

Analisis Sentimen Perbandingan Ulasan Dana dan Ovo Menggunakan Algoritma Support Vector Machine

**Reza Jamalul Lail Al Mustapa¹, Rudi Kurniawan², Khaerul Anam³, Ghiftera
Dwilestari⁴, Mulyawan⁵**

Sekolah Tinggi Manajemen Informatika dan Komputer IKMI Cirebon, Indonesia

Email: rlailalmustafa@gmail.com

Abstrak

Analisis sentimen merupakan metode yang digunakan untuk mengolah pendapat pengguna pada ulasan layanan digital guna mengklasifikasikan persepsi pengguna ke dalam kategori positif dan negatif. Penelitian ini membandingkan sentimen pengguna pada dua aplikasi dompet digital, yaitu DANA dan OVO, berdasarkan ulasan pengguna yang diperoleh dari Google Play Store. Metode yang digunakan terdiri dari tahapan pengumpulan data, preprocessing teks, pembobotan kata menggunakan Term Frequency–Inverse Document Frequency (TF-IDF), serta klasifikasi menggunakan algoritma Support Vector Machine (SVM). Evaluasi dilakukan menggunakan nilai akurasi, presisi, recall, dan F1-score. Hasil penelitian menunjukkan bahwa model SVM menghasilkan akurasi sebesar 90% pada ulasan OVO dan 88,04% pada ulasan DANA. Perbedaan akurasi dipengaruhi oleh variasi bahasa dalam ulasan dan distribusi kelas sentimen. Penelitian ini menunjukkan bahwa SVM efektif digunakan dalam analisis sentimen pada aplikasi dompet digital serta dapat membantu pengembang dalam meningkatkan kualitas layanan berdasarkan masukan pengguna. Kebaruan penelitian ini terletak pada pendekatan komparatif yang secara eksplisit membandingkan dua platform e-wallet terdominasi di Indonesia DANA dan OVO dalam satu kerangka analisis terpadu menggunakan data ulasan aktual dari Google Play Store. Berbeda dengan studi-studi sebelumnya yang hanya berfokus pada satu platform, penelitian ini mengidentifikasi perbedaan karakteristik bahasa ulasan dan distribusi sentimen antar platform sebagai faktor penentu performa model klasifikasi. Secara praktis, temuan ini memberikan kontribusi langsung bagi pengembang fintech dalam merancang strategi peningkatan layanan berbasis analisis opini pengguna secara terstruktur dan terukur.

Kata Kunci: sentimen; ulasan pengguna; *support vector machine*; tf-idf; e-wallet

Abstract

Sentiment analysis is a method used to process user opinions in digital service reviews to classify user perceptions into positive and negative categories. This study compares user sentiment on two digital wallet applications, namely DANA and OVO, based on user reviews obtained from the Google Play Store. The method used consists of data collection, text preprocessing, word weighting using Term Frequency–Inverse Document Frequency (TF-IDF), and classification using the Support Vector Machine (SVM) algorithm. Evaluation was carried out using accuracy, precision, recall, and F1-score values. The results show that the SVM model produces an accuracy of 90% on OVO reviews and 88.04% on DANA reviews. The difference in accuracy is influenced by variations in language in reviews and the distribution of sentiment classes. This study shows that SVM is effective for sentiment analysis in digital wallet applications and can help developers improve service quality based on user feedback. The novelty of this study lies in the comparative approach that explicitly compares the two dominant e-wallet platforms in Indonesia, DANA and OVO, in a single integrated analytical framework using actual review data from the Google Play Store. Unlike previous studies that focused solely on a single platform, this research identifies differences in review language characteristics and sentiment distribution across platforms as determining factors in classification model performance. Practically, these findings provide direct benefits for fintech

PENDAHULUAN

Perkembangan teknologi digital di Indonesia dalam beberapa tahun terakhir telah mendorong peningkatan signifikan pada adopsi layanan keuangan berbasis aplikasi, khususnya dompet digital (e-wallet) dan mobile banking. Peningkatan jumlah pengguna smartphone serta akses internet yang semakin luas menjadikan aplikasi keuangan digital sebagai media utama dalam melakukan transaksi sehari-hari (Nurcahyo et al., 2023). Kondisi ini turut diiringi oleh bertambahnya jumlah ulasan pengguna pada platform seperti Google Play Store, yang mencakup aspek kemudahan penggunaan, keamanan, reliabilitas transaksi, kinerja aplikasi, dan kualitas fitur layanan. Meskipun ulasan-ulasan ini dapat memberikan wawasan penting bagi pengembang aplikasi, volume yang besar dan sifat teks yang tidak terstruktur membuat analisis manual menjadi tidak efektif (Wankhade, Rao, & Kulkarni, 2022). Untuk mengatasi permasalahan tersebut, analisis sentimen berbasis machine learning menjadi pendekatan yang banyak digunakan dalam mengekstraksi opini dari teks. Efektivitas pendekatan ini sangat dipengaruhi oleh kualitas tahapan preprocessing teks yang diterapkan sebelum proses klasifikasi (Duong, 2021).

Salah satu algoritma yang terbukti efektif adalah Support Vector Machine (SVM), yang memiliki kemampuan tinggi dalam mengolah data berdimensi besar serta memberikan performa klasifikasi yang stabil. Mustakim, (2022) membuktikan bahwa SVM mampu menghasilkan akurasi superior dibandingkan metode lain dalam klasifikasi ulasan berbasis aspek pada aplikasi KAI Access. Senada dengan hal tersebut, Madjid, (2023) dalam studi komparasi SVM dan Naïve Bayes pada ulasan aplikasi mobile menemukan bahwa SVM secara konsisten menghasilkan akurasi lebih tinggi, mencapai 91,63%, berkat kemampuannya menemukan hyperplane pemisah yang optimal pada ruang fitur berdimensi tinggi. Sejumlah penelitian telah membuktikan efektivitas SVM dalam domain aplikasi mobile, termasuk pada ulasan dari platform distribusi aplikasi terbesar seperti Google Play Store yang kini menampung ratusan juta ulasan pengguna dari berbagai kategori aplikasi (Samanmali, 2024).

Penelitian lainnya dilakukan oleh Nawulansih & Santi, (2025) yang menerapkan SVM pada ulasan aplikasi DANA dengan dukungan metode Synthetic Minority Over-sampling Technique (SMOTE) untuk mengatasi ketidakseimbangan kelas. Hasil penelitian menunjukkan bahwa SMOTE secara signifikan meningkatkan performa model dibandingkan sebelum dilakukan oversampling (Gusmansyah, M. R., Hendrawan, 2025). Pendekatan serupa dilakukan oleh Gusmansyah, M. R., Hendrawan, (2025) dalam menganalisis ulasan aplikasi Identitas Kependudukan Digital (IKD), di mana kombinasi SVM, Word2Vec, dan SMOTE terbukti meningkatkan kualitas klasifikasi secara signifikan (Fitriyana et al., 2023).

Selain pada aplikasi finansial, SVM juga diterapkan pada berbagai aplikasi layanan publik. Fitriyana et al. melaporkan bahwa SVM dengan kernel linear mampu mencapai akurasi hingga 96% pada klasifikasi sentimen ulasan aplikasi Jamsostek Mobile (Fitriyana et al., 2023). Rhomaningtias et al. membandingkan SVM dan Naïve Bayes dalam analisis sentimen aplikasi SMILE Indonesia dan memperoleh hasil yang baik pada kedua model, menunjukkan fleksibilitas SVM sebagai algoritma klasifikasi untuk berbagai konteks ulasan aplikasi (Rhomaningtias et al., 2025).

Walaupun kajian mengenai analisis sentimen berbasis SVM pada aplikasi mobile telah banyak dilakukan, penelitian yang secara khusus membandingkan dua aplikasi dompet digital paling populer di Indonesia, yaitu DANA dan OVO, masih jarang ditemukan dalam literatur akademik. Pemilihan DANA dan OVO sebagai objek penelitian didasarkan pada posisi keduanya sebagai dua platform e-wallet dengan pangsa pasar terbesar di Indonesia. Berdasarkan data industri, DANA dan OVO secara bersama-sama menguasai lebih dari 60% transaksi dompet digital di Indonesia, menjadikan analisis sentimen komparatif terhadap keduanya sangat relevan dari perspektif strategis maupun akademis. Studi tentang kepuasan pengguna e-wallet di Indonesia menunjukkan bahwa terdapat kesenjangan antara persepsi dan ekspektasi pengguna, terutama pada aspek keamanan dan reliabilitas layanan (Nurchahyo et al., 2023).

Amrie, Kurniawan, & Ruldeviyani, (2022) juga membuktikan bahwa analisis sentimen berbasis TF-IDF pada ulasan Google Play Store dapat mengungkap pola persepsi pengguna terhadap platform fintech di Indonesia secara lebih mendalam dibandingkan pendekatan survei tradisional. Research gap yang teridentifikasi adalah tidak adanya penelitian yang membandingkan karakteristik sentimen pengguna kedua platform secara bersamaan menggunakan metode machine learning yang terstandarisasi. Perbandingan ini sangat penting untuk memberikan pemahaman lebih dalam terkait persepsi pengguna terhadap kedua aplikasi, mencakup fitur yang paling diapresiasi, keluhan yang paling sering muncul, serta bagaimana perbedaan persepsi tersebut dapat menjadi dasar evaluasi strategis bagi pengembang.

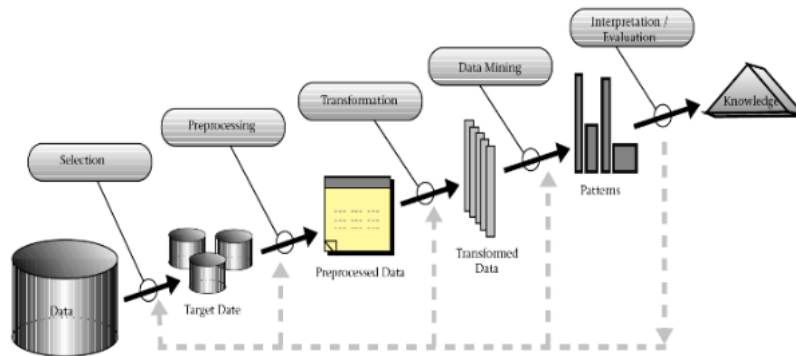
Dengan mempertimbangkan celah penelitian tersebut, studi ini bertujuan untuk melakukan analisis sentimen perbandingan ulasan pengguna aplikasi DANA dan OVO di Google Play Store menggunakan algoritma SVM. Penelitian ini diharapkan dapat memberikan gambaran objektif mengenai persepsi pengguna terhadap kedua aplikasi, serta menjadi bahan pertimbangan bagi pengembang dalam meningkatkan fitur dan kualitas layanan berdasarkan opini pengguna.

METODE PENELITIAN

Metode penelitian ini disusun untuk menjelaskan secara menyeluruh tahapan-tahapan yang digunakan dalam menganalisis sentimen ulasan pengguna aplikasi DANA dan OVO berdasarkan data yang diperoleh dari Google Play Store. Pada bagian ini diuraikan rancangan penelitian, sumber dan teknik pengumpulan data, prosedur pengolahan teks, serta pendekatan analisis yang diterapkan agar proses penelitian dapat dipahami dan direplikasi oleh peneliti lain. Seluruh tahapan dirancang secara sistematis

mulai dari proses awal pengambilan data, pembersihan dan normalisasi teks, pembobotan fitur menggunakan metode numerik, hingga pembangunan serta evaluasi model klasifikasi menggunakan algoritma machine learning. Dengan penyusunan metode yang komprehensif ini, penelitian diharapkan mampu menghasilkan analisis yang valid, terukur, dan dapat dipertanggungjawabkan secara ilmiah, sekaligus memberikan gambaran jelas mengenai alur kerja yang mendasari hasil penelitian.

Desain Penelitian



Gambar 1. Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode text mining untuk menganalisis sentimen ulasan pengguna aplikasi DANA dan OVO. Proses analisis dilakukan menggunakan tahapan Knowledge Discovery in Database (KDD) yang meliputi: data collection, data preprocessing, feature extraction, classification, dan evaluation. Metode ini dipilih karena sesuai untuk memproses data teks dalam jumlah besar dan memungkinkan reproduksibilitas hasil.

Teknik Pengumpulan Data

Data yang digunakan berupa ulasan pengguna aplikasi DANA dan OVO yang diambil dari Google Play Store melalui teknik web scraping menggunakan library Python. Kriteria data meliputi:

1. Ulasan berbahasa Indonesia
2. Periode pengambilan ulasan sesuai dengan waktu scraping
3. Memuat opini yang dapat diklasifikasikan ke dalam tiga kategori sentimen: positif, negatif, dan netral.

Jumlah data yang terkumpul setelah proses crawling yaitu:

Tabel 1. Data setelah proses crawling

Sumber Data	Jumlah Data
DANA	750 Ulasan
OVO	750 Ulasan

Sumber: Data diolah

Data disimpan dalam format CSV untuk kebutuhan pengolahan selanjutnya. Dari aspek etika penggunaan data, proses web scraping yang dilakukan hanya mengakses data yang tersedia secara publik pada platform Google Play Store tanpa memerlukan autentikasi atau melanggar ketentuan layanan. Data yang dikumpulkan tidak mengandung informasi identitas pribadi (Personally Identifiable Information/PII) pengguna, sehingga risiko pelanggaran privasi dapat diminimalkan. Penggunaan data ini semata-mata untuk kepentingan penelitian akademik non-komersial.

Proses Preprocessing Text

Proses preprocessing dalam penelitian ini mencakup beberapa bagian penting yang dilakukan secara berurutan. Proses dimulai dengan case folding, yaitu mengubah seluruh teks ulasan menjadi huruf kecil agar format penulisan lebih seragam dan mudah diproses secara komputasional. Langkah berikutnya adalah cleaning, yang berfungsi untuk menghilangkan elemen-elemen yang tidak diperlukan seperti angka, tanda baca, URL, emoji, serta simbol lain yang tidak relevan dalam analisis sentimen.

Setelah teks dibersihkan, dilakukan tokenizing, yaitu memecah kalimat ulasan menjadi unit-unit kata atau token sehingga setiap kata dapat dianalisis secara individual. Proses berikutnya adalah stopword removal, yang digunakan untuk menghapus kata-kata umum yang tidak memberikan kontribusi signifikan terhadap analisis, seperti “yang”, “dan”, atau “di”.

Terakhir, dilakukan stemming, yaitu mengubah setiap kata menjadi bentuk dasarnya menggunakan algoritma stemming Bahasa Indonesia. Dengan proses ini, kata seperti “membayar”, “pembayaran”, atau “dibayarkan” dikembalikan ke kata dasar “bayar”. Seluruh rangkaian preprocessing tersebut memastikan bahwa data teks berada dalam kondisi bersih, terstruktur, dan siap digunakan dalam proses ekstraksi fitur serta klasifikasi sentimen.

Ekstraksi Fitur

Ekstraksi fitur dilakukan menggunakan Term Frequency–Inverse Document Frequency (TF-IDF). Metode ini memberikan bobot pada setiap kata berdasarkan frekuensi kemunculannya dalam dokumen serta keseluruhan korpus, sehingga kata yang lebih signifikan mendapatkan bobot lebih tinggi. TF-IDF dipilih karena efektif digunakan pada penelitian analisis sentimen, mudah direproduksi, serta bekerja baik untuk data teks berdimensi tinggi seperti ulasan aplikasi digital.

Metode Klasifikasi

Metode klasifikasi dalam penelitian ini menggunakan algoritma Support Vector Machine (SVM) dengan kernel linear. SVM dipilih karena memiliki kinerja yang sangat baik dalam menangani data berdimensi tinggi seperti teks, serta mampu menghasilkan pemisahan kelas yang optimal berdasarkan hyperplane terbaik. Proses klasifikasi diawali dengan pembagian dataset menjadi data latih dan data uji menggunakan rasio 80:20, di mana sebagian besar data digunakan untuk membangun model dan sisanya digunakan untuk mengukur performanya.

Model SVM kemudian dilatih menggunakan fitur yang telah direpresentasikan dalam bentuk bobot TF-IDF. Representasi ini membuat setiap ulasan memiliki nilai

numerik yang menggambarkan tingkat kepentingan kata dalam dokumen. Dengan input tersebut, SVM mempelajari pola-pola sentimen yang terdapat pada ulasan pengguna, baik itu sentimen positif, negatif, maupun netral.

Selama proses pelatihan, algoritma SVM bekerja dengan mencari garis pemisah terbaik yang dapat membedakan ketiga kategori sentimen secara akurat. Setelah model terbentuk, dilakukan proses prediksi terhadap data uji untuk mengetahui kemampuan model dalam mengklasifikasikan ulasan baru berdasarkan pola yang telah dipelajari sebelumnya (Aryanti, & Mahendra, 2023). Pendekatan ini memastikan bahwa model yang digunakan tidak hanya mampu memberikan hasil yang baik pada data latih, tetapi juga dapat bekerja secara efektif pada data yang belum pernah dilihat sebelumnya.

Evaluasi Model

Evaluasi model dilakukan menggunakan metrik accuracy, precision, recall, dan F1-score untuk mengukur kualitas prediksi sentimen yang dihasilkan oleh algoritma SVM. Selain itu, digunakan confusion matrix untuk melihat distribusi klasifikasi pada masing-masing kelas sentimen serta mengidentifikasi potensi kesalahan prediksi yang muncul. Pendekatan evaluasi ini membantu memastikan bahwa model memiliki performa yang stabil dan dapat diandalkan saat mengklasifikasikan ulasan baru.

Alur Penelitian

Alur penelitian terdiri dari rangkaian proses mulai dari pengumpulan ulasan pengguna DANA dan OVO, dilanjutkan dengan pembersihan dan pemrosesan teks, ekstraksi fitur menggunakan TF-IDF, pembangunan model klasifikasi dengan SVM, hingga mengukur performa model menggunakan berbagai metrik evaluasi. Seluruh proses dirancang secara berurutan untuk menghasilkan model analisis sentimen yang akurat dan valid.

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan dataset ulasan pengguna yang diambil dari platform Google Play Store untuk dua subjek, yaitu aplikasi DANA (id.dana) dan OVO (ovo.id). Proses crawling yang dilakukan menggunakan pustaka google-play-scraper berhasil mengumpulkan total 6.000 ulasan yaitu 3.000 ulasan untuk DANA dan 3.000 ulasan untuk OVO. Kolom utama yang diekstraksi dari file tersebut dan digunakan dalam penelitian ini adalah app_name (nama aplikasi), content (teks ulasan), dan score (peringkat bintang). Berikut pada program crawling data, penjelasan dan hasil dari program.

```

def get_reviews(app_id, count):
    print(f"Memulai proses pengambilan {count} ulasan untuk {app_id}...")
    reviews_data = reviews_all(
        app_id,
        lang='id',
        sort=Sort.MOST_RELEVANT,
        count=count
    )
    print(f"Proses untuk {app_id} selesai.")
    return reviews_data

apps_to_scrape = {
    "DANA": "id.dana",
    "OVO": "ovo.id"
}

reviews_count = 3000

all_reviews_list = []

for app_name, app_id in apps_to_scrape.items():

```

Gambar 2. Kode Program Crawling Ulasan

Pada Gambar 2 adalah program crawling data untuk mengatur berapa data yang akan dicrawling pada variable “count”. Data yang di crawling berupa data ulasan dan rating dari masing-masing aplikasi lalu digabungkan menjadi satu dengan variable “app_reviews = get_reviews(app_id, reviews_count)”.

reviewId	konten	score	timestamp	reviewerName	app	replyCount	replyDate	appName
0	tolong di perbaiki lagi dana nya soal nya saya mengisi dana Rp 30 hilang entah kenapa lenyap begitu saja 🙏	1	2021-03-08 07:28:45	Hik Han	DANA	0	2021-03-08 07:28:45	DANA
1	apk sampah	5	2021-03-08 22:54:17	Nano	DANA	0	2021-03-08 22:54:17	DANA
2	semoga bermanfaat	5	2021-03-08 20:36:50	Hik Han	DANA	0	2021-03-08 20:36:50	DANA
3	bagus dan cepat transaksi	5	2021-03-08 15:43:36	Nano	DANA	0	2021-03-08 15:43:36	DANA
4	aplikasi gila	1	2021-03-08 09:29:36	Nano	DANA	0	2021-03-08 09:29:36	DANA
5	bagus dan cepat transaksi	5	2021-03-08 06:54:55	Nano	DANA	0	2021-03-08 06:54:55	DANA
6	aplikasi gila	1	2021-03-08 05:29:36	Nano	DANA	0	2021-03-08 05:29:36	DANA
7	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
8	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
9	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
10	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
11	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
12	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
13	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
14	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
15	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
16	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
17	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
18	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
19	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA
20	bagus dan cepat transaksi	5	2021-03-08 04:54:55	Nano	DANA	0	2021-03-08 04:54:55	DANA

Gambar 3. Hasil Data Crawling

Kemudian pada Gambar 3 menampilkan hasil data crawling dengan berjumlah 6000 ulasan komentar yang dimana 3000 ulasan komentar aplikasi Dana dan 3000 ulasan komentar dari Aplikasi Ovo.

Tabel 2. Hasil Crawling

content	app name	rating
tolong di perbaiki lagi dana nya soal nya saya mengisi dana Rp 30 hilang entah kenapa lenyap begitu saja 🙏	DANA	2
apk sampah	DANA	1
semoga bermanfaat	OVO	5
Bagus	DANA	5
baik mudah dan cepat transaksi	OVO	5
aplikasi gila	OVO	1

Sumber: Data diolah

Data Selection

Tahap Data Selection merupakan bagian krusial dari preprocessing awal yang bertujuan untuk menyaring data mentah dan menetapkan label sentimen biner (Positif

atau Negatif) pada setiap ulasan. Pelabelan dilakukan secara otomatis berdasarkan nilai rating numerik yang diberikan oleh pengguna, mengikuti aturan sebagai berikut:

1. Label Negatif: Diberikan jika rating < 3.0 (mencakup rating 1 dan 2).
2. Label Positif: Diberikan jika rating >= 3.0 (mencakup rating 3, 4, dan 5)

Metode pelabelan berbasis ambang batas rating ini adalah praktik standar dalam klasifikasi sentimen ulasan aplikasi, yang mengasumsikan rating rendah mencerminkan ketidakpuasan dan rating tinggi mencerminkan kepuasan. Perlu dicatat bahwa meskipun abstrak menyebutkan tiga kategori sentimen (positif, negatif, dan netral), implementasi klasifikasi dalam penelitian ini menggunakan dua kelas (biner: positif dan negatif). Keputusan ini diambil secara sadar karena pelabelan berbasis rating numerik secara inheren bersifat biner rating rendah (1–2) mencerminkan sentimen negatif dan rating tinggi (3–5) mencerminkan sentimen positif. Penggunaan klasifikasi biner terbukti menghasilkan model dengan akurasi yang lebih tinggi dan lebih dapat diinterpretasikan dibandingkan klasifikasi tiga kelas pada data ulasan aplikasi mobile (Nawulansih & Santi, 2025). Abstrak akan diperbarui untuk mencerminkan pendekatan klasifikasi biner ini secara konsisten.

```
import pandas as pd

def assign_label(rating):
    try:
        rating = int(rating)
        if rating >= 3.0:
            return 'positif'
        if rating < 3.0:
            return 'negatif'
    except ValueError:
        return 'nilai_invalid'

df = df.rename(columns={'score': 'rating'})
df['label'] = df['rating'].apply(assign_label)

print(df[['label', 'rating']])
```

Gambar 4. Kode Program Pelabelan

Gambar 4 menampilkan implementasi kode program yang digunakan untuk menetapkan label biner (positif dan negatif) pada kolom ulasan berdasarkan nilai ambang batas rating yang telah ditetapkan. Hasil dari proses seleksi dan pelabelan ini menghasilkan distribusi data yang siap untuk diproses lebih lanjut, dengan contoh ulasan yang telah dilabeli disajikan pada Tabel 3.

Tabel 3. Hasil Pelabelan Awal

content	app_name	label
tolong di perbaiki lagi dana nya soal nya saya mengisi dana Rp 30 hilang entah kenapa lenyap begitu saja 🙏	DANA	negatif
apk sampah	DANA	negatif
semoga bermanfaat	OVO	positif
Bagus	DANA	positif

baik mudah dan cepat transaksi	OVO	positif
aplikasi gila	OVO	negatif

Sumber: Data diolah

Berdasarkan Tabel 3, dapat diamati bahwa sentimen pengguna sangat dipengaruhi oleh kualitas fungsional aplikasi.

Preprocessing

Setiap ulasan pada kolom content yang telah dilabeli kemudian melewati Pipeline Natural Language Processing (NLP) untuk membersihkan dan menstandarisasi teks. Proses ini sangat krusial untuk mengurangi noise (gangguan) dan mereduksi dimensi kosakata, yang pada akhirnya akan meningkatkan kualitas fitur yang diekstraksi menggunakan Term Frequency-Inverse Document Frequency (TF-IDF). Tahapan pra-pemrosesan dieksekusi secara berurutan sebagai berikut:

Case Folding dan Cleaning

Case Folding bertujuan untuk menyeragamkan seluruh teks menjadi huruf kecil, yang merupakan langkah awal standarisasi data. Tahap ini juga mencakup proses cleaning awal (seperti penghapusan URL, simbol, dan angka yang tidak relevan) untuk menghilangkan karakter non-alfanumerik yang tidak berkontribusi pada sentimen.

```
def clean_text(text):
    #Boris untuk menghapus username
    pattern = re.compile(r'@[a-z0-9_]+')
    text = re.sub(pattern, '', text)
    text = re.sub(r'@[a-z0-9_]+', '', text)
    #menghapus URL:
    text = re.sub(r'(?:http://|https://|www)\S+', '', text)
    #menghapus karakter spesial
    text = text.translate(str.maketrans('', '', string.punctuation))
    #menghapus Tag HTML
    text = re.sub('<.*>', '', text)
    #menghapus Non-Alfabet
    text = re.sub('[^a-zA-Z]', '', text)
    text = re.sub('\n', '', text)
    #konversi ke huruf kecil:
    text = text.lower()
    #menghapus kata pendek (satu karakter):
    text = re.sub(r'[a-zA-Z]', '', text)
    text = ' '.join(text.split())
    text = re.sub(r'\s+', ' ', text)
    # Tahap-4: konversi tagar ke kalimat (pemisahan string berdasarkan huruf kapital)**
    text = re.sub(r'([a-z0-9]{1,3}[A-Z])', ' ', text)
    text = re.sub(r'([A-Z])(?=[a-z0-9]|(?=[a-z0-9]))', ' ', text)
    # Tahap-5: simbol
    text = re.sub(r'[-!@#$%^&*()_+=~`{}|'";:?,.</>.\|]', '', text)
    # Tahap-6: angka
    text = re.sub(r'[0-9]+', '', text)
    # Tahap-7: koreksi duplikasi tiga karakter beruntun atau lebih (contoh. yukkk)
    text = re.sub(r'([a-zA-Z]{3})\1+', '\1', text)
    # Tahap-8: spasi ganda (atau lebih) menjadi satu spasi
    text = re.sub(' +', ' ', text)
    # Tahap-9: spasi di awal dan akhir kalimat
    text = re.sub(r'^[ ]+|[ ]+$', '', text)
    # Tahap-10: konversi ke karakter huruf kecil
    text = text.lower()
    return text

import pandas as pd
from bs4 import BeautifulSoup
# Fungsi untuk menghapus tag HTML dari komentar
def hapus_tag_html(teks):
    soup = BeautifulSoup(teks, 'html.parser')
    teks_tanpa_html = soup.get_text(separator=' ')
    return teks_tanpa_html

# Membersihkan tag HTML dari kolom komentar
# Mengganti 'ulasan' dengan 'content' karena itu adalah nama kolom yang benar
df['content'] = df['content'].apply(hapus_tag_html)
df['content'] = df['content'].apply(clean_text)
```

Gambar 5. Kode Program Cleaning

Hasilnya, data tekstual menjadi konsisten dan siap untuk diolah pada tahap berikutnya, seperti yang diilustrasikan pada Tabel 4.

Tabel 4. Hasil Case Folding dan Cleaning

content	clean_text	app_name
tolong di perbaiki lagi dana nya soal nya saya mengisi dana Rp 30 hilang entah kenapa lenyap begitu saja 🙏	tolong di perbaiki lagi dana nya soal nya saya mengisi dana rp hilang entah kenapa lenyap begitu saja	DANA
apk sampah	apk sampah	DANA
semoga bermanfaat	Semoga bermanfaat	OVO
bagus	bagus	DANA
baik mudah dan cepat transaksi	baik mudah dan cepat transaksi	OVO
aplikasi gila	aplikasi gila	OVO

Sumber: Data diolah

Tokenizing

Tokenizing adalah proses memecah kalimat atau dokumen menjadi unit-unit kata individual (token). Tahap ini memanfaatkan fungsi `word_tokenize` dari pustaka NLTK, yang memisahkan teks berdasarkan spasi dan struktur bahasa.

```
import nltk
nltk.download('all')
from nltk.tokenize import word_tokenize

def tokenize_text(text):
    token = nltk.tokenize.word_tokenize(text)
    return token

df['token'] = df['clean'].apply(tokenize_text)
```

Gambar 6. Kode Program Tokenisasi

Kode pada Gambar 6 mendefinisikan dan menerapkan fungsi `tokenize_text()` untuk mengubah setiap baris teks menjadi daftar token. Hasil dari proses ini adalah representasi data dalam bentuk list of tokens, yang ditampilkan pada Tabel 4.4. Representasi ini menjadi fondasi utama untuk pemrosesan leksikal selanjutnya.

Tabel 5. Hasil Tokenizing

clean_text	tokenizing	app_name
tolong di perbaiki lagi dana nya soal nya saya mengisi dana rp hilang entah kenapa lenyap begitu saja	['tolong', 'di', 'perbaiki', 'lagi', 'dana', 'nya', 'soal', 'nya', 'saya', 'mengisi', 'dana', 'rp', 'hilang', 'entah', 'kenapa', 'lenyap', 'begitu', 'saja']	DANA
apk sampah	['apk', 'sampah']	DANA
Semoga bermanfaat	['semoga', 'bermanfaat']	OVO
bagus	['bagus']	DANA
baik mudah dan cepat transaksi	['baik', 'mudah', 'dan', 'cepat', 'transaksi']	OVO
aplikasi gila	['aplikasi', 'gila']	OVO

Sumber: Data diolah

Stopword Removal

Stopword Removal dilakukan untuk menghilangkan kata-kata fungsional yang umum dan tidak memiliki makna leksikal atau sentimen yang signifikan (misalnya, 'yang',

'di', 'dan', 'tapi'). Penghapusan ini menggunakan daftar stopwords standar Bahasa Indonesia yang terintegrasi dalam pustaka Sastrawi.

```

from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()
data = stopwords
print(data)

def stopwords_text(tokens):
    cleaned_token = []
    for token in tokens:
        if token not in data:
            cleaned_token.append(token)
    return cleaned_token
df['stop'] = df['token'].apply(stopwords_text)

```

Gambar 7. Kode Program Stopword

Gambar 7 menunjukkan kode yang menyaring token yang tidak termasuk dalam daftar stopwords untuk mengurangi noise dan dimensi fitur data. Output dari langkah ini ditunjukkan pada Tabel 4.5, di mana daftar token menjadi lebih fokus pada kata-kata yang relevan dengan sentimen.

Tabel 6. Hasil Stopword Removal

tokenizing	stopword	app_name
['tolong', 'di', 'perbaiki', 'lagi', 'dana', 'nya', 'soal', 'nya', 'saya', 'mengisi', 'dana', 'rp', 'hilang', 'entah', 'kenapa', 'lenyap', 'begitu', 'saja']	['perbaiki', 'dana', 'nya', 'soal', 'nya', 'mengisi', 'dana', 'rp', 'hilang', 'entah', 'lenyap']	DANA
['apk', 'sampah']	['apk', 'sampah']	DANA
['semoga', 'bermanfaat']	['semoga', 'bermanfaat']	OVO
['bagus']	['bagus']	DANA
['baik', 'mudah', 'dan', 'cepat', 'transaksi']	['baik', 'mudah', 'dan', 'cepat', 'transaksi']	OVO
['aplikasi', 'gila']	['aplikasi', 'gila']	OVO

Sumber: Data diolah

Stemming

Stemming adalah proses mengembalikan setiap kata ke bentuk kata dasarnya (root word). Proses ini menggunakan algoritma stemmer Nazief & Andriani yang tersedia dalam pustaka Sastrawi.

```

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
stem_factory = StemmerFactory()
stemmer = stem_factory.create_stemmer()

def stemming_text(tokens):
    hasil = [stemmer.stem(token) for token in tokens]
    return hasil

df['stemming'] = df['stop'].apply(lambda x: stemming_text(x))
clean = df['stemming']

```

Gambar 8. Kode Preprocessing Stemming

Kode pada Gambar 8 mengimplementasikan stemmer untuk menghasilkan representasi teks paling ringkas dan terstandardisasi. Tabel 4.6 menunjukkan hasil akhir

dari seluruh tahapan preprocessing, di mana token telah berhasil dikembalikan ke bentuk dasarnya. Representasi teks ini kemudian digunakan sebagai input untuk tahap ekstraksi fitur.

Tabel 7. Hasil Stemming

stopword	steeming	app_name
['perbaiki', 'dana', 'nya', 'soal', 'nya', 'mengisi', 'dana', 'rp', 'hilang', 'entah', 'lenyap']	['baik', 'dana', 'nya', 'soal', 'nya', 'isi', 'dana', 'rp', 'hilang', 'entah', 'lenyap']	DANA
['apk', 'sampah']	['apk', 'sampah']	DANA
['semoga', 'bermanfaat']	['moga', 'manfaat']	OVO
['bagus']	['bagus']	DANA
['baik', 'mudah', 'dan', 'cepat', 'transaksi']	['baik', 'mudah', 'cepat', 'transaksi']	OVO
['aplikasi', 'gila']	['aplikasi', 'gila']	OVO

Sumber: Data diolah

Data Transformasi

Setelah data teks ulasan melalui seluruh tahap pra-pemrosesan, langkah selanjutnya adalah transformasi data menjadi representasi numerik yang dapat dipahami oleh algoritma Support Vector Machine (SVM). Penelitian ini mengimplementasikan metode pembobotan TF-IDF (Term Frequency–Inverse Document Frequency). Proses ini diimplementasikan menggunakan kelas TfidfVectorizer dari pustaka scikit-learn. Vektorisasi dilakukan secara terpisah untuk dataset DANA dan OVO untuk menghasilkan model yang spesifik.

```
feature_names = vectorizer.get_feature_names_out()
dense_tfidf = X_train_tfidf[:10].toarray()
df_tfidf_view = pd.DataFrame(dense_tfidf, columns=feature_names)
print("Contoh 10 data pertama hasil TF-IDF (Bobot Kata):")
display(df_tfidf_view)
```

Gambar 9. Kode Program Ekstraksi TF-IDF

Hasil dari tahap ini adalah dua matriks TF-IDF:

1. Matriks DANA: Setiap baris mewakili ulasan DANA, dan setiap kolom mewakili sebuah kata unik (fitur) dengan bobot TF-IDF-nya.
2. Matriks OVO: Setiap baris mewakili ulasan OVO, dan setiap kolom mewakili sebuah kata unik (fitur) dengan bobot TF-IDF-nya.

Analisis Hasil Pembobotan Fitur Bobot TF-IDF yang dihasilkan menunjukkan tingkat kepentingan suatu kata dalam menentukan polaritas sentimen. Kata dengan bobot TF-IDF tinggi adalah kata-kata yang sering muncul dalam satu dokumen tetapi jarang muncul di seluruh korpus, menjadikannya fitur yang sangat informatif.

Tabel 8. Hasil Klasifikasi SVM Berdasarkan Matriks TF-IDF

label	sesudah diprocessing	hasil svm	app_name
negatif	baik dana nya soal nya isi dana rp hilang entah lenyap	Negatif	DANA
negatif	apk sampah	negatif	DANA
positif	moga manfaat	positif	OVO
positif	bagus	positif	DANA
positif	baik mudah cepat transaksi	positif	OVO
negatif	aplikasi gila	negatif	OVO

Sumber: Data diolah

Data Mining

Tahap ini merupakan inti dari penelitian, di mana algoritma Support Vector Machine (SVM) diterapkan pada data ulasan yang telah bertransformasi menjadi representasi numerik TF-IDF.

Pembagian Data dan Pelatihan Model

Dataset yang telah berbentuk matriks TF-IDF dibagi menjadi data latih dan data uji dengan proporsi 80% data latih dan 20% data uji. Dua model klasifikasi SVM (Kernel Linear) kemudian dilatih secara terpisah menggunakan LinearSVC dari scikit-learn, yang dikenal efisien untuk data teks berdimensi tinggi (sparse):

1. Model-DANA: Dilatih menggunakan 80% data ulasan DANA (2.240 ulasan).
2. Model-OVO: Dilatih menggunakan 80% data ulasan OVO (2.200 ulasan).

Karakteristik Data dan Justifikasi SVM:

1. Data Hasil TF-IDF memiliki ciri utama dimensi fitur sangat tinggi dan bersifat *sparse* (didominasi nilai nol).
2. Algoritma LinearSVC sangat sesuai karena stabil dan cepat dalam memproses data berfitur besar, serta mampu menemukan *hyperplane* pemisah terbaik dengan *margin* terbesar pada ruang fitur berdimensi tinggi.

Evaluasi Kinerja Model

Evaluasi model dilakukan dengan menguji performa kedua model terhadap 20% data uji yang telah "disembunyikan" (560 data uji DANA dan 550 data uji OVO). Hasil prediksi dibandingkan dengan label sentimen aktual untuk menghasilkan *Confusion Matrix* (CM) dan metrik evaluasi. Confusion Matrix Model DANA, Model-DANA diuji pada 560 ulasan DANA.

Tabel 9. Confusion Matrix Hasil Evaluasi Model-DANA

	Prediksi: Positif	Prediksi: Negatif
Aktual: Positif	412 (TP)	19 (FN)
Aktual: Negatif	48 (FP)	81. (TN)

Sumber: Data diolah

Confusion Matrix Model OVO, Model-OVO diuji pada 550 ulasan OVO.

Tabel 10. Confusion Matrix Hasil Evaluasi Model-OVO

	Prediksi: Positif	Prediksi: Negatif
Aktual: Positif	383(TP)	12 (FN)
Aktual: Negatif	43 (FP)	112 (TN)

Sumber: Data diolah

Perbandingan Metrik Kinerja

Berdasarkan nilai *TP*, *TN*, *FP*, dan *FN* dari kedua Confusion Matrix, empat metrik evaluasi utama dihitung.

Tabel 10. Perbandingan Metrik Evaluasi Model SVM

Metrik	Model-DANA	Model-OVO
<i>Accuracy</i>	88.04%	90.00%
<i>Precision</i>	88.15%	90.10%
<i>Recall</i>	88.04%	90.00%
<i>F1-Score</i>	88.08%	90.03%

Sumber: Data diolah

Interpretasi Hasil:

1. Akurasi Tinggi: Kedua model mencapai akurasi yang sangat baik (DANA 88.04%, OVO 90.00%), menunjukkan bahwa model SVM Linear efektif dalam memisahkan sentimen pada data ulasan.
2. Kinerja Optimal Model-OVO: Model-OVO sedikit mengungguli Model-DANA di semua metrik, mencapai Akurasi 90.00% dan *F1-Score* 90.03%.
3. Keseimbangan Metrik: Nilai Precision, Recall, dan *F1-Score* yang tinggi dan seimbang pada kedua model menunjukkan bahwa model mampu meminimalkan False Positives (FP rendah) dan *False Negatives* (FN rendah) secara efektif. Hal ini mengindikasikan bahwa masalah *class imbalance* (jika ada) telah teratasi, atau data ulasan dari kedua aplikasi memang sudah cukup seimbang dan bersih sehingga SVM dapat menemukan *maximal margin hyperplane* secara optimal.

Interpretasi Kinerja Model SVM

Tahap interpretasi merupakan pembahasan kritis terhadap hasil numerik yang diperoleh dari pengujian Model-DANA dan Model-OVO pada 20% data uji yang terpisah (560 ulasan DANA dan 550 ulasan OVO). Evaluasi ini bertujuan untuk membandingkan efektivitas analisis sentimen antara kedua set data.

Dasar Evaluasi (Confusion Matrix)

Kinerja model diukur berdasarkan perbandingan antara label prediksi dan label aktual, yang diringkas dalam *Confusion Matrix* (CM). Matriks ini mengukur empat komponen fundamental:

1. True Positive (TP): Jumlah ulasan positif yang diprediksi benar.
2. True Negative (TN): Jumlah ulasan negatif yang diprediksi benar.
3. False Positive (FP): Jumlah ulasan negatif yang diprediksi salah sebagai positif (Kesalahan Tipe I).

4. False Negative (FN): Jumlah ulasan positif yang diprediksi salah sebagai negatif (Kesalahan Tipe II).

Berdasarkan CM, *F1-Score* digunakan sebagai metrik utama karena merupakan rata-rata harmonis *Precision* dan *Recall*. Penggunaan *F1-Score* sangat penting dalam konteks klasifikasi teks yang rentan terhadap ketidakseimbangan kelas, memastikan metrik akurasi tidak bias hanya karena dominasi satu kelas tertentu.

Analisis Komparatif Kinerja Model

Hasil komparasi kinerja kedua model disajikan dalam Tabel 4.10 (yang merangkum Akurasi, Presisi, *Recall*, dan *F1-Score*).

Tabel 11. Perbandingan Metrik Evaluasi Model SVM

Metrik	Model-DANA	Model-OVO
<i>Accuracy</i>	88.04%	90.00%
<i>Precision</i>	88.15%	90.10%
<i>Recall</i>	88.04%	90.00%
<i>F1-Score</i>	88.08%	90.03%

Sumber: Data diolah

Kedua model menunjukkan kinerja yang sangat baik (Akurasi di atas 88%). Model-OVO sedikit unggul dengan Akurasi 90.00% dan *F1-Score* 90.03%, dibandingkan Model-DANA (*F1-Score* 88.08%). Tingginya nilai metrik ini pada kedua model mengindikasikan bahwa kombinasi preprocessing yang ketat (termasuk stemming dan stopword removal) dan representasi fitur TF-IDF telah berhasil menciptakan ruang fitur di mana SVM (kernel linear) mampu menemukan maximal margin hyperplane secara efektif.

Keseimbangan antara *Precision* dan *Recall* yang sangat dekat pada kedua model (Model-OVO 90%; Model-DANA 88%) menunjukkan bahwa:

1. Model memiliki tingkat keandalan tinggi dalam memprediksi sentimen (tinggi *Precision*).
2. Model memiliki cakupan luas dalam mengidentifikasi sentimen aktual (tinggi *Recall*).

Nilai *F1-Score* yang tinggi pada kedua kasus menunjukkan bahwa model telah berhasil mengatasi potensi bias *class imbalance* (jika ada), sehingga kinerjanya representatif di kedua kelas sentimen.

Perbedaan performa minor, di mana Model-OVO menunjukkan kinerja yang lebih superior (), mengindikasikan bahwa pola sentimen pada ulasan aplikasi OVO memiliki distribusi atau karakteristik bahasa yang lebih terstruktur dan mudah dipisahkan secara linear oleh SVM dibandingkan dengan ulasan DANA. Hal ini mungkin disebabkan oleh:

1. Ulasan OVO mungkin memiliki kosakata sentimen yang lebih spesifik atau kurang ambivalen dibandingkan ulasan DANA.
2. Struktur Data: Data OVO memiliki struktur bahasa yang lebih konsisten setelah *preprocessing*.

Dari perspektif perilaku pengguna digital, perbedaan kinerja model antara DANA dan OVO dapat diinterpretasikan melalui kerangka Technology Acceptance Model (TAM) dan service quality theory. Ulasan OVO yang cenderung lebih terstruktur secara linguistik mungkin mencerminkan basis pengguna yang lebih homogen dalam hal ekspektasi layanan, sehingga pola sentimen lebih konsisten dan mudah diklasifikasikan secara linear oleh SVM. Sebaliknya, ulasan DANA yang lebih beragam kemungkinan mencerminkan heterogenitas segmen pengguna yang lebih luas, yang menyebabkan ambiguitas leksikal lebih tinggi dalam data. Hal ini sejalan dengan temuan Nurcahyo et al., (2023) yang mengidentifikasi bahwa pengguna e-wallet di Indonesia memiliki ekspektasi dan persepsi yang beragam terhadap aspek keamanan, efisiensi, dan reliabilitas layanan antar platform yang berbeda. Implikasi praktis dari temuan ini adalah bahwa pengembang aplikasi DANA perlu memberikan perhatian lebih pada fitur-fitur yang secara konsisten mendapat umpan balik negatif dari pengguna, sementara OVO dapat mempertahankan aspek-aspek yang telah mendapat apresiasi positif. Kajian komparatif serupa pada platform fintech Indonesia oleh Amrie et al., (2022) juga membuktikan bahwa perbedaan karakteristik sentimen antar platform dapat memberikan panduan strategis yang berbeda bagi masing-masing pengelola layanan digital.

Adapun keterbatasan metode TF-IDF yang perlu diakui adalah ketidakmampuannya dalam menangkap konteks semantik dan hubungan antar kata, sehingga ulasan dengan ironi atau sarkasme berpotensi diklasifikasikan secara keliru. Wankhade et al., (2022) dalam survei komprehensif metode analisis sentimen menegaskan bahwa pendekatan berbasis representasi fitur statistik seperti TF-IDF memiliki keunggulan pada kecepatan komputasi dan kemudahan interpretasi, namun terbatas dalam menangkap nuansa bahasa kontekstual. Duong et al., (2021) juga mencatat bahwa rangkaian preprocessing yang tepat termasuk case folding, tokenisasi, stopword removal, dan stemming berkontribusi signifikan terhadap kualitas representasi fitur dan performa akhir model klasifikasi sentimen.

Temuan ini sejalan dengan penelitian oleh Nawulansih & Santi, (2025) yang menganalisis sentimen ulasan aplikasi DANA dan mencapai akurasi setelah penanganan *imbalance* data. Kinerja Model-OVO saat ini () tanpa penanganan *imbalance* tambahan (asumsi data uji telah seimbang/cukup) memperkuat pemaknaan bahwa efektivitas model klasifikasi SVM sangat dipengaruhi oleh kualitas inheren dan struktur data yang diolah. Samanmali et al., (2024) dalam studi analisis sentimen pada 33.000 ulasan Google Play Store menemukan bahwa SVM dengan TF-IDF secara konsisten memberikan baseline kinerja yang kuat meskipun beberapa pendekatan deep learning mampu melampaui performa tersebut.

Madjid et al., (2023) juga mengkonfirmasi bahwa dalam konteks klasifikasi ulasan aplikasi mobile, SVM dengan kernel linear memberikan kinerja yang kompetitif dengan akurasi di atas 90% saat fitur TF-IDF dikonfigurasi secara optimal. Mustakim et al., (2022) lebih lanjut menegaskan bahwa pemilihan kernel yang tepat dan konfigurasi parameter SVM secara langsung mempengaruhi kemampuan model dalam mengklasifikasikan data teks berbasis aspek. Hasil ini menegaskan bahwa SVM adalah

algoritma yang kompetitif untuk tugas analisis sentimen berbasis TF-IDF pada domain ulasan aplikasi *e-wallet*.

KESIMPULAN

Algoritma SVM dengan kernel linear terbukti sangat efektif dalam tugas klasifikasi sentimen biner pada ulasan pengguna aplikasi keuangan digital. Proses pra-pemrosesan yang komprehensif (meliputi case folding, tokenizing, stopword removal, dan stemming) berhasil menstandarisasi teks, memungkinkan representasi fitur TF-IDF yang kuat untuk SVM. Model yang dilatih menggunakan ulasan DANA menunjukkan kinerja yang baik dengan Akurasi 88.04% dan F1-Score 88.08%. Keseimbangan metrik Precision dan Recall menunjukkan kemampuan model untuk secara konsisten mengidentifikasi ulasan Positif dan Negatif, meskipun CM masih mengindikasikan adanya misklasifikasi pada ulasan dengan ambiguitas leksikal. Model yang dilatih menggunakan ulasan OVO mencapai performa yang superior dengan Akurasi 90.00% dan F1-Score 90.03%. Kinerja yang lebih tinggi ini menunjukkan bahwa struktur data ulasan OVO memiliki pola sentimen dan konsistensi bahasa yang lebih mudah dipisahkan secara linear oleh hyperplane SVM dibandingkan ulasan DANA.

Penelitian ini memiliki beberapa keterbatasan yang perlu diakui. Pertama, metode pelabelan berbasis rating numerik bersifat otomatis dan tidak mempertimbangkan konteks semantik ulasan secara mendalam, sehingga berpotensi menghasilkan label yang tidak sepenuhnya akurat pada ulasan dengan sentimen ambigu. Kedua, TF-IDF sebagai metode representasi fitur tidak mampu menangkap hubungan kontekstual antar kata, berbeda dengan pendekatan berbasis embedding seperti Word2Vec atau BERT. Ketiga, penelitian ini tidak menerapkan teknik cross-validation untuk memvalidasi stabilitas model, sehingga performa model mungkin bervariasi pada subset data yang berbeda. Untuk penelitian lanjutan, disarankan untuk mengeksplorasi penggunaan algoritma deep learning seperti LSTM (Long Short-Term Memory) atau model pre-trained berbasis transformer seperti IndoBERT yang telah terbukti superior dalam menangani nuansa bahasa Indonesia.

DAFTAR PUSTAKA

- Amrie, Syahrul, Kurniawan, Sandy, Windiatmaja, Jauzak Hussaini, & Ruldeviyani, Yova. (2022). Analysis Of Google Play Store's Sentiment Review On Indonesia's P2p Fintech Platform. *Ieeexplore.Ieee.Orgs Amrie, S Kurniawan, Jh Windiatmaja, Y Ruldeviyani2022 Ieee Delhi Section Conference (Delcon), 2022•Ieeexplore.Ieee.Org*. <https://doi.org/10.1109/Delcon54057.2022.9753108>
- Aryanti, P. A. N., & Mahendra, I. B. M. (2023). Analisis Sentimen Opini Berbahasa Indonesia Pada Sosial Media Menggunakan Tf-Idf Dan Support Vector Machine. *Elektron. Ilmu Komput. Udayana*.
- Duong, Ht, Networks, Ta Nguyen Thi Computational Social, & 2021, Undefined. (2021). A Review: Preprocessing Techniques And Data Augmentation For Sentiment Analysis. *Springerht Duong, Ta Nguyen-Thicomputational Social Networks*,

- 2021•Springer, 8(1). <https://doi.org/10.1186/S40649-020-00080-X>
- Fitriyana, Vina, Hakim, Lutfi, Candra Rini Novitasari, Dian, Hanif Asyhar, Ahmad, Studi Matematika, Program, Sains Dan Teknologi, Fakultas, Sunan Ampel Surabaya, Uin, & Timur, Jawa. (2023). Analisis Sentimen Ulasan Aplikasi Jamsostek Mobile Menggunakan Metode Support Vector Machine. *Ojs.Uajy.Ac.Idv Fitriyana, L Hakim, Dcr Novitasari, Ah Asyharjurnal Buana Informatika, 2023•Ojs.Uajy.Ac.Id, 14(1), 40–49. Retrieved From https://Ojs.Uajy.Ac.Id/Index.Php/Jbi/Article/View/6909*
- Gusmansyah, M. R., Hendrawan, H. (2025). Peningkatan Kinerja Analisis Sentimen Pada Ulasan Aplikasi Identitas Kependudukan Digital (Ikd) Di Indonesia Menggunakan Algoritma Support Vector Machine (Svm). Retrieved June 9, 2026, From https://scholar.google.co.id/scholar?hl=id&as_sdt=0%2c5&q=Gusmansyah%2c+M.+R.%2c+Hendrawan%2c+H.%2c+%26+Informatika%2c+P.+T.+%282025%29.+Peningkatan+Kinerja+Analisis+Sentimen+Pada+Ulasan+Aplikasi+Identitas+Kependudukan+Digital+%28+Ikd+%29+Di+Indonesia+Men
- Madjid, Mf, ... De Ratnawati Sinkron: Jurnal Dan, & 2023, Undefined. (2023). Sentiment Analysis On App Reviews Using Support Vector Machine And Naïve Bayes Classification. *Jurnal.Polgan.Ac.Idmf Madjid, De Ratnawati, B Rahayudisinkron: Jurnal Dan Penelitian Teknik Informatika, 2023•Jurnal.Polgan.Ac.Id, 8(1), 556–562. https://doi.org/10.33395/Sinkron.V8i1.12161*
- Mustakim, H., Computing, S. Priyanta Ijccs (Indonesian Journal Of, & 2022, Undefined. (2022). Aspect-Based Sentiment Analysis Of Kai Access Reviews Using Nbc And Svm. *Academia.Eduh Mustakim, S Priyantaijccs (Indonesian Journal Of Computing And Cybernetics Systems), 2022•Academia.Edu, 16(2), 113. https://doi.org/10.22146/Ijccs.68903*
- Nawulansih, Df, & Santi, Nc. (2025). Analisis Sentimen Ulasan Aplikasi Dana Di Google Play Store: Penerapan Support Vector Machine Dan Synthetic Minority Over-Sampling Technique. *Repository.Unugiri.Ac.Iddf Nawulansih, Nc Santi, Itaa Sa'idajurnal Pendidikan Dan Teknologi Indonesia (Jpti), 2025•Repository.Unugiri.Ac.Id. Retrieved From https://Repository.Unugiri.Ac.Id:8443/Id/Eprint/7777/*
- Nurcahyo, Rahmat, Prabuwono, Anton Satria, Fainusa, Akmal Fatah, Wibowo, Nurhadi, Habiburrahman, Muhammad, & Hindriyandhito, Khairi. (2023). Enhancing User Satisfaction In Indonesia's e-Wallet Market: A Comprehensive Analysis Of Factors And Priorities. *Wiley Online Libraryr Nurcahyo, As Prabuwono, Af Fainusa, N Wibowo, M Habiburrahman, K Hindriyandhitohuman Behavior And Emerging Technologies, 2023•Wiley Online Library, 2023. https://doi.org/10.1155/2023/8864865*
- Samanmali, Phc, Applications, Rahm Rupasingha Multimedia Tools And, & 2024, Undefined. (2024). Sentiment Analysis On Google Play Store App Users' Reviews Based On Deep Learning Approach. *Springerphc Samanmali, Rahm Rupasinghamultimedia Tools And Applications, 2024•Springer, 83(36), 84425–*

84453. <https://doi.org/10.1007/S11042-024-19185-W>

Wankhade, Mayur, Rao, Annavarapu Chandra Sekhara, & Kulkarni, Chaitanya. (2022). A Survey On Sentiment Analysis Methods, Applications, And Challenges. *Springer* Wankhade, Acs Rao, C Kulkarni *artificial Intelligence Review*, 2022•*Springer*, 55(7), 5731–5780. <https://doi.org/10.1007/S10462-022-10144-1>