

OPTIMASI ANALISIS SENTIMEN APLIKASI GLINTS MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM)

Fanny Rahmasari^{1*}, Nining Rahaningsih², Raditya Danar Dana³, Cep Lukman Rohmat⁴

^{1,2,3,4} STMIK IKMI Cirebon, Jl. Perjuangan No.10B, Karyamulya, Kec. Kesambi, Kota Cirebon, Jawa Barat 45135, Telp: (0231) 490480

Received: 12 Desember 2024

Accepted: 14 Januari 2025

Published: 20 Januari 2025

Keywords:

Analisis Sentimen, Support Vector Machine (SVM), Aplikasi Glints, Klasifikasi Sentimen.

Correspondent Email:

fannyrhmsri1102@gmail.com

Abstrak. Teknologi informasi telah mengubah cara orang mencari pekerjaan, dan aplikasi seperti Glints adalah salah satu contohnya. Namun, lebih banyak ulasan pengguna membuat analisis sentimen sulit. Pengelolaan fitur yang relevan dan pemilihan parameter yang ideal adalah masalah utama. Dengan menggunakan algoritma *Support Vector Machine (SVM)*, penelitian ini mengoptimalkan analisis sentimen ulasan Glints. Sebanyak 2000 ulasan dari Playstore dikumpulkan melalui scraping, dengan 69,2% positif, 16,6% netral, dan 14,2% negatif. Dalam proses pra-pemrosesan, case folding dan transformasi fitur menggunakan *TF-IDF* dengan unigram dan bigram dilakukan. Model *SVM* memiliki tingkat akurasi tinggi sebesar 92 persen, presisi sebesar 87%, *recall* sebesar 86%, dan *F1-Score* sebesar 86%. Implementasi berbasis Streamlit memungkinkan analisis sentimen dalam waktu nyata. Hasil ini membantu pengembang Glints meningkatkan layanan yang diberikan oleh pengguna.

Abstract. *Information technology has changed the way people look for jobs, and applications like Glints are one example of this. However, more user reviews make sentiment analysis difficult. The management of relevant features and the selection of ideal parameters are the main issues. By using the Support Vector Machine (SVM) algorithm, this research optimizes the sentiment analysis of Glints reviews. A total of 2000 reviews from the Playstore were collected through scraping, with 69.2% positive, 16.6% neutral, and 14.2% negative. In the preprocessing stage, case folding and feature transformation using TF-IDF with unigrams and bigrams were performed. The SVM model has a high accuracy rate of 92%, a precision of 87%, a recall of 86%, and an F1 score of 86%. Streamlit-based implementation enables real-time sentiment analysis. This result helps Glints developers improve the services provided by users.*

1. PENDAHULUAN

Teknologi informasi telah menghasilkan transformasi cara orang mengelola karier dan mencari pekerjaan. Glints, sebuah aplikasi yang memfasilitasi koneksi profesional, pengembangan keterampilan, dan pencarian pekerjaan, adalah salah satu platform yang mendukung proses ini. Data ulasan dan pendapat tentang aplikasi ini terus meningkat seiring dengan jumlah pengguna yang

menggunakannya. Analisis sentimen menjadi sangat penting untuk memahami pendapat pengguna dan meningkatkan layanan aplikasi. Berbagai metode dan algoritma telah digunakan dalam analisis sentimen untuk mengklasifikasikan opini menjadi positif, negatif, atau netral. *Support Vector Machine (SVM)* adalah salah satu algoritma yang paling umum digunakan untuk mengklasifikasikan teks, terutama untuk data berukuran besar

seperti ulasan pengguna. Namun, penerapan *SVM* masih menghadapi masalah, seperti memilih parameter yang ideal dan menangani fitur yang relevan untuk mening.

Beberapa penelitian tentang analisis sentimen menggunakan algoritma *Support Vector Machine (SVM)* telah dilakukan. Contohnya penelitian oleh Maulana dengan studinya yang berjudul “Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM)” menunjukkan bahwa algoritma *SVM* lebih akurat daripada algoritma *Naive Bayes* dan *KNN*. Studi ini mengevaluasi model menggunakan metode *K-Cross validation*, yang memungkinkan peneliti menemukan model dengan akurasi terbaik. Studi ini, bagaimanapun, hanya membahas aplikasi Pluang dan tidak membahas optimasi algoritma *SVM* lebih lanjut yang dapat digunakan pada aplikasi seperti Glints [1]. Selanjutnya penelitian oleh Sugihartono dengan studinya yang berjudul “Penerapan Metode *Support Vector Machine* Dalam *Classifikasi* Ulasan Pengguna Aplikasi Mobile JKN” memberikan penjelasan tentang proses yang termasuk pengumpulan data ulasan, preprocessing, ekstraksi fitur, dan evaluasi kinerja model *SVM*. Penelitian ini menekankan betapa pentingnya pembagian data yang tepat antara data pelatihan dan pengujian jika kita ingin meningkatkan akurasi model. Meskipun demikian, penelitian ini tidak membahas metode optimasi tambahan yang dapat meningkatkan kinerja *SVM* contohnya, penggunaan metode pengolahan bahasa alami yang lebih canggih atau pemilihan fitur yang lebih efisien [2]. Selanjutnya penelitian oleh Angraena dengan studinya yang berjudul “Analisis Sentimen Pengguna Aplikasi Google Meet Menggunakan Algoritma *Support Vector Machine*” Penelitian ini membahas penggunaan algoritma *SVM* untuk menganalisis sentimen pengguna terhadap aplikasi *Google Meet*. Penelitian ini menemukan bahwa *SVM* dapat digunakan secara efektif untuk mengklasifikasikan sentimen positif, negatif, dan netral dari ulasan pengguna. Hasil penelitian menunjukkan bahwa *SVM* memberikan akurasi yang lebih tinggi dalam analisis sentimen, dengan fokus pada pengolahan data teks dan ekstraksi fitur yang relevan. Namun, penelitian ini tidak membahas

secara menyeluruh metode optimasi yang dapat digunakan untuk meningkatkan akurasi lebih lanjut contohnya, penggunaan metode pengolahan bahasa alami yang lebih kompleks atau pemilihan fitur yang lebih canggih [3]. Studi tersebut menunjukkan bahwa algoritma *SVM* memiliki potensi yang luar biasa untuk analisis sentimen. Namun, penelitian lebih lanjut diperlukan untuk mengatasi masalah dengan optimasi parameter dan meningkatkan akurasi klasifikasi sentimen untuk aplikasi seperti Glints. Dengan menggunakan algoritma *Support Vector Machine (SVM)* untuk melakukan optimasi analisis sentimen pada aplikasi Glints, penelitian ini bertujuan untuk mengisi celah tersebut. Optimisasi yang tepat diharapkan dapat meningkatkan kinerja klasifikasi sentimen aplikasi Glints, memberikan informasi lebih lanjut tentang pendapat pengguna.

Tujuan penelitian ini adalah untuk menganalisis sentimen pengguna aplikasi Glints yang digunakan oleh algoritma *Support Vector Machine (SVM)* dan mengoptimalkan parameter *SVM* untuk meningkatkan akurasi klasifikasi sentimen melalui peningkatan akurasi analisis. Penelitian ini diharapkan dapat membantu mengembangkan teknik analisis sentimen untuk platform layanan karier yang lebih baik.

2. TINJAUAN PUSTAKA

2.1 Analisis Sentimen

Analisis sentimen adalah teknik untuk mengidentifikasi dan mengklasifikasikan pendapat yang terkandung dalam teks, baik itu positif, negatif, atau netral. Ini sangat penting dalam e-commerce karena ulasan konsumen tentang produk dapat memengaruhi keputusan mereka untuk membeli produk. Handayani [4] merinci bahwa analisis sentimen menjadi alat yang efektif untuk membantu pelanggan memahami pendapat publik tentang produk tertentu karena semakin banyak ulasan produk, konsumen seringkali kesulitan membuat kesimpulan yang tepat.

2.2 Algoritma Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah salah satu algoritma yang paling

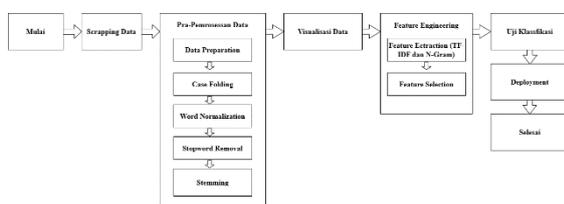
banyak digunakan dalam analisis sentimen karena dapat menangani data yang kompleks dan berukuran besar. Sugihartono [2] menggambarkan proses penerapan *SVM*, mulai dari pengumpulan data, preprocessing, dan evaluasi kinerja model. Penelitian ini menunjukkan bahwa *SVM* dapat dengan akurat mengklasifikasikan sentimen ulasan pengguna, yang menjadikannya pilihan yang tepat untuk penelitian ini.

2.3 Pemrosesan Bahasa Alami (NLP)

Bidang studi *Natural Language Processing (NLP)* berkonsentrasi pada bagaimana bahasa manusia berinteraksi dengan komputer. Teknik pengolahan bahasa natural (*NLP*) digunakan dalam analisis sentimen untuk mengubah teks mentah menjadi data yang dapat dianalisis oleh algoritma seperti *SVM*. Teknik *NLP* termasuk tokenisasi, stemming, lemmatisasi, dan pembersihan teks untuk memastikan bahwa data siap untuk dianalisis [5].

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan Knowledge Discovery in Databases (KDD) untuk mengoptimalkan analisis sentimen aplikasi Glints dengan menggunakan algoritma Support Vector Machine (*SVM*). Dalam penelitian ini, beberapa tahapan KDD termasuk pra-pemrosesan data, visualisasi data, teknik fitur, dan uji klasifikasi [6].



Gambar 3. 1 Metode Penelitian

3.1 Scrapping Data

Pada tahap ini, data ulasan dikumpulkan dari aplikasi Glints yang tersedia di Playstore. Sebanyak 2000 data ulasan terbaru dikumpulkan melalui teknik web scraping, yang dilakukan pada September 2024. Data yang dikumpulkan terdiri dari ulasan teks pengguna yang dikategorikan menjadi positif, negatif, dan netral. Selanjutnya, data disimpan dalam format

yang sesuai, seperti CSV atau JSON, sehingga dapat diproses lebih lanjut di langkah berikutnya.

3.2 Pra-Pemrosesan Data

Sebelum analisis sentimen dilakukan, pra-pemrosesan data bertujuan untuk membersihkan dan mempersiapkan data. Proses ini terdiri dari beberapa langkah berikut:

1. Data Preparation

mengorganisir data yang dihasilkan dari scraping secara seragam dan terstruktur. Ini termasuk menyusun ulang data dalam bentuk tabel dengan kolom seperti ID Ulasan, Teks Ulasan, dan Label Sentimen [7].

2. Case Folding

Mengubah semua huruf dalam teks menjadi huruf kecil agar tidak berubah karena kapitalisasi. Sebagai contoh, kata "baik" diganti menjadi "baik" [8].

3. Word Normalization

Menormalisasi kata-kata yang tidak baku menjadi bentuk baku, seperti "gk" [9].

4. Stopword Removal

Untuk analisis sentimen, hapus kata-kata umum seperti "yang", "di", dan "dan". Tujuan menghapus stopwords adalah untuk mengurangi dimensi fitur dan meningkatkan efisiensi pemrosesan [10].

5. Stemming

Mengubah kata ke bentuk dasar, seperti mengubah kata "pekerjaan" menjadi "kerja", adalah proses yang membantu mengurangi jumlah kata yang berbeda sehingga analisis dapat dilakukan dengan lebih konsisten [11].

3.3 Visualisasi Data

Setelah tahap *pra-pemrosesan* selesai, data divisualisasikan untuk memahami distribusi dan pola sentimen. Diagram batang dapat digunakan untuk menunjukkan frekuensi sentimen positif, negatif, dan netral. Selain itu, *word cloud* digunakan untuk menampilkan kata-kata yang paling sering muncul dalam ulasan, memberikan gambaran umum tentang topik yang sering didiskusikan oleh pengguna. Visualisasi ini membantu memahami sebaran data, menemukan masalah potensial dalam kumpulan data, dan memastikan bahwa data siap untuk tahap analisis berikutnya [12].

3.4 Feature Engineering

Feature Engineering digunakan untuk mengekstrak dan memilih fitur yang relevan dari data teks, yang mencakup:

1. Feature Extraction

Untuk mengkonversi teks menjadi representasi numerik, metode *TF-IDF* (*Term Frequency-Inverse Document Frequency*) dan *N-Gram* digunakan.

$$TF-IDF(t, d) = TF(t, d) \times \log \left(\frac{N}{DF(t)} \right)$$

Di mana :

- **TF (Term Frequency)** adalah jumlah kata *t* yang muncul dalam dokumen *d*
- **DF (Document Frequency)** adalah jumlah dokumen dengan kata *t*
- **N** adalah total jumlah dokumen

2. Feature Selection

Untuk meningkatkan kinerja algoritma klasifikasi, pilih fitur yang paling penting. Tujuan dari proses ini adalah untuk mengurangi jumlah fitur yang tidak penting sehingga meningkatkan akurasi dan efisiensi komputasi model.

3.5 Uji Klasifikasi

Pada tahap ini, algoritma *Support Vector Machine* (*SVM*) digunakan untuk mengkategorikan data ulasan ke dalam kategori sentimen positif, negatif, atau netral. Proses ini termasuk:

1. Training dan Testing

Data dibagi menjadi dua bagian: set pelatihan (*training set*) dan set uji (*test set*). Set pelatihan digunakan untuk melatih model *SVM*, sementara set uji digunakan untuk mengukur kinerja model.

2. Evaluasi Model

Untuk menilai kinerja model, metrik seperti akurasi, presisi, recall, dan skor F1 digunakan. Rumus untuk masing-masing metrik adalah sebagai berikut:

- **Akurasi**

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Presisi**

$$Presisi = \frac{TP}{TP + FP}$$

- **Recall**
- **F1-Score** TP

$$F1-Score = \frac{2 \times Presisi \times Recall}{Presisi + Recall}$$

Di mana :

- **TP (True Positive)** : Prediksi positif yang tepat
- **TN (True Negative)** : Prediksi negatif yang tepat.
- **FP (False Positive)** : Prediksi positif yang tidak akurat.
- **FN (False Negative)** : Prediksi negatif yang tidak akurat.

3.6 Deployment

Setelah model *SVM* diuji dan dioptimalkan, model terbaik dipasang pada aplikasi Glints menggunakan streamlit untuk analisis sentimen secara *real-time*. Model ini dapat dipantau secara otomatis untuk ulasan dan memberikan kontribusi penting untuk meningkatkan kualitas layanan.

4. HASIL DAN PEMBAHASAN

4.1 Hasil Scrapping Data

Pada bulan September 2024, proses scrapping data digunakan untuk mengumpulkan 2000 data ulasan terbaru dari aplikasi Glints yang tersedia di *Playstore*. Data yang dikumpulkan terdiri dari ulasan pengguna dengan berbagai jenis sentimen, termasuk positif, negatif, dan netral. Untuk pemrosesan lebih lanjut, data kemudian disimpan dalam format CSV.

```
from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.glints.candidate',
    lang='id',
    country='id',
    sort=Sort.NEWEST,
    count=2000,
    filter_score_with=None
)
```

Gambar 4.1 Code program scrapping data

reviewId	userName	userImage	content	score	timestamp	reviewCreatedDate	is	replyContent	replyDate	replyScore
u730808-f68-64c4-7a-7b270a9e927c	Neska M. Hani		Siapa bilang... Malah enak!	5	1693100430	2024-08-27 08:01:51	0		2024-08-27 08:01:51	1.000
980x44c-830c-4131-930e9d000000022	Jheni Ayu		Kecep banget... mending sih...!	5	1693100430	2024-08-27 08:01:51	0		2024-08-27 08:01:51	1.000
228032d-f16-840-5505-c04f8b30c	Rudhita		memainkan	5	1693100430	2024-08-27 08:01:51	0		2024-08-27 08:01:51	1.000
118368f-9c10-910c-7a-7b270a9e927c	Kecep net		sempet macem... malah enak... mending sih...!	5	1693100430	2024-08-27 08:01:51	0		2024-08-27 08:01:51	1.000
6027793-150f-45c7-930e9d000000022	Alya		Sehring banget... cepat banget... mending!	5	1693100430	2024-08-27 08:01:51	0		2024-08-27 08:01:51	1.000

Gambar 4.2 Hasil Scrapping Data

4.2 Pra-Pemrosesan Data

Data yang diperoleh melalui scraping diproses dalam tahap pra-pemrosesan data, yang terdiri dari beberapa langkah, seperti:

1) Data Preparation

Untuk memastikan bahwa data yang akan digunakan dalam proses analisis sentimen siap dan berkualitas baik, tahap persiapan data melibatkan sejumlah langkah sistematis, seperti pelabelan data, pemanggilan dataset, pengecekan duplikat data, pengecekan *missing values*, dan pemilihan kolom yang sesuai. Tujuan dari proses ini adalah untuk menghindari masalah yang dapat mempengaruhi hasil analisis dan untuk memastikan bahwa data yang digunakan sesuai untuk tahap selanjutnya.

a. Pelabelan

Proses pelabelan sentimen dalam penelitian ini dilakukan dengan bantuan Ibu Hermin, M.Pd., seorang pakar bahasa Indonesia yang ahli. Berdasarkan makna dan konteksnya, teks dalam dataset diberi label positif, netral, atau negatif. Teks yang mengandung pujian atau kepuasan diberi label positif, sedangkan teks yang mengandung keluhan atau ketidakpuasan diberi label negatif. Teks yang netral diberi label netral. Setelah pelabelan, verifikasi dilakukan untuk memastikan label konsisten dan akurat. Data yang sudah dilabeli digunakan untuk analisis tambahan. Misalnya, mereka digunakan untuk melatih model analisis sentimen.

Tabel 1.1 Kategori sentimen

Label	Deskripsi
Positif	Ulasan yang menunjukkan seberapa puas atau menghargai Anda dengan aplikasi
Netral	ulasan yang tidak emosi tetapi penuh informasi.
Negatif	Ketidakpuasan, keluhan, atau kritik dalam komentar.

b. Pemanggilan Dataset

Proses pemanggilan dataset dilakukan oleh library pandas Python. Kode program berikut digunakan untuk memasukkan dataset ke lingkungan pemrograman.

```
data = pd.read_csv('scrapped_data_glints2.csv')
data.head()
✓ 0.2s
```

Gambar 4.3 Pemanggilan dataset

	userName	score	at	label	content
0	Nanda Mmhmarsel	5	10/7/2024 5:41	positif	baik
1	Zhen Zen	2	10/7/2024 4:41	negatif	Kurang lengkap masalah info loker
2	Rachmad Rudianto	5	10/7/2024 2:10	positif	memuaskan
3	Bang zait	5	10/7/2024 1:39	positif	sangat mudah untuk mengirim email/ cv
4	Alan Walansari	5	10/6/2024 16:46	netral	Semoga cepet dapat pekerjaan, aminnnn

Gambar 4.4 Hasil pemanggilan dataset

c. Cek Data Duplikat

Pada titik ini, dataset yang dihasilkan dari proses pelabelan yang dilakukan oleh pakar bahasa Indonesia diintegrasikan ke dalam proyek untuk dianalisis lebih lanjut. Proses pemanggilan dataset dilakukan oleh library pandas Python. Kode program berikut digunakan untuk memasukkan dataset ke lingkungan pemrograman:

```
# Cek data duplikat
data.duplicated().sum()
✓ 0.0s
```

0

Gambar 4.5 Cek duplikat data

Semua data dalam dataset adalah unik dan tidak ada pengulangan, karena hasil eksekusi kode menunjukkan nilai 0. Oleh karena itu, data yang digunakan untuk analisis sentimen dinilai dengan baik dan siap untuk tahap pemrosesan berikutnya.

d. Cek Missing Values

dilakukan pengecekan terhadap data yang hilang atau tidak ada nilainya. Akurasi model dan proses pelatihan dapat dipengaruhi oleh data yang tidak memiliki nilai. Oleh karena itu, memastikan bahwa dataset tidak memiliki nilai yang tidak ada

adalah langkah penting dalam proses prapemrosesan data.

```
# Cek missing values
data.isnull().sum()
✓ 0.0s

userName    0
score       0
at          0
label       0
content     0
dtype: int64
```

Gambar 4. 6 Cek *missing values*

Hasilnya menunjukkan bahwa dataset memiliki data lengkap dalam setiap kolom dan tidak ada nilai kosong.

e. Pemilihan Kolom

Pada tahap optimasi analisis sentimen aplikasi Glints yang menggunakan algoritma *Support Vector Machine (SVM)*, pemilihan kolom dilakukan untuk memastikan bahwa data yang digunakan relevan dengan tujuan penelitian. Tujuan pemilihan kolom adalah untuk menyederhanakan dataset dengan menambahkan hanya atribut yang diperlukan untuk analisis sentimen, seperti teks ulasan dan label sentimen.

```
data.drop(columns = ['userName', 'at', 'label', 'score'], inplace = True)
data.columns = ['content', 'label_num']
data.head()
✓ 0.0s
```

Gambar 4.7 Code program pemilihan kolom

	content	label_num
0	baik	2
1	Kurang lengkap masalah info loker	0
2	memuaskan	2
3	sangat mudah untuk mengirim email/ cv	2
4	Semoga cepet ² dapet pekerjaan,, aminnnn	1

Gambar 4. 8 Hasil pemilihan kolom

Dalam dataset, hanya teks ulasan dan label sentimen yang diperlukan untuk membangun model analisis sentimen menggunakan *Support Vector Machine (SVM)* dapat ditemukan dengan memilih kolom ini. Kolom lain, seperti *userName*, *at*, dan *score*, dihapus karena mereka tidak

memberikan kontribusi langsung terhadap klasifikasi sentimen.

2) Case Folding

Untuk mengoptimalkan analisis sentimen aplikasi Glints menggunakan algoritma *Support Vector Machine (SVM)*, *case folding* adalah tahap penting dalam prapemrosesan data. *Case folding* digunakan untuk menghilangkan elemen yang tidak penting seperti *URL*, angka, dan tanda baca dari teks, menyederhanakannya menjadi huruf kecil. Dengan mengurangi variasi teks yang tidak diperlukan, langkah ini membantu meningkatkan akurasi model.

```
# Membandingkan before dan after case folding
raw_sample = data['content'].iloc[3]
case_folding = caseFolding(raw_sample)

print('Raw data\t:', raw_sample)
print('Case folding\t:', case_folding)

print('\n')

raw_sample = data['content'].iloc[4]
case_folding = caseFolding(raw_sample)

print('Raw data\t:', raw_sample)
print('Case folding\t:', case_folding)

Raw data      : sangat mudah untuk mengirim email/ cv
Case folding  : sangat mudah untuk mengirim email cv

Raw data      : Semoga cepet2 dapet pekerjaan,, aminnnn
Case folding  : semoga cepet2 dapet pekerjaan aminnnn
```

Gambar 4. 9 Hasil sebelum & sesudah *case folding*

Tahap *case folding* membuat teks yang digunakan untuk melatih model *SVM* lebih bersih dan seragam. Ini memungkinkan model untuk mengenali pola sentimen lebih mudah tanpa terganggu oleh perubahan format atau elemen yang tidak penting.

3) Word Normalization

Daftar kata tidak baku dan padanannya dalam bahasa Indonesia digunakan untuk melakukan normalisasi kata dalam penelitian ini. Bentuk baku ditetapkan untuk kata-kata slang, singkatan, atau penulisan yang tidak sesuai dengan Kamus Besar Bahasa Indonesia (KBBI).

```
# Membandingkan before after word normalization
raw_data = data['content'].iloc[65]
word_normal = text_normalize(raw_data)

print('Raw Data\t:', raw_data)
print('Word Normalization\t:', word_normal)
✓ 0.0s

Raw Data      : The glintts Alhamdulillah sudah dapet pekerjaan dari apk glints
Word Normalization : terima kasih glints alhamdulillah sudah dapat pekerjaan dari aplikasi glints
```

Gambar 4. 10 Hasil sebelum & sesudah *word normalization*

Proses normalisasi kata membantu mengurangi kompleksitas data dan meningkatkan kinerja model *Support Vector Machine (SVM)* dengan mengurangi jumlah variasi kata yang tidak perlu. Dengan normalisasi yang tepat, model dapat mengenali pola sentimen dengan lebih baik dan menghasilkan klasifikasi yang lebih akurat.

4) Stopword Removal

Dalam analisis sentimen, penghapusan stopword adalah tahap prapemrosesan. Tahap ini bertujuan untuk menghilangkan kata-kata umum dalam teks yang tidak memberikan makna atau sentimen yang signifikan. Kata-kata seperti "yang", "dan", "di", "ke", atau "dari" sering muncul dalam teks, tetapi tidak membantu membedakan sentimen positif, netral, atau negatif. Dengan menghilangkan stopword, model bisa lebih fokus pada kata-kata yang lebih relevan.

```
raw_sample = data['content'].iloc[65]
case_folding = caseFolding(raw_sample)
stopword_removal = remove_stop_word_and_detect_sentiment(case_folding)

print('Raw Data \t\t:', raw_data)
print('Case Folding \t\t:', case_folding)
print('Stopword Removal \t\t:', stopword_removal)

✓ 0.0s
```

Raw Data : thx glintts alhamdulillah sudah dapat pekerjaan dari apk glintts
Case Folding : thx glintts alhamdulillah sudah dapat pekerjaan dari apk glintts
Stopword Removal : thx glintts alhamdulillah dapat pekerjaan apk glintts

Gambar 4.11 Hasil sebelum & sesudah *stopword removal*

Hal ini memastikan bahwa hanya kata-kata yang terkait dengan sentimen yang dipertahankan dalam dataset melalui proses penghapusan *stopword*. Oleh karena itu, model *Support Vector Machine (SVM)* dapat berkonsentrasi pada data penting.

5) Stemming

Mengubah kata menjadi bentuk dasarnya (kata dasar) dengan menghapus imbuhan (awalan, sisipan, dan akhiran) disebut stemming. Stemming membantu dalam analisis sentimen karena menyederhanakan variasi kata yang memiliki makna yang sama. Akibatnya, model *Support Vector Machine (SVM)* menjadi lebih efisien dan akurat. Karena kata-kata dengan akar yang sama diwakili oleh satu bentuk dasar, stemming mengurangi jumlah fitur yang ada dalam

teks. Pustaka sastra yang mendukung bahasa Indonesia digunakan untuk proses stemming dalam penelitian ini.

```
raw_sample = data['content'].iloc[65]
case_folding = caseFolding(raw_sample)
stopword_removal = remove_stop_word_and_detect_sentiment(case_folding)
text_stemming = stemming(stopword_removal)

print('Raw Data \t\t:', raw_sample)
print('Case Folding \t\t:', case_folding)
print('Stopword Removal \t\t:', stopword_removal)
print('Stemming \t\t:', text_stemming)

✓ 40s
```

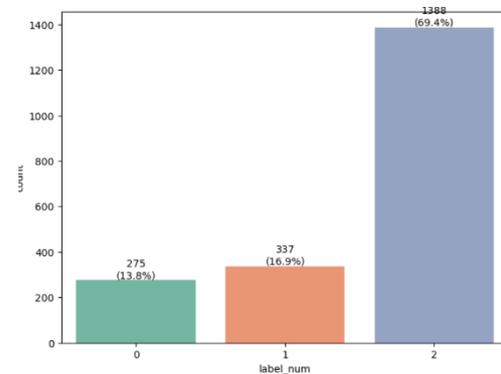
Raw Data : Thx glintts alhamdulillah sudah dapat pekerjaan dari apk glintts
Case Folding : thx glintts alhamdulillah sudah dapat pekerjaan dari apk glintts
Stopword Removal : thx glintts alhamdulillah dapat pekerjaan apk glintts
Stemming : thx glintts alhamdulillah dapat kerja apk glintts

Gambar 4.12 Hasil sebelum & sesudah *stemming*

Dengan menyelaraskan kata-kata dalam bentuk dasar, proses stemming meningkatkan akurasi model analisis sentimen karena membuat model lebih fokus pada pola sentimen yang relevan tanpa terganggu oleh variasi kata yang memiliki arti yang sama.

4.3 Visualisasi Data

Untuk memahami distribusi sentimen dalam dataset ulasan aplikasi Glints, visualisasi data digunakan.



Gambar 4.13 Distribusi sentimen

Grafik batang di atas menunjukkan jumlah evaluasi berdasarkan tiga kategori sentimen: positif(2), netral(1), dan negatif(0). Dataset dengan 1.384 ulasan dengan sentimen positif, yang mencakup 69,2% dari data, menunjukkan bahwa sebagian besar pengguna menyukai aplikasi Glints. Sebanyak 333 ulasan dengan sentimen netral, yang merupakan 16,6% dari data, menunjukkan tanggapan yang tidak terlalu memihak, baik positif maupun negatif. 283 ulasan negatif, atau 14,2% dari data, menunjukkan bahwa, meskipun jumlah ulasan

menggunakan *TF-IDF*. Tujuan dari seleksi fitur adalah untuk memilih fitur-fitur yang paling relevan dan berdampak terhadap klasifikasi sentimen. Untuk melakukan seleksi fitur ini, metode *chi-square* (χ^2) digunakan. Metode ini mengevaluasi hubungan antara setiap fitur dan label target. Metode ini hanya mempertahankan fitur yang memiliki nilai signifikansi tinggi untuk prediksi sentimen. Proses ini meningkatkan kinerja model dan mengurangi kemungkinan *overfitting* karena jumlah fitur yang terlalu banyak.

jumlah fitur	akurasi	presisi	recall	f1_score	confusion_matrix
1	0.85	0.85	0.85	0.85	[[100 0 0] [0 100 0] [0 0 100]]
2	0.85	0.85	0.85	0.85	[[100 0 0] [0 100 0] [0 0 100]]
3	0.85	0.85	0.85	0.85	[[100 0 0] [0 100 0] [0 0 100]]
4	0.85	0.85	0.85	0.85	[[100 0 0] [0 100 0] [0 0 100]]

Gambar 4. 18 Hasil *feature selection*

Hasil teknik fitur ini menunjukkan bahwa kombinasi *TF-IDF N-Gram* dan metode seleksi fitur memberikan representasi data terbaik proses ini secara signifikan meningkatkan akurasi dan generalisasi model terhadap data uji.

4.5 Uji Klasifikasi

A. Persiapan Library

Pada awal kode program, berbagai pustaka diimpor untuk mendukung analisis sentimen. Pembagian data menjadi set pelatihan dan pengujian adalah contohnya. Penggunaan *train_test_split* dari *sklearn.model_selection* untuk membagi dataset menjadi dua bagian data pelatihan dan data pengujian adalah penting untuk evaluasi model. Selanjutnya, algoritma *Support Vector Machine (SVM)* diimpor dari *sklearn.svm* melalui *SVC*. Kemudian dibangun model klasifikasi sentimen berdasarkan data yang telah diproses. Terakhir, pustaka *classification_report*, *accuracy_score*, dan *confusion_matrix* diimpor dari *sklearn.metrics* untuk mengevaluasi kinerja model yang dibangun. Berbagai metrik evaluasi tersedia di buku ini, termasuk akurasi, presisi, *recall*, dan *confusion matrix*.

```
# Import library
import random
from sklearn.model_selection import train_test_split

# Import Algoritma SVM
from sklearn.svm import SVC

from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
```

Gambar 4. 19 Library klasifikasi

B. Membagi Data Train dan Data Test

Proses analisis sentimen membagi data yang telah diproses menjadi dua bagian: data pelatihan (*training*) dan data pengujian (*testing*). Tahap *engineering feature* termasuk teknik ekstraksi fitur seperti *TF-IDF* dan *N-Gram* serta metode seleksi fitur menggunakan metode *SelectKBest*. Dengan menggunakan kode program sebelumnya, data dibagi menjadi dua kelompok: kelompok pelatihan (*training*) dan kelompok pengujian (*testing*). Pada bagian pertama, variabel *x* diberi nilai dari fitur yang telah dipilih sebelumnya, *selected_x*. Variabel *y* diberi nilai dari kolom *label_num* dari dataset, yang merupakan label target untuk klasifikasi (dalam hal ini, analisis sentimen). Fungsi *train_test_split* library *scikit-learn* digunakan untuk membagi data *x* dan *y* menjadi dua bagian. Sepuluh persen data digunakan untuk pelatihan (*training*) dan sepuluh persen sisanya untuk pengujian. Dengan parameter *test_size = 0.1*, sepuluh persen data akan digunakan untuk menguji model dan sepuluh persen akan digunakan untuk melatih model. Argumen *random_state=2* digunakan setiap kali kode digunakan untuk memastikan bahwa pembagian data adalah reproduktif dan konsisten. Setelah pemisahan data, kode ini menunjukkan jumlah data yang digunakan untuk pelatihan dan pengujian dengan fungsi *len()*. Data fitur dan target untuk pelatihan dan pengujian masing-masing adalah jumlah sampel yang dimiliki oleh *x_train*, *x_test*, *y_train*, dan *y_test*. Pembagian ini sangat penting karena set pelatihan digunakan untuk melatih model, sedangkan set pengujian digunakan untuk mengukur kinerja model setelah dilatih.

```
x = selected_x
y = data.label_num

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=2)

# Menampilkan jumlah data training dan data testing
print('Banyaknya X_train : ', len(x_train))
print('Banyaknya X_test : ', len(x_test))
print('Banyaknya y_train : ', len(y_train))
print('Banyaknya y_test : ', len(y_test))

Banyaknya X_train : 1800
Banyaknya X_test : 200
Banyaknya y_train : 1800
Banyaknya y_test : 200
```

Gambar 4. 20 Train test split

C. Pelatihan dan Pengujian Model

Penyiapan model *Support Vector Machine (SVM)* dimulai dengan menggunakan pustaka *scikit-learn*. Parameter *class_weights* diberi bobot 1 pada kelas 0 dan 2, dan kelas 1 diberi bobot 2 untuk menangani ketidakseimbangan kelas dalam dataset. Selanjutnya, model *SVM* dibuat menggunakan kernel Fungsi *Basis Radial (RBF)*, jenis kernel yang biasa digunakan untuk masalah klasifikasi non-linear. Parameter *gamma* kernel diatur oleh *gamma='scale'* untuk menyesuaikan model dengan data, dan tingkat penalti untuk kesalahan klasifikasi ditentukan oleh parameter *C=100*. Proses pelatihan model setelah model disiapkan. *Fit(x_train, y_train)* digunakan untuk melatih model dengan data pelatihan yang telah disiapkan sebelumnya. Oleh karena itu, prediksi sentimen pada dataset yang tersedia dapat dilakukan dengan menggunakan model *SVM* siap pakai ini.

```
# Menyiapkan model SVM
class_weights = (0: 1, 1: 2, 2: 1)

model = SVC(kernel='rbf', C=100, gamma='scale', class_weight=class_weights)

# Melatih model pada data yang sudah di-resample
model.fit(x_train, y_train)
```

Gambar 4. 21 Persiapan model SVM

D. Evaluasi Model

Setelah model dilatih, beberapa metrik digunakan untuk menilai kinerjanya. Ini termasuk metrik akurasi, presisi, recall, dan *FI-score*. Metrik presisi mengukur akurasi prediksi positif dengan membandingkan jumlah prediksi benar untuk kelas tertentu dengan jumlah prediksi total untuk kelas tersebut.

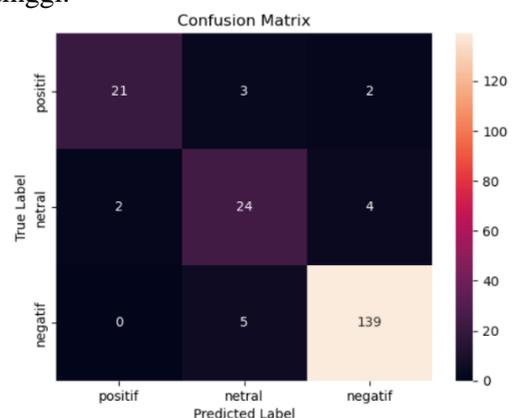
```
# Evaluasi model pada data pengujian
y_pred = model.predict(x_test)

# Menampilkan hasil evaluasi
print(f'Classification Report: \n(classification_report(y_test, y_pred))')
```

Classification Report:				
	precision	recall	f1-score	support
0	0.91	0.81	0.86	26
1	0.75	0.80	0.77	30
2	0.96	0.97	0.96	144
accuracy				0.92
macro avg				0.87
weighted avg				0.92

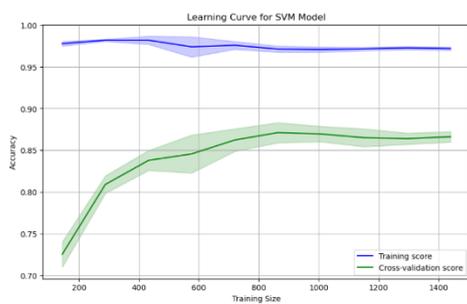
Gambar 4. 22 Classification report

Hasil evaluasi model yang ditampilkan dalam laporan klasifikasi menunjukkan bagaimana model klasifikasi bekerja untuk setiap kelas menggunakan metrik seperti *precision*, *recall*, skor *f1*, dan dukungan. Untuk kelas 0, ketepatan model adalah 0,91, yang menunjukkan bahwa 91% dari prediksi yang dibuat oleh model untuk kelas 0 adalah benar, dan *recall* adalah 0.81, yang menunjukkan bahwa model berhasil memprediksi 81% dari data yang sebenarnya yang termasuk dalam laporan klasifikasi. Secara keseluruhan, model ini memiliki ketepatan sebesar 0.92, yang berarti 92 persen dari semua prediksi yang dibuat benar. Untuk rata-rata, makro rata-rata menunjukkan nilai ketepatan sebesar 0.87, *recall* sebesar 0.86, dan nilai *f1*-sebesar 0.86, yang menunjukkan bahwa model mampu menjaga performa yang baik secara keseluruhan pada semua kelas, meskipun kelas-kelas tersebut memiliki distribusi yang berbeda. Sebaliknya, rata-rata berat menunjukkan nilai sec Model ini menunjukkan kinerja yang luar biasa, terutama dalam memprediksi kelas 2, dengan keseimbangan yang baik antara akurasi dan ketepatan dan akurasi yang tinggi.



Gambar 4. 23 Confusion matrix

Confusion matrix yang terdiri dari tiga kategori sentimen: positif, netral, dan negatif menunjukkan bagaimana model berfungsi dalam analisis sentimen untuk aplikasi Glints. Model berhasil mengklasifikasikan 21 data ke dalam kategori positif dengan benar, tetapi terdapat kesalahan pada 3 data netral dan 2 data negatif. Model juga berhasil mengklasifikasikan 24 data ke dalam kategori netral dengan benar, tetapi salah mengklasifikasikan 2 data positif dan 4 data negatif. Dalam kategori negatif, model berhasil mengklasifikasikan 139 data dengan benar, meskipun ada 5 data yang salah diprediksi.



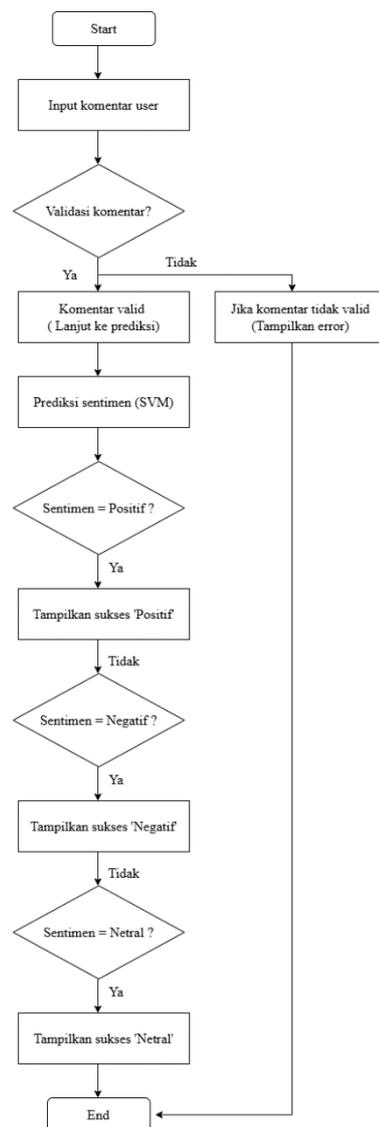
Gambar 4. 24 Learning curve

Bagaimana model *Support Vector Machine* (SVM) melakukan analisis sentimen pada aplikasi Glints berdasarkan jumlah data latih yang digunakan, ditunjukkan oleh kurva pembelajaran ini. Grafik ini terdiri dari dua garis utama: Skor Pelatihan (berwarna biru) dan Skor *Cross-Validation* (berwarna hijau). Untuk garis *cross-validation*, pada awal pelatihan, dengan data latih yang lebih sedikit, skor validasi lebih rendah, mencapai sekitar 0.80. Skor pelatihan hampir 1.0 (100%) menunjukkan bahwa model SVM sangat mampu mempelajari pola pada data latih. Garis ini cukup stabil, menunjukkan bahwa model tidak kesulitan menyesuaikan dengan data latih. Kesimpulannya, model SVM dapat mempelajari pola data latih dan menggeneralisasi pada data validasi. Skor validasi meningkat secara signifikan seiring bertambahnya data latih hingga mendekati skor training (di atas 0.90), menunjukkan bahwa model semakin mampu menggeneralisasi pola dari data baru. Dengan data yang cukup besar (lebih dari 2000 sampel), model berhasil mengimbangi skor pelatihan dan

validasi dengan akurasi kira-kira 0.92, menunjukkan bahwa model tidak melakukan *overfitting* atau *underfitting*.

4.6 Deployment

Pada saat ini, hasil penelitian diterapkan dalam bentuk aplikasi web sederhana berbasis platform Streamlit. Dengan memasukkan teks ulasan ke dalam aplikasi, pengguna dapat secara langsung mengakses hasil analisis sentimen.

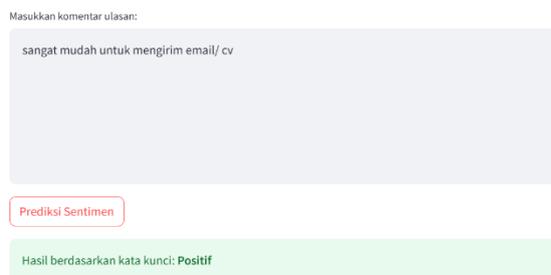


Gambar 4. 25 Flowchart deployment

Alur kerja sistem analisis sentimen ini digambarkan dalam flowchart. Alur kerja ini dimulai dengan input komentar dan berakhir dengan algoritma *Support Vector Machine* (SVM) yang mengklasifikasikan sentimen.

Pengguna dapat meninggalkan ulasan atau komentar tentang aplikasi Glints dengan mengisikan komentar. Sistem memastikan apakah komentar yang dimasukkan benar selama proses validasi komentar. Pesan error akan ditampilkan oleh sistem jika komentar tidak valid, seperti kosong atau tidak memenuhi syarat. Prediksi Sentimen dilakukan oleh sistem menggunakan model yang sudah dilatih (*SVM*) setelah komentar dianggap valid. Model ini menentukan apakah komentar positif, negatif, atau netral. Menurut prediksinya, jika komentar menunjukkan sentimen positif, sistem akan menampilkan pesan "Sukses Positif", pesan "Sukses Negatif", dan pesan "Sukses Netral". Jika sentimen netral, sistem akan menampilkan pesan "Sukses Netral".

Prediksi Sentimen Ulasan



Gambar 4.26 Prediksi sentimen positif

Gambar 4.26 menunjukkan hasil analisis sentimen yang menemukan bahwa komentar bernilai positif, yang ditandai dengan pesan sukses yang memberi tahu pengguna bahwa komentar tersebut bernilai positif.

Prediksi Sentimen Ulasan



Gambar 4.27 Prediksi sentimen negatif

Gambar 4.27 menunjukkan hasil analisis sentimen menemukan bahwa komentar bersifat negatif, menghasilkan pesan error yang memberi tahu pengguna bahwa komentar tersebut memiliki sentimen negatif.

Prediksi Sentimen Ulasan



Gambar 4.28 Prediksi sentimen netral

Gambar 4.28 menunjukkan hasil analisis sentimen menemukan bahwa komentar netral, dengan pesan informasi, dan tidak mengandung sentimen ekstrem.

5. KESIMPULAN

Hasil penelitian menghasilkan beberapa kesimpulan berikut:

- Data yang dikumpulkan dari aplikasi Glints, yang tersedia di *Playstore*, terdiri dari 2000 ulasan pengguna dengan berbagai jenis sentimen, termasuk positif, negatif, dan netral, melalui proses *scraping* data yang berhasil.
- Tahap *preprocessing* data seperti case folding berhasil menghasilkan teks yang digunakan untuk melatih model *SVM* menjadi lebih bersih dan seragam, yang memudahkan model untuk mengidentifikasi pola sentimen.
- Menurut diagram batang, dataset mengandung 1.384 ulasan dengan sentimen positif (69,2%), 333 ulasan dengan sentimen netral (16,6%), dan 283 ulasan dengan sentimen negatif (14,2%). Ini menunjukkan bahwa mayoritas pengguna menyukai aplikasi Glints.
- Analisis cloud kata positif menemukan fitur pengguna favorit, seperti "bantu", "bagus", "kerja", dan "mudah". Sementara itu, analisis cloud kata negatif menemukan keluhan pengguna tentang pengalaman, masalah teknis, atau antarmuka aplikasi.
- Dengan akurasi keseluruhan sebesar 92%, presisi sebesar 87%, recall sebesar 86%, dan F1-score sebesar 86%, evaluasi model *SVM* menunjukkan bahwa model ini mampu

beroperasi dengan baik pada semua kelas sentimen.

Saran untuk penelitian lanjutan adalah sebagai berikut:

- Untuk meningkatkan kinerja model SVM, pelajari metode optimasi tambahan seperti pengoptimalan partikel swarm atau algoritma genetika.
- Untuk meningkatkan akurasi klasifikasi sentimen, gunakan teknik pembelajaran kelompok seperti *voting* atau *stacking*.
- Untuk memberikan saran yang lebih spesifik bagi pengembang, kami melakukan penelitian lebih lanjut tentang komponen yang memengaruhi perasaan pengguna terhadap aplikasi Glints.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang berpartisipasi dalam penelitian ini. Pertama, penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Selain itu, penulis mengucapkan terima kasih kepada ibu Ghiftera Dwilestari, S.I.Kom., M.Kom., yang bertugas sebagai ketua Program Studi Teknik Informatika di STMIK IKMI Cirebon, atas bantuan yang diberikan selama proses penelitian. Selain itu, penulis mengucapkan terima kasih kepada teman-teman dan keluarga atas doa, inspirasi, dan bantuan yang mereka berikan selama proses penelitian dan penulisan jurnal. Penulis juga berterima kasih kepada orang-orang yang tidak dapat disebutkan namanya yang telah membantu dan mendukung penelitian ini untuk berjalan lancar. Akhir kata, penulis berharap jurnal ilmiah ini akan membantu kemajuan ilmu pengetahuan dan teknologi dan bermanfaat bagi masyarakat.

DAFTAR PUSTAKA

- [1] B. A. Maulana, M. J. Fahmi, A. M. Imran, and N. Hidayati, "Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM)," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 375–384, Feb. 2024, doi: 10.57152/malcom.v4i2.1206.
- [2] T. Sugihartono and R. R. C. Putra, "PENERAPAN METODE SUPPORT VECTOR MACHINE DALAM CLASSIFIKASI ULASAN PENGGUNA APLIKASI MOBILE JKN," *SKANIKA Sist. Komput. dan Tek. Inform.*, vol. 7, no. 2, pp. 144–153, Jul. 2024, doi: 10.36080/skanika.v7i2.3193.
- [3] D. Angraina and A. Putri, "Analisis Sentimen Pengguna Aplikasi Google Meet Menggunakan Algoritma Support Vector Machine," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 3, pp. 472–478, 2022, doi: 10.37859/coscitech.v3i3.4260.
- [4] R. Nurfitriana Handayani, "Optimasi Algoritma Support Vector Machine Untuk Analisis Sentimen Pada Ulasan Produk Tokopedia Menggunakan Pso," *Media Inform.*, vol. 20, no. 2, pp. 97–108, 2021.
- [5] B. S. Rintyarna *et al.*, "Modelling Service Quality of Internet Service Providers during COVID-19: The Customer Perspective Based on Twitter Dataset," *Informatics*, vol. 9, no. 1, pp. 1–12, 2022, doi: 10.3390/informatics9010011.
- [6] R. Wahyudi *et al.*, "Analisis Sentimen pada review Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *J. Inform.*, vol. 8, no. 2, 2021, [Online]. Available: <http://ejournal.bsi.ac.id/ejournal/index.php/ji>
- [7] R. Merdiansah and A. Ali Ridha, "Sentiment Analysis of Indonesian X Users Regarding Electric Vehicles Using IndoBERT," *J. Ilmu Komput. dan Sist. Inf. (JIKOMSI)*, vol. 7, no. 1, pp. 221–228, 2024.
- [8] M. Alfin, Z. Abidin, and P. M. N. S. A. Basid, "Peringkasan Multi Dokumen Berbahasa Indonesia Menggunakan Metode Recurrent Neural Network," *Techno.Com*, vol. 23, no. 1, pp. 187–197, 2024, doi: 10.62411/tc.v23i1.9605.
- [9] A. R. Lubis and M. K. M. Nasution, "Twitter Data Analysis and Text Normalization in Collecting Standard Word," *J. Appl. Eng. Technol. Sci.*, vol. 4, no. 2, pp. 855–863, 2023, doi: 10.37385/jaets.v4i2.1991.
- [10] M. S. F. Fayaza and F. F. Farhath, "Towards Stopwords Identification in Tamil Text Clustering," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 12, pp. 524–529, 2021, doi: 10.14569/IJACSA.2021.0121267.
- [11] R. I. Agustin, "Komparasi Algoritma Naive Bayes Dan Svm Untuk Analisis Sentimen Twitter Korupsi Bansos Beras Masa Pandemi," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 912–918, 2024, doi: 10.23960/jitet.v12i2.4020.
- [12] I. T. Julianto, "Analisis Sentimen Terhadap Sistem Informasi Akademik Institut Teknologi Garut," *J. Algoritm.*, vol. 19, no. 1, pp. 449–456, 2022, doi: 10.33364/algoritma/v.19-1.1112.