PROGRAM STUDI SARJANA SISTEM INFORMASI

PROPOSAL PROYEK AKHIR MATA KULIAH

11S4037 - VISUALISASI DATA

KLASIFIKASI TEKS PADA JUDUL BERITA

OLEH:

12S16008	**Alfendo S. P. Situmorang**
12S16030	**Boas Demeson Pangaribuan**
12S16050	**Reinheart Christian Simanungkalit**
12S17041	**Dewi Purnama Napitupulu**

PROGRAM STUDI SARJANA SISTEM INFORMASI

FAKULTAS TEKNOLOGI INFORMATIKA DAN ELEKTRO

INSTITUT TEKNOLOGI DEL

2020

DAFTAR ISI

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Setiap orang tentunya tidak asing dengan berita. Berita adalah sebuah informasi yang sifatnya fakta yang sedang terjadi maupun sudah terjadi dan disampaikan melalui perantara media, baik itu media elektronik maupun mefia like cetak. Penyampaian berita juga bisa melalui mulut ke mulut dan harus merupakan sebuah kenyataan, bukan karangan fiktif atau cerita yang dibuat-buat. Namun, saat ini yang menjadi fokus penulis adalah berita yang berbentuk teks. Setiap Teks berita tentunya memiliki isi yang berbeda satu dengan yang lainnya. Satu teks berita tentunya memuat konten yang berbeda dengan berita yang lain. Hal yang mempengaruhi perbedaan isi setiap berita adalah topik atau kategori utama berita tersebut. Setiap berita tentunya dapat dikelompokkan kedalam suatu kategori sehingga siapapun yang ingin membaca sebuah berita mengetahui berita yang akan dibacanya dengan mengetahui kategori berita tersebut terlebih dahulu.

Berita Tentunya memiliki judul. Judul bertujuan untuk merepresentasikan apa yang dimuat dalam suatu teks berita secara keseluruhan. Melalui kata yang dimuat pada sebuah judul berita, tentunya kita pun dapat mengetahui kategori berita tersebut. Terdapat sebuah metode pada Pemrosesan Bahasa Alami, yang dapat mengklasifikasikan berita. Namun, harus ditentukan terlebih dahulu apa yang digunakan untuk mengklasifikasikan berita tersebut. Pada kasus ini penulis berfokus pada judul berita. Sebelum melangkah lebih jauh, maka harus dipahami apa maksud dari klasifikasi dalam Pemrosesan Bahasa Alami. Klasifikasi merupakan teknik dalam penambangan data untuk mengelompokkan data berdasarkan keterikatan data terhadap data sampel. Meskipun klasifikasi merupakan suatu teknik dalam penambangan data, namun dapat diterapkan pada Pemrosesan Bahasa Alami apabila data yang digunakan adalah data yang berbentuk teks. Karena, pada kasus ini penulis memiliki focus untuk mengklasifikasikan teks pada judul berita.

1.2 Tujuan

Tujuan dari proyek pemrosesan bahasa alami ini adalah mengklasifikasikan sebuah berita dengan mengggunakan teks yang ada pada judul berita.

1.3 Ruang Lingkup

Ruang lingkup pada proyek ini adalah, penulis akan menggunakan judul pada teks berita sebagai data utama untuk melakukan klasifikasi.

BAB 2
LANDASAN TEORI

2.1 *Natural Languange Processing*

NLP merupakan sebuah cabang ilmu *Artificial Inteligent* yang berfokus pada pengolahan Bahasa natural. Bahasa natural merupakan bahasa umum yang digunakan untuk berkomunikasi antar manusia. Sedangkan untuk bahasa komputer sendiri, diperlukan sebuah pemrosesan tersendiri agar apa yang dimaksud oleh pengguna dapat dipahami oleh komputer [1].

Area utama pada NLP, diantaranya [2]:

a. *Question Answering System* (QAS), merupakan kemampuan komputer untuk menjawab pertanyaan yang diajukan oleh penggunanya. Dalam hal ini, pengguna tidak lagi memasukkan kata kata kunci berkenaan dengan jawaban yang diinginkan melainkan pengguna memasukkan pertanyaan secara langsung ke dalam komputer.
b. *Summarization*, merupakan pembuatan ringkasan dari kumpulan dokumen atau email sehingga pengguna dibantu untuk mengkonversikan dokumen teks yang besar ke dalam sebuah slide presentasi.
c. *Machine Translation*, salah satu hasil aplikasi yang telah menarapkan cabang ilmu ini adalah *Google Translate*, dimana output dari aplikasi yang dibuat dapat memahami bahasa manusia sekaligus menerjemahkannya ke dalam bahasa lain seperti dari bahasa Indonesia ke dalam bahasa Inggris, atau sebaliknya.
d. *Speech Recognition*, merupakan cabang ilmu NLP yang cukup sulit. Hal ini dikarenakan komputer harus memahami bahasa manusia yang diucapkan. Dan bentuk kalimat yang sering digunakan dalam cabang ilmu ini adalah kalimat tanya dan perintah.
e. *Document Classification*, adalah salah satu cabang ilmu NLP yang paling sukses. Pekerjaan yang dilakukan melalui cabang ilmu ini yaitu menentukan dimana tempat terbaik dokumen yang baru diinputkan ke dalam sistem. Hal ini sangat berguna pada aplikasi *spam filtering, news article classification,* dan *movie review.*

2.2 Klasifikasi Teks

Klasifikasi teks adalah tugas penting di *Natural Language Processing* dengan banyak aplikasi, seperti sebagai pencarian web, pencarian informasi, peringkat dan klasifikasi dokumen (Deerwester et al., 1990; Pang dan Lee, 2008). Klasifikasi teks merupakan proses penentuan kategori suatu dokumen teks sesuai dengan karakteristik dari teks tersebut. Dalam prosesnya klasifikasi teks terdiri dari 5 komponen yaitu pengumpulan data, *pre-processing*, ekstraksi fitur, klasifikasi dan evaluasi. Tahapan *pre-processing* terdiri dari *case folding, tokenizing, filtering* dan *stemming*. Selain itu pembentukan kamus, pemilihan fitur dan pembobotannya merupakan bagian dari tahap ini.

2.3 Berita

Berita adalah sebuah informasi yang bersifat nyata atau fakta, baik yang sedang terjadi maupun sudah terjadi. Berita disampaikan melalui perantara media, terutama pada media sosial. Umumnya suatu berita yang disampaikan, harus disajikan berdasarkan fakta agar tidak ada pembaca yang salah dalam mengartikan suatu berita. Berita juga pada umumnya mempengaruhi masyarakat secara luas, oleh karena itu berita harus disajikan dengan hati-hati.

2.4 *Preprocessing*

Tahap *preprocessing* bertujuan untuk mempersiapkan teks yang akan diringkas menjadi data yang siap diproses pada tahap selanjutnya [3]. *Preprocessing* sangat penting dalam melakukan analisis sentimen, terutama untuk media sosial yang pada umumnya berisi kata-kata yang tidak formal dan tidak terstruktur serta memiliki banyak *noise*. Gambar 1 merupakan jalur dari proses *preprocessing* :

Gambar 1 Tahapan *Preprocessing*

2.4.1 *Case Folding*

Case Folding merupakan langkah selanjutnya pada *Preprocessing* teks. *Data Cleaning* berfungsi untuk mengubah huruf yang terdapat di dalam teks menjadi *lowercase* semua [4]. *Case folding* adalah proses membarui huruf-huruf yang ada dalam suatu teks menjadi huruf kecil (Rustiana & Rahayu, 2017). Contoh dari prosedur *case folding* adalah mengubah kata "Nilai" menjadi kata "nilai" yang memiliki huruf kecil semua [5].

2.4.2 *Tokenization*

Proses tokenisasi adalah pemecahan kata-kata yang ada di suatu kalimat (Robinson, 2014). Tokenisasi dilakukan dengan memisahkan setiap kata dengan spasi. Contoh dari tokenisasi adalah memecah kalimat "saya pergi ke kantor polisi" menjadi kumpulan kata-kata "saya", "pergi", "ke", "kantor", "polisi" [5]. *Tokenizing* merupakan langkah untuk memotong dokumen menjadi potongan-potongan kecil yang disebut token dan terkadang disertai langkah untuk membuang karakter tertentu seperti tanda baca (Manning, Raghavan, dan Schultze, 2009). Tokenisasi yang dilakukan adalah memisah kalimat berdasarkan karakter spasi (" "). Berikut adalah contoh dari *tokenizing* atau *Tokenization* [4].

Sebelum	Sesudah
Sarapan di pagi hari sangat penting bagi kesehatan.	'Sarapan', 'di', 'pagi', 'hari', 'sangat', 'penting', 'bagi', 'kesehatan'

Gambar 2 Contoh Tokenisasi

2.4.3 *Stopword Removal*

Tahap ini berfungsi untuk menghilangkan kata-kata yang tidak penting dalam proses klasifikasi dan penentuan alasan, seperti kata: "yang", "tetapi", "atau", "ke", "di", "dengan", dan sebagainya [6]. *Stopword* merupakan kumpulan kata umum. *Stopword* harus dibuang untuk memudahkan pengolahan teks (Raulji & Saini, 2016). Pada penelitian ini digunakan *stopword* milik Talla F. Z. yang tersedia di https://github.com/masdevid/ID-Stopwords . Contoh dari proses *stopword removal* adalah menghapus kata-kata "saya" dan "ke" dari kalimat "saya pergi ke kantor polisi" [5].

2.4.4 *Stemming*

Langkah selanjutnya untuk membuat kata dalam teks sebagai kata dasar disebut dengan *stemming*. *Stemming* mampu menaikkan 10 sampai 50 kali jumlah dokumen yang ingin didapat (Sandhya et al, 2011). Pada beberapa penelitian teks berbahasa indonesia dapat digunakan *stemmer* Bahasa Indonesia milik Sastrawi yang tersedia di https://github.com/sastrawi/sastrawi. Contoh dari *stemmer* milik Sastrawi adalah kata "seekor" menjadi "ekor" dan "menginap" menjadi "inap" [5].

2.5 *Feature Extraction*

Komputer (mesin) tidak dapat mengolah data selain data numerik, sehingga dibutuhkan langkah untuk mengekstrak "kata" menjadi numerik. Metode ekstraksi fitur digunakan sebagai tahap awal dalam metode komputasi guna merepresentasikan data secara menyeluruh. Metode ekstraksi fitur yang efektif akan menuntun ke sebuah peningkatan performa model menjadi lebih baik. Selain itu pula, metode ekstraksi fitur berguna untuk menggali informasi potensial dan merepresentasikan sebuah sampel asli sebagai vektor fitur yang akan digunakan sebagai input untuk metode *machine learning* pada tahap selanjutnya [7]. Secara umum terdapat 3 teknik ekstraksi fitur, yaitu:

1. *Bag of Word* (TF, TFIDF)
2. *Word Embedding* (Glove, Word2vec, FastText)
3. *Character Embedding*

Ketiga teknik diatas dapat digunakan dalam penelitian analisis sentimen ataupun case penelitian lainnya dalam topik NLP, seperti *Language Modelling*, dan *Named Entity Recognition*. Ketiga pendekatan diatas memiliki perbedaan yang sangat signifikan. BoW (*Bag of Word*) akan merubah term (kata) dalam sebuah kalimat menjadi skalar, *Word Embedding* akan merubah sebuah kata menjadi vektor dengan dimensi tertentu, sedangkan *Character Embedding* akan merubah sebuah huruf menjadi vektor dengan dimensi tertentu.

Word embedding adalah sebuah fungsi parameter yang memetakan setiap kata ke dalam vektor berdimensi tinggi. Keunggulan *word embedding* tidak membutuhkan anotasi, dapat langsung diturunkan dari korpus tak teranotasi. *Word embedding* dapat dibuat langsung dari dataset yang dimiliki atau menggunakan *pre-trained word embedding* yang telah tersedia. *Pre-trained word*

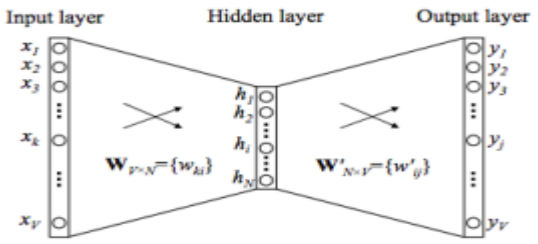
embedding ini adalah *word embedding* yang telah dilatih menggunakan dataset yang besar pada domain permasalahan tertentu yang dapat digunakan untuk menyelesaikan permasalahan lain yang serupa. Penggunaan *word embedding* ini harus disesuaikan dengan domain dari kasus yang dimiliki. Misalkan permasalahan pada domain biomedik tidak akan cocok menggunakan *pretrained word embedding* dari korpus berita atau Wikipedia [8].

2.5.1 Word2vec

Word2vec merupakan salah satu algoritma word embedding yang memetakan setiap kata dalam teks ke dalam vektor. Algoritma word2vec ini diciptakan oleh Mikolov dkk. pada tahun 2013. Sejak kemunculannya, model *word embedding* ini banyak digunakan dalam penelitian NLP. Word2vec merepresentasikan kata ke dalam vektor yang dapat membawa makna semantik dari kata tersebut. Model *word embedding* ini merupakan salah satu aplikasi *unsupervised learning* menggunakan neural network yang terdiri dari sebuah *hidden layer* dan *fully connected layer*. Dimensi dari matriks bobot pada setiap layer adalah jumlah dengan kata dalam korpus dikalikan dengan jumlah hidden neuron pada hidden layer-nya. Matriks bobot pada *hidden layer* dari model yang telah dilatih digunakan untuk mentransformasikan kata ke dalam vektor. Matriks bobot ini seperti *lookup table*, di mana setiap baris mewakili setiap kata dan kolom mewakili vektor dari kata tersebut. Wod2vec mengandalkan informasi lokal dari bahasa. Semantik yang dipelajari dari kata tertentu dipengaruhi oleh kata-kata sekitarnya. Model ini mendemonstrasikan kemampuan untuk mempelajari pola linguistik sebagai hubungan linear antarvektor kata. Terdapat dua algoritma word2vec yaitu *Continuous Bagof-Word* (CBOW) dan *Skip-gram* [8].

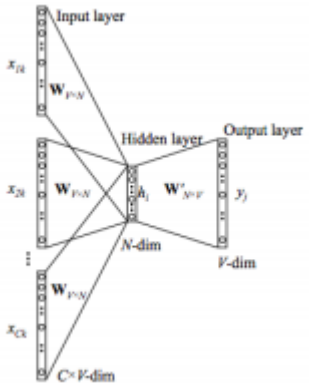
a. CBOW

   Model ini menggunakan konteks untuk memprediksi target kata. CBOW memiliki waktu *training* lebih cepat dan memiliki akurasi yang sedikit lebih baik untuk *frequent words*.



Gambar 3 *One Word Context* CBOW

Sumber: Rong, 2016 **(**Jurnal TEKNOKOMPAK: Perbandingan Kinerja *Word Embedding* Word2vec, Glove, Dan *FastText* Pada Klasifikasi Teks, **Arliyanti Nurdin, dkk., 2020)**

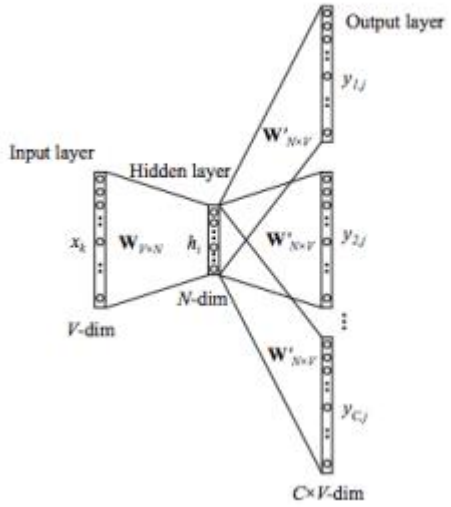


Gambar 4 *Mutiple Context Words* CBOW

Sumber: Rong, 2016 **(**Jurnal TEKNOKOMPAK: Perbandingan Kinerja *Word Embedding* Word2vec, Glove, Dan *FastText* Pada Klasifikasi Teks, **Arliyanti Nurdin, dkk., 2020)**

b. Skip-Gram

Model ini menggunakan sebuah kata untuk memprediksi target konteks. Skip-Gram bekerja dengan baik dengan data pelatihan yang jumlahnya sedikit dan dapat merepresentasikan kata-kata yang dianggap langka.



Gambar 5 *Skip-Gram*

Sumber: Rong, 2016 **(**Jurnal TEKNOKOMPAK: Perbandingan Kinerja *Word Embedding* Word2vec, Glove, Dan *FastText* Pada Klasifikasi Teks, **Arliyanti Nurdin, dkk., 2020)**

2.5.2 GloVe

Berbeda dengan word2vec yang hanya mengandalkan informasi lokal dari kata dengan *local context window* (CBOW dan Skip-gram), algoritma GloVe juga menggabungkan informasi *co-occurrence* kata atau statistik global untuk memperoleh hubungan semantik antarkata dalam korpus. GloVe menggunakan metode *global matrix factorization,* matriks yang mewakili kemunculan atau ketiadaan kata-kata dalam suatu dokumen (Pennington, Socher and Manning, 2014). Word2vec adalah model *feedforward neural network* sehingga sering disebut sebagai *neural word embeddings*, sedangkan GloVe adalah model *log-bilinear* atau secara sederhana dapat disebut sebagai model berbasis hitungan. GloVe mempelajari hubungan katakata dengan menghitung seberapa sering kata-kata muncul bersama satu sama lain dalam sebuah korpus yang diberikan. Rasio probabilitas kemunculan kata-kata memiliki potensi untuk mengkodekan beberapa bentuk makna serta membantu meningkatkan kinerja pada permasalahan analogi kata. Pelatihan model GloVe bertujuan untuk mempelajari vektor kata sedemikian rupa sehingga dot product katakata tersebut sama dengan logaritma probabilitas katakata untuk muncul bersama atau

probabilitas *cooccurence*-nya. Algoritma GloVe terdiri dari langkah – langkah berikut (Selivanov, 2020) [8]:

1. Mengumpulkan statistik *word co-occurence* dalam bentuk sebuah matriks *word co-occurence* X. Setiap elemen Xij merepresentasikan berapa kali kata i muncul dalam konteks kata j.
2. Tentukan *soft constraints* untuk setiap pasangan kata :

   $$w_i^T w_j + b_i + b_j = \log(X_{ij})$$

   Di mana wi – vektor kata utama, wj- vektor kata konteks, bi,bj bias skalar untuk kata-kata utama dan kata-kata konteks.
3. Tentukan sebuah *cost function*.

   $$J = \sum_{i=1}^{V}\sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - logX_{ij})^2$$

   Di mana f fungsi pembobotan yang membantu kita mencegah belajar hanya dari pasangan kata yang sangat umum. Fungsi tersebut didefinisikan sebagai berikut:

   $$f(X_{ij}) = \begin{cases} (\frac{x_{ij}}{x_{max}})^a & if\ X_{ij} < XMAX \\ 1 & otherwise \end{cases}$$

2.5.3 FastText

FastText adalah metode *word embedding* yang merupakan pengembangan dari word2vec. Metode ini mempelajari representasi kata dengan mempertimbangkan informasi *subword*. Setiap kata direpresentasikan sebagai sekumpulan karakter ngram. Dengan demikian dapat membantu menangkap arti kata-kata yang lebih pendek dan memungkinkan embedding untuk memahami sufiks dan prefiks dari kata. Representasi vektor dikaitkan dengan setiap karakter ngram, sedangkan kata-kata direpresentasikan sebagai jumlah dari representasi vektor tersebut. Setelah kata direpresentasikan dengan karakter n-gram, model Skipgram dilatih untuk mempelajari *embedding* vektor dari kata. Pada umumnya model yang mempelajari representasi kata ke dalam vektor mengabaikan morfologi kata, setiap kata memiliki vektor yang berbeda. Hal ini menjadi keterbatasan untuk merepresentasikan kata dari bahasa dengan kosakata yang besar dan memiliki banyak katakata langka. *FastText* memiliki kinerja yang baik, dapat melatih model pada dataset yang besar dengan cepat dan dapat memberikan representasi kata yang tidak muncul dalam data

latih. Jika kata tidak muncul selama pelatihan model, kata tersebut dapat dipecah menjadi n-gram untuk mendapatkan *embedding* vektornya [8].

2.6 Metode Klasifikasi

Klasifikasi adalah bentuk dasar dari analisis data. Klasifikasi juga biasa disebut teknik yang digunakan untuk menentukan anggota kelompok dari data yang telah tersedia. Konsep dasar dari klasifikasi adalah sejumlah data yang mempunyai struktur data yang hampir sama atau serupa akan menghasilkan klasifikasi yang hampir sama atau serupa juga. Klasifikasi adalah metode data mining yang banyak diterapkan di berbagai bidang [9].

Dalam memasuki era *Artificial Intelligence* (AI), terdapat banyak masalah yang lebih kompleks dan rumit sedang dihadapi oleh para peneliti. *Deep Learning* adalah pendekatan yang ampuh untuk menjembatani kesenjangan antara pemikiran manusia dan logika komputer. *Deep Learning* adalah bidang keilmuan dari *Machine Learning* yang memiliki algoritma-algoritma yang digunakan untuk menyelesaikan masalah dengan volume data yang sangat besar dan banyak komputasi [10]. Ada beberapa metode *Machine Learning,* khususnya dalam bidang keilmuan *Deep Learning* yang dapat digunakan untuk melakukan klasifikasi teks, antara lain:

1. Metode *Convolutional Neural Network (CNN)*
2. Metode *Recurrent Neural Networks*
3. Metode *Maximum Entropy*

2.6.1 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) pada awalnya dikembangkan oleh LeCun, dkk. (1998) untuk klasifikasi angka tulisan tangan dan sekarang dianggap sebagai sistem untuk semua jenis pekerjaan yang terkait dengan klasifikasi atau ekstraksi fitur gambar. Hal tersebut termasuk dalam kelas *feed forward artificial neural network* (ANN). Mekanisme tersebut menggunakan model *perceptron multi-layer* setidaknya untuk *prepocessing* [10]. CNN memiliki kemampuan untuk mengekstrak fitur dari data yang disediakan sebagai masukan.

Convolution Layer bertanggung jawab untuk ekstraksi fitur dari vektor kata yang disediakan sebagai masukan. Output dari lapisan ini, digabungkan dalam kasus model berikut, diumpankan ke lapisan penggabungan maks global dengan penghentian diaktifkan. Lapisan ini selanjutnya diikuti oleh 3 lapisan padat yang terhubung sepenuhnya, lapisan terakhir dengan satu neuron, yang

bertanggung jawab untuk klasifikasi. Kami telah menggunakan putus sekolah di lapisan penyatuan maks global untuk mengurangi overfitting, yang sudah rendah karena besar Himpunan data. Namun, pada penerapannya, perbedaannya antara akurasi validasi dan akurasi pelatihan berkurang, juga mengarah ke konvergensi yang lebih baik.

2.6.3 *Maximum Entropy*

Maximum Entropy juga merupakan metode *supervised learning* yang digunakan dengan sangat baik dalam banyak aplikasi pemrosesan bahasa alami. Terkadang memberikan kinerja yang lebih baik daripada Klasifikasi Naïve Bayes dalam klasifikasi teks. *Maximum Entropy* adalah pengklasifikasi probabilistik. Hal ini dibedakan dari pengklasifikasi Naïve Bayes karena Naïve Bayes menganggap bahwa suatu peristiwa adalah independen satu sama lain sedangkan pendekatan *Maximum Entropy* tidak menganggap suatu peristiwa addalah independen. Metode ini memilih data yang paling sesuai dengan *data test* dan memiliki entropi maksimum diantara data-data tersebut. Metode ini dapat digunakan untuk masalah yang berbeda seperti klasifikasi teks, analisis sentimen, deteksi bahasa, dan klasifikasi gambar [11].

Pada penelitian ini dipilih algoritma pembelajaran mesin *Maximum Entropy*. Algoritma *Maximum Entopy* dipilih karena memiliki akurasi lebih tinggi daripada *Naïve Bayes* pada penelitian (Masithoh, 2016) [5].Teknik yang dipakai guna mencari kemungkinan dengan nilai *entropy* paling tinggi (Ahmad, 2011) disebut dengan *Maximum Entropy*. Nilai *entropy* dipakai untuk mendapatkan nilai *Maximum Entropy*. Rumus *Maximum Entropy* pada Persamaan berikut [5].

$$Entropy(X) = -\Sigma i=1nP(Xi)\log 2P(Xi)$$

Keterangan:

- Entropy(X) = Himpunan informasi dari suatu kejadian x
- P(X) = Probabilitas dari kemunculan kejadian x

Proses klasifikasi pada metode *Maximum Entropy* hanya menggunakan informasi kemunculan dari suatu fitur dalam sebuah dokumen (Anggraeni, 2008). Secara garis besar, metode *Maximum Entropy* mencari distriusi probabilitas yang paling sama dengan menggunakan asumsi minimal. Pada kasus klasifikasi teks, *Maximum Entropy* menggunakan rumus pada Persamaan berikut [5].

$$P(c|d) = 1Z(d)\exp(\Sigma\lambda i f i(d,c))$$

Keterangan:

- P(c|d) = Probabilitas kemunculan *term* d di kelas c
- Z(d) = Derajat kepangkatan *term* d
- λ = Parameter
- $fi(d,c)$ = Probabilitas kemunculan *term* d di kelas c

Nilai Z(d) dapat dihitung menggunakann rumus pada Persamaan berikut [4].

- $Z(d)=\Sigma\exp[fo](\Sigma\lambda i f i(d,c))$

2.6.4 Teknik Klasifikasi Teks

Keuntungan dan kerugian dari berbagai teknik klasifikasi dirangkum pada hasil analisis di bawah ini [12]:

Teknik	Kelebihan	Kekurangan
Statistical Approaches:		
Naive Bayes	• Cepat dan sangat mudah diterapkan. • Kebutuhan memori rendah.	• Mengasumsikan bahwa fitur kelas saling independen.
Decision Tree	• Mudah diinterpretasikan dan dibaca. • Akurasi meningkat seiring waktu.	• Cenderung overfit. • Penghitungan tumbuh secara eksponensial dengan peningkatan data.
Support Vector Machines	• Berfungsi paling baik pada kumpulan data kecil.	• Mengasumsikan bahwa data bersifat independen bersyarat.
Maximum Entropy	• Tidak membuat asumsi apa pun sehubungan dengan dependensi dalam data. • Memberikan akurasi dan efisiensi tinggi.	• Kebutuhan memori tinggi. • Implementasi yang kompleks.

Knowledge/Lexicon based approaches:	• Pendekatan berbasis leksikon lebih disukai daripada pendekatan statistik, ketika sentimen yang diungkapkan oleh kata-kata tertentu adalah domain tertentu atau dalam kasus konteks linguistik.	• Ontologi spesifik domain harus didefinisikan dengan baik.
Hybrid approaches:	• Peningkatan presisi, akurasi, *recall*, dan *f-score*.	• Menuntut pengetahuan lanjutan tentang semua pendekatan. • Kompleksitas dalam implementasi.
Other Approaches :		
Emoticon-based approach	• Lebih mudah diterapkan karena jumlah emotikon yang terbatas.	• Deteksi sarkasme diperlukan dengan pose menantang.
Volume-based approach	• Mudah diimplementasikan karena bergantung pada frekuensi kuantitatif.	• Jika jumlah kemunculannya lebih sedikit, maka menjadi kurang akurat dan sulit untuk diklasifikasikan.
Novel approaches	• Kebaruan dan inovasi	• Digunakan untuk proyek dan studi tertentu.

2.7 Python

Python adalah bahasa pemrograman tingkat tinggi sekaligus bahasa pemrograman yang interpretatif dan multiguna. Bahasa pemrograman ini pertama kali muncul pada tahun 1991 dan dirancang oleh Guido van Rossum di CWI, Amsterdam. *Python* dikenal sebagai bahasa pemrograman yang mudah untuk dipelajari baik untuk pemula maupun pengguna yang sudah *expert*. *Python* dapat digunakan untuk mengembangkan berbagai macam keperluan perangkat lunak. Untuk mendukung pengembangan perangkat lunak, *python* menyediakan *standard library*.

Hal tersebut dikarenakan *python* lebih menekankan pada keterbacaan (*readability*) dari kode. Kemudahan untuk membaca kode dapat membantu dalam memahami *syntax* [13].

Python merupakan bahasa pemrograman tingkat tinggi. Hal ini disebabkan karena kode yang dituliskan akan di-*compile* menjadi *byte code* dan dieksekusi sehingga Python cocok digunakan untuk *scripting language*, aplikasi web dan lain sebagainya. Hal lain yang menjadikan bahasa ini menjadi bahasa pemgrograman tingkat tinggi adalah Python dapat di-*extend* kedalam bahasa C dan C++ serta bahasa pemgrograman ini memiliki struktur konstruksi yang kuat (blok kode, fungsi, *class*, *modules*, dan *packages*) dan serta konsisten menggunakan konsep *Object Oriented Programming* (OOP) (Kuhlman, 2015) [14].

BAB 3 METODE DAN JADWAL PENELITIAN

3.1 Metode Penelitian

Metode Penelitian ini merupakan pedoman melakukan penelitian seperti langkah-langkah yang digunakan, pengumpulan data serta analisis data serta instrumen penelitian yang digunakan.

Berikut ini merupakan penjelasan terhadap *flow* metodologi.

1. Permusan Masalah

   Pada tahap ini menjabarkan mengenai hal yang melatar belakangi permasalahan terkait dengan *Text Classification.* Perumusan masalah melalui studi tentang *Text Classification* seperti buku dan jurnal penelitian.
2. Pengumpulan Data

   Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan proyek.
3. Analisis

   Pada tahap ini akan dilakukan analisis terhadap judul berita yang sudah di proses pada langkah sebelumnya, menganalisis pemilihan algoritma yang akan dipakai dan pemilihan metode pendekatan *Text Classification* yang diterapkan pada proyek.
4. Desain

   Pada tahap ini perancangan dilakukan berdasarkan hasil analisis yang telah dilakukan sebelumnya.
5. Implementasi

   Tahap ini dilakukan untuk membangun sistem berdasarkan perancangan yang telah dilakukan sebelumnya.
6. Pengujian dan Evaluasi

   Pada tahap ini dilakukan pengujian untuk menentukan apakah sistem yang telah dibangun berhasil mencapai tujuan atau tidak.
7. Kesimpulan dan Saran

   Pada akhir proyek, dilakukan pengambilan kesimpulan terkait *Text Classification* pada pengambilan judul berita dan saran untuk penelitian selanjutnya.

3.2 **Jadwal Penelitian**

Adapun yang menjadi jadwal penelitian ditunjukkan pada Tabel

Tabel 1 Jadwal Penelitian

Kegiatan	Minggu																															
	1				2				3				4				5				6				7				8				
Perumusan masalah																																
Pengumpulan Data																																
Analisis																																
Desain																																
Implementasi																																
Pengujian dan Evaluasi																																
Kesimpulan dan Saran																																

DAFTAR PUSTAKA

[1] S. D, "Natural Language Proccesing," *Binus Universitu,* 2013.

[2] R. R, R. Rainer and R. Potter, "Introduction to Information Technology, Second Edition," *New York: John Wiley & Sons,* 2003.

[3] A. P. Widyassar, . S. Rustad, . G. F. Shidik, E. Noersasongko, . A. Syukur , A. Affandy and D. R. I. M. Setiadi, "Review of automatic text summarization techniques & methods," *Journal of King Saud University –Computer and Information Sciences,* p. 8, 2020.

[4] P. P. A. M. A. F. Alvandi Fadhil Sabily, "Analisis Sentimen Pemilihan Presiden 2019 pada Twitter menggunakan Metode Maximum Entropy," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 3, no. 5, p. 4205, 2019.

[5] P. P. A. S. A. Albert Bill Alroy, "Klasifikasi Hoaks Menggunakan Metode Maximum Entropy Dengan Seleksi Fitur Information Gain," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 3, no. 9, p. 9292, 2019.

[6] E. W. Ira Zulfa, "Sentimen Analisis Tweet Berbahasa Indonesia dengan Deep Belief Network," *IJCCS,* vol. 11, no. 2, pp. 187-198, 2017.

[7] Z.-H. Y. X. C. K. C. ,. X. L. Yu-An Huang, "Sequence-based prediction of protein-protein interactions using weighted sparse representation model combined with global encoding," *BMC Bioinformatics,* 2016.

[8] B. A. S. A. A. B. Z. A. Arliyanti Nurdin, "Perbandingan Kinerja Word Embedding Word2vec,Glove, dan FastText pada Klasifikasi Teks," *Jurnal TEKNOKOMPAK,* vol. 14, no. 2, pp. 74-79, 2020.

[9] S. Z, S. Q, Z. X, S. H, X. B and Y. , "Pattern Recognit," vol. 4, pp. 1623-1637, 2015.

[10] D. S. Pulkit Mehndiratta, "Identification of Sarcasm in Textual Data: A Comparative Study," *Journal of Data and Information Science,* vol. 4, no. 4, pp. 56-83, 2019.

[11] R. Suman and J. SIngh, "Sentimen Analysis of Tweets Using Support Vector Machine," *International Journal of Computer Science and Mobile Applications,* vol. 5, no. 10, pp. 83-91, October 2017.

[12] A. M. B. A. G. B. P. A. M. Nethravathi B., "Study of Techniques Used in Sentiment Analysis of Social Media Data," *MAT Journals,* vol. 5, no. 3, pp. 21-28, 2019.

[13] T. W. H. &. S. P. Perkasa, "RANCANG BANGUN PENDETEKSI GERAK MENGGUNAKAN," *Journal of Control and Network Systems,* vol. 3, no. 2, p. 92, 2014.

[14] Luthfi E. T, Suryono S and Utami E, "Analisis Sentiment Pada Twitter Dengan Menggunakan Metode Naïve Bayes Classifier," *Seminar Nasional Geotik,* pp. 9-15, 2018.