

**IMPLEMENTASI *WORD2VEC* PADA ANALISIS SENTIMEN TERHADAP
ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE
*SUPPORT VECTOR MACHINE***

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



**DISUSUN OLEH :
DENNY IVAN RIFAI
NIM 3260200018**

**FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2024

LEMBAR PENGESAHAN PEMBIMBING

Proposal Tugas Akhir dengan judul “IMPLEMENTASI *WORD2VEC* PADA ANALISIS SENTIMEN TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE*” ini disusun oleh:

Nama : Denny Ivan Rifai

NIM : 32602000018

Program Studi : Teknik Informatika

Telah disetujui oleh Dosen Pembimbing pada :


Hari : Senin

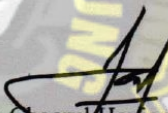
Tanggal : 4 Desember 2023

Mengesahkan,

Pembimbing I

Pembimbing II


Badie'ah, S.T., M. Kom
NIDN. 0619018701


Sam Farisa Chaerul Hayiana, S.T., M. Kom
NIDN. 0628028602


Mengetahui,

Ketua Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Islam Sultan Agung



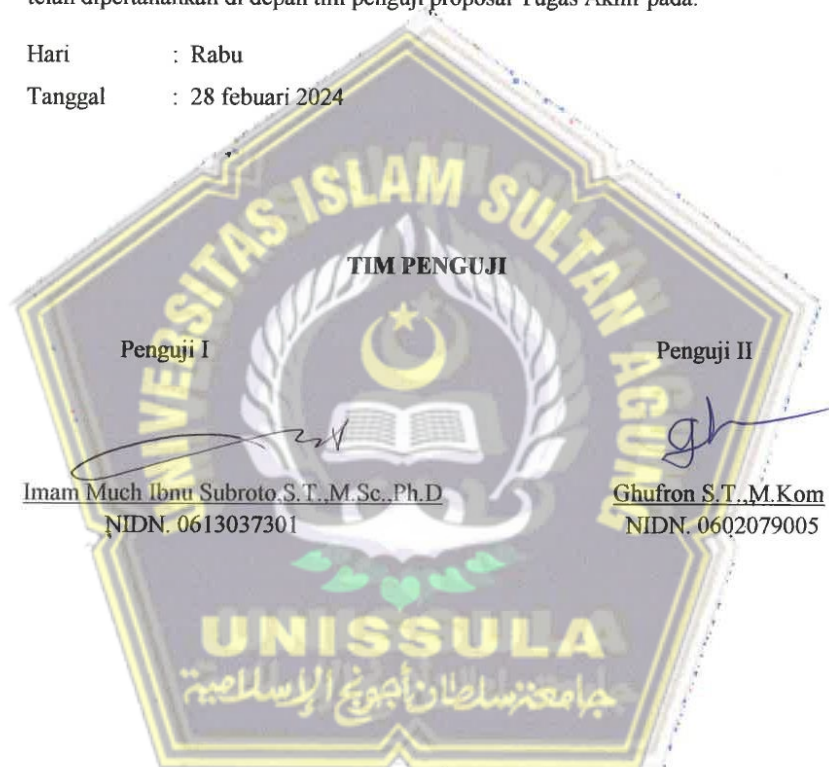

I. Sri Mulyono, M.Eng
NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

Proposal tugas akhir dengan judul “IMPLEMENTASI *WORD2VEC* PADA ANALISIS SENTIMEN TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE*” ini telah dipertahankan di depan tim penguji proposal Tugas Akhir pada:

Hari : Rabu

Tanggal : 28 febuari 2024



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Denny Ivan Rifai

NIM : 32602000018

Judul Tugas Akhir : Implementasi *Word2vec* Pada Analisis Sentimen Terhadap Ulasan Pengguna Aplikasi Tiktok Menggunakan Metode *Support Vector Machine*

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 7 Februari 2024

Yang Menyatakan,



10000
METERAI
TEMPEL
DCC17ALX076683303

Denny Ivan Rifai

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Implementasi *Word2vec* Pada Analisis Sentimen Terhadap Ulasan Pengguna Aplikasi Tiktok Menggunakan Metode *Support Vector Machine*” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, S.T., M.T.
3. Dosen pembimbing I penulis Badie'ah, S.T., M.Kom yang telah meluangkan waktu dan memberi ilmu.
4. Dosen pembimbing II penulis Sam Farisa Chaerul Haviana, S.T., M.Kom yang memberikan banyak nasehat dan saran.
5. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini,
6. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan rendah hati, penulis menyadari bahwa laporan masih memiliki banyak kekurangan dalam hal kuantitas, kualitas, dan ilmu pengetahuan. Oleh karena itu, penulis mengharapkan kritikan dan saran yang membangun untuk membantu laporan ini menjadi lebih baik di masa depan.

Semarang, 25 Januari 2024



Denny Ivan Rifai

DAFTAR ISI

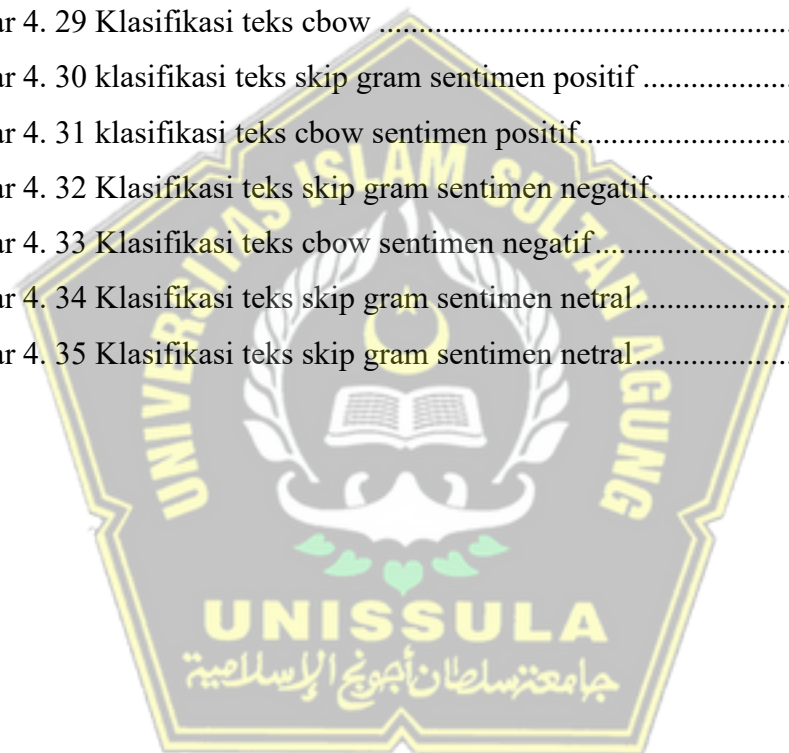
HALAMAN JUDUL.....	
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	viii
DAFTAR TABEL.....	ix
ABSTRAK.....	x
BAB I.....	
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
BAB II.....	5
TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	8
2.2.1 Text Mining.....	8
2.2.2 Analisis Sentimen.....	8
2.2.3 <i>Support Vector Machine</i>	9
2.2.4 Model <i>Word2Vec</i>	10
BAB III.....	15
METODOLOGI PENELITIAN.....	15
3.1 Metode Penelitian.....	15
3.2.1 Pemodelan Sistem.....	15

3.2.2	Analisis sistem	18
3.2	Perancangan <i>User Interface</i>	19
3.2.1	Halaman Menampilkan Dataset	19
3.2.2	Halaman Jumlah Dataset.....	20
3.2.3	Halaman Menampilkan Akurasi.....	20
3.2.4	Halaman Klasifikasi Skip Gram dan <i>Cbow</i>	21
3.3	Analisis Kebutuhan	23
BAB IV		26
HASIL DAN ANALISIS PENELITIAN		26
4.1	Hasil Penelitian.....	26
4.1.1	Persiapan Data.....	26
4.1.2	<i>Data Preprocessing</i>	27
4.1.3	Membuat Model <i>Word2Vec</i>	31
4.1.4	Membagi Data Menjadi Data Latih dan Data Uji	33
4.1.5	Metode <i>Support Vector Machine</i>	34
4.1.6	Simpan Model.....	35
4.1.7	Klasifikasi Teks.....	35
4.2	Hasil Perancangan	37
4.3	Hasil Uji Coba Sistem	42
BAB V.....		44
KESIMPULAN DAN SARAN.....		44
5.1	Kesimpulan.....	44
5.2	Saran.....	44
DAFTAR PUSTAKA.....		45
LAMPIRAN.....		49

DAFTAR GAMBAR

Gambar 2. 1 Model arsitektur <i>continous bags of words</i> (Cbow)	12
Gambar 2. 2 Model arsitektur skip gram	13
Gambar 3. 1 <i>Flowchart</i> pemodelan sistem.....	14
Gambar 3. 2 <i>Flowchart</i> Klasifikasi Teks	18
Gambar 3. 3 Halaman Utama.....	19
Gambar 3. 4 Menampilkan Jumlah Dataset	20
Gambar 3. 5 Halaman Menampilkan Akurasi.....	21
Gambar 3. 6 Klasifikasi Teks Skip Gram.....	22
Gambar 3. 7 Klasifikasi Teks Cbow.....	22
Gambar 4. 1 Mengambil Dataset Melalui <i>Web Scarping</i>	26
Gambar 4. 2 Hasil dari <i>web Scraping</i>	27
Gambar 4. 3 Menampilkan Dataset.....	27
Gambar 4. 4 Tahap <i>Cleaning</i> data.....	28
Gambar 4. 5 Tahap <i>Case Folding</i>	28
Gambar 4. 6 Tahap <i>Tokenizing</i>	29
Gambar 4. 7 Tahap <i>formalisasi</i> teks.....	29
Gambar 4. 8 Tahap <i>Stopwords</i>	29
Gambar 4. 9 Tahap <i>Stemming</i>	30
Gambar 4. 10 Ubah Sentimen Menjadi Label.....	30
Gambar 4. 11 model skip gram	31
Gambar 4. 12 model cbow (<i>Continuous Bag Of Words</i>).....	31
Gambar 4. 13 Kolom baru Skip gram	32
Gambar 4. 14 Kolom baru cbow (<i>Continous Bag Of Words</i>)	32
Gambar 4. 15 model skip gram.....	33
Gambar 4. 16 model cbow (<i>continous bag of words</i>)	33
Gambar 4. 17 Pembagian dataset skip gram	33
Gambar 4. 18 pembagian dataset cbow (<i>continous bag of words</i>)	33
Gambar 4. 19 klasifikasi model skip gram menggunakan svm	34
Gambar 4. 20 klasifikasi model cbow menggunakan svm.....	34

Gambar 4. 21 Simpan model klasifikasi skip gram	35
Gambar 4. 22 Simpan model klasifikasi Cbow.....	35
Gambar 4. 23 klasifikasi teks model skip gram metode svm.....	35
Gambar 4. 24 klasifikasi teks model cbow metode svm.....	36
Gambar 4. 25 Halaman Utama.....	37
Gambar 4. 26 Menampilkan Jumlah dataset	38
Gambar 4. 27 Menampilkan hasil akurasi.....	38
Gambar 4. 28 Klasifikasi teks skip gram	39
Gambar 4. 29 Klasifikasi teks cbow	39
Gambar 4. 30 klasifikasi teks skip gram sentimen positif	40
Gambar 4. 31 klasifikasi teks cbow sentimen positif.....	40
Gambar 4. 32 Klasifikasi teks skip gram sentimen negatif.....	40
Gambar 4. 33 Klasifikasi teks cbow sentimen negatif.....	41
Gambar 4. 34 Klasifikasi teks skip gram sentimen netral.....	41
Gambar 4. 35 Klasifikasi teks skip gram sentimen netral.....	41



DAFTAR TABEL

Tabel 4. 1 Hasil uji klasifikasi teks Skip Gram.....	42
Tabel 4. 2 Hasil uji klasifikasi teks Cbow (<i>Continouos Bag Of Words</i>)	42



ABSTRAK

Penggunaan media sosial semakin berkembang pesat, dengan aplikasi tiktok menjadi salah satu platform yang sangat populer. Analisis sentimen merupakan opini seseorang terhadap suatu topik, produk atau layanan tertentu. Analisis sentimen digunakan untuk menganalisis opini atau sentimen terhadap suatu topik tertentu, apakah opini tersebut cenderung positif, negatif dan netral. Peneliti menggunakan model *word2vec* yaitu *cbow* (*continuous bag of words*) dan *skip gram* untuk membandingkan akurasi serta membuat klasifikasi teks dengan jumlah dataset 900 terbagi menjadi 300 negatif, 300 positif dan 300 netral serta menerapkan metode *Support Vector Machine* (SVM) untuk mengklasifikasikan sentimen dari ulasan pengguna aplikasi tiktok. Metode penelitian ini melibatkan langkah-langkah utama, yaitu pengumpulan dataset ulasan pengguna TikTok, *preprocessing* data, dan implementasi model *Word2Vec*. Setelah representasi vektor kata-kata berhasil dihasilkan, SVM diterapkan untuk melatih dan menguji model analisis sentimen. Hasil akurasi dari model *cbow* (*continuous bag of words*) 0.66 sedangkan hasil model *skip gram* 0.68. Hasil akurasi cukup rendah namun untuk dapat meningkatkan hasil akurasi perlu meningkatkan jumlah dataset, fokus pada tahap *preprocessing*, yaitu memperbaiki kata yang tidak baku serta pengecekan kata yang lebih detail.

Kata kunci: analisis sentimen, *word2vec*, klasifikasi, *support vector machine*

ABSTRACT

The use of social media is growing rapidly, with the TikTok application becoming one of the most popular platforms. Sentiment analysis is a person's opinion on a particular topic, product or service. Sentiment analysis is used to analyze opinions or sentiments on a particular topic, whether these opinions tend to be positive, negative or neutral. Researchers use the *word2vec* model, namely *cbow* (*continuous bag of words*) and *skip gram* to compare accuracy and create text classification with a dataset of 900 divided into 300 negative, 300 positive and 300 neutral and apply the *Support Vector Machine* (SVM) method to classify sentiment from reviews TikTok application users. This research method involves main steps, namely collecting a dataset of TikTok user reviews, data preprocessing, and implementing the *Word2Vec* model. After the vector representation of words is successfully generated, SVM is applied to train and test the sentiment analysis model. The accuracy results from the *cbow* (*continuous bag of words*) model were 0.66 while the results from the *skip gram* model were 0.68. The accuracy results are quite low, but to be able to increase the accuracy results, it is necessary to increase the number of datasets, focus on the preprocessing stage, namely correcting non-standard words and checking words in more detail.

Keywords: sentiment analysis, *word2vec*, classification, support vector machine

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan Teknologi sekarang sudah sangat berkembang, banyak media yang dapat digunakan untuk mendapatkan informasi atau komunikasi, media tersebut sangat mudah digunakan dengan tersambung ke jaringan internet. Perkembangan teknologi sangat pesat sehingga mempermudah manusia untuk melakukan komunikasi dengan salah satu teknologi *smartphone*, perangkat *smartphone* ini sangat memudahkan masyarakat untuk mengakses media kapan pun dan dimana saja. Banyak media yang dapat digunakan oleh masyarakat untuk melakukan suara, filter, menambahkan *effect*, dan stiker. Aplikasi TikTok ini mendorong pengguna untuk berkreatifitas dalam membuat video pendek berdurasi kurang lebih 2 menit. Aplikasi ini juga sekarang menjadi aplikasi yang banyak diminati oleh masyarakat, bahkan selama masa pandemi Pengguna Aplikasi TikTok naik 20% dibandingkan sebelumnya. (Alfiah Zulqornain dkk., 2021)

Dari viral nya Aplikasi TikTok pengguna mempunyai beberapa pandangan yang mana bisa dilihat dari ulasan Aplikasi tiktok diplaystore, terdapat penilaian negatif, netral dan positif terhadap pengguna aplikasi TikTok.(Rahman dkk., 2023)

Analisis sentimen merupakan opini seseorang terhadap suatu topik, produk atau layanan tertentu. Analisis sentimen digunakan untuk menganalisis opini atau sentimen terhadap suatu topik tertentu, apakah opini tersebut cenderung positif, negatif dan netral. Google Play Store menyediakan tempat dimana pengguna dapat menggunakan aplikasi langsung dari pengembang dan juga proses ulasan pengguna aplikasi dapat langsung dilakukan melalui kolom komentar yang telah disediakan. Penggunaan Aplikasi Tiktok memberi tanggapan dan opini melalui kolom komentar pada google play store.(Suryati dkk., 2023)

Implementasi *Word2Vec*, sebuah pemodelan bahasa alami yang mampu mengubah kata-kata menjadi vector. Dengan memahami konteks dan makna kata-kata dalam ulasan pengguna, *word2vec* dapat membantu mengatasi tantangan analisis sentimen yang kompleks. Implementasi *Word2Vec* menggunakan dua representasi konteks yang berbeda yaitu CBOW (*Continuous Bag Of Words*) dan Skip-Gram. (Sahbuddin dkk., 2022)

Penggunaan metode *Support vector machine* (SVM) dalam klasifikasi sentimen memberikan pendekatan yang teruji dan efektif. SVM memiliki kemampuan untuk mengklasifikasi ulasan pengguna ke dalam kategori positif, negatif dan netral dengan tingkat akurasi tinggi. (Septiawan dkk., 2023)

Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan *word2vec* pada analisis sentimen terhadap ulasan pengguna aplikasi tiktok dengan metode *support vector machine*. Data yang akan diambil dari ulasan pengguna Aplikasi TikTok yang ada di playstore. *Word2vec* mempunyai dua Model Cbow (*Continuous Bag Of Words*) dan Skip-gram. Penelitian ini akan membandingkan kinerja model mana yang lebih baik antara Cbow (*Continuous Bag Of Words*) dan Skip-Gram dengan menggunakan metode *Support Vector Machine*, SVM Metode pembelajaran mesin (*supervised learning*) yang digunakan untuk mengklasifikasikan komentar negatif, netral dan positif pada ulasan pengguna Aplikasi tiktok. (Normah dkk., 2022)

1.2 Perumusan Masalah

Bagaimana menerapkan Metode *Support Vector Machine* untuk mengimplentasikan *Word2Vec* pada model skip-gram dan *cbow* (*Continuous Bag of Words*) untuk membandingkan kinerja model yang mana lebih baik dengan menggunakan Data ulasan pengguna Aplikasi TikTok.

1.3 Batasan Masalah

- 1 Data yang diambil dari Ulasan Pengguna Aplikasi TikTok di PlayStore.
- 2 Data yang akan diambil di Ulasan PlayStore sebanyak 900 Data dari tahun 2018 sampai tahun 2023 (positif, netral dan negatif).
- 3 Penelitian ini akan mengkategorikan sentimen ulasan menjadi tiga kategori positif, netral dan negatif.
- 4 Penelitian ini akan memfokuskan *Word2Vec* dengan model Skip-gram dan Cbow (*Continuous bag of words*) untuk membandingkan kinerja model yang lebih baik dalam data ulasan Aplikasi Tiktok menggunakan metode *Support Vector Machine*.

1.4 Tujuan

Tujuan dari penelitian ini adalah mengimplementasikan *Word2Vec* pada analisis sentimen ulasan pengguna Aplikasi TikTok dengan membandingkan algoritma Skip-Gram dan CBOW (*Continuous bag of words*) mengenai kinerja model mana lebih baik dengan menggunakan metode *Support Vector Machine*.

1.5 Manfaat

1. Peneliti memperoleh pengalaman dalam menerapkan teknik-teknik pemrosesan bahasa alami (Natural Language Processing/NLP) dan pembelajaran mesin (Machine Learning) dalam konteks analisis sentimen.
2. Penelitian ini tidak hanya bermanfaat untuk memahami sentimen pengguna TikTok, tetapi juga dapat diterapkan pada aplikasi dan platform lainnya.

1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan oleh penulis dalam sebuah pembuatan laporan tugas akhir adalah sebagai berikut:

BAB 1 PENDAHULUAN :Pada bab ini penulis mengutarakan latar belakang pemilihan judul, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, serta sistematika penulisan.

BAB 2 TINJAUAN PUSTAKAN DAN DASAR TEORI :Bab ini memuat penelitian-penelitian sebelumnya dan dasar teori yang berguna untuk membantu sebuah penulis untuk memahami bagaimana teori yang berhubungan dengan *Word2vec* dan *Support Vector Machine* (SVM).

BAB 3 METODE PENELITIAN :Bab ini memuat proses analisis proses pembuatan sistem klasifikasi teks yang mencakup perancangan sistem dan desain antar muka

BAB 4 HASIL DAN ANALISIS :Pada bab ini penulis mengungkapkan hasil penelitian yaitu hasil Implementasi *word2vec* pada klasifikasi analisis sentimen terhadap ulasan aplikasi tiktok diplaystore menggunakan metode *support vector machine*.

BAB 5 KESIMPULAN DAN SARAN :Bab ini penulis memaparkan kesimpulan proses penelitian dari awal hingga akhir

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Permasalahan yang dihadapi oleh para penikmat film disebabkan kan informasi atau opini untuk menjadikan referensi film yang akan ditonton maka penelitian analisis sentimen sangat mempermudah pengguna. Analisis sentimen adalah jenis *pemrosesan* bahasa alami untuk melacak *mood* publik tentang produk atau topik tertentu. untuk menghasilkan kalimat *review* diperlukan algoritma yang akan memproses dan memetakan informasi menjadi sebuah *vektor* dengan menggunakan metode *word2vec*. Membutuhkan metode SVM (*Support Vector Machine*) untuk mengklasifikasikan paragraf atau dokumen. Hasil dari penelitian ini metode ekstraksi fitur *word2vec* yang digabungkan dengan klasifikasi SVM dengan jumlah data sebanyak 10.000 ulasan. Dapat menyimpulkan tahap *preprocessing*, dimensi *word2vec*, dan model klasifikasi mempengaruhi hasil performansi pada analisis sentimen level aspek pada ulasan film.

Hasil disetiap skenario terbaik yaitu menggunakan tahap *preprocessing lemmatization*, *word2vec* menggunakan 300 dimensi, dan model klasifikasi SVM linier dengan hasil terbaik sebesar 78,74% f1-score. Penggunaan jumlah dimensi *word2vec* yang ada pada data train mempengaruhi akurasi dari klasifikasi. (Jihad dkk., 2021)

Pada penelitian ini, LSTM dipilih untuk digunakan sebagai metode klasifikasi sentimen. Dataset yang digunakan adalah dataset movie *review* yang terdiri dari 25.000 dokumen *review*, dengan panjang rata-rata per *review* adalah 233 kata. Penelitian ini menggunakan metode CBOW dan Skip-Gram pada *word2vec* untuk membentuk *vektor representasi* dari setiap kata (*word vector*) dicorpus. Beberapa ukuran dimensi dari *word vector* yaitu dimensi 50, 60, 100, 200, dan 500 di gunakan, untuk mengetahui pengaruh nya terhadap akurasi yang dihasilkan. Akurasi terbaik

diperoleh pada ukuran dimensi *word vector* 100 sebesar 88.17% dan akurasi terendah 85.86% pada ukuran dimensi vector 500.(Widayat, 2021)

Pada era society 5.0 teknologi informasi semakin berkembang pesat, salah satunya pada bidang transportasi. Fenomena jasa transportasi online menjadi sangat populer dikalangan masyarakat. Dengan adanya fenomena tersebut, banyak masyarakat yang beropini terhadap jasa transportasi online baik itu komentar positif atau komentar negatif. Tujuan dari penelitian ini untuk melakukan analisis sentimen terhadap komentar atau pengguna aplikasi jasa transportasi online pada Gojek dan Grab di google playstore. Tahapan proses penelitian ini adalah pengumpulan data, pelabelan data, preprocessing data, ekstrasi fitur serta klasifikasi sentimen menggunakan algoritma *Support Vector Machine* (SVM). Pengumpulan data dilakukan dengan *web scaping*. Pelabelan dataset dibagi menjadi dua kelas, yaitu kelas positif dan kelas negatif. *Word2Vec Text Embedding* digunakan sebagai fitur ekstrasi fitur model sebagai representasi kata kedalam bentuk vektor. Arsitektur yang digunakan *word2vec* ialah skip-gram. Algoritma *Support Vector Machine* (SVM) digunakan untuk proses klasifikasi data untuk mengetahui tingkat akurasi terhadap sentimen yang digunakan. Hasil dari pengujian yang dilakukan terhadap klasifikasi analisis sentimen pada aplikasi transportasi online menunjukkan hasil performa yang cukup baik untuk aplikasi Gojek mendapatkan tingkat akurasi sebesar 87%, presisi sebesar 93%, dan re-call sebesar 84%. Sedangkan aplikasi Grab mendapatkan tingkatan akurasi sebesar 82%, presisi sebesar 89%, dan re-call sebesar 83%.(Suryati dkk., 2023)

Gojek Adalah salah satu aplikasi transportasi *online* yang memiliki pengguna bisa dikatakan banyak di Indonesia. Namun dalam system ini pasti memiliki banyak kekurangan yang dirasakan penggunanya. Dengan menganalisis kekurangan dari aplikasi perusahaan dapat mengetahui kekurangan dari aplikasi dan bagaimana cara memperbaikinya. Tujuan dari penelitian ini yaitu untuk melakukan analisis sentiment dengan menggunakan data ulasan yang terdapat pada Google Play guna

mengetahui perbandingan keakurasian antara metode *Support Vector Machine* untuk mengklasifikasikan ulasan dari dua kategori yaitu ulasan positif dan negatif. Kemudian dibandingkan dengan metode *Decision Tree*. Hasil dari klasifikasi diperoleh hasil akurasi sebesar 90.20% untuk metode *support vector machine* sedangkan 89.80% untuk metode *Decision tree*. Jadi disimpulkan bahwa metode *Support vector machine* memiliki akurasi nilai yang lebih tinggi dibandingkan metode *Decision tree*.(Rokhman dkk., 2021)

Pinjaman online merupakan jasa keuangan yang menyediakan layanan berbasis online. Seiringan dengan perkembangan pinjaman online tersebut, banyak pinjaman *online illegal* bermunculan dan sering melakukan pelanggaran, seperti membocorkan informasi pribadi nasabah dan melakukan penyalahgunaan data dengan menjalankan aksi ekstrem seperti melakukan teror kepada nasabah yang melakukan transaksi pinjaman online. Hal ini tentunya mendapat banyak komentar dari kalangan masyarakat khususnya di media sosial twitter. Penelitian ini bertujuan untuk melakukan Analisa sentiment guna melihat fenomena apa yang sedang terjadi di kalangan masyarakat mengenai pinjaman online. Data yang digunakan merupakan tweet atau retweet dari media sosial twitter dengan #pinjamanonline #pinjol. hasil penelitian dan pengujian metode Support Vector Machine untuk mengklasifikasi pinjaman online dengan tweet berbahasa Indonesia menggunakan ekstraksi fitur TF-IDF didapat kesimpulan bahwa penggunaan metode *Support Vector Machine* tingkat nilai *Accuracy* sebesar 86.6%, dengan *Precision* sebesar Positif sebesar 86 %, Netral 1.00% dan Negatif sebesar 87%, Recall Positif sebesar 90%, Netral 87% dan Negatif sebesar 26% dan *F1-Score* Positif sebesar 88% Netral 42% dan Negatif sebesar 86%.(Lestari, 2022)

2.2 Dasar Teori

2.2.1 Text Mining

Text Mining ialah bagian dari data mining, yaitu proses untuk mendapatkan pengetahuan menggunakan alat analisis dimana pengguna berintraksi dengan sekumpulan dokumen dari waktu ke waktu. Teks mining merupakan teknik yang digunakan untuk menangani *classification*, *information extraction* dan *information retrieval*. Konsep teks mining ini dilakukan untuk klasifikasi yaitu dokumen tekstual dengan tujuan untuk mengklasifikasi dokumen yang sesuai topik pembahasan. Perbedaan antara data mining dan teks mining terletak pada *preprocessing*, pada data mining *preprocessing* berfokus pada penomoran (*indexing*) dan normalisasi data, sedangkan *text mining* berfokus pada identifikasi dan ekstraksi fitur. (Isnain dkk., 2021)

2.2.2 Analisis Sentimen

Analisis sentimen adalah proses untuk menentukan sentimen, sikap atau opini seseorang terhadap suatu topik atau produk berdasarkan teks atau data yang ada. Tujuan dari analisis sentimen untuk memahami perasaan yang terkandung dalam suatu teks, apakah itu positif, negatif dan netral. Analisis sentimen dilakukan untuk mendeteksi opini terhadap suatu subjek dan objek (misalnya individu, organisasi ataupun produk) dalam sebuah kumpulan data. Besarnya pengaruh dan manfaat dari analisis sentiment menyebabkan penelitian ataupun aplikasi mengenai analisis sentimen berkembang pesat. (Hasri dkk., 2022) Proses ini dilakukan untuk memperoleh sebuah informasi sentimen yang tersirat dari suatu opini. Dilakukan analisis sentimen untuk mengetahui kecenderungan penilaian terhadap permasalahan dari sudut pandang orang lain, Umumnya apakah mereka mempunyai opini negatif, netral dan positif terhadap suatu masalah. Dalam beberapa penelitian, sentiment dapat dikategorikan sebagai big data dimana ukuran dari data teks tersebut semakin banyak dan makna konteksnya semakin beragam. (Sidik dkk., 2022)

2.2.3 Support Vector Machine

Support Vector Machine (SVM) adalah salah satu metode klasifikasi yang banyak dikembangkan saat ini. Konsep dasar metode ini adalah memaksimalkan batas *hyperplane* yang memisahkan suatu set data. *Support Vector Machine* (SVM) merupakan salah satu metode klasifikasi dengan menggunakan *machine learning* (*supervised learning*) yang memprediksi kelas berdasarkan model atau pola dari hasil proses *training*. Klasifikasi dilakukan dengan mencari *hyperplane* atau garis pembatas (*decision boundary*) yang memisahkan antara suatu kelas dengan kelas lain. *Hyperplane* yang dicari adalah yang memberikan jarak paling jauh dari setiap titik data.

Dalam klasifikasi (*Support vector machine*) SVM dapat digunakan untuk klasifikasi *linearly separable* dan *non-linearly separable*.

1. SVM Linear

Mencari *hyperplane* yang memisahkan dua kelas dengan cara mencari garis yang paling baik memisahkan antara dua kelas. Rumus dasar SVM sebagai berikut:

$$f(x) = w \cdot x + b \quad \dots \dots \dots (1)$$

Dimana x adalah input (vektor fitur), w adalah vektor bobot, dan b adalah bias. Untuk membuat klasifikasi, maka ditambahkan dengan fungsi kondisional seperti berikut:

$$y = \{1 \text{ if } f(x) \geq 0, -1 \text{ Otherwise}\} \dots \dots \dots (2)$$

2. SVM Non linear

Jika data tidak *linearly separable*, maka akan digunakan *kernel trick* untuk mengubah data ke dalam dimensi yang lebih tinggi sehingga dapat disepara dengan *linear*. Rumus dasar dari SVM *non linear* sebagai berikut:

$$f(x) = \sum(a_i y_i K(x_i, x) + b) \dots \dots \dots (1)$$

Dimana x adalah input (vektur fitur), a adalah vektor bobot, y adalah label kelas, k adalah *kernel* yang digunakan (misalnya radial basis *function*, *polynomial*, atau *sigmoid*), b adalah bias.

SVM mencari *hyperplane* yang memberikan jarak paling jauh dari setiap titik data. Untuk menemukan nilai a_i dan b yang optimal digunakan metode *Quadratic Programming (QP)* yang mencari solusi yang meminimalkan fungsi *objective* dengan memperhatikan kendala yang ditentukan. (Septiawan dkk., 2023)

Pada dasarnya SVM digunakan untuk klasifikasi dua kelas untuk mencari *hyperplane* dengan *margin* optimal. Dalam hal lebih dua kelas maka diperlukan pendekatan yang berbeda dari dua kelas lainnya. apabila pada penelitian ini menggunakan tiga kelas yaitu positif, negatif dan netral. Ada beberapa metode *Support vector machine multi class* yaitu *One Against One (OAO)* dan *One Against All (OAA)*. Prinsip dasar dari *One Against One* adalah membangun $(k-1)/2$ model svm biner (k adalah jumlah kelas), tempat setiap model klasifikasi dilatih berdasarkan data dua kelas. Sedangkan konsep *OAA* ialah bahwa k model SVM biner dibangun (k adalah jumlah kelas), tempat setiap model klasifikasi ke- i dilatih menggunakan semua data untuk menemukan solusi masalah. (Nurkholis dkk., 2022)

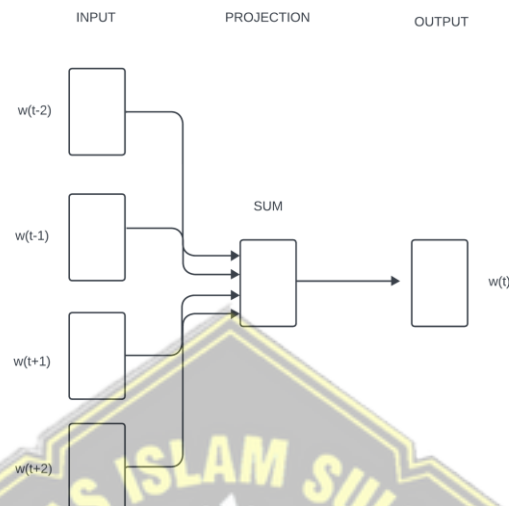
2.2.4 Model *Word2Vec*

Word2Vec adalah sebuah metode atau algoritma dalam bidang pemrosesan bahasa alami (*nature language processing*) yang digunakan untuk menciptakan representasi vektor kata (*word embeddings*) dari teks. Algoritma ini dikembangkan oleh Tomas Mikolov dan timnya pada tahun 2013. Tujuan dari *word2vec* adalah untuk menghubungkan setiap kata dalam teks ke dalam ruang vektor. Dimana kata-kata dengan makna yang mirip atau sering kali muncul bersama-sama akan memiliki representasi vektor yang berdekatan satu sama lain. Representasi vektor ini memungkinkan komputer untuk mengenali dan memahami hubungan

semantik antara kata-kata. Dengan memanfaatkan *word2vec*, kita dapat menghasilkan vektor yang merepresentasikan kata-kata dalam bentuk numerik. Representasi vektor ini bermanfaat dalam berbagai tugas pemrosesan bahasa alami, seperti analisis sentimen, klasifikasi teks, pencarian informasi, dan penerjemahan mesin (Pradana dkk., 2023)

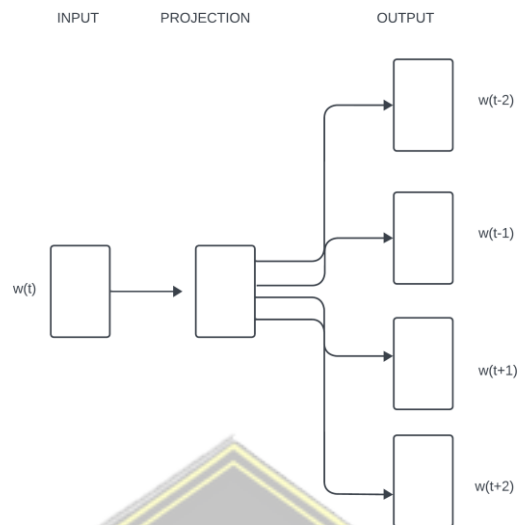


Dibawah ini ialah penjelasan model *word2vec* yaitu Cbow (*continuous bags of words*) dan Skip-Gram.



Gambar 2. 1 Model arsitektur *continuous bags of words* (Cbow)

Continuous Bag of Words (CBOW) Pada arsitektur CBOW, model berusaha memprediksi kata target berdasarkan konteks kata di sekitarnya. model arsitektur yang didasarkan pada *neural network* di mana *non-linear hidden layer* dihilangkan dan *projection layer* dibagikan ke semua kata. Pada model *Continuous Bag-of-Words*, representasi terdistribusi dari konteks (kata-kata di sekitarnya) digabungkan untuk memprediksi kata di tengah.



Gambar 2. 2 arsitektur model *skip gram*

Skip-Gram adalah model berusaha memprediksi konteks kata-kata disekitar kata target yang diberikan. model lain yang sangat mirip dengan CBOW adalah model Skip-gram. Namun, dalam memprediksi kata saat ini berdasarkan konteksnya, ia mencoba memaksimalkan klasifikasi kata berdasarkan kata lain dalam kalimat yang sama. Skip-gram memiliki performa yang lebih baik dari CBOW pada sebagian besar evaluasi, tapi CBOW lebih cepat untuk dilatih.

Model Skip-Gram memaksimalkan rata-rata *log probability* yang dapat dilihat pada persamaan berikut. (Kurniawan, 2020)

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t) \dots\dots\dots (1)$$

w_t = kata *center*

w_{t+j} = kata setelah kata *center*

c = *ukuran training context*

Nilai vektor dokumen atau dalam penelitian ini merupakan nilai vektor cuitan (ulasan). Nilai vektor dokumen didapatkan dengan menghitung nilai rata-rata vektor dari semua kata di dalam satu cuitan

(ulasan). Berikut persamaan untuk menghitung nilai vektor dokumen. (Kurniawan, 2020)

$$vec_{c,j} = \frac{1}{\Pi_j} \sum_{i=1}^m \omega_i v(w_i) \dots \dots \dots (2)$$

$vec_{c,j}$ = vector dokumen j pada class c

Π_j = jumlah fitur dalam j

$V(w_i)$ = vektor kata dari fitur w

ω_i = nilai vektor dari fitur w

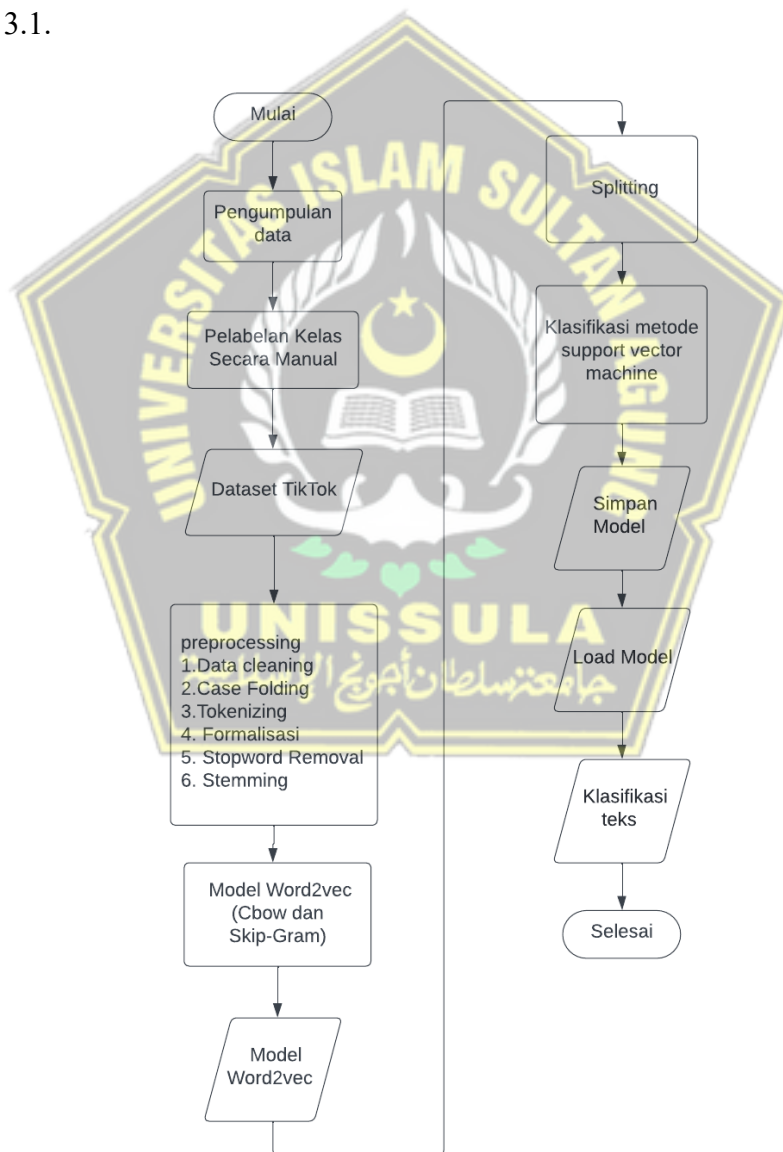


BAB III METODOLOGI PENELITIAN

3.1 Metode Penelitian

3.2.1 Pemodelan Sistem

Penelitian ini akan dibangun model analisis sentimen dengan menggunakan model *Word2Vec* (Cbow dan Skip Gram) dan klasifikasi menggunakan Support Vector Machine (SVM). Alur sistem pada gambar 3.1.



Gambar 3. 1 *Flowchart* Pemodelan Sistem

Berdasarkan dari gambar 3.1 maka dijelaskan tahapan-tahapan sebagai berikut:

1. Pengumpulan data

Pengumpulan data menggunakan web scraping di google colab dengan menggunakan bahasa python. Tujuan dari *web scraping* ialah untuk mendapatkan data kemudian melakukan ekstraksi informasi yang dimiliki data tersebut.

2. Pelabelan kelas sentimen secara manual

Pelabelan kelas sentimen data ulasan pengguna dikelompokkan menjadi tiga kelas yaitu positif negatif dan netral. Pelabelan dilakukan secara manual untuk memudahkan dalam mengolah data dengan membaca maksud dari kalimat yang ada dari ulasan tersebut. Berdasarkan pada kelas negatif bermakna mengenai ejekan, keluhan dan ketidakpuasan mengenai aplikasi tersebut, sedangkan kelas netral teks yang bermakna tidak menyatakan perasaan atau penilaian emosional positif dan negatif dan yang terakhir pengelompokan kelas positif bermakna pujian, saran dan mendukung dan menyatakan setuju.

3. *Preprocessing* data

Tahapan selanjutnya perlu melakukan pembersihan data dengan tujuan supaya data digunakan pada tahap selanjutnya. Adapun tahapan yang dilakukan penelitian ini sebagai berikut.

a. *Data Cleaning*

Data yang sudah diberikan label akan melalui tahap pembersihan seperti pembersihan tanda baca, simbol, *emoticon*, link, karakter, angka, dan lainnya

b. *Case folding* dilakukan perubahan seluruh huruf menjadi kecil (*lowercase*) yang ada pada dokumen.

c. *Tokenizing* ialah proses yang dilakukan untuk memisahkan kalimat menjadi kumpulan kata(token).

d. Tahap formalisasi dilakukan untuk pengubah penggunaan kata tidak baku menjadi baku sesuai dengan KBBI

- e. *Stop words Remove* ialah menghapus kata-kata umum yang mungkin tidak memberikan informasi sentimen yang berarti, seperti dan, atau, dari
- f. Proses stemming dilakukan perubahan kata yang berimbuhan menjadi kata dasar

4. Data *Split*

Setelah dataset melakukan *preprocessing*, dataset akan dipisah menjadi dua kategori yaitu data latih dan data uji sebanyak 900 data ulasan aplikasi tiktok dengan label positif, negatif dan netral. Data uji sebanyak 180 sedangkan data latih 720.

5. *Word2Vec* (Skip-Gram dan Cbow)

Pada tahap ini data akan dilatih, data ulasan aplikasi Tiktok menggunakan model *word2vec*. Pelatihan *word2vec* memerlukan data dengan jumlah kata yang lengkap, sehingga pada penelitian ini menggunakan data ulasan aplikasi Tiktok. *word2vec* yaitu Skip-Gram dan Cbow (*Continuous Bag of Words*). Model *word2vec* dari data latih ini akan melalui tahap selanjutnya. Model ini akan mengonversi kata-kata dalam ulasan menjadi representasi vektor numerik.

6. Klasifikasi *Support vector machine* (SVM)

Metode SVM adalah metode *supervised learning*. Setelah dataset dibersihkan ditahap *preprocessing*, selanjutnya setiap kata pada dataset menjadi nilai vektor pada model *word2vec* yang sudah dilatih. SVM akan mencoba membuat garis *hyperplane* untuk memisahkan kalimat yang memiliki sentimen positif negatif dan netral. *Margin* dari garis *hyperplane* akan dimaksimalkan dengan mengidentifikasi *support vector*. SVM ini akan memprediksi hasil akurasi. Hasil klasifikasi model, seperti yang dihasilkan dari pelatihan dengan algoritma SVM dengan model *word2vec*, menghasilkan model yang dapat digunakan untuk memprediksi sentimen pada data pengujian.

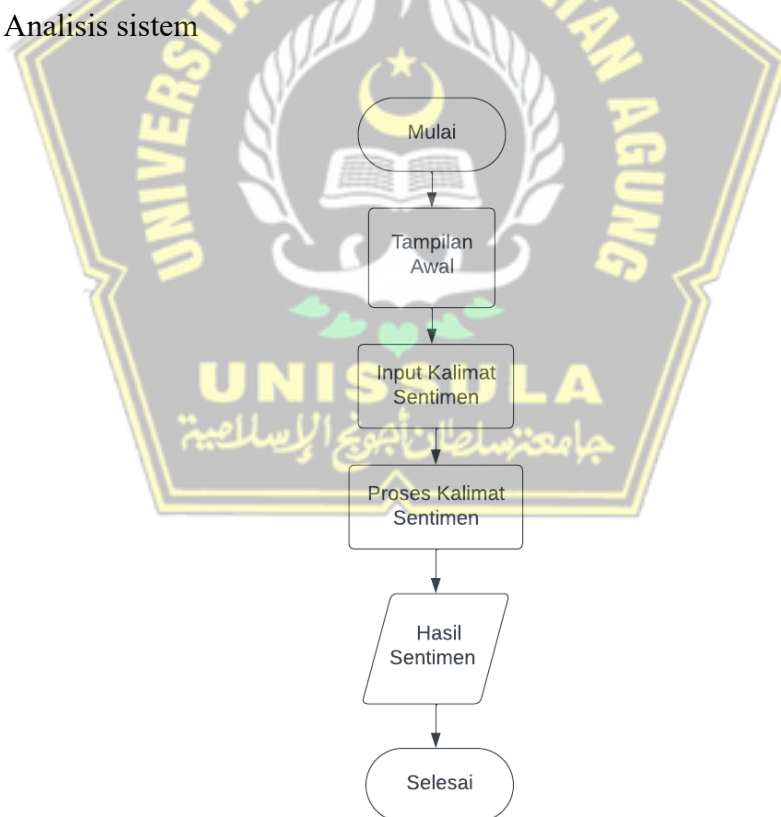
7. Simpan Model

Tahapan ini ketika sudah *preprocessing*, *split* data, model *word2vec* dan klasifikasi *svm* maka akan mengetahui akurasi *cbow* dan *skip gram* disimpan di *joblib*. *Joblib* digunakan untuk menyimpan dan memuat objek Python ke dan dari file. Ini sering digunakan dalam proses serialisasi dan deserialisasi objek.

8. Klasifikasi teks

Klasifikasi teks proses mengkategorikan teks ke dalam kelas atau label berdasarkan pada isinya. Tujuan dari klasifikasi teks adalah untuk memahami dan mengidentifikasi sentimen, topik atau jenis informasi yang terkandung dalam teks.

3.2.2 Analisis sistem



Gambar 3. 2 *Flowchart* klasifikasi teks

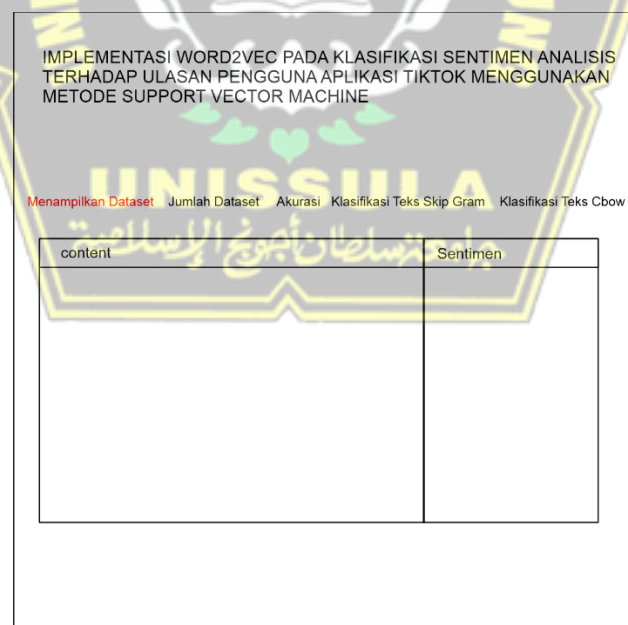
Pada gambar 3.2 adalah flowchart klasifikasi teks yang menggambarkan proses sistem klasifikasi teks. Ada beberapa tahapan pada gambar 3.2 yaitu:

1. Tampilan awal ini menampilkan dataset, jumlah dataset, akurasi, dan klasifikasi teks cbow dan skip gram.
2. Tahapan input kalimat sentimen, memasukan kalimat sentimen berupa positif, negatif dan netral.
3. Selanjutnya tahap proses kalimat sentimen, tahapan ini ialah proses untuk mencocokkan apakah kalimat yang di inputkan kategori positif, negatif dan netral, sesuai proses data yang dilatih.
4. Setelah diinputkan maka akan keluar hasil sentimennya. apakah positif, negatif atau netral

3.2 Perancangan *User Interface*

3.2.1 Halaman Menampilkan Dataset

Halaman menampilkan dataset bertujuan untuk menampilkan data dan sentimen yang dikategorikan menjadi tiga positif negatif dan netral.

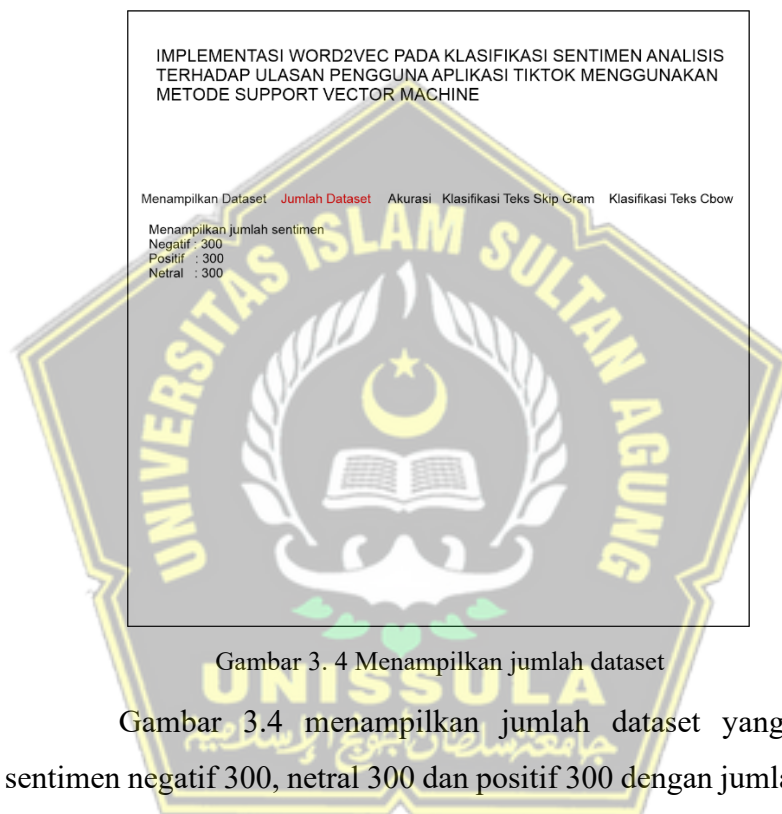


Gambar 3. 3 Halaman Utama

Gambar 3.3 menunjukkan halaman awal dimana pengguna melihat judul serta apa saja ditampilkan awal seperti jumlah dataset, akurasi, klasifikasi teks Cbow dan Skip Gram.

3.2.2 Halaman Jumlah Dataset

Ketika sudah melihat halaman utama atau menampilkan data, ketika ditekan jumlah dataset maka akan menampilkan gambar 3.4.

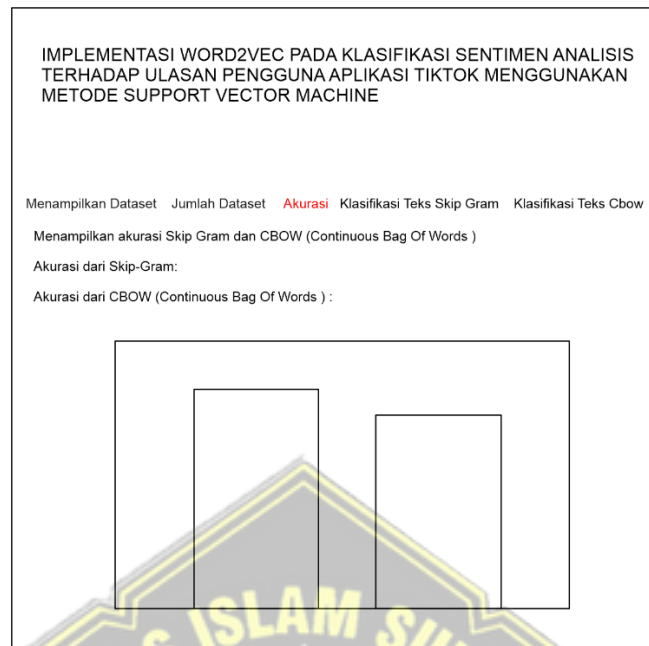


Gambar 3. 4 Menampilkan jumlah dataset

Gambar 3.4 menampilkan jumlah dataset yang mana berisi sentimen negatif 300, netral 300 dan positif 300 dengan jumlah 900 dataset.

3.2.3 Halaman Menampilkan Akurasi

Selanjutnya ketika masuk ke menu lainnya seperti akurasi akan menampilkan jumlah akurasi antara Cbow dan Skip gram.



Gambar 3. 5 Halaman menampilkan akurasi

Gambar 3.5 menampilkan hasil akurasi antara cbow (*Continuous Bag of Words*) dan Skip Gram.

3.2.4 Halaman Klasifikasi Skip Gram dan Cbow (*Continuous Bag of Words*)

Halaman selanjutnya yaitu klasifikasi teks dimana user bisa memasukan kalimat mengenai sentimen analisis yang dikategorikan menjadi tiga yaitu positif negatif dan netral seperti gambar 3.6 dan gambar 3.7.

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Menampilkan Dataset Jumlah Dataset Akurasi **Klasifikasi Teks Skip Gram** Klasifikasi Teks Cbow

Klasifikasi teks untuk Skip Gram

Masukan teks untuk analisis sentimen:

Gambar 3. 6 Klasifikasi teks Skip Gram

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Menampilkan Dataset Jumlah Dataset Akurasi Klasifikasi Teks Skip Gram **Klasifikasi Teks Cbow**

Klasifikasi teks untuk Cbow

Masukan teks untuk analisis sentimen:

Gambar 3. 7 Klasifikasi teks Cbow

Dari gambar diatas Gambar 3.6 dan Gambar 3.7 mengenai klasifikasi teks user bisa memasukan kalimat sentimen seperti kalimat negatif, positif dan netral maka *output* dari klasifikasi teks sesuai dengan kalimat yang dimasukan.

3.3 Analisis Kebutuhan

Pada tahap pengembangan, peneliti menganalisis semua perangkat lunak yang diperlukan untuk mengembangkan aplikasi. Berikut adalah perangkat lunak yang digunakan dalam pengembangan sistem adalah:

1. Python 3.12.1

Bahasa Pemrograman Python digunakan untuk berbagai macam keperluan pengembangan software (perangkat lunak) dan dapat bahasa ini juga bisa dijalankan di berbagai macam platform sistem operasi seperti Windows, Linux, Apple, dan sebagainya. python versi 3.12.1 terbaru biasanya terdapat perbaikan bug dan peningkatan kinerja dari versi sebelumnya. Mungkin ada penambahan fitur baru, perbaikan keamanan, atau peningkatan dalam standar library Python. Perbaikan dan peningkatan di versi Python seperti ini dapat membantu pengembang dalam menulis kode yang lebih efisien, aman, dan dapat diandalkan. (Musa ddk, 2023)

2. *Library Pandas* 1.5.3

Pandas adalah salah satu library python yang menyediakan struktur data dan alat analisis data yang kuat, memungkinkan pengguna untuk memuat, memanipulasi dan menganalisis data dengan mudah seperti tabel atau spreadsheet. Beberapa fitur utama dari pandas DataFrame, Pembacaan dan penulisan data dan fungsi analisis data.

3. *Scikit-Learn* 1.2.2

Scikit – Learn adalah modul python yang mengintegrasikan berbagai algoritma pembelajaran mesin *state-of-the-art* untuk masalah yang diawasi dan tidak diawasi skala menengah. Penekanan diberikan pada kemudahan penggunaan, kinerja, dokumentasi, dan konsistensi API. Library ini menyediakan berbagai algoritma untuk tugas-tugas seperti klasifikasi, pengelompokan, pemrosesan data, pengurangan dimensi, dan masih banyak lagi (Riadi Silitonga dkk., 2019)

4. *Google Colaboratory*

Google Colaboratory yang dikenal luas sebagai *google colab* adalah layanan sumber terbuka yang disediakan oleh google kepada siapa pun yang memiliki akun Gmail. *Google colab* menyediakan CPU untuk penelitian kepada orang-orang yang tidak memiliki sumber daya yang cukup.(Kanani dkk., 2019)

5. *Visual Studio Code*

Visual Studio Code adalah alat pemrograman yang kuat, mudah dikonfigurasi, dan sangat fungsional, dirancang untuk pemula dan berpengalaman pengembang sama. Menggunakan *Visual Studio Code* akan memungkinkan membuat situs web dinamis, serta aplikasi web yang bersih, menggunakan bahasa yang didukung, seperti *C#, C++, Java, PHP, Python, SQL, Visual Basic* dan *XML*.

6. *Library Numpy 1.23.5*

Library numpy yang digunakan untuk bekerja dengan data. Perpustakaan *numpy* memungkinkan untuk menangani daftar nilai yang besar, di sini daftar disebut array. Ini memberikan kekuatan cara untuk melakukan aritmatika array. Dapat menggunakan array untuk menangani kumpulan input yang besar, dan untuk menangani matriks dan aljabar linier, salah satu cabang matematika yang telah menjadi penting dalam semua aspek ilmu komputer.(Nelli dkk, 2023)

7. *Library Gensim 4.3.2*

Library Gensim adalah perpustakaan pemodelan topik yang digunakan terutama untuk analisis kesamaan dokumen. Ini menyediakan berbagai algoritma untuk proses teks ringkasan, termasuk *TextRank* dan *Latent Analisis Semantik (LSA)* (Dubey dkk., 2023)

8. *Streamlit* 1.30.0

Streamlit adalah sebuah framework *open-source* yang digunakan untuk membuat aplikasi web interaktif dengan cepat dan mudah menggunakan Python. Dengan *Streamlit*, dapat membuat aplikasi web yang berinteraksi dengan data, visualisasi, model pembelajaran mesin, dan banyak lagi.(Kamaludin dkk., 2023)



BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil Penelitian

4.1.1 Persiapan Data

Data yang peneliti ambil dari ulasan aplikasi tiktok menggunakan *Web scraping* teknik ini digunakan untuk mengambil informasi dari berbagai situ web. File didownload dalam bentuk *excel*, selanjutnya di labelin secara manual dengan membaca kalimat atau komentar satu persatu, data yang diambil 900 dikategorikan positif 300, negatif 300 dan netral 300.

```
[ ] result, continuation_token = reviews(
    'com.ss.android.ugc.trill',
    lang='id',
    country='id',
    sort=Sort.MOST_RELEVANT,
    count=10000,
    filter_score_with=None
)
df_busu = pd.DataFrame(np.array(result), columns=['review'])
df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))

[ ] new_df = df_busu[['userName', 'score', 'at', 'content']]
sorted_df = new_df.sort_values(by='at', ascending=False)
sorted_df.to_excel("Ulasan_My_XL_10000_Data.xlsx", index=False)
```

Gambar 4. 1 Mengambil Dataset Melalui *Web Scarping*

Dataset yang diambil menggunakan kode python untuk mengambil ulasan (reviews) dari aplikasi dengan paket ID “com.ss.android.ugc.trill”. `reviews ()` disini digunakan untuk mengambil ulasan dari aplikasi dengan paket ID tertentu. Tahap selanjutnya membuat dataFrame pandas hasil ulasan yang diperoleh dari fungsi ‘`reviews ()`’. Data yang akan diambil *UserName, Score, At, Content*. Disimpan dalam bentuk *excel* dengan jumlah dataset 10000.

```
[ ] sorted_df
```

	userName	score	at	content
2287	Nur Alsah	2	2024-02-06 14:27:06	Kadang gabisa post komen, sekiranya bisa kepot...
2307	oniy	1	2024-02-06 13:55:30	Jelek banget, kenapa fitur foto tiba-tiba hila...
3017	alie abduallah	4	2024-02-06 13:37:51	Aplikasi nya bagus cuma saran aku hilangin big...
1676	Ayu Wandira	2	2024-02-06 13:33:40	Tiktok knpa sih suka kembali sendiri. Padahal ...
3696	Jois Mongan	2	2024-02-06 13:28:50	lagu yang di jadi kan favorit hilang semua dar...
...
7926	Abdillah Mahbubi	1	2023-08-13 22:34:14	halo permisi, mau sedikit komplain nih abis up...
2762	Alifa Kayla	3	2023-08-13 07:09:42	Maaf klo saya kasih bintang 3 soalnya habis di...
7043	Muhammad Zaki	5	2023-08-13 06:55:01	aplikasi Tik tok sangat bagus apalagi di tik t...
9843	Nyonya Sakit	2	2023-08-12 06:45:11	Ini kenapa tiktok sering bgt ngeframe? Lagi no...
9939	salamun yara	2	2023-08-12 03:59:28	Versi sebelumnya masih ada sedikit bug pas buk...

10000 rows x 4 columns

Gambar 4. 2 Hasil dari *web Scraping*

Hasil dari *web scraping* yang diambil dari kolom *UserName*, *Score*, *At*, *Content* dengan data yang berjumlah 10000 sesuai dengan kode yang diatas.

4.1.2 *Data Preprocessing*

Tahap selanjutnya *preprocessing*, sebelum data melalui tahap *preprocessing* dataset diambil 900 data dari tahun 2018 sampai 2023 selanjutnya diberikan label sentimen yang mengandung kalimat negatif, positif atau netral. Tahapan *preprocessing* sebagai berikut.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 900 entries, 0 to 899
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   content     900 non-null    object
1   Sentimen    900 non-null    object
dtypes: object(2)
memory usage: 14.2+ KB
```

Gambar 4. 3 Menampilkan Dataset

Gambar 4.3 menampilkan jumlah dataset dengan kolom yang berisi kolom *content* dan *Sentimen* dengan *dtype object*.


```

def cleaningcontent(content):
    content = re.sub(r'@[A-Za-z0-9]+', ' ', content)
    content = re.sub(r'#[A-Za-z0-9]+', ' ', content)
    content = re.sub(r'http\S+', ' ', content)
    content = re.sub(r'[0-9]+', ' ', content)
    content = re.sub(r"[-()\#/@;:<>{}'+=|.!?,_]", " ", content)
    content = content.strip(' ')
    return content

def removeEmoji(text):
    emoji_pattern = re.compile("[
        u"\U0001F600-\U0001F64F"
        u"\U0001F300-\U0001F5FF"
        u"\U0001F680-\U0001F6FF"
        u"\U0001F700-\U0001F77F"
        u"\U0001F780-\U0001F7FF"
        u"\U0001F800-\U0001F8FF"
        u"\U0001F900-\U0001F9FF"
        u"\U0001FA00-\U0001FA6F"
        u"\U0001FA70-\U0001FAFF"
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
    ]+", flags=re.UNICODE)

```

Gambar 4. 4 Tahap *Cleaning* data

Gambar 4.4 mengenai tahapan *cleaning* data yaitu yang pertama fungsi *cleaningcontent* mengganti *username*, *hashtag*, *http*, angka dan karakter khusus menjadi spasi dan spasi yang berlebihan dikurangi selanjutnya tahapan *removeEmoji* menghapus emoji seperti emot, simbol, map simbol, alat-alat teknis, kategori tambahan, emoji warna kulit, alat game kartu, simbol catur dan dingbat.

```

def casefoldingText(ulasan):
    content = ulasan.lower()
    return content

df['cleaning'] = df['content'].apply(cleaningcontent)
df['RemoveEmoji'] = df['cleaning'].apply(removeEmoji)
df['CaseFolding'] = df['RemoveEmoji'].apply(casefoldingText)

```

Gambar 4. 5 Tahap *Case Folding*

Gambar 4.5 menerapkan *case folding* mengubah huruf yang kapital menjadi huruf kecil sehingga dataset menjadi huruf kecil agar menjadi sama. Tahap selanjutnya kolom *content* di *cleaningcontent*, data disimpan dikolom baru yaitu 'Cleaning', selanjutnya kolom 'Cleaning' dibersihkan atau *removeEmoji* di simpan dikolom yang baru yaitu kolom 'RemoveEmoji', setelah melalui tahap pembersihan selanjutnya huruf kapital dijadi kecil semua disimpan di kolom baru yaitu 'Case Folding'.

```

nltk.download('punkt')
def tokenizingText(content):
    content = word_tokenize(content)
    return content
df['Tokenizing'] = df['CaseFolding'].apply(tokenizingText)

```

Gambar 4. 6 Tahap *Tokenizing*

Tahap selanjutnya *tokenizing* yaitu tahap untuk mengubah kalimat-kalimat tersebut menjadi kata-kata atau token-token. kolom *casefolding* di *tokenizing* disimpan dikolom yang baru bernama *tokenizing*.

```

def convertToSlangword(content):
    kamusSlang = eval(open("/content/drive/MyDrive/combined_slang_words.txt").read())
    pattern = re.compile(r'\b(' + '|'.join(kamusSlang.keys()+r')\b')
    contentt = []
    for kata in content:
        filterSlang = pattern.sub(lambda x: kamusSlang[x.group()],kata)
        contentt.append(filterSlang.lower())
    content = contentt
    return content
df['Formalisasi'] = df['Tokenizing'].apply(convertToSlangword)

```

Gambar 4. 7 Tahap formalisasi teks

Tahap selanjutnya formalisasi teks digunakan untuk mengubah kata-kata slang atau tidak formal menjadi kata-kata yang formal atau baku sesuai KBBI. Fungsi pertama membaca kamus *slang* dari *file* teks yang disimpan *'/content/drive/MyDrive/combined_slang_words.txt'* file ini berisi pasangan kata slang dan kata formal yang sesuai. Hasil formalisasi disimpan dalam kolom baru *Formalisasi* dalam *DataFrame*.

```

nltk.download('stopwords')
from nltk.corpus import stopwords

daftar_stopword = stopwords.words('indonesian')

daftar_stopword.extend(["yg","dg","rt","tik","po","c"])
daftar_stopword = set(daftar_stopword)

def stopwordText(words):
    return [word for word in words if word not in daftar_stopword]

df['Stopword Removal'] = df['Formalisasi'].apply(stopwordText)

```

Gambar 4. 8 Tahap *Stopwords*

Gambar 4.8 menunjukkan *stopwords* menghapus kata-kata yang tidak penting dari hasil formalisasi teks. Pertama-tama kode mendownload sumber daya *stopwords* dari NLTK (*Natural Language Toolkit*)

menggunakan perintah `nlTK.download(stopwords)`. Kode diatas menambahkan *stopwords* tambahan seperti *yg,dg,rt,tik,tok*. Fungsi dari `stopwordsText()` kemudian diterapkan dikolom ‘formalisasi’ dan hasilnya disimpan dalam kolom baru ‘Stopword Removal’.

```
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}
for document in df['Stopword Removal']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ''

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":",term_dict[term])

def stemmingText(document):
    return [term_dict[term] for term in document]

df['Stemming'] = df['Stopword Removal'].swifter.apply(stemmingText)
```

Gambar 4. 9 Tahap *Stemming*

Proses *stemming* dilakukan perubahan kata yang berimbuhan menjadi kata dasar, kolom ‘Stopwords Removal’ melalui tahap *Stemming*. Pertama mendefinisikan fungsi *stemmed_wrapper* fungsi ini digunakan untuk menerapkan stemming pada sebuah *term* atau kata. Kode membuat sebuah kamus (‘*term_dict*’) yang akan digunakan untuk memetakan setiap kata ke kata dasarnya. Pada kolom ‘Stopword Removal’ kode akan memeriksa apakah kata tersebut ada didalam ‘*term_dict*’ jika belum ada, kata tersebut akan dimasukan ke kamus dan dilakukan stemming. Hasil ‘Stopword Removal’ disimpan dalam kolom baru ‘*stemming*’ dalam *DataFrame*.

```
label_mapping = {'Positif': 1, 'Netral': 0, 'Negatif': -1}
df['label'] = df['Sentimen'].map(label_mapping)
```

Gambar 4. 10 Ubah Sentimen Menjadi Label

Tahap ini mengubah kolom ‘Sentimen’ menjadi ‘label’ yang mana isinya positif di beri nilai 1, negatif bernilai -1 dan netral bernilai 0.

4.1.3 Membuat Model *Word2Vec*

Tahap selanjutnya sesudah melakukan *preprocessing* diantaranya *cleaning* data, menghapus emoji, *case folding*, *tokenizing*, formalisasi teks, *stopwords* dan *stemming*, tahapan ini membuat model cbow dan skip gram.

```
word2vec_model_skipgram = Word2Vec(vector_size=100, window=5, min_count=4, sg=1, workers=30, epochs=100)
word2vec_model_skipgram.build_vocab(df['Stemming'], progress_per=10)
word2vec_model_skipgram.train(df['Stemming'],
total_examples=word2vec_model_skipgram.corpus_count, epochs=word2vec_model_skipgram.epochs)
```

Gambar 4. 11 model skip gram

```
word2vec_model_cbow = Word2Vec(vector_size=100, window=5, min_count=4, sg=0, workers=30, epochs=100)
word2vec_model_cbow.build_vocab(df['Stemming'], progress_per=10)
word2vec_model_cbow.train(df['Stemming'],
total_examples=word2vec_model_cbow.corpus_count, epochs=word2vec_model_cbow.epochs)
```

Gambar 4. 12 model cbow (*Continuous Bag Of Words*)

Pada gambar 4.11 dan gambar 4.12 melatih model skip gram dan Cbow (*continuous Bag Of Words*) menggunakan *Word2vec* dari *library Gensim*, baris pertama menginisialisasi model, `vector_size=100` yaitu jumlah dimensi vektor fitur yang dihasilkan, dalam hal ini vektor fitur memiliki dimensi 100, `Window=5` ukuran jendela yang digunakan dalam model skip gram, `min_count=4` ialah jumlah minimum kemunculan kata yang diperlukan untuk mempertimbangkan kata tersebut dalam pembentukan vektor. Kata-kata yang muncul kurang dari 4 kali akan diabaikan. `sg=1` menentukan model, nilai 1 mengindikasikan pengguna skip gram dan `sg=0` mengindikasikan cbow, `wordkcers=30` jumlah thread yang digunakan untuk melatih model, `epochs=100` menunjukkan jumlah *iterasi* yang digunakan saat melatih model. setelah model skip gram dibuat, `word2vec_model_skipgram.build_vocab(df['Stemming'], progress_per=10)` dan `word2vec_model_cbow.build_vocab(df['Stemming'], progress_per=10)` baris ini membangun kosakata model yang sudah *dipreprocessing*, `build_vocab` digunakan untuk membangun kosakata model dengan mengambil daftar kata-kata dari data, `progress_per10` menampilkan jumlah 10 kosakata, selanjutnya data dilatih dengan memperbarui bobot vektor kata berdasarkan konteks data dalam teks.

```
df_word2vec = pd.DataFrame(columns=['skipgram'])

for document in df['stemming']:
    document_vector = np.mean([word2vec_model_skipgram.wv[word] for word in document if word in word2vec_model_skipgram.wv], axis=0)
    df_word2vec = df_word2vec.append({'skipgram': document_vector}, ignore_index=True)

df = pd.concat([df, df_word2vec], axis=1)
```

Gambar 4. 13 Kolom baru Skip gram

```
df_word2vec = pd.DataFrame(columns=['cbow'])

for document in df['stemming']:
    document_vector = np.mean([word2vec_model_cbow.wv[word] for word in document if word in word2vec_model_cbow.wv], axis=0)
    df_word2vec = df_word2vec.append({'cbow': document_vector}, ignore_index=True)

df = pd.concat([df, df_word2vec], axis=1)
```

Gambar 4. 14 Kolom baru cbow (*Continuous Bag Of Words*)

Pada gambar 4.13 dan gambar 4.14 barisan pertama membuat kolom baru yang diberi nama “ skipgram” atau “cbow” digunakan untuk menyimpan vektor fitur, selanjutnya `document_vector` ini berisi menghitung vektor fitur untuk setiap dokumen dengan mengambil rata-rata vektor fitur dari setiap kata dalam dokumen, vektor fitur kata diambil dari dari model `word2vec`, `word2vec_model_skipgram.wv[word]` untuk skip gram sedangkan untuk cbow `word2vec_model_cbow.wv[word]` jika kata tersebut ada dalam kosakata maka (`if word in word2vec_model_skipgram.wv` atau `if word in word2vec_model_cbow.wv`) nilai vektor tersebut kemudian dirata-ratakan menggunakan fungsi `np.mean` dengan `axis=0`, selanjutnya baris `df_word2vec=df_word2vec.append({'skipgram':document_vector}, ignore_index=true)` dan `df_word2vec=df_word2vec.append({'cbow':document_vector}, ignore_index=true)` baris ini menambahkan vektor fitur yang telah dihitung ke dataframe sebagai baris baru. Fungsi dari `append` digunakan untuk menambah baris, `ignore_index=true` memastikan bahwa indeks baris dalam dataframe diabaikan sehingga indeks baru akan otomatis ditetapkan. Baris terakhir menggabungkan dataframe awal dengan dataframe baru proses ini dilakukan menggunakan `concat` dari `pandas`.


```
word2vec_model_skipgram.save("/content/drive/MyDrive/Colab Notebooks/word2vec_model_skipgram")
word2vec_model_skipgram.wv.most_similar('baik')
word2vec_model_skipgram.wv['baik']
```

Gambar 4. 15 model skip gram

```
word2vec_model_cbow.save("/content/drive/MyDrive/Colab Notebooks/word2vec_model_cbow")
word2vec_model_cbow.wv.most_similar('baik')
word2vec_model_cbow.wv['baik']
```

Gambar 4. 16 model cbow (*continuous bag of words*)

Pada gambar 4.15 dan gambar 4.16 baris pertama menyimpan model antara model skip gram dan cbow, baris kedua memanggil kata-kata yang paling mirip dengan kata baik berdasarkan model *word2vec*. Nantinya akan memberikan daftar kata-kata yang paling mirip dengan baik. baris ketiga memanggil vektor representasi kata 'baik' dalam ruang vektor yang dipelajari oleh model. vektor ini akan mengukur kedekatan atau kesamaan antara kata 'baik' dengan kat lainnya.

4.1.4 Membagi Data Menjadi Data Latih dan Data Uji

Tahap dataset melewati pemodelan Skip gram dan Cbow (*Continuous Bag Of Words*) tahap selanjutnya membagi dataset menjadi dua yaitu data latih dan data uji.

```
X = df['skipgram']
y = df['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 4. 17 Pembagian dataset skip gram

```
x = df['cbow']
Y = df['label']
x_train, x_test, Y_train, Y_test = train_test_split(x, Y, test_size=0.2, random_state=42)
```

Gambar 4. 18 pembagian dataset cbow (*continuous bag of words*)

Gambar 4.17 dan Gambar 4.18 Tahapan ini mengimport fungsi `from sklearn.model_selection import train_test_split` dari modul `model_selection` di library `scikit-learn`, `X=df['skipgram']`, `y=df['label']` pertama menginisialisakan untuk untuk skipgram dan untuk cbow

`X=df['cbow'],y=df['label'].test_size=0,2` dalam hal ini 20% data akan digunakan untuk data uji dan 80% digunakan untuk data latih, `random_state=42` digunakan untuk menentukan *seed* untuk *generator* secara acak, `X_train` ialah data latih yang berisi fitur sedangkan `x_test` berisi data uji yang berisi fitur, `y_train` label data latih yang sesuai dengan `X_train`, sedangkan `y_train` label data uji yang sesuai dengan `x_test`.

4.1.5 Metode *Support Vector Machine*

Setelah tahap pembagian dataset antara model cbow dan skip gram sudah selesai dengan jumlah data latih 720 dan data uji 180.tahap selanjutnya klasifikasi menggunakan metode *support vector machine* antara model skip gram dan cbow.

```

classifier_skipgram = SVC()
classifier_skipgram.fit(X_train.tolist(), y_train)
predictions = classifier_skipgram.predict(X_test.tolist())
accuracy = round(accuracy_score(y_test, predictions), 2)
report = classification_report(y_test, predictions, target_names=['negatif', 'positif', 'netral'])
print("Akurasi model SKIP GRAM Metode SVM:", accuracy)
print("Laporan Klasifikasi:")
print(report)

```

Gambar 4. 19 klasifikasi model skip gram menggunakan svm

```

classifier_cbow = SVC()
classifier_cbow.fit(x_train.tolist(), Y_train)
predictions = classifier_cbow.predict(x_test.tolist())
accuracy = round(accuracy_score(y_test, predictions), 2)
report = classification_report(y_test, predictions, target_names=['negatif', 'positif', 'netral'])
print("Akurasi model CBOW Metode SVM:", accuracy)
print("Laporan Klasifikasi:")
print(report)

```

Gambar 4. 20 klasifikasi model cbow menggunakan svm

Gambar 4.19 dan Gambar 4.20 Mengklasifikasi model skip gram dan cbow. pertama menginisialisasi model klasifikasi svm menggunakan kelas SVC, selanjutnya kode melatih model klasifikasi SVM menggunakan data latih `X_train.tolist` (format *series pandas*) dan label data latih `y_train` model akan belajar untuk mengklasifikasi data berdasarkan fitur yang diberikan dalam `X_train`,setelah model dilatih,model tersebut untuk melakukan prediksi pada data uji `x_test.tolist` hasil prediksi disimpan divariabel `predictions`,menghitung akurasi prediksi dengan membandingkan hasil prediksi (`predictions`) dengan label dari data uji

(y_{test}) akurasi dihitung sebagai jumlah prediksi, laporan klasifikasi mencakup presisi, recall dan f1-score.

4.1.6 Simpan Model

Setelah tahap klasifikasi menggunakan metode support vektor machine model cbow (*Continuous Bag Of Words*) dan skip gram. Tahap selanjutnya kedua model disimpan.

```
joblib.dump(classifier_skipgram, "/content/drive/MyDrive/Classroom/skipgram1.pkl")
```

Gambar 4. 21 Simpan model klasifikasi skip gram

```
joblib.dump(classifier_cbow, "/content/drive/MyDrive/Classroom/cbow1.pkl")
```

Gambar 4. 22 Simpan model klasifikasi Cbow

Pada gambar 4.21 dan gambar 4.22 hasil klasifikasi menggunakan *support vector machine* dengan model cbow dan skip gram disimpan terpisah, fungsi dari `joblib.dump` digunakan untuk menyimpan objek `classifier_cbow` dan `classifier_skipgram` ke dalam file model klasifikasi yang telah dilatih dapat disimpan dan digunakan kembali tanpa perlu melatih ulang.

4.1.7 Klasifikasi Teks

Sesudah model disimpan menggunakan *joblib*, tahap selanjutnya menggunakan model tersebut untuk klasifikasi teks dengan memprediksi sentimen pada teks baru menggunakan model klasifikasi yang telah dilatih.

```
classifier_skipgram = joblib.load("/content/drive/mydrive/classroom/skipgram1.pkl")
new_text = "aplikasinya sangat buruk mempengaruhi aplikasi yang lain menjadi lambat"
tokenized_new_text = word_tokenize(new_text.lower())
new_text_vector = np.mean([word2vec_model_skipgram.wv[word] for word in tokenized_new_text if word in word2vec_model_skipgram.wv], axis=0)
new_text_prediction = classifier_skipgram.predict([new_text_vector])[0]
if new_text_prediction == 1:
    print("Sentimen: Positif")
elif new_text_prediction == 0:
    print("Sentimen: Netral")
else:
    print("Sentimen: Negatif")
print(new_text_prediction)
```

Gambar 4. 23 klasifikasi teks model skip gram metode *svm*

```

classifier_cbow = joblib.load("/content/drive/MyDrive/Classroom/cbow1.pkl")
new_text = "aplikasinya sangat buruk mempengaruhi aplikasi yang lain menjadi lambat"
tokenized_new_text = word_tokenize(new_text.lower())
new_text_vector = np.mean([word2vec_model_cbow.wv[word] for word in tokenized_new_text if word in word2vec_model_cbow.wv], axis=0)
new_text_prediction = classifier_cbow.predict([new_text_vector])[0]
if new_text_prediction == 1:
    print("Sentimen: Positif")
elif new_text_prediction == 0:
    print("Sentimen: Netral")
else:
    print("Sentimen: Negatif")
print(new_text_prediction)

```

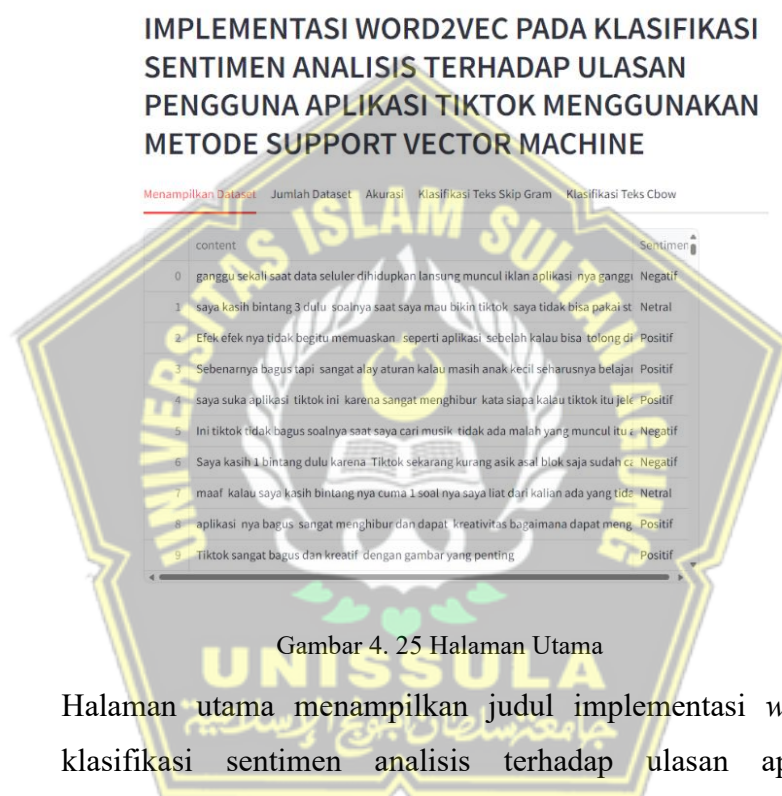
Gambar 4. 24 klasifikasi teks model cbow metode svm

Pada gambar 4.23 kode melakukan prediksi sentimen teks baru menggunakan model klasifikasi yang telah disimpan `word2vec_skipgram.pkl`. sedangkan gambar 4.24 telah disimpan di `word2vec_cbow.pkl`. pertama model klasifikasi svm yang telah dilatih. Melakukan *preprocessing* teks baru, `new_text` diinisialisasi, teks tersebut kemudian diubah menjadi huruf kecil dan *tokenizing*, `new_text_vector`, vektor rata-rata dari vektor kata untuk setiap kata dalam teks baru dihitung menggunakan model (skip gram atau cbow) yang sudah dilatih. `new_text_prediction`, memprediksi sentimen yang sudah dilatih untuk melakukan prediksi sentimen. hasil prediksi sentimen disimpan di `new_text_prediction`, selanjutnya hasil sentimen merepresentasikan kelas sentimen (1 untuk positif, 0 untuk netral dan negatif untuk -1).

4.2 Hasil Perancangan

Setelah membuat model *word2vec* diklasifikasi menggunakan metode *support vector machine*. Tahap selanjutnya mengimplementasi model melalui *platform website*, rancangan sudah dibuat di BAB 3 yang mana akan dikembangkan pada platform website. Hasil dari Perancangan sebagai berikut.

1. Menampilkan Halaman Utama



Gambar 4. 25 Halaman Utama

Halaman utama menampilkan judul implementasi *word2vec* pada klasifikasi sentimen analisis terhadap ulasan aplikasi tiktok menggunakan metode *support vector machine* serta menampilkan Dataset yang berisi kolom content dan sentimen, ada empat fitur lainnya seperti jumlah dataset, akurasi, klasifikasi teks skip gram dan klasifikasi teks cbow.

2. Menampilkan Jumlah Dataset

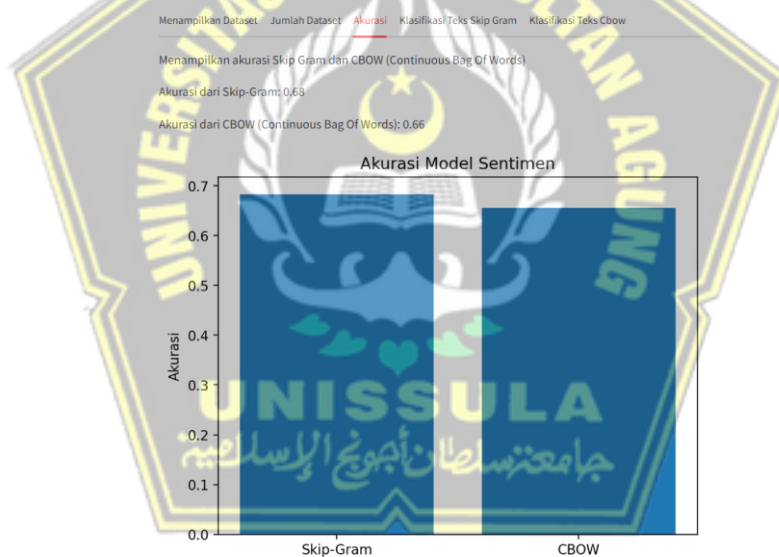
IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE



Gambar 4. 26 Menampilkan Jumlah dataset

Selanjutnya halaman untuk menampilkan jumlah dataset yang berjumlah 900 data dengan kategori sentimen negatif 300, sentimen positif 300 dan sentimen netral 300.

3. Menampilkan Akurasi



Gambar 4. 27 Menampilkan hasil akurasi

Halaman ini menampilkan akurasi Cbow (*continuous Bag Of Words*) dan skip gram yang mana hasil akurasi dari Cbow 0.68 dan skip gram 0.66, serta menampilkan diagram akurasi cbow dan skip gram.

4. Menampilkan Klasifikasi Teks Skip Gram dan Cbow (*Continuous Bag Of Words*)

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Menampilkan Dataset Jumlah Dataset Akurasi Klasifikasi Teks Skip Gram Klasifikasi Teks Cbow

Klasifikasi teks untuk Skip Gram

Masukkan teks untuk analisis sentimen:

Gambar 4. 28 Klasifikasi teks skip gram

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Menampilkan Dataset Jumlah Dataset Akurasi Klasifikasi Teks Cbow Klasifikasi Teks Skip Gram

Klasifikasi teks untuk Cbow

Masukkan teks untuk analisis sentimen:

Gambar 4. 29 Klasifikasi teks cbow

Klasifikasi teks skip gram dan cbow ini *user* tinggal memasukan kalimat yang mempunyai sentimen negatif, positif atau netral nantinya sistem akan mendeteksi kalimat tersebut apakah positif, netral atau negatif.

5. Halaman klasifikasi teks skip gram dan cbow (*Continuous Bag Of Words*) memasukan kalimat positif.

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Menampilkan Dataset Jumlah Dataset Akurasi Klasifikasi Teks Skip Gram Klasifikasi Teks Cbow

Klasifikasi teks untuk Skip Gram

Masukkan teks untuk analisis sentimen:

Aplikasi ini sangat bermanfaat, kreatif dan mendapatkan banyak informasi penting untuk kita

Sentimen: Positif

Gambar 4. 30 klasifikasi teks skip gram sentimen positif

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Menampilkan Dataset Jumlah Dataset Akurasi Klasifikasi Teks Skip Gram Klasifikasi Teks Cbow

Klasifikasi teks untuk Cbow

Masukkan teks untuk analisis sentimen:

Aplikasi ini sangat bermanfaat, kreatif dan mendapatkan banyak informasi penting untuk kita

Sentimen: Positif

Gambar 4. 31 klasifikasi teks cbow sentimen positif

Pada gambar 4.26 dan gambar 4.27 *user* memasukan kalimat ‘aplikasi ini sangat bermanfaat, kreatif dan mendapatkan banyak informasi penting untuk kita’ dari kalimat tersebut yang telah dikirim, sistem akan memproses dengan memberikan hasil *output* yaitu positif.

6. Halaman klasifikasi teks skip gram dan cbow (*Continuous Bag Of Words*) memasukan kalimat negatif.

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Menampilkan Dataset Jumlah Dataset Akurasi Klasifikasi Teks Skip Gram Klasifikasi Teks Cbow

Klasifikasi teks untuk Skip Gram

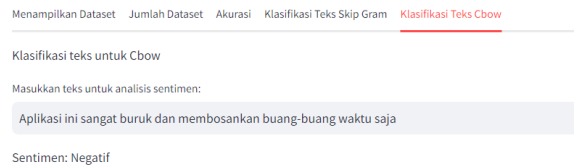
Masukkan teks untuk analisis sentimen:

Aplikasi ini sangat buruk dan membosankan buang-buang waktu saja

Sentimen: Negatif

Gambar 4. 32 Klasifikasi teks skip gram sentimen negatif

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

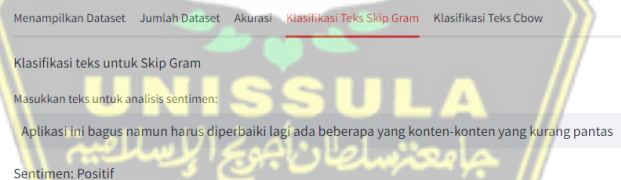


Gambar 4. 33 Klasifikasi teks cbow sentimen negatif

Pada gambar 4.28 dan gambar 4.29 *user* memasukan kalimat ‘Aplikasi ini sangat buruk dan membosankan buang-buang waktu saja’ dari kalimat tersebut yang telah dikirim, sistem akan memproses dengan memberikan hasil *output* yaitu negatif.

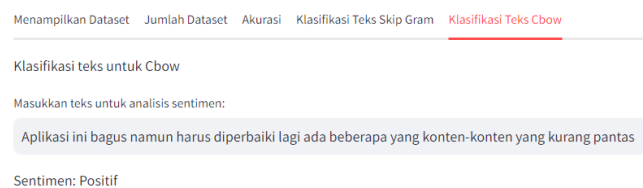
7. Halaman klasifikasi teks skip gram dan cbow (*Continuous Bag Of Words*) memasukan kalimat netral.

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE



Gambar 4. 34 Klasifikasi teks skip gram sentimen netral

IMPLEMENTASI WORD2VEC PADA KLASIFIKASI SENTIMEN ANALISIS TERHADAP ULASAN PENGGUNA APLIKASI TIKTOK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE



Gambar 4. 35 Klasifikasi teks skip gram sentimen netral

Pada gambar 4.30 dan gambar 4.31 *user* memasukan kalimat ‘Aplikasi ini bagus namun harus diperbaiki lagi ada beberapa konten-kontent yang kurang pantas’ dari kalimat tersebut yang telah dikirim, sistem akan memproses dengan memberikan hasil *output* yaitu netral namun hasil *output* positif oleh karena itu disebabkan kurangnya dataset dan tingkat akurasi yang tidak sempurna.

4.3 Hasil Uji Coba Sistem

Hasil uji klasifikasi teks antara model skip gram dan cbow (*continuous bag of words*) dengan *metode support vector machine* mempunyai kesimpulan sebagai berikut.

Tabel 4. 1 Hasil uji klasifikasi teks Skip Gram

No	Kalimat	Hasil sentimen	Kesimpulan
1	aplikasi ini sangat bermanfaat, kreatif dan mendapatkan banyak informasi penting untuk kita	Positif	Sistem mampu mendeteksi dengan baik untuk sentimen positif
2	Aplikasi ini sangat buruk dan membosankan buang-buang waktu saja	Negatif	Sistem mampu mendeteksi dengan baik untuk sentimen negatif
3	Aplikasi ini bagus namun harus diperbaiki lagi ada beberapa konten-kontent yang kurang pantas	Netral	Sistem mampu mendeteksi dengan baik untuk sentimen netral

Tabel 4. 2 Hasil uji klasifikasi teks Cbow (*Continouos Bag Of Words*)

No	Kalimat	Hasil sentimen	Kesimpulan
1	aplikasi ini sangat bermanfaat, kreatif dan mendapatkan banyak informasi penting untuk kita	Positif	Sistem mampu mendeteksi dengan baik untuk sentimen positif
2	Aplikasi ini sangat buruk dan membosankan buang-buang waktu saja	Negatif	Sistem mampu mendeteksi dengan baik untuk sentimen negatif
3	Aplikasi ini bagus namun harus diperbaiki lagi ada beberapa konten-kontent yang kurang pantas	Netral	Sistem mampu mendeteksi dengan baik untuk sentimen netral

Dari hasil uji dengan menggunakan data ulasan aplikasi tiktok yang berjumlah 900 dengan kategori sentimen negatif 300, positif 300 dan netral 300 dengan menggunakan model skip gram dan cbow (*continouos bag of words*) dengan metode *support vector machine* menghasilkan klasifikasi teks. Tahap model skip gram metode svm mampu mengklasifikasi teks dengan baik dengan kategori sentimen positif, negatif dan netral. Tahap model cbow metode svm mampu mengklasifikasi teks dengan kategori sentimen positif, negatif dan netral dengan baik.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan tahapan proses yang sudah dilakukan di bab-bab sebelumnya dapat disimpulkan sebagai berikut. Dataset yang digunakan dari tahun 2018 sampai 2023 penelitian ini menggunakan dataset dari ulasan aplikasi tiktok yang ada di playstore dengan mendapatkan dataset tersebut melalui proses web scraping, selanjutnya diberikan label sentimen secara manual dengan kategori positif, negatif dan netral jumlah dataset sebanyak 900. penerapan *word2vec* sebagai ekstraksi fitur menggunakan model skip gram dan cbow menggunakan metode *support vector machine*, mendapatkan akurasi skip gram 0.68 dan cbow (*continuous bag of words*) 0.66.

5.2 Saran

Saran yang dapat dilakukan untuk peneliti selanjutnya adalah meningkatkan dimensi pada *word2vec* untuk model skip gram dan cbow (*Continuous Bag Of Words*) yang mana bisa memungkinkan untuk meningkatkan akurasi, serta penambahan dataset yang lebih banyak agar dapat meningkatkan akurasi, dan fokus pada tahap *preprocessing*, yaitu memperbaiki kata yang tidak baku serta pengecekan kata yang lebih detail, serta membangun sistem klasifikasi teks yang lebih baik dan akurat dalam memberikan kategori sentimen positif, negatif ataupun netral.

DAFTAR PUSTAKA

- Pandu Adikara, P. (2021). *Analisis Sentimen Tanggapan Masyarakat Aplikasi Tiktok Menggunakan Metode Naïve Bayes dan Categorical Propotional Difference (CPD)*. 5(7), 2886–2890.
<http://j-ptiik.ub.ac.id>
- Nagpure, P. R. (2023). *WEBAPP FOR MULTIPLE APPROACH OF TEXT SUMMARIZATION*. 04, 1970–1975.
- Alita, D. (2022). Penerapan Metode Naïve Bayes Classifier Dan Support Vector Machine Pada Analisis Sentimen Terhadap Dampak Virus Corona Di Twitter. *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, 3(2), 145–160.
<http://jim.teknokrat.ac.id/index.php/informatika>
- Marga, N. S. (2021). Sentimen Analisis Publik Terhadap Kebijakan Lockdown Pemerintah Jakarta Menggunakan Algoritma Svm. *Jurnal Data Mining dan Sistem Informasi*, 2(1), 31.
<https://doi.org/10.33365/jdmsi.v2i1.1021>
- Astuti, W. (2021). Analisis Sentimen Terhadap Ulasan Film Menggunakan Algoritma Random Forest. *e-Proceeding of Engineering*, 8(4), 10153–10165.
- Setywan, H. (2023). *PREDIKSI GENRE ANIME PADA POSTER DENGAN METODE TRANSFER*. 4(5), 1041–1052.
- Padole, M. (2019). Deep learning to detect skin cancer using google colab. *International Journal of Engineering and Advanced Technology*, 8(6), 2176–2183.
<https://doi.org/10.35940/ijeat.F8587.088619>

- Kurniawan, F. W. (2020). Analisis Sentimen Twitter Bahasa Indonesia dengan Word2Vec. *Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(2), 4704–4713.
<https://openlibrary.telkomuniversity.ac.id/home/catalog/id/159923/slug/analisis-sentimen-twitter-bahasa-indonesia-dengan-word2vec.html%0A/home/catalog/id/159923/slug/analisis-sentimen-twitter-bahasa-indonesia-dengan-word2vec.html>
- Lestari, T. P. (2022). Analisis Text Mining pada Sosial Media Twitter Menggunakan Metode Support Vector Machine (SVM) dan Social Network Analysis (SNA). *Jurnal Informatika Ekonomi Bisnis*, 4(3), 65–71. <https://doi.org/10.37034/infec.v4i3.146>
- Musa, P. (2023). *Book · May 2023* (Nomor May).
- Nelli, F. (2023). The NumPy Library. *Python Data Analytics*, 45–72.
https://doi.org/10.1007/978-1-4842-9532-8_3
- Maulana, R. (2022). Analisa Sentimen Perkembangan Vtuber Dengan Metode Support Vector Machine Berbasis SMOTE. *Jurnal Teknik Komputer AMIK BSI*, 8(2), 174–180.
<https://doi.org/10.31294/jtk.v4i2>
- Munandar, A. (2022). *Comparison of Kernel Support Vector Machine Multi-Class in PPKM*. 5(158), 1–3.
- ... (2023). Analisis Sentimen Pemindahan Ibu Kota Indonesia pada Media Sosial Twitter menggunakan Metode LSTM dan Word2Vec. ... *Teknologi Informasi dan ...*, 7(5), 2389–2397.
<https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12731%0Ahttps://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/12731/5789>

- Widjaja, A. (2023). *PENERAPAN ALGORITMA NAÏVE BAYES DALAM ANALISIS SENTIMEN PENGGUNA TWITTER TERHADAP APLIKASI Application of the Naïve Bayes Algorithm in the Analysis of Twitter User Sentiments for the M-Tix 21 Cineplex Application*. 2(April), 277–286.
- Noor Hapsari, I. (2019). Analisis Dan Penerapan Datamining Untuk Mendeteksi Berita Palsu (Fake News) Pada Social Media Dengan Memanfaatkan Modul Scikit Learn. *Undergraduate Theses of Information Systems*.
- Arsi, P. (2021). Perbandingan Metode Support Vector Machine Dan Decision Tree Untuk Analisis Sentimen Review Komentar Pada Aplikasi Transportasi Online. *Journal of Information System Management (JOISM)*, 3(1), 1–7.
<https://doi.org/10.24076/joism.2021v3i1.341>
- Agustian, S. (2022). Support Vector Machine Method with Word2vec for Covid-19 Vaccine Sentiment Classification on Twitter. *Journal of Informatics and Telecommunication Engineering*, 6(1), 288–297. <https://doi.org/10.31289/jite.v6i1.7534>
- Chairani, C. (2023). Perbandingan Akurasi Metode Deteksi Ujaran Kebencian dalam Postingan Twitter Menggunakan Metode SVM dan Decision Trees yang Dioptimalkan dengan Adaboost. *Teknika*, 17(2), 287 – 299–287 – 299.
<https://jurnal.polsri.ac.id/index.php/teknika/article/view/6877>
- Hasan, F. N. (2022). Analisis Sentimen Terhadap Pembelajaran Daring Dengan Algoritma Naive Bayes Classifier. *Jurnal Linguistik Komputasional (JLK)*, 5(1), 34.

<https://doi.org/10.26418/jlk.v5i1.79>

Suryati Submitted, E. (2023). Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi dan Sistem Informasi*, 4(1), 96–106.

<https://doi.org/10.33365/jtsi.v4i1.2445>

Widayat, W. (2021). Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning. *Jurnal Media Informatika Budidarma*, 5(3), 1018.

<https://doi.org/10.30865/mib.v5i3.3111>

