

Analisis Kepuasan Pengguna Aplikasi Gojek Berdasarkan Ulasan Menggunakan SVM

Nurintan Vermata Sari Sihombing¹, Bobby Agustinus Hutabarat², Cili Rosenna Simarmata³,
Arina Prima Silalahi⁴

^{1,2,3,4}Fakultas Ilmu Komputer, Universitas Methodist Indonesia

Info Artikel

Histori Artikel:

Received, 15 Juni 2026

Revised, 26 Juli 2026

Accepted, 3 Agustus 2026

Keywords:

Analisa Sentimen,
Klasifikasi,
SVM Klasifikasi,
Gojek,
Kaggle,
Python.

ABSTRAK

Perkembangan layanan transportasi digital mendorong meningkatnya ulasan pengguna di platform seperti Google Play Store, sehingga diperlukan metode otomatis untuk menganalisis sentimen tersebut. Penelitian ini bertujuan untuk menganalisis sentimen pengguna aplikasi Gojek menggunakan algoritma Support Vector Machine (SVM) berbasis representasi fitur TF-IDF. Dataset yang digunakan berjumlah 12.500 ulasan yang diperoleh dari platform Kaggle, terdiri dari lima kategori sentimen: sangat negatif, negatif, netral, positif, dan sangat positif, masing-masing berjumlah 2.500 data. Tahapan preprocessing yang diterapkan meliputi cleansing, case folding, tokenizing, stopword removal, dan stemming menggunakan bahasa pemrograman Python. Data dibagi dengan rasio 80:20 menggunakan stratified split, menghasilkan 10.000 data latih dan 2.500 data uji. Model SVM dilatih menggunakan kernel linear dengan parameter $C=1,0$ dan representasi TF-IDF dengan $\max_features=5.000$. Hasil evaluasi menunjukkan akurasi sebesar 51,32%, dengan kelas sangat positif memperoleh performa terbaik (F1-Score = 0,6908) dan kelas netral memperoleh performa terendah (F1-Score = 0,4283). Hasil ini mengindikasikan bahwa klasifikasi sentimen lima kelas dengan SVM dan TF-IDF masih memiliki keterbatasan dalam membedakan kelas-kelas yang memiliki perbedaan semantik tipis, sehingga diperlukan pengembangan lebih lanjut seperti normalisasi kata tidak baku dan penggunaan metode word embedding.

This is an open access article under the [CC BY-SA](#) license.



Penulis Koresponden:

Nurintan Vermata Sari Sihombing
Fakultas Ilmu Komputer,
Universitas Methodist Indonesia, Medan,
Jl. Hang Tuah No.8, Medan - Sumatera Utara.
Email: nvssihombing@gmail.com

1. PENDAHULUAN

Perkembangan teknologi digital telah membawa perubahan yang signifikan dalam berbagai aspek kehidupan masyarakat, termasuk pada sektor transportasi. Kehadiran layanan transportasi online memberikan kemudahan bagi masyarakat dalam melakukan aktivitas sehari-hari karena dapat diakses melalui smartphone secara cepat dan efisien. Tingginya penggunaan internet dan smartphone di Indonesia turut mendorong pertumbuhan layanan transportasi online yang semakin pesat. Salah satu layanan transportasi online yang banyak digunakan di Indonesia adalah Gojek. Gojek merupakan platform digital yang menyediakan berbagai layanan seperti transportasi, pengantaran makanan, pengiriman barang, dan pembayaran digital. Seiring dengan meningkatnya jumlah pengguna aplikasi Gojek, perusahaan dituntut untuk terus menjaga serta meningkatkan kualitas layanan agar dapat memenuhi kebutuhan dan kepuasan pengguna. [1]

Melalui Google Play Store, masyarakat dapat memberikan penilaian langsung yang merefleksikan tingkat kepuasan, kritik, maupun pengalaman mereka selama menggunakan aplikasi. Dokumen opini ini

tidak hanya memandu calon pengguna lain, tetapi juga menjadi instrumen evaluasi esensial bagi tim pengembang untuk membenahi kualitas platform. Mengingat pertumbuhan jumlah ulasan yang sangat eksponensial, pemrosesan manual menjadi tidak relevan lagi. Pendekatan analisis sentimen (*sentiment analysis*) kemudian diterapkan sebagai teknik otomatisasi guna mengenali pola teks dan mengklasifikasikannya ke dalam kategori opini positif, netral, atau negatif[2].

Algoritma *Support Vector Machine* (SVM) kerap menjadi pilihan utama dalam studi analisis sentimen karena kemampuannya yang adaptif terhadap data teks berskala besar dan berdimensi tinggi. Efektivitas metode ini terkonfirmasi melalui sejumlah riset terdahulu yang menguji ulasan aplikasi Gojek. Salah satu literatur mencatat bahwa SVM mampu meraih persentase akurasi sebesar 87,98%, unggul tipis dari metode KNN yang berada di angka 82,14%. Eksplorasi ilmiah lainnya juga memperlihatkan dominasi SVM atas algoritma *Random Forest*, dengan capaian nilai akurasi hingga 96%. Berdasarkan rekam jejak tersebut, penggunaan SVM terbukti menjadi solusi yang efisien dan valid untuk mengolah klasterisasi opini digital[3].

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk menganalisis sentimen pengguna aplikasi Gojek berdasarkan 12.500 data ulasan yang diperoleh dari Google Play Store menggunakan algoritma *Support Vector Machine* (SVM). Dataset yang digunakan terdiri dari lima kategori sentimen, yaitu sangat positif, positif, netral, negatif, dan sangat negatif. Hasil penelitian diharapkan dapat memberikan gambaran mengenai persepsi pengguna terhadap aplikasi Gojek serta menjadi bahan evaluasi dalam meningkatkan kualitas layanan yang diberikan.

2. METODE PENELITIAN

2.1 Analisa Sentimen

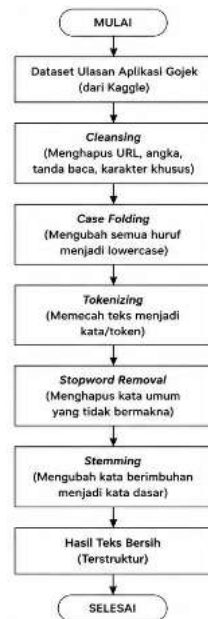
Analisis sentimen beroperasi di bawah payung *text mining* untuk membedah aspek emosional, sudut pandang, dan sikap pengguna terhadap suatu layanan atau produk[4]–[7]. Melalui proses komputasi, teks diidentifikasi dan diklasifikasikan ke dalam kategori sentimen yang spesifik, yaitu positif, netral, atau negatif. Dalam pengerjaan tugas akhir ini, penerapan analisis sentimen difungsikan sebagai alat ukur kepuasan konsumen dan penilai persepsi publik terhadap performa aplikasi Gojek lewat data tekstual dari Google Play Store[8].

2.2 Aplikasi Gojek

Gojek merupakan sebuah ekosistem teknologi asal Indonesia yang memulai kiprahnya pada tahun 2010 di bawah kepemimpinan Nadiem Makarim. Platform ini memelopori integrasi multi-layanan dalam satu pintu aplikasi, seperti moda transportasi berbasis daring (GoRide dan GoCar), jasa pengiriman logistik (GoSend), platform pemesanan makanan (GoFood), serta solusi pembayaran nontunai (GoPay). Statusnya sebagai salah satu *super-app* raksasa di Asia Tenggara didukung oleh tingginya loyalitas jutaan pengguna aktif harian. Pertumbuhan skala bisnis yang masif ini menuntut manajemen perusahaan untuk secara kontinu memperkuat standar pelayanan agar selaras dengan kebutuhan pasar.[9]

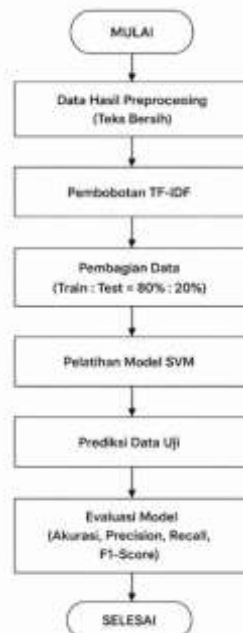
2.3 Text Mining

Text mining merupakan langkah analisis teks dengan tujuan mengekstrak informasi yang bernilai dari data teks yang tidak terstruktur[4]–[6], [10]. Di ranah *text mining*, terdapat berbagai kegiatan seperti menganalisis teks, mengekstrak informasi, mengelompokkan, mengategorikan, serta menerapkan pembelajaran mesin[7]. Sebelum melakukan klasifikasi menggunakan metode machine learning, langkah awal yang harus dilakukan adalah melakukan text preprocessing terhadap data teks tersebut.[11]

Gambar 1. *Flowchart Text Processing*

2.4 Support Vector Machine (SVM)

Sebagai algoritma pembelajaran mesin yang diperkenalkan oleh Vapnik dan Cortes pada tahun 1995, *Support Vector Machine* (SVM) memiliki reputasi yang solid dalam ranah klasifikasi dokumen teks. Prinsip kerja utama dari metodologi ini berpusat pada penentuan posisi *hyperplane* paling optimal guna memisahkan kelompok data antar-kelas dengan batas jarak (*margin*) yang paling maksimal. Karakteristik ini membuat SVM sangat andal dan efisien ketika diimplementasikan pada data berdimensi tinggi, seperti pada ekstraksi fitur tekstual, yang sering kali menghasilkan akurasi klasifikasi lebih superior dibanding pendekatan alternatif lainnya.[12]

Gambar 2. *Flowchart SVM Classification*

Tahapan-tahapan yang perlu dilakukan dalam klasifikasi menggunakan algoritma SVM adalah sebagai berikut [3]:

1. Lakukan text preprocessing pada data ulasan, meliputi cleansing, case folding, tokenizing, stopwords removal, dan stemming.
2. Ubah data teks hasil preprocessing menjadi representasi vektor numerik menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency).
3. Hitung bobot TF-IDF untuk setiap kata dalam dokumen menggunakan rumus berikut:

$$TF-IDF(t,d) = TF(t,d) \times IDF(t)$$

Keterangan:

TF(t,d) = Frekuensi kemunculan kata t dalam dokumen d

IDF(t) = $\log(N / df(t))$, dimana N adalah jumlah total dokumen dan df(t) adalah jumlah dokumen yang mengandung kata t

TF-IDF(t,d) = Nilai bobot kata t pada dokumen d

4. Tentukan kernel SVM yang digunakan. Kernel yang umum digunakan antara lain kernel linear, polynomial, dan Radial Basis Function (RBF).
5. Latih model SVM menggunakan data training dengan fungsi keputusan sebagai berikut:

$$f(x) = \text{sign}(w \cdot x + b)$$

Keterangan:

w = Vektor bobot (weight vector)

x = Vektor fitur input

b = Nilai bias

6. Klasifikasikan data testing menggunakan model yang telah dilatih dan evaluasi performa model menggunakan metrik akurasi, presisi, recall, dan F1-score.

2.5 Framework Penelitian

Adapun alur atau tahapan yang dilakukan dalam penelitian ini diilustrasikan pada Gambar 3.



Gambar 3. *Framework Penelitian*

Tahapan awal pada penelitian ini yaitu mengambil dataset ulasan pengguna aplikasi Gojek yang berjumlah 12.500 data dari platform *Kaggle*. Dataset tersebut merupakan ulasan yang berasal dari Google Play Store dengan lima kategori label sentimen, yaitu sangat positif, positif, netral, negatif, dan sangat negatif. Setelah dataset diperoleh, maka akan dilakukan text preprocessing untuk mengubah data ulasan mentah menjadi data terstruktur menggunakan bahasa pemrograman Python. Setelah data bersih diperoleh, akan dilakukan ekstraksi fitur menggunakan TF-IDF, kemudian dilanjutkan dengan penerapan algoritma Support Vector Machine (SVM) untuk mendapatkan hasil klasifikasi sentimen.

3. HASIL DAN PEMBAHASAN

3.1 Dataset

Penelitian ini memanfaatkan dataset ulasan pengguna Gojek pada Google Play Store yang diperoleh dari laman resmi *Kaggle*, di mana data tersebut telah dipastikan bersih melalui serangkaian pemrosesan. Secara keseluruhan, terdapat 12.500 rekaman ulasan yang dikelompokkan ke dalam lima kategori kelas sentimen dengan porsi berimbang, yakni tepat 2.500 data untuk tiap-tiap kelasnya. Karakteristik dan sebaran dari data tersebut telah dirangkum dalam Tabel 3.1.

Tabel 3.1 Distribusi Dataset

No	Kelas Sentimen	Label	Jumlah
1	sangat negatif	1	2.500
2	negatif	2	2.500
3	netral	3	2.500
4	positif	4	2.500
5	sangat positif	5	2.500
	Total		12.500

3.2 Preprocessing Text

Sebelum memasuki fase komputasi model, data ulasan mentah harus disaring terlebih dahulu melalui tahapan *text preprocessing* agar polanya menjadi lebih terstruktur. Rangkaian aktivitas ini meliputi teknik *Cleansing*, *Case Folding*, *Tokenizing*, *Stopword Removal*, dan *Stemming*.

3.2.1 Cleansing

Tahapan *cleansing* difungsikan untuk membersihkan teks ulasan dari noise atau gangguan visual seperti URL, angka, tanda baca, serta karakter khusus yang kurang relevan. Langkah ini menjamin teks berada dalam kondisi optimal sebelum diproses lebih jauh. Dokumentasi hasil pembersihan untuk 5 sampel data ulasan dicantumkan secara lengkap pada Tabel 3.2.

Tabel 3.2 Hasil Cleansing

No	Dok	Review Asli	Hasil Cleansing
1	D1	Untuk gofood diskon diperlambat gada sengaja pengiriman pembayaran cod	Untuk gofood diskon diperlambat gada sengaja pengiriman pembayaran cod
2	D2	dengan ada nya ini terselasaikn hal yg urgent dapet transportasi sangat membantu	dengan ada nya ini terselasaikn hal yg urgent dapet transportasi sangat membantu
3	D3	gabisa chat sama cs nya gatau kenapa	gabisa chat sama cs nya gatau kenapa
4	D4	gojek lagi masalah apa dah internet lancar tapi dibilangnya tidak tersambung	gojek lagi masalah apa dah internet lancar tapi dibilangnya tidak tersambung
5	D5	driver ramah interaktif mantab pokoknya	driver ramah interaktif mantab pokoknya

3.2.2 Case Folding

Proses *case folding* bekerja dengan mengubah seluruh komponen huruf kapital menjadi huruf kecil. Langkah ini krusial untuk menyamakan persepsi kata pada sistem, sehingga perbedaan penulisan huruf besar di awal atau di tengah kata tidak memicu bias klasifikasi. Contoh perubahan bentuk kata dari tahapan *case folding* ini disajikan pada Tabel 3.3.

Tabel 3.3 Hasil Case Folding

No	Dok	Hasil Cleansing	Hasil Case Folding
1	D1	Untuk gofood diskon diperlambat gada sengaja pengiriman pembayaran cod	untuk gofood diskon diperlambat gada sengaja pengiriman pembayaran cod
2	D2	dengan ada nya ini terselasaikn hal yg urgent dapet transportasi sangat membantu	dengan ada nya ini terselasaikn hal yg urgent dapet transportasi sangat membantu
3	D3	gabisa chat sama cs nya gatau kenapa	gabisa chat sama cs nya gatau kenapa
4	D4	gojek lagi masalah apa dah internet lancar tapi dibilangnya tidak tersambung	gojek lagi masalah apa dah internet lancar tapi dibilangnya tidak tersambung
5	D5	driver ramah interaktif mantab pokoknya	driver ramah interaktif mantab pokoknya

3.2.3 Tokenizing

Proses tokenisasi bertugas memilah teks ulasan menjadi kumpulan kata tunggal atau token. Setiap komponen kata dipisahkan secara sistematis dan dikelompokkan sebagai objek mandiri di dalam sebuah

daftar (*list*). Langkah ini sangat krusial agar model dapat mengenali dan mengevaluasi bobot setiap kata secara individual pada tahapan berikutnya. Visualisasi teks pasca-tokenisasi ini termuat dalam. Tabel 3.4.

Tabel 3.4 Hasil Tokenizing

No	Dok	Hasil Case Folding	Hasil Tokenizing
1	D1	untuk gofood diskon diperlamb at gada sengaja pengirima n pembayara n cod	['untuk','gofood','diskon','diperlambat','gada','sengaja','pengiriman','pembayaran','cod']
2	D2	dengan ada nya ini terselasaik n hal yg urgent dapat transportas i sangat membantu	['dengan','ada','nya','ini','terselasaikn','hal','yg','urgent','dapat','transportasi']
3	D3	gabisa chat sama cs nya gatau kenapa	['gabisa','chat','sama','cs','nya','gatau','kenapa']
4	D4	gojek lagi masalah apa dah internet lancar tapi dibilangny a tidak tersambun g	['gojek','lagi','masalah','apa','dah','internet','lancar','tapi','dibilangnya']
5	D5	driver ramah interaktif mantab pokoknya	['driver','ramah','interaktif','mantab','pokoknya']

3.2.4 Stopword Removal

Proses *stopword removal* berfungsi mengeliminasi kata-kata yang tidak memiliki makna yang signifikan bagi model klasifikasi, seperti "yang", "dan", "di", "ke", dan "dari". Untuk menyaring kata-kata tersebut, penelitian ini menggunakan daftar *stopword* bahasa Indonesia yang disediakan oleh *library* NLTK. Contoh perbandingan teks sebelum dan sesudah proses eliminasi kata ini tercantum dalam Tabel 3.5.

Tabel 3.5 Hasil Stopword Removal

No	Dok	Hasil Tokenizing	Hasil Stopword
1	D1	['untuk','gofood','diskon','diperlambat','gada','sengaja','pengiriman','pembayaran','cod']	gofood diskon diperlambat gada sengaja pengiriman pembayaran cod
2	D2	['dengan','ada','nya','ini','terselasaikn','hal','yg','urgent','dapat','transportasi']	nya terselasaikn yg urgent dapat transportasi sangat membantu
3	D3	['gabisa','chat','sama','cs','nya','gatau','kenapa']	gabisa chat cs nya gatau
4	D4	['gojek','lagi','masalah','apa','dah','internet','lancar','tapi','dibilangnya']	gojek dah internet lancar dibilangnya tersambung
5	D5	['driver','ramah','interaktif','mantab','pokoknya']	driver ramah interaktif mantab pokoknya

3.2.5 Stemming

Proses *stemming* dilakukan untuk mengembalikan setiap kata berimbuhan ke bentuk kata dasar aslinya. Riset ini mengimplementasikan *library* PySastrawi untuk mengeksekusi pemotongan imbuhan pada teks berbahasa Indonesia. Langkah ini efektif dalam menyatukan variasi kata yang berakar sama agar dikenali sebagai satu entitas tunggal oleh algoritma klasifikasi. Visualisasi dari draf teks yang telah melalui proses *stemming* ini terangkum pada Tabel 3.6

Tabel 3.6 Hasil Stemming

No	Dok	Hasil Stopword	Hasil Stemming
1	D1	gofood diskon diperlambat gada sengaja pengiriman pembayaran cod	gofood diskon lambat gada sengaja kirim bayar cod
2	D2	nya terselasaikn yg urgent dapat transportasi sangat membantu	nya terselasaikn yg urgent dapat transportasi sangat bantu
3	D3	gabisa chat cs nya gatau	gabisa chat cs nya gatau
4	D4	gojek dah internet lancar dibilangnya tersambung	gojek dah internet lancar bilang sambung
5	D5	driver ramah interaktif mantab pokoknya	driver ramah interaktif mantab pokok

3.3 Pembobotan Dokumen dengan TF-IDF

Pasca-fase pemrosesan awal teks (*preprocessing*), data tekstual dikonversi ke dalam bentuk representasi matriks numerik melalui pendekatan *Term Frequency-Inverse Document Frequency* (TF-IDF). Skema pembobotan ini bekerja secara simultan dengan mengorelasikan frekuensi kemunculan term dalam suatu dokumen spesifik (*Term Frequency*) terhadap tingkat kekhasan atau kelangkaan term tersebut di seluruh korpus (*Inverse Document Frequency*). Dalam pengerjaan ini, konfigurasi argumen `max_features=5000` diterapkan guna membatasi ekstraksi hanya pada 5.000 kosakata dengan nilai bobot tertinggi yang akan disuplai sebagai fitur input algoritma

Rumus TF-IDF yang digunakan adalah sebagai berikut:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

$$\text{IDF}(t) = \log\left(\frac{N}{\text{df}(t)}\right)$$

Di mana:

- $\text{TF}(t, d)$: Kuantitas kemunculan token t di dalam dokumen d .
- $\text{IDF}(t)$: Logaritma rasio jumlah dokumen keseluruhan (N) terhadap jumlah dokumen yang memuat token t ($\text{df}(t)$).
- $\text{TF-IDF}(t, d)$: Nilai bobot akhir dari token t pada dokumen d .

Tabel 3.7 Contoh Pembobotan TF-IDF (Sebagian Kata)

Kata	D1	D2	D3	D4	D5	DF	IDF
gofood	1	0	0	0	0	1	0.699
diskon	1	0	0	0	0	1	0.699
lambat	1	0	0	0	0	1	0.699
bantu	0	1	0	0	0	1	0.699
gabisa	0	0	1	0	0	1	0.699
gojek	0	0	0	1	0	1	0.699
driver	0	0	0	0	1	1	0.699
pokok	0	0	0	0	1	1	0.699

Berdasarkan Tabel 3.7, seluruh kata pada contoh ini hanya muncul pada satu dokumen ($\text{df} = 1$) sehingga memiliki nilai IDF yang sama, yaitu 0,699. Semakin sedikit dokumen yang mengandung suatu kata, semakin tinggi nilai IDF yang diperoleh, sedangkan kata yang muncul di banyak atau seluruh dokumen akan memiliki nilai IDF yang semakin rendah atau mendekati 0.

3.4 Pembagian Data Latih dan Data Uji

Melalui metode *train-test split*, dataset yang tersedia dipisahkan ke dalam komponen data latih dan data uji dengan perbandingan bobot 80:20. Untuk menjamin akurasi pemodelan, argumen *stratify* diaplikasikan agar distribusi lima kategori sentimen tetap merata pada kedua kelompok data. Pembagian kuantitas data secara spesifik untuk masing-masing kelas dirangkum pada Tabel 3.8.

Tabel 3.8 Pembagian Data Latih dan Data Uji

No	Kelas Sentimen	Data Latih (80%)	Data Uji (20%)
1	sangat negatif	2.000	500
2	negatif	2.000	500
3	netral	2.000	500
4	positif	2.000	500
5	sangat positif	2.000	500
	Total	10.000	2.500

Berdasarkan Tabel 3.8, data latih berjumlah 10.000 data yang digunakan untuk melatih model SVM, dan data uji berjumlah 2.500 data yang digunakan untuk mengukur performa model. Setiap kelas sentimen memiliki 2.000 data latih dan 500 data uji.

3.5 Pengujian Sistem

Setelah bobot fitur TF-IDF diperoleh dari data latih dan data uji, algoritma SVM disuplai dengan data latih untuk fase pembelajaran model. Proses pengujian ini berjalan di bawah pengaturan parameter regularisasi $C=1.0$ dengan basis *kernel linear*. Langkah berikutnya melibatkan penggunaan model tersebut untuk memprediksi kecenderungan kategori sentimen pada subset data uji. Sebagai tahap akhir, efektivitas performa klasifikasi diukur secara matematis mengacu pada nilai akurasi, *precision*, *recall*, dan nilai *F1-score*.

3.5.1 Akurasi

Model SVM yang dilatih dalam penelitian ini menghasilkan akurasi sebesar 51,32% pada data uji yang berjumlah 2.500 data. Akurasi dihitung menggunakan rumus berikut:

$$\begin{aligned}\text{Akurasi} &= (\text{Jumlah Prediksi Benar} / \text{Total Data Uji}) \times 100\% \\ &= (1.283 / 2.500) \times 100\% = 51,32\%\end{aligned}$$

3.5.2 Precision, Recall, dan F1-Score

Selain mengukur tingkat akurasi umum, efektivitas model ini juga diuji berdasarkan skor *precision*, *recall*, serta *F1-score* di setiap kelas sentimen. Rekapitulasi hasil matrik evaluasi sistem ini dapat dicermati secara lengkap pada Tabel 3.9.

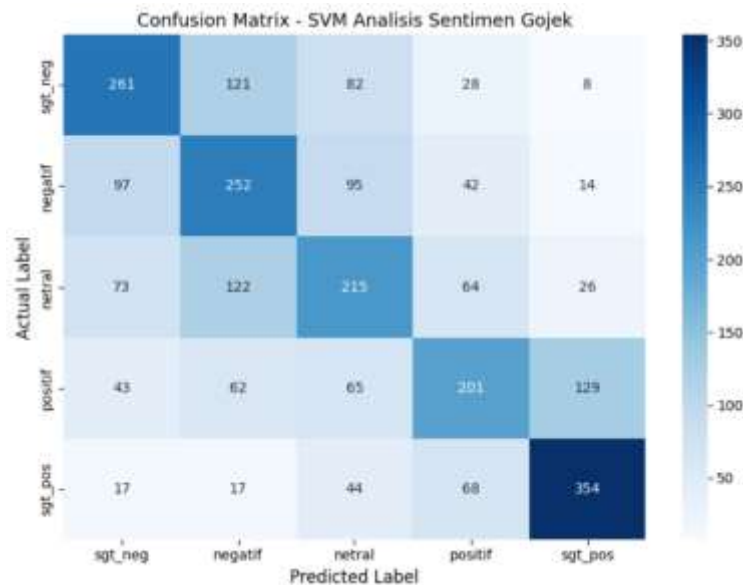
Tabel 3.9 Hasil Evaluasi Model SVM (Classification Report)

Kelas Sentimen	Precision	Recall	F1-Score	Support
sangat negatif	0,43	0,50	0,47	500
negatif	0,43	0,43	0,43	500
netral	0,50	0,40	0,45	500
positif	0,53	0,52	0,53	500
sangat positif	0,67	0,71	0,69	500
Rata-rata (weighted)	0,51	0,51	0,50	2.500

Berdasarkan Tabel 3.9, menunjukkan bahwa kategori sentimen sangat_positif mencatatkan performa paling dominan dengan raihan skor *Precision* (0,6726), *Recall* (0,7100), dan *F1-Score* (0,6908) yang melampaui kelompok sentimen lainnya. Temuan empiris ini mengonfirmasi superioritas model dalam mengenali karakter ulasan berkategori sangat positif. Di sisi lain, penurunan performa paling signifikan terlihat pada kelas netral yang hanya mengantongi nilai *F1-Score* sebesar 0,4283. Rendahnya skor tersebut merefleksikan adanya hambatan atau bias pada algoritma dalam memisahkan teks ulasan netral dari intervensi kelas sentimen lain.

3.5.3 Confusion Matrix

Komparasi hasil klasifikasi antara label berpasangan (aktual vs prediksi) dipetakan secara detail oleh *confusion matrix* pada Tabel 3.10. Nilai pada lajur diagonal utama menandakan total data yang berhasil ditebak secara benar, sementara nilai di luar lajur tersebut menandakan kesalahan klasterisasi.



Dari hasil komputasi, kuantitas tebakan model yang akurat pada diagonal utama meliputi: sangat_negatif (261), negatif (257), netral (219), positif (203), dan sangat_positif (354) dari basis 500 sampel data pengujian di setiap kelas. Fenomena error atau salah prediksi paling banyak dijumpai pada variasi kelas yang berdekatan secara makna bahasa, seperti fluktuasi data antara kategori negatif dengan netral, serta klaster data positif dengan sangat_positif.

3.6 Pembahasan

Dari hasil evaluasi, kolaborasi metode TF-IDF dan model SVM *kernel linear* memperoleh nilai akurasi sebesar 51,32% dalam memetakan lima kategori ulasan pelanggan Gojek. Catatan performa klasifikasi ini dipicu oleh beberapa poin esensial di bawah ini.

Pertama, kompleksitas klasifikasi lima kelas. Tugas klasifikasi sentimen lima kelas jauh lebih sulit dibandingkan klasifikasi biner. Perbedaan semantik antar kelas yang berdekatan, seperti negatif dengan sangat_negatif, atau positif dengan sangat_positif, sangat tipis sehingga sulit diidentifikasi hanya berdasarkan representasi TF-IDF.

Kedua, keunggulan pada kelas ekstrem. Model menunjukkan performa terbaik pada kelas sangat_positif dengan F1-Score 0,6908. Ulasan sangat positif cenderung menggunakan kata-kata yang khas dan tegas, seperti 'mantap', 'puas', 'luar biasa', sehingga lebih mudah dikenali oleh model.

Ketiga, kelemahan pada kelas tengah. Kelas netral dan negatif menghasilkan F1-Score terendah. Ulasan pada kelas ini seringkali mengandung campuran sentimen positif dan negatif, sehingga sulit diklasifikasikan secara tepat.

Keempat, potensi peningkatan. Akurasi model dapat ditingkatkan dengan menerapkan normalisasi kata tidak baku (slang), penggunaan Word Embedding seperti Word2Vec atau FastText, serta optimasi hyperparameter SVM menggunakan Grid Search CV.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada bab ini, dapat disimpulkan bahwa:

Model SVM dengan kernel linear berhasil mengklasifikasikan ulasan pengguna aplikasi Gojek ke dalam lima kelas sentimen dengan akurasi sebesar 51,32%. Tahapan preprocessing text (cleansing, case folding, tokenizing, stopwords removal, dan stemming) berhasil mengubah data teks mentah menjadi representasi yang terstruktur dan siap diolah oleh model machine learning.

Pembagian data dilakukan dengan rasio 80:20 menggunakan stratified split, menghasilkan 10.000 data latih dan 2.500 data uji dengan distribusi kelas yang seimbang.

Kelas sangat_positif memperoleh performa terbaik dengan F1-Score 0,6908, sedangkan kelas netral memperoleh F1-Score terendah sebesar 0,4283, yang mengindikasikan bahwa model mengalami kesulitan dalam membedakan ulasan yang memiliki perbedaan semantik tipis, khususnya pada kelas-kelas yang berdekatan seperti negatif dengan netral, dan positif dengan sangat positif.

Akurasi sebesar 51,32% menunjukkan bahwa klasifikasi sentimen lima kelas menggunakan SVM dengan representasi TF-IDF masih memiliki keterbatasan. Peningkatan performa dapat dilakukan melalui penerapan normalisasi kata tidak baku, penggunaan metode word embedding yang lebih kontekstual, serta optimasi hyperparameter model SVM.

4.2 Saran

Penelitian ini masih memerlukan pengembangan lebih lanjut. Sebagai rekomendasi yang dapat menjadi kontribusi bagi penelitian selanjutnya, adalah:

Berdasarkan penelitian yang dilakukan, diharapkan penelitian selanjutnya dapat memperkaya proses preprocessing dengan menambahkan tahapan normalisasi kata tidak baku (slang normalization) menggunakan kamus kata informal Bahasa Indonesia. Mengingat data ulasan dari Google Play Store banyak mengandung bahasa gaul dan singkatan, penanganan kata tidak baku yang lebih baik diharapkan dapat meningkatkan kualitas representasi teks dan performa klasifikasi model secara keseluruhan.

Berdasarkan hasil penelitian yang menunjukkan akurasi SVM sebesar 51,32% dengan representasi TF-IDF, diharapkan penelitian selanjutnya dapat mengeksplorasi penggunaan metode word embedding seperti Word2Vec atau FastText sebagai alternatif representasi fitur teks. Metode ini mampu menangkap hubungan semantik antar kata secara lebih mendalam dibandingkan TF-IDF, sehingga diharapkan dapat membantu model dalam membedakan kelas-kelas sentimen yang memiliki kemiripan makna, seperti kelas negatif dengan sangat negatif, atau kelas positif dengan sangat positif.

Berdasarkan penelitian yang dilakukan, diharapkan penelitian selanjutnya dapat melakukan optimasi hyperparameter model SVM menggunakan teknik Grid Search CV atau Random Search, serta membandingkan performa SVM dengan algoritma klasifikasi lainnya seperti Random Forest, Naive Bayes, atau model berbasis deep learning seperti BERT yang telah terbukti unggul dalam pemrosesan bahasa alami. Perbandingan ini diharapkan dapat memberikan gambaran yang lebih komprehensif dalam memilih metode terbaik untuk analisis sentimen ulasan pengguna berbahasa Indonesia.

REFERENSI

- [1] A. Q. Sari, Y. L. Sukestiyarno, and A. Agoestanto, “Unnes Journal of Mathematics,” *Unnes J. Math.*, vol. 6, no. 2, pp. 168–177, 2017.
- [2] M. M. Maarif and N. Setiyawati, “Analisis Sentimen Review Aplikasi LinkedIn di Google Play Store Menggunakan Support Vector Machine,” *Progresif J. Ilm. Komput.*, vol. 20, no. 1, p. 454, 2024, doi: 10.35889/progresif.v20i1.1614.
- [3] U. J. I. Performa, D. A. N. Perbandingan, and R. Mysql, “Jurnal Informatika Terpadu HIVE-HADOOP,” *J. Inform. Terpadu*, vol. 8, no. 1, pp. 1–7, 2022.
- [4] T. R. Baruara, H. G. Manullang, and A. P. Silalahi, “ANALISA ALGORITMA K-MEDOIDS DALAM,” vol. 4, no. 2, pp. 1–10, 2024.
- [5] L. M. Barus, H. G. Simanullang, and A. P. Silalahi, “Classification of English-Language Racial Hate Speech Using Random Forest,” *J. Intell. Comput. Inf. Syst.*, vol. 1, no. 1, pp. 1–17, 2025.
- [6] H. G. Simanullang, A. P. Silalahi, and G. P. Pangaribuan, “PENGELOMPOKAN KOMENTAR PENGGUNA JKN MOBILE MENGGUNAKAN METODE K-MEDOIDS,” 2025.
- [7] H. G. Simanullang, S. Wijono, B. Kristianto, W. Sulisty, and A. P. Silalahi, “Detecting Cyberbullying Using Machine Learning Approaches,” in *2024 Ninth International Conference on Informatics and Computing (ICIC)*, 2024, pp. 1–8. doi: 10.1109/ICIC64337.2024.10956842.
- [8] I. Maulana, W. Apriandari, and A. Pambudi, “Analisis Sentimen Berbasis Aspek Terhadap Ulasan Aplikasi Mypertamina Menggunakan Support Vector Machine,” *IDEALIS Indones. J. Inf. Syst.*, vol. 6, no. 2, pp. 172–181, 2023, doi: 10.36080/idealis.v6i2.3022.
- [9] Y. Khoiruddin, A. Fauzi, and A. M. Siregar, “Menggunakan Algoritme Naïve Bayes Dan Support Vector Machine,” *Progresif J. Ilm. Komput.*, vol. 19, no. 1, p. 391, 2023, doi: 10.35889/progresif.v19i1.1173.
- [10] S. M. Sipayung, M. Yohanna, H. G. Manullang, and R. Mandala, “ANALISIS SENTIMEN MENGGUNAKAN K-NEAREST NEIGHBOR TERHADAP FILM NGERI-NGERI SEDAP,” vol. 4, no. 2, pp. 42–52, 2024.
- [11] T. Tukino and F. Fifi, “Penerapan Support Vector Machine Untuk Analisis Sentimen Pada Layanan Ojek Online,” *J. Desain Dan Anal. Teknol.*, vol. 3, no. 2, pp. 104–113, 2024, doi: 10.58520/jddat.v3i2.59.
- [12] 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, 2024, doi: 10.36080/skanika.v7i2.3193.