terjadi karena adanya persamaan pengucapan suatu kata sehingga dapat menyebabkan beragam makna dalam kalimat. Contoh "memberi tahu", dari contoh tersebut dapat ditemui dua makna yaitu memberikan sesuatu berupa benda, dan menginformasikan sesuatu. Terdapat pula ambiuitas gramatikal, yaitu jenis kalimat ambigu yang menyangkut hubungan intrabahsa atau makna yang muncul sebagai akibat berfungsinya sebuah kata dalam kalimat[15]. Contoh "Ditunggangi", makna kata yang diberi awalan "Di" ini dapat mempunyai dua makna seperti "Kuda itu ditunggangi oleh pamanku" yang berarti paman menaiki seekor kuda dan "Acara itu ditunggangi oleh kepentingan suatu partai" yang berarti sebuah acara yang dicampuri oleh suatu organisasi. Terakhir terdapat ambiguitas leksikal yaitu kalimat ambigu dimana terdapat kata dasar yang mempunyai makna lebih dari satu tergantung penempatanya dalam kalimat. Contoh "Bulan", dimana kata bulan dapat mempunyai makna tempat, seperti "Dia pernah mendarat di Bulan". Dan bulan dapat mempunyai makna waktu, seperti "Kita bertemu pada bulan Mei lalu". Khusus untuk ambiguitas leksikal dapat digunakan untuk *word sense disambiguation* karena itu perlu dapat menentukan *sense* yang tepat dalam mengolahnya.

2.2 Word Sense Disambiguation

Word sense disambiguation (WSD) adalah kemampuan mengidentifikasi arti kata dalam kalimat tertentu [11]. Dalam *natural language processing* penelitian tentang WSD dimulai sejak tahun 1940, pada tahun 1949 Zipf membuat sebuah teori yang diberi nama *"Law Of Meaning"*. Teori tersebut berisi tentang hubungan diantara kata-kata yang sering digunakan dengan yang tidak sering digunakan. Dimana untuk kata-kata yang sering digunakan memiliki *sense* yang sering digunakan dibandingkan dengan yang tidak sering digunakan [11]. Pada tahun 1957 Masterman memberikan teori tentang menemukan arti atau makna sebenarnya sebuah kata yang dipresentasikan dalam *Roget's International Thesaurus*. Pada tahun 1990 para peneliti banyak menggunakan prosedur ekstraksi pengetahuan otomatis serta menerapkan algoritma lesk untuk mendisambiguasi kata dalam konteks kalimat.

Word Senses Disambiguation banyak berperan dalam berbagai peneliatian dibidang *machine translation* (MT), *sematic mapping* (SM), *sematic annotation* (SA), dan *ontology learning* (OL) [11]. Dalam beberapa penelitian terkait *word sense disambiguation* terdapat tiga pendekatan yang banyal ditemukan untuk mempelajari *word sense disambiguation* yaitu *knowledge-based*, *unsupervised*, dan *supervised*. Dari ketiga pendekatan tersebut, terdapat peran pembelajaran mesin yang digunakan untuk menerapkan pendekatan-pendekatan itu kedalam WSD. Dimana metode-metode dalam pembelajaran mesin sering digunakan dalam WSD, diantaranya seperti Naïve Bayes, *K-Nearest Neighbor* (KNN), *Linear multilayer perceptron*, dan SVM.

Dalam penelitian WSD Bahasa Indonesia sendiri sudah ada yang menerapkan dalam berbagai metode. Khusus untuk penelitian WSD Bahasa Indonesia dengan menerapkan *Naïve bayes* sudah dilakukan oleh Uliniansyah, Mohammad & Ishizaki, Shun dengan judul penelitian *"A Word Sense Disambiguation System Using Modified Naive Bayesian Algorithms for Indonesian Language"*. Penelitian oleh Uliniansyah dan

Ishizaki ini membahas tentang penerapan algoritma *naïve bayes* terhadap *word senses disambiguation* dalam Bahasa Indonesia. Dalam menerapkan algoritma *naïve bayes*, data *training* diklasifikasi kedalam sepuluh kategori makna atau sense seusai dengan konteks kalimat. Sedangkan untuk kata yang ambigu sebanyak lima kata polisemi yaitu bintang, bunga, garis, kursi, lapangan. Kata polisemi merupakan kata – kata yang memiliki makna lebih dari satu. Masing – masing kata ambigu memiliki makna atau sense dengan lebih dari dua sense. Contoh bintang dapat berarti bentuk, nama, level, benda langit dan medali. Penelitian ini menganalisa kalimat ambigu dalam berbagai konteks. Memodifikasi algoritma *naïve bayes* ke dalam beberapa versi. Tujuan untuk dapat diterapkan dalam *word sense disambigua*tion untuk beberapa kasus. Sebagai contoh kata bunga memiliki idiom atau gabungan kata yang mempunyai arti sendiri yaitu suku bunga. Salah satu fokus penelitian ini yaitu dapat mengklasifikasi kata idiom tersebut. Dalam proses klasifkasi menggunakan algoritma *naïve bayes* yang sudah dimodifikasi. Kata – kata polisemi yang bersatu sebagai sebuah idiom akan dihitung probabilitasnya dari setiap kata polisemi. Hasil kedua probabilitas tersebut akan dihitung sebagai satu nilai probabilitas dari kata idiom. Nilai probabilitas tersebut akan dipilih yang terbaik seperti pada klasifikasi *naïve bayes* umum. Setelah terpilih nilai probabilitas terbaik maka program dapat menentukan sense yang tepat dari suatu kata idiom. Sebagai evaluasi penelitian hasil akurasi yang didapat adalah 73%-99%. Hasil tersebut didapat dari berbagai metode pengujian. Mulai dari mencari sense dari suatu kata polisemi hingga mencari *sense* kata idiom. Hasil penelitian ini dapat digunakan untuk perkembangan penelitian *natural language processing* dalam Bahasa Indonesia.

2.3 Text Processing

Dalam studi penambangan teks atau *Natural Languange Processing* (NLP) diperlukan cara atau tahapan awal yang diperlukan agar teks dapat diubah menjadi lebih terstruktur. Salah satu penerapan dari penambangan teks disebut *text processing*. Tahapan ini akan menyeleksi data yang diproses pada setiap dokumen. Proses dalam tahapan ini meliputi *tokenizing* atau tokenisasi, *lowercase*, *filtering* atau *remove stopwords*, dan *stemming*. Tokenisasi merupakan tahapan pemecahan sebuah dokumen teks yang terdiri dari sekumpulan kalimat menjadi bagian-bagian kata yang disebut dengan token. Selanjutnya *lowercase* merupakan tahapan dimana mengubah semua huruf kapital pada sebuah dokumen teks menjadi huruf kecil. Tahap selanjutnya adalah *filtering*(*remove stopwords)*.Tahapan ini merupakan tahap mengambil serta menyimpan seluruh kata penting dari tahap tokenisasi dengan menggunakan algortima *stoplist* (membuang kata yang tidak terpakai atau *stopwords*). *Stopwords* sendiri merupakan kata-kata umum yang muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh *stopword* dalam Bahasa Indonesia adalah "yang", "dan", "di", "dari", dll. Tahap terakhir adalah *stemming*. *Stemming* merupakan proses untuk mengembalikan kata yang mendapat imbuhan ke bentuk dasar kata tersebut. Contoh "berkebun" menjadi kata "kebun". Dengan menghilangkan imbuhan "ber-" diharapkan dapat mencari ketepatan semua variasi kata untuk menghasilkan dokumen yang paling relevan. Tahap text processing ini ditujukan untuk menghasilkan kalimat yang dapat diolah dalam proses klasifikasi. Sehingga akan menghasilkan nilai akurasi yang tinggi.

2.4 Algoritma Naïve Bayes

Naïve Bayes merupakan salah satu metode yang ada dalam *machine learning* dan masuk kedalam pendekatan *supervised*. Dimana teori atau metode ini ditemukan oleh ilmuwan Inggris bernama Thomas Bayes. Dasar dari metode ini adalah ilmu probabilitas dan statiska. Algoritma naïve bayes sendiri digunakan untuk memprediksi peluang di masa depan dengan mengandalkan pengalaman pada masa lalu. Ciri utama dari algoritma naïve bayes adalah adanya independensi yang kuat dari masing-masing kondisi [5]. Penggunaan algoritma ini tidak memerlukan jumlah data latih yang banyak untuk menentukan estimasi parameter dalam proses pengklasifikasian. Karena yang dibutuhkan hanya varian dari suatu variabel dalam kelas. Tidak menggunakan keseluruhan dari matriks kovarian [5]. Adapun tahapan dari proses algoritma naïve bayes menghitung jumlah kelas atau label, menghitung jumlah kasus, menghitung semua variabel dan terakhir membandingkan hasil. Secara umum hasil dari algoritma ini mudah untuk diterapkan serta mempunyai nilai akurasi yang tinggi. Namun dikarenakan adanya keterikatan antar atribut dalam kelas dapat membuat nilai akurasinya berkurang. Secara umum rumus perhitungan naïve bayes ditulis

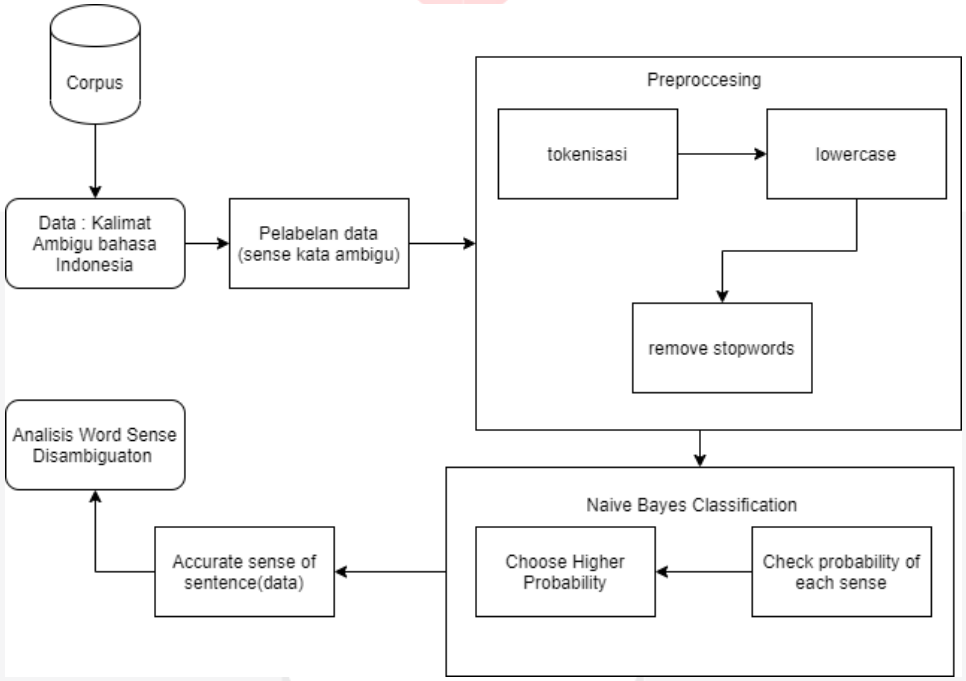
$$P(H|X) = \frac{P(H)P(H)}{P(X)})$$

Keterangan:

X = Data sample dengan kelas (label) yang tidak diketahui
H = Hipotesa bahwa X adalah data dengan klas (label) C
P(H|X) = Peluang bahwa hipotesa benar (valid) untuk data sampel X yang diamati
P(X|H) = Peluang data sample X, bila diasumsikan bahwa hipotesa benar (valid).
P(H) = Peluang dari hipotesa H
P(X) = Peluang data sample yang diamati

3. Sistem yang Dibangun

Pada bab ini menjelaskan tentang skema atau alur dari rancangan program dalam penelitian. Dimulai dengan tahapan awal penelitian hingga menganalisis kinerja program. Tahap awal yaitu mengumpulkan data yang akan digunakan dari korpus serta membuat label terhadap setiap data *training*. Dilanjutkan dengan *preprocessing* yaitu untuk mengolah kalimat Bahasa Indonesia menjadi data yang dapat diolah dengan algoritma. Selanjutanya adalah tahap mengklasifikasi data dengan algoritma *naïve bayes*. Setelah diklasifikasi akan mendapatkan *sense* terbaik dari kalimat tes atau data *test*. Tahap selanjutnya adalah menghitung akurasi dari algoritma *naïve bayes*. Terakhir adalah mengalisis kinerja dari algoritma *naïve bayes* terhadap *word sense disambiguation* Bahasa Indonesia.



Gambar 1 : Rancangan Sistem

Berdasarkan gambar 1 diagram sistem yang dibangun pada penelitian ini bertujuan untuk dapat menerapkan penggunaan algoritma *naïve bayes* terhadap *word sense disambiguation* dalam Bahasa Indonesia. Dalam penelitian ini data yang digunakan mengambil dari korpus Bahasa Indonesia baik yang ada dalam artikel – artikel *online*, media social dan membuat kalimat sendiri. Data yang berupa kumpulan kalimat Bahasa Indonesia yang memiliki kata atau makna ambigu akan diberi label manual sesuai dengan makna yang tepat. Data yang diberi label akan diolah karena masih dalam bentuk data *raw* atau kotor melalui tahapan *preprocessing*. Selanjutnya dalam data akan diproses dengan klasifikasi *naïve bayes.* Setelah melalui klasifikasi *naïve bayes* penelitian ini menganalisis penerapan *naïve bayes* terhadap *word sense disambiguation* dalam Bahasa Indonesia.

3.1 Dataset

Tahap awal dimulai dengan mengumpulkan data yang berasal dari kumpulan kalimat Bahasa Indonesia yang teridikasi ambigu. Data atau kalimat didapatkan dari artikel – artikel yang ada dalam media *online*, media sosial twitter, dan membuat kalimat sendiri. Setelah itu data akan diberi id serta mendapatkan label untuk sense dalam kalimat ambigu. Berikut contoh dataset seperti pada Tabel dibawah:

Tabel 1 : Proses Pelabelan Data

ID	Sense	Kalimat
001	Bulan: Tempat	Neil Amstrong adalah orang pertama yang mendaratan kaki di Bulan
002	Bulan: Waktu	Aku bertemu dia pada bulan Januari
003	Bulan: Waktu	Musim hujan terjadi pada bulan Oktober hingga April

3.2 Prepocessing

Tahap ini data berupa kalimat yang masih mentah akan diolah serta dilakukan pembersihan dalam kalimat tersebut agar terbentuk kata-kata yang diperlukan untuk proses klasifikasi dengan algoritma *naïve bayes*. Langkah *preprocessing* ini berfungsi meningkatkan keefektifan algortima *naïve bayes*. Selain itu juga tahap ini digunakan untuk menjadi parameter yang mengukur tingkat ketepatan akurasi klasifikasi algoritma. Pada tahap *preprocessing* ini meliputi *Tokenizing*, *lowercase*, dan *Remove stop words*.

Tokenizing adalah tahap dimana kalimat yang telah diiput untuk diolah akan dipisahkan setiap karakter penyusun dari kalimat tersebut. Dimana *tokenizing* memang bertujuan untuk membuat kata-kata penyusun kalimat terpisah agar dapat diolah menggunakan algoritma *naïve bayes*. *Lowercase* merupakan tahap dimana merubah semua alphabet besar menjadi alphabet kecil. *Remove stop words* merupakan tahap menghilangkan kata-kata yang tidak dipakai. Perlunya langkah ini agar dapat menentukan kata-kata yang mengandung makna ambigu serta mempercepat proses perhitungan nanti. Lebih jelas akan ditunjukkan melalui tabel dibawah :

Tabel 2 : Tahap Preprocessing

Tahapan	Input	Output	Penjelasan
Tokenizing	Neil Amstrong adalah orang pertama yang mendaratkan kaki di Bulan	“Neil”, “Amstrong”, “adalah”, “orang”, “pertama”, “yang”, “mendaratkan”, “kaki”, “di”, “Bulan”	Memisahkan kata – kata sebagai struktur kalimat.
Lowercase	Neil Amstrong adalah orang pertama yang mendaratkan kaki di Bulan	neil amstrong adalah orang pertama yang mendaratkan kaki di bulan	Mengubah semua huruf besar ke dalam huruf kecil.
Remove Stopwords	neil amstrong adalah orang pertama yang mendaratkan kaki di bulan	[neil amstrong orang pertama mendaratkan kaki bulan]	Menghilangkan kata (adalah, yang, di).

3.3 Naïve Bayes Classifier

Setelah melewati tahap preprocessing maka data yang akan digunakan telah menjadi data yang bersih sehingga dapat diuji dengan klasifikasi naïve bayes. Tujuan pengujian ini adalah untuk dapat mengklasifikasikan suatu data (kalimat – kalimat yang terdapat kata atau makna ambigu) kedalam makna sesungguhnya. Dalam metode *Naive Bayes Classifier* dilakukan proses pengklasifikasian teks berdasarkan data latih yang sebelumnya sudah disimpan [14]. Pada implementasinya terdapat tiga tahap yaitu membuat daftar vocab data, melatih program dengan beragam varian kalimat dalam data *train* dan membuat klasifikasi.

Sebelum program dapat melakukan atau menemukan sense yang tepat pada suatu kalimat. Perlu disiapkan langkah pengolahan data untuk memermudah program dalam mengklasifikasi. Tahap pertama dalam implementasi sistem yang dibuat adalah membuat vocab data. Vocab data merupakan suatu kumpulan kata yang dibuat untuk memudahkan program dalam mengklasifikasi. Kumpulan kata tersebut

disimpan dalam vocab.txt Vocab data memanfaatkan modul *python* yaitu *regex*. *Regex* merupakan modul dalam python yang berupa deretan karakter yang digunakan untuk pencarian string atau teks dengan menggunakan pola. Tahap awal penggunaan *regex* adalah membuat pola terlebih dahulu. Kemudian dicocokkan dengan teks atau tulisan yang tersedia. Jika dijumpai *string* yang cocok dengan pola, maka string tersebut pun bisa diekstrak atau diambil. Penggunaan vocab data ini ditujukan untuk mencari kata ambigu dalam kalimat. Serta digunakan untuk memberi bobot nilai pada kata-kata yang ada dalam kalimat yang terdapat dalam vocab.txt. Pemberian bobot nilai digunakan untuk menghitung nilai *probability* dari kalimat.

Tahapan kedua adalah melatih program dengan menggunakan data *train*. Pada implementasi data *train* yang telah diberi label secara manual, dimana label terdiri dari dua *sense*. Kedua label akan digunakan sebagai kelas data untuk mengelompokkan kalimat-kalimat sesuai dengan *sense* yang sebenarnya. Kedua kelas data tersebut akan ditampung kedalam bentuk *dictionary* yang telah dibuat. Data *train* bertujuan untuk membuat model *probability* yang digunakan untuk mengklasifikasi *sense* pada data *test*. Perhitungan klasifikasi *naïve bayes* tidak bergantung pada jumlah atau seberapa banyak data yang digunakan. Perhitungan nilai *probability* dalam *naïve bayes* dipengaruhi oleh jumlah varian kalimat yang ada dalam data *train*.

Klasifikasi *naïve bayes* menggunakan *prior probability* (yaitu nilai probabilitas yang diyakini benar sebelum melakukan eksperimen) dari setiap label yang merupakan frekuensi masing-masing label pada data latih dan kontribusi dari masing-masing fitur [14]. Berdasarkan dari ciri alami dari sebuah model probabilitas, klasifikasi Naive Bayes bisa dibuat lebih efisien dalam bentuk pembelajaran *supervised* [14]. Sebagai sebuah model klasifikasi yang dihitung adalah p(H|X), berupa peluang untuk hipotesa valid dari data sampel X yang diamati. Dalam peneltian ini dapat dituliskan P(kata-i|sense-j) = P(sense-j|kata-i) P(kata-i)/ P(sense-j). Contoh perhitungan dapat dilihat pada tabel dibawah :

Tabel 3 : Data latih

No	Kalimat Latih	Sense
1	Aku pergi ke Yogyakarta pada bulan Juni	Waktu
2	Pada malam hari aku melihat bulan	Satelit
3	Bulan terasa sangat besar	Satelit

Perhitungan data latih maupun data uji sudah diterapkan dengan tahap – tahap sebelumnya yaitu pelabelan data dan preprocessing teks. Sehingga data latih yang berupa kalimat – kalimat memiliki kata ambigu bulan sudah bisa diolah sebagai data dalam tahap klasifikasi *naïve bayes*.

Tabel 4 : Perhitungan *Naïve Bayes*

No	Kalimat Uji	Sense
1	Pada malam di bulan Juni aku pergi ke Jakarta	Count prior prob : Peluang setiap sense dari kata bulan/ Pulang semua kata bulan (P{bulan|waktu}) : 1/3 (P{bulan|satelit}): 2/3 Vocab : Pergi, Juni, Malam, Terang, Melihat, Besar *Kata-kata unik dari dari testing Rumus menghitung Probability: $P(j) = \frac{count(j,i)+1}{count(i)+V}$ *Menerapkan rumus *naïve bayes word sense disambiguation* dari Stanford.edu

		Conditional probability : Peluang setiap kata pada kalimat tes dengan kata bulan memiliki sense waktu : P(Pergi|waktu) = (1+1)/(7+6) = 2/13 P(Malam|waktu) = (1+1)/(7+6) = 2/13 P(Juni|waktu) = (1+1)/(7+6) = 2/13 P(Jakarta|waktu) = (0+1)/(7+6) = 1/13 Peluang setiap kata pada kalimat tes dengan kata bulan memiliki sense waktu : P(Pergi|satelit) = (0+1)/(10+6) = 1/16 P(Malam|satelit) = (1+1)/(10+6) = 2/16 P(Juni|satelit) = (0+1)/(10+6) = 1/16 P(Jakarta|satelit) = (0+1)/(10+6)= 1/16 Choosing class: Menghitung *probability* setiap kelas P(waktu)= (P{bulan|waktu})*P(Pergi|waktu)*P(Malam|waktu)*P(Juni|waktu)*P(Jakarta|waktu) = 1/3 * 2/13 * 2/13 * 2/13*1/13 = 0.00009 P(satelit)= (P{bulan|satelit})*P(Pergi|satelit)*P(Malam|satelit)*P(Juni|satelit) *P(Jakarta|satelit) = 2/3 * 1/16 * 2/16 * 1/16 *1/16= 0.00002 Memilih nilai *probability* tertinggi Dari hasil diatas *Probability* tertinggi adalah P(waktu) maka kalimat tersebut mempunyai sense waktu.

3.4 Evaluasi Sistem

Setelah dilakukakan klasifikasi dengan algoritma *naïve bayes.* Algortima menghasil prediksi dari data *test.* Prediksi berupa file dalam bentuk txt. Hasil prediksi tersebut akan digunakan untuk mengevaluasi algoritma *naïve bayes*. Cara untuk mengavaluasinya menggunakan perbandingan jawaban antara hasil prediksi algoritma dengan jawaban sesungguhnya yang telah disiapkan. Setiap jawaban benar dari hasil prediksi yang telah dicocokan dengan jawaban sesungguhnya. Maka akan mendapatkan nilai 1. Nilai tersebut akan dijumlahkan sebanyak jumlah jawaban prediksi. Lalu akan dibagi dengan jumlah jawaban hasil prediksi, sehingga menghasilkan nilai akurasi dari algortima. Hasil tersebut digunakan untuk mengevaluasi sistem yang dibuat. Tahapan selanjutnya adalah menganalisis nilai akurasi yang didapat. Analisis bertujuan untuk menemukan faktor – faktor yang dapat memengaruhi algoritma *naïve bayes* dalam *word sense disambiguation* Bahasa Indonesia.

4. Evaluasi

4.1 Hasil dan Analisis Pengujian

Berdasarkan hasil penelitian dengan jumlah data uji sebanyak 20 kalimat yang terdapat kata ambigu. Dimana data tes yang digunakan menggunakan kata “bulan” sebagai kata yang dianggap ambigu. Arti kata “bulan” memiliki dua *sense* kata yaitu bulan memiliki arti waktu serta bulan memiliki arti satelit.

Penelitian ini menggunakan bahasa pemrograman *python* sebagai pengimplementasiaan klasifikasi *naïve bayes*. Sedangkan untuk data berupa teks berbahasa Indonesia formal. Dalam *preprocessing* teks menggunakan *library python* yaitu NLTK dan Pysastrawi. Penelitian juga menggunakan modul dalam

python yaitu *regex,* dimana modul ini digunakan untuk mencari pola *string* tertentu dari kumpulan teks. Modul ini akan mengekstrak *string* jika cocok dengan pola yang sudah dibuat. Penggunaan modul ini bertujuan untuk memermudah klasifikasi algoritma *naïve bayes* untuk mendapatkan nilai *probability* dari data *test*. Setelah dilakukan klasifikasi dengan *naïve bayes* didapatkan hasil sebagai berikut:

Tabel 5 : Hasil Percobaan Pertama

Prediksi	Sense sebenarnya		Total sense data	Akurasi
	Satelit	Waktu		
Satelit	7	3	Bulan : satelit = 10	80%
Waktu	2	8	Bulan : waktu = 10	
Total Data	9	11	20	

Tabel 6 : Hasil Percobaan Kedua

Prediksi	Sense sebenarnya		Total sense data	Akurasi
	Satelit	Waktu		
Satelit	13	0	Bulan : satelit = 13	85%
Waktu	2	5	Bulan : waktu = 7	
Total Data	15	5	20	

Tabel 7 : Hasil Percobaan Ketiga

Prediksi	Sense sebenarnya		Total sense data	Akurasi
	Satelit	Waktu		
Satelit	7	1	Bulan : satelit = 8	65%
Waktu	5	7	Bulan : waktu = 12	
Total Data	12	8	20	

Hasil penelitian diatas mendapatkan nilai akurasi dari tiga pengujian dengan kasus yang berbeda – beda. Tabel 3 dengan kasus jumlah *sense* atau data tes berjumlah sama masing – masing 10 data. Tabel 4 dengan kasus jumlah sense "satelit" lebih banyak dibandingkan sense "waktu". Sedangkan tabel 5 dengan kasus jumlah sense "waktu: lebih banyak dibanding dengan jumlah sense "satelit".

Penelitian yang telah dilakukan dengan melihat perbandingan hasil akurasi tabel 5, tabel 6 dan tabel 7. Ketidakseimbangan jumlah *sense* dapat memengaruhi nilai akurasi. Dimana dalam data *train* sebanyak 105 kalimat. Dengan perbandingan jumlah kalimat yang mengandung makna satelit sebanyak 58 kalimat. Sedangkan kalimat yang mengandung makna waktu sebanyak 47 kalimat. Dengan perbandingan lebih banyak kalimat bermakna satelit. Maka lebih banyak varian kalimat yang memiliki *sense* satelit. Dalam permasalahan *words sense disambiguation* dimana varian dari variabel kelas akan lebih berpengaruh terhadap hasil akurasi. Dikarenakan dalam *word sense disambiguation* ini, variabel kelas hanya ada satu. Oleh karena itu variasi kalimat ambigu sesuai dengan jumlah kalimat dalam data *training*. Sehingga jumlah data atau kalimat dalam data *train* memiliki pengaruh terhadap hasil akurasi. Selain itu penentuan kata dalam vocab data memiliki pengaruh untuk dapat menentukan predikisi. Sebagai contoh dapat dilihat pada tabel dibawah :

Tabel 8 : Kesalahan dalam Prediksi

Kalimat	Sense prediksi	Sense sesungguhnya
300022 <tag>bulan<tag/> ini saya akan menghadap pemimpin perusahaan karena kenaikan pangkat.	Satelit	Waktu
300028 Pertemuan keduanya terjadi sangat singkat. Entah karena suatu hal mereka tidak akan saling bertemu lagi dalam beberapa <tag>bulan<tag/> ke depan.	Satelit	Waktu

300039 Beberapa hari ke depan selama musim hujan, tak ada satupun nampak cahaya <tag>bulan<tag/> pada malam hari.	Waktu	Satelit

Berdasarkan contoh pada tabel 8 kesalahan yang terjadi dalam pembobotan kata mengakibatkan kesalahan predikisi algoritma naïve bayes. Kesalahan terjadi seperti pada kalimat 300022 tidak adanya kata – kata yang masuk kedalam vocab data. Sehingga program kekurangan bobot untuk menentukan *sense* yang tepat. Program menghitung setiap kata dalam kalimat 300022. Namun adanya kekurangan indikator yang menguatkan nilai perhitungan. Menyebabkan program kurang tepat dalam memprediksi *sense*. Selain adanya kekurangan bobot, program menemukan bobot ganda. Maksud dari bobot ganda adalah dimana ditemukan kata yang sama muncul pada dua atau lebih kalimat dengan sense yang berbeda. Contoh ditemukan kata "depan" pada kalimat 300028, dan 300039. Kata "depan" ini akan dihitung sebagai salah satu bobot didalam kalimat dengan makna satelit serta waktu. Program akan menggunakan kata "depan" sebagai salah satu indikator dalam menentukan *sense* suatu kalimat. Dalam kasus kalimat 300028 dan 300039 kata "depan" akan dihitung sebagai bagian dari vocab data yang ada kalimat dengan kata bulan memiliki *sense* waktu ataupun satelit. Sehingga menyebabkan program akan memiliki perhitungan ganda dalam menghitung kata "depan" tersebut. Jadi dapat disimpulkan bahwa permasalahan penentuan kata pada vocab data, dimana kata tersebut dijumpai pada kalimat – kalimat dengan *sense* yang berbeda. Maka akan mempengaruhi proses klasifikasi pada sistem. Permasalahan tersebut juga dapat mempengaruhi tingkat akurasi sistem, sehingga mengakibatkan kinerja sistem tidak berjalan secara maksimal. Sebagai hasil evaluasi akhir dari sistem yang dibangun, nilai akurasi yang didapat sebesar 76.67%. Nilai tersebut berasal dari perhitungan rata - rata dari ketiga percobaan yang telah dilakukan.

5. Kesimpulan

Kesimpulan yang didapat dari penelitian yang bertujuan untuk mengiplementasikan algortima *naïve bayes* terhadap *word sense disambiguation* dalam Bahasa Indonesia telah berhasil dilakukan. Berdasarkan hasil dan analisis yang telah dilakukan. Didapatkan bahwa klasifikasi *word sense disambiguation* Bahasa Indonesia dapat dilakukan dengan menggunakan algoritma *naïve bayes*. Dengan membuat *sense* dari kata yang ambigu menjadi kelas dalam algortima *naïve bayes*. Maka kalimat – kalimat akan dikelompokkan sesuai dengan kelasnya. Sehingga saat proses klasifikasi, sistem hanya akan menentukan data *test* sesuai dengan kelas yang sebenarnya. Serta dengan adanya tahap *preproccesing* yang dapat mengubah kalimat menjadi data yang siap untuk diolah dalam klasifikasi. Serta dengan adanya vocab data dapat memermudah sistem dalam mengklasifikasi kalimat – kalimat untuk menemukan *sense* yang sebenarnya. Semakin banyak data *train* yang digunakan akan memengaruhi hasil kerja sistem. Nilai akurasi yang didapat akan semakin tinggi. Hal ini akan menandakan bahwa sistem berhasil melakukan klasifikasi. Akurasi tertinggi yang didapat adalah 85%, dan akurasi terendah adalah 65%. Sedangkan sebagai evaluasi sistem digunakan nilai akurasi rata – rata penelitian yaitu 76.67%. Hasil penelitian ini dapat digunakan sebagai bahan studi untuk penelitian *word senses disambiguation* Bahasa Indonesia. Kedepanya dapat diterapkan kedalam penelitian serupa dengan menggunakan metode – metode lain. Serta tidak hanya mengidentifkasi kata dengan dua *sense* yang berbeda.