

## Klasifikasi Sentimen Ulasan Pengguna Aplikasi Mobile Menggunakan Algoritma Support Vector Machine

Syifa Amara Dhestiyani<sup>1\*</sup>

<sup>1</sup>Universitas Telkom, Bandung - Indonesia

\*e-mail : syifaamdh@student.telkomuniversity.ac.id

### Abstrak

Pertumbuhan aplikasi mobile meningkatkan volume ulasan pengguna yang memuat opini penting bagi pengembang. Penelitian ini bertujuan mengklasifikasikan sentimen ulasan aplikasi menjadi positif, negatif, dan netral menggunakan algoritma Support Vector Machine (SVM). Dataset terdiri dari 5.000 ulasan yang diambil dari Google Play Store, kemudian melalui proses preprocessing meliputi tokenisasi, stopword removal, dan stemming. Fitur diekstraksi menggunakan metode TF-IDF, kemudian dilatih dan diuji menggunakan SVM dengan K-fold cross-validation. Hasil menunjukkan model SVM mampu mencapai akurasi 87%, presisi 85%, recall 86%, dan F1-score 85,5%. Analisis menunjukkan performa dipengaruhi panjang ulasan dan penggunaan istilah informal. Temuan ini menegaskan efektivitas SVM dalam klasifikasi sentimen ulasan aplikasi mobile serta memberikan dasar bagi pengembang untuk memahami opini pengguna secara otomatis.

**Kata Kunci:** Klasifikasi Sentimen, Support Vector Machine, Ulasan Aplikasi Mobile, TF-IDF, Analisis Opini

### Abstract

The growth of mobile applications has increased the volume of user reviews, which contain valuable opinions for developers. This study aims to classify mobile app reviews into positive, negative, and neutral sentiments using the Support Vector Machine (SVM) algorithm. The dataset consists of 5,000 reviews collected from Google Play Store, processed through tokenization, stopword removal, and stemming. Features are extracted using TF-IDF and trained with SVM using K-fold cross-validation. Results show that the SVM model achieves 87% accuracy, 85% precision, 86% recall, and 85.5% F1-score. Performance is influenced by review length and informal language usage. These findings confirm SVM's effectiveness in mobile app sentiment classification and provide a foundation for developers to automatically understand user opinions.

**Keywords:** Sentiment Classification, Support Vector Machine, Mobile App Reviews, TF-IDF, Opinion Analysis

## **Pendahuluan**

Perkembangan pesat teknologi mobile telah mendorong proliferasi aplikasi yang digunakan dalam kehidupan sehari-hari, mulai dari komunikasi, hiburan, hingga produktivitas. Setiap aplikasi digital menarik perhatian pengguna yang kemudian memberikan ulasan terkait pengalaman mereka. Ulasan ini tidak hanya berisi penilaian kualitas aplikasi, tetapi juga kritik, saran, dan opini subjektif yang dapat memberikan masukan penting bagi pengembang. Volume ulasan yang sangat besar menyebabkan analisis manual menjadi tidak efisien, mahal, dan rentan terhadap bias. Oleh karena itu, pengembangan metode otomatis untuk mengekstraksi dan mengklasifikasikan opini pengguna menjadi suatu kebutuhan strategis dalam pengembangan aplikasi mobile (Pang & Lee, 2008; Medhat, Hassan, & Korashy, 2014; Liu, 2020).

Analisis sentimen merupakan pendekatan yang memungkinkan pemahaman opini pengguna melalui pemisahan ulasan ke dalam kategori positif, negatif, dan netral. Pendekatan berbasis machine learning terbukti efektif karena mampu menangkap pola teks yang kompleks dan dimensi fitur yang tinggi. Salah satu algoritma yang sering digunakan adalah Support Vector Machine (SVM), yang dikenal unggul dalam klasifikasi teks karena kemampuannya memaksimalkan margin pemisah antar kelas serta menangani data berdimensi tinggi (Joachims, 1998; Cortes & Vapnik, 1995). Meskipun banyak penelitian sebelumnya telah menggunakan SVM untuk analisis sentimen, penelitian ini menambahkan novelty berupa penerapan preprocessing teks adaptif untuk ulasan aplikasi mobile berbahasa Indonesia, termasuk tokenisasi, normalisasi istilah informal, penghilangan stopword khusus konteks aplikasi, serta stemming berbasis kamus domain. Pendekatan ini diharapkan meningkatkan performa model SVM dalam menangkap nuansa bahasa sehari-hari yang sering muncul dalam ulasan pengguna (Kurniawan et al., 2021; Fadli & Nugroho, 2022).

Rumusan masalah dalam penelitian ini berfokus pada dua aspek utama. Pertama, bagaimana cara mengklasifikasikan sentimen ulasan aplikasi mobile menjadi kategori positif, negatif, dan netral secara akurat. Kedua, seberapa efektif algoritma SVM dengan penerapan preprocessing adaptif dalam meningkatkan performa klasifikasi dibandingkan dengan metode standar yang umum digunakan.

Tujuan penelitian ini adalah membangun model klasifikasi sentimen ulasan aplikasi mobile menggunakan SVM yang dilengkapi dengan preprocessing adaptif untuk menangani variasi bahasa sehari-hari dan istilah informal. Selain itu, penelitian ini bertujuan mengevaluasi performa model secara komprehensif melalui metrik akurasi, presisi, recall, dan F1-score, sehingga dapat mengukur kemampuan model dalam mengklasifikasikan ulasan secara tepat.

Manfaat penelitian mencakup penyediaan dasar bagi pengembang aplikasi mobile berbahasa Indonesia untuk melakukan analisis sentimen secara otomatis. Penelitian ini juga memberikan informasi strategis yang dapat digunakan untuk meningkatkan kualitas fitur dan layanan aplikasi, sehingga opini dan pengalaman pengguna dapat direspons secara lebih cepat dan tepat, mendukung pengambilan keputusan berbasis data (Prasetyo et al., 2020; Rakhman et al., 2021).

## **Metode**

### ***Desain Penelitian***

Penelitian ini menggunakan pendekatan kuantitatif eksperimental berbasis komputasional. Penelitian ini menguji kemampuan algoritma Support Vector Machine (SVM) dalam mengklasifikasikan sentimen ulasan pengguna aplikasi mobile ke dalam tiga kategori, yaitu positif, negatif, dan netral. Pendekatan kuantitatif digunakan karena penelitian mengukur performa model melalui nilai akurasi, presisi, recall, dan F1-score. Pendekatan eksperimental digunakan karena penelitian membandingkan hasil klasifikasi setelah data melalui tahapan preprocessing adaptif, ekstraksi fitur TF-IDF, pelatihan model, dan evaluasi performa.

Penelitian ini berfokus pada klasifikasi sentimen ulasan aplikasi mobile berbahasa Indonesia. Fokus tersebut dipilih karena ulasan pengguna aplikasi sering memuat bahasa informal, singkatan, typo, slang, emoji, dan ekspresi subjektif. Kondisi ini membuat proses klasifikasi sentimen membutuhkan tahap preprocessing yang lebih sesuai dengan karakteristik bahasa pengguna. Studi terbaru pada ulasan aplikasi Telegram di Google Play Store juga menempatkan preprocessing data dan SVM sebagai komponen penting dalam klasifikasi sentimen positif dan negatif.

### ***Dataset Penelitian***

Data penelitian berupa 5.000 ulasan pengguna aplikasi mobile berbahasa Indonesia yang diperoleh dari Google Play Store. Ulasan tersebut mencakup opini pengguna terhadap fitur aplikasi, kinerja sistem, kemudahan penggunaan, stabilitas aplikasi, tampilan antarmuka, kecepatan layanan, dan pengalaman penggunaan. Google Play Store dipilih karena platform tersebut menyediakan ulasan tekstual dan rating pengguna yang dapat digunakan sebagai sumber data analisis sentimen. Beberapa penelitian terdahulu pada ulasan aplikasi Info BMKG, TikTok, Telegram, dan aplikasi lain juga menggunakan data ulasan dari Google Play Store untuk membangun model klasifikasi sentimen berbasis TF-IDF dan SVM.

Data yang digunakan dalam penelitian ini memenuhi beberapa kriteria. Pertama, ulasan ditulis dalam bahasa Indonesia. Kedua, ulasan memuat opini atau pengalaman pengguna terhadap aplikasi. Ketiga, ulasan memiliki struktur teks yang dapat diproses melalui tahapan preprocessing. Keempat, data tidak memuat duplikasi yang identik. Kelima, data tidak memuat teks kosong atau hanya berisi simbol. Kriteria ini digunakan agar model memperoleh data yang relevan untuk proses klasifikasi sentimen.

Setiap ulasan dikategorikan ke dalam tiga kelas sentimen, yaitu positif, negatif, dan netral. Ulasan positif berisi kepuasan, apresiasi, rekomendasi, atau pengalaman baik terhadap aplikasi. Ulasan negatif berisi keluhan, kritik, ketidakpuasan, gangguan teknis, atau pengalaman buruk. Ulasan netral berisi pernyataan informatif, ambigu, atau tidak menunjukkan kecenderungan emosi yang kuat. Proses pelabelan dapat dilakukan menggunakan kombinasi rating pengguna dan validasi manual. Rating tinggi, misalnya 4 sampai 5, dapat menjadi indikasi awal sentimen positif. Rating rendah, misalnya 1 sampai 2, dapat menjadi indikasi awal sentimen negatif. Rating 3 dapat menjadi indikasi awal sentimen netral, tetapi tetap memerlukan pemeriksaan isi teks.

Dataset dibagi menjadi data latih dan data uji dengan proporsi 80:20. Data latih digunakan untuk membangun model SVM, sedangkan data uji digunakan untuk mengukur kemampuan model dalam mengklasifikasikan data baru. Pembagian data dilakukan secara stratified agar distribusi kelas positif, negatif, dan netral tetap seimbang pada data latih dan data uji. Strategi ini membantu model mempelajari pola setiap kelas secara proporsional.

### ***Tahapan Preprocessing Data***

Case folding dilakukan dengan mengubah seluruh huruf dalam ulasan menjadi huruf kecil. Tahap ini bertujuan menyamakan representasi kata yang sama tetapi ditulis dengan format berbeda. Misalnya, kata “Bagus”, “BAGUS”, dan “bagus” akan diproses sebagai satu bentuk kata yang sama. Dengan cara ini, sistem mengurangi duplikasi fitur yang tidak diperlukan.

Cleaning data dilakukan dengan menghapus elemen yang tidak relevan, seperti URL, tanda baca berlebihan, angka yang tidak memiliki makna sentimen, karakter khusus, spasi ganda, dan simbol nontekstual. Tahap ini penting karena ulasan pengguna sering memuat karakter acak, emotikon, atau simbol yang dapat menambah noise pada dataset. Cleaning membantu model memproses teks yang lebih bersih dan terstruktur.

Tokenisasi dilakukan dengan memecah kalimat ulasan menjadi unit kata. Misalnya, kalimat “aplikasi ini sering error” akan dipecah menjadi token “aplikasi”, “ini”, “sering”, dan “error”.

Tokenisasi menjadi dasar bagi proses penghapusan stopword, normalisasi kata, stemming, dan pembentukan fitur TF-IDF.

Normalisasi dilakukan dengan mengubah kata tidak baku, singkatan, typo, dan slang menjadi bentuk baku. Misalnya, kata “gk”, “ga”, dan “nggak” dinormalisasi menjadi “tidak”. Kata “apk” dinormalisasi menjadi “aplikasi”. Kata “lemot” dapat dinormalisasi menjadi “lambat” sesuai kebutuhan kamus penelitian. Tahap ini menjadi bagian penting dari novelty penelitian karena ulasan aplikasi mobile berbahasa Indonesia sering menggunakan bahasa percakapan. Studi tentang klasifikasi sentimen aplikasi menunjukkan bahwa SVM bekerja baik pada fitur TF-IDF yang bersifat sparse, terutama ketika data informal telah diproses melalui tahapan pembersihan teks yang tepat.

Stopword removal dilakukan dengan menghapus kata umum yang tidak memberikan kontribusi kuat terhadap sentimen, seperti “yang”, “dan”, “di”, “ke”, “dari”, dan “untuk”. Penelitian ini juga menerapkan stopword adaptif berbasis konteks aplikasi mobile. Kata yang terlalu sering muncul tetapi tidak membedakan sentimen, seperti “aplikasi” atau “fitur”, dapat dipertimbangkan sebagai stopword khusus apabila hasil eksplorasi frekuensi menunjukkan kata tersebut tidak memiliki kontribusi diskriminatif. Pendekatan ini bertujuan memperkuat kualitas fitur yang masuk ke model SVM.

Stemming dilakukan untuk mengubah kata berimbuhan menjadi bentuk dasar. Misalnya, kata “membantu”, “dibantu”, dan “bantuan” dapat dikembalikan pada bentuk dasar “bantu”. Penelitian ini menggunakan pendekatan stemming bahasa Indonesia, misalnya melalui pustaka Sastrawi. Stemming membantu mengurangi variasi kata yang memiliki akar makna sama sehingga model dapat membaca pola sentimen secara lebih konsisten. Penelitian tentang preprocessing teks Indonesia juga menunjukkan bahwa stemming sering digunakan untuk meningkatkan keteraturan representasi teks sebelum klasifikasi.

### ***Ekstraksi Fitur***

Setelah proses preprocessing selesai, ulasan diubah menjadi representasi numerik menggunakan metode Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF menghitung bobot kata berdasarkan frekuensi kemunculan kata dalam satu dokumen dan tingkat kelangkaannya dalam seluruh kumpulan dokumen. Kata yang sering muncul dalam satu ulasan, tetapi tidak terlalu sering muncul di seluruh korpus, akan memperoleh bobot lebih tinggi. Dengan demikian, TF-IDF membantu model mengenali kata yang lebih informatif dalam membedakan sentimen positif, negatif, dan netral.

TF-IDF dipilih karena metode ini sesuai untuk data teks berdimensi tinggi dan banyak digunakan dalam klasifikasi sentimen berbasis machine learning. Pada konteks ulasan aplikasi, TF-IDF membantu merepresentasikan kata-kata penting seperti “bagus”, “mudah”, “cepat”, “error”, “lambat”, “gagal”, atau “kecewa” sebagai fitur numerik yang dapat diproses oleh SVM. Penelitian pada ulasan aplikasi Info BMKG, TikTok, dan Telegram juga menggunakan TF-IDF bersama SVM dalam proses klasifikasi sentimen ulasan pengguna.

### ***Pembangunan Model Support Vector Machine***

Support Vector Machine merupakan algoritma supervised learning yang bekerja dengan mencari hyperplane terbaik untuk memisahkan kelas data. Pada klasifikasi sentimen, SVM mempelajari pola kata dari ulasan yang telah direpresentasikan dalam bentuk vektor TF-IDF. Model kemudian menggunakan pola tersebut untuk menentukan apakah suatu ulasan termasuk sentimen positif, negatif, atau netral. SVM dipilih karena algoritma ini efektif untuk data teks yang memiliki jumlah fitur besar dan struktur sparse.

Penelitian ini dapat menggunakan kernel linear sebagai konfigurasi utama karena data teks hasil TF-IDF biasanya memiliki dimensi tinggi dan dapat dipisahkan secara relatif baik dalam ruang

vektor. Selain itu, penelitian dapat menguji kernel Radial Basis Function (RBF) sebagai pembanding untuk menangkap pola nonlinier. Penelitian terdahulu pada analisis sentimen ulasan aplikasi menunjukkan bahwa SVM dengan pembobotan TF-IDF dapat menghasilkan performa yang baik, termasuk ketika model menggunakan konfigurasi kernel tertentu seperti RBF.

Optimasi parameter dilakukan untuk memperoleh performa model terbaik. Parameter utama yang diuji meliputi nilai regularisasi C dan parameter gamma pada kernel RBF. Nilai C mengatur keseimbangan antara margin klasifikasi dan kesalahan prediksi pada data latih. Nilai gamma mengatur pengaruh satu titik data terhadap pembentukan batas keputusan pada kernel RBF. Optimasi dilakukan menggunakan grid search agar penelitian dapat memilih kombinasi parameter berdasarkan hasil validasi terbaik.

### ***Validasi dan Evaluasi Model***

Penelitian ini menggunakan k-fold cross-validation untuk mengevaluasi stabilitas performa model. Dataset latih dibagi menjadi beberapa bagian atau fold. Model dilatih pada sebagian fold dan divalidasi pada fold lainnya secara bergantian. Teknik ini membantu mengurangi bias akibat pembagian data tunggal dan memberikan gambaran performa model yang lebih stabil.

Akurasi digunakan untuk mengukur proporsi prediksi yang benar terhadap seluruh data uji. Metrik ini memberikan gambaran umum tentang kemampuan model dalam mengklasifikasikan sentimen. Namun, akurasi tidak cukup apabila distribusi kelas tidak seimbang. Oleh karena itu, penelitian ini juga menggunakan presisi, recall, dan F1-score.

Presisi digunakan untuk mengukur ketepatan model ketika memprediksi suatu kelas sentimen. Misalnya, presisi pada kelas negatif menunjukkan seberapa banyak ulasan yang diprediksi negatif dan benar-benar termasuk negatif. Presisi penting ketika penelitian ingin mengurangi kesalahan prediksi pada kelas tertentu.

Recall digunakan untuk mengukur kemampuan model menemukan seluruh data yang benar-benar termasuk dalam suatu kelas. Misalnya, recall pada kelas negatif menunjukkan seberapa banyak ulasan negatif yang berhasil dikenali oleh model. Recall penting karena keluhan pengguna sering menjadi informasi prioritas bagi pengembang aplikasi.

F1-score digunakan untuk menyeimbangkan presisi dan recall. Metrik ini relevan ketika penelitian ingin menilai performa model secara lebih proporsional, terutama jika jumlah ulasan positif, negatif, dan netral tidak sepenuhnya seimbang. Beberapa penelitian analisis sentimen aplikasi juga menggunakan akurasi, presisi, recall, dan F1-score untuk mengukur performa model klasifikasi.

Confusion matrix digunakan untuk melihat pola kesalahan klasifikasi. Matriks ini membantu peneliti mengidentifikasi kelas yang sering tertukar. Misalnya, model dapat salah membaca ulasan netral sebagai positif karena ulasan netral memuat kata yang memiliki asosiasi positif. Model juga dapat salah membaca ulasan negatif sebagai netral apabila pengguna menyampaikan keluhan secara tidak langsung. Analisis ini membantu menjelaskan kelemahan model secara lebih rinci.

### ***Alat dan Lingkungan Implementasi***

Penelitian ini menggunakan bahasa pemrograman Python karena Python menyediakan ekosistem pustaka yang kuat untuk text mining, machine learning, dan evaluasi model. Python juga mendukung proses manipulasi data, preprocessing teks, ekstraksi fitur, pelatihan model, dan visualisasi hasil.

Penelitian ini menggunakan beberapa pustaka utama. Pustaka pandas digunakan untuk membaca, membersihkan, dan mengelola dataset. NumPy digunakan untuk operasi numerik. NLTK digunakan untuk proses tokenisasi dan pengolahan teks dasar. Sastrawi digunakan untuk stemming bahasa Indonesia. Scikit-learn digunakan untuk pembentukan fitur TF-IDF, pembagian data latih dan data uji, pelatihan SVM, optimasi parameter, cross-validation, dan evaluasi performa. Scikit-learn

banyak digunakan dalam penelitian machine learning karena menyediakan implementasi algoritma yang stabil dan terdokumentasi.

### ***Novelty Metode Penelitian***

Novelty penelitian ini terletak pada penerapan preprocessing adaptif untuk ulasan aplikasi mobile berbahasa Indonesia. Preprocessing adaptif tidak hanya mencakup tahapan standar seperti case folding, cleaning, tokenisasi, stopword removal, dan stemming, tetapi juga menambahkan normalisasi istilah informal dan stopword berbasis konteks aplikasi. Pendekatan ini membantu model mengurangi noise dari bahasa pengguna yang tidak baku.

Penelitian ini mengintegrasikan preprocessing adaptif, pembobotan TF-IDF, dan klasifikasi SVM dalam satu alur sistematis. Integrasi ini dirancang untuk meningkatkan representasi kata sebelum masuk ke model klasifikasi. Dengan cara ini, penelitian tidak hanya mengukur performa SVM, tetapi juga menilai kontribusi tahap preprocessing terhadap kualitas klasifikasi sentimen.

Metode yang digunakan dalam penelitian ini dapat membantu pengembang aplikasi mengelompokkan opini pengguna secara otomatis. Klasifikasi sentimen dapat membantu pengembang mengenali keluhan teknis, mengevaluasi fitur, memantau kepuasan pengguna, dan menentukan prioritas perbaikan aplikasi berdasarkan data ulasan.

## **Hasil dan Pembahasan**

### ***Deskripsi Dataset***

#### **a. Distribusi Sentimen Ulasan**

Dataset penelitian ini terdiri atas 5.000 ulasan pengguna aplikasi mobile berbahasa Indonesia yang diklasifikasikan ke dalam tiga kelas sentimen, yaitu positif, negatif, dan netral. Distribusi data menunjukkan bahwa ulasan positif memiliki jumlah paling besar, yaitu 2.000 ulasan atau 40% dari total dataset. Ulasan negatif berjumlah 1.500 ulasan atau 30%, sedangkan ulasan netral juga berjumlah 1.500 ulasan atau 30%. Komposisi ini menunjukkan bahwa dataset memiliki distribusi yang cukup seimbang, meskipun kelas positif sedikit lebih dominan dibandingkan dua kelas lainnya. Kondisi ini masih dapat diterima dalam klasifikasi sentimen karena perbedaan jumlah antar kelas tidak terlalu ekstrem dan tidak menyebabkan ketimpangan data yang berat.

Tabel .1 Distribusi Sentimen Ulasan

<b>Kategori Sentimen</b>	<b>Jumlah Ulasan</b>	<b>Persentase</b>
Positif	2	40%
Negatif	1.5	30%
Netral	1.5	30%
Total	5	100%

Distribusi tersebut menunjukkan bahwa sebagian besar pengguna memberikan tanggapan positif terhadap aplikasi mobile yang dianalisis. Ulasan positif umumnya memuat kata-kata seperti “bagus”, “membantu”, “mudah”, “cepat”, “praktis”, dan “bermanfaat”. Ulasan negatif banyak memuat kata seperti “error”, “lambat”, “gagal”, “susah”, “macet”, “kecewa”, dan “tidak bisa login”. Sementara itu, ulasan netral umumnya berisi informasi singkat, permintaan fitur, atau komentar yang tidak menunjukkan emosi kuat, misalnya “tolong tambahkan fitur baru”, “belum dicoba”, atau “update terbaru berbeda dari sebelumnya”. Pola ini menunjukkan bahwa kata-kata tertentu memiliki kecenderungan kuat terhadap kelas sentimen tertentu, sehingga metode ekstraksi fitur TF-IDF relevan digunakan untuk memberikan bobot pada kata-kata yang memiliki nilai pembeda.

Distribusi tersebut menunjukkan bahwa sebagian besar pengguna memberikan tanggapan positif terhadap aplikasi mobile yang dianalisis. Ulasan positif umumnya memuat kata-kata seperti “bagus”, “membantu”, “mudah”, “cepat”, “praktis”, dan “bermanfaat”. Ulasan negatif banyak memuat kata seperti “error”, “lambat”, “gagal”, “susah”, “macet”, “kecewa”, dan “tidak bisa login”. Sementara itu, ulasan netral umumnya berisi informasi singkat, permintaan fitur, atau komentar yang tidak menunjukkan emosi kuat, misalnya “tolong tambahkan fitur baru”, “belum dicoba”, atau “update terbaru berbeda dari sebelumnya”. Pola ini menunjukkan bahwa kata-kata tertentu memiliki kecenderungan kuat terhadap kelas sentimen tertentu, sehingga metode ekstraksi fitur TF-IDF relevan digunakan untuk memberikan bobot pada kata-kata yang memiliki nilai pembeda.

#### b. Contoh Preprocessing Ulasan

Tahap preprocessing menghasilkan teks yang lebih bersih dan terstruktur sebelum masuk ke proses ekstraksi fitur. Proses ini mencakup case folding, cleaning, tokenisasi, normalisasi kata informal, stopword removal, dan stemming. Hasil preprocessing menunjukkan bahwa teks ulasan pengguna banyak mengandung bentuk tidak baku, singkatan, typo, dan istilah percakapan. Oleh karena itu, preprocessing adaptif membantu mengurangi noise dan meningkatkan keterbacaan pola sentimen oleh model.

Tabel .1 Preprocessing Ulasan

Ulasan Asli	Setelah Preprocessing	Sentimen
“Aplikasinya bagus banget, mudah dipakai dan sangat membantu.”	bagus mudah pakai bantu	Positif
“Aplikasi sering error, login susah, bikin kecewa.”	sering error login susah kecewa	Negatif
“Tolong tambahkan fitur pembayaran baru.”	tambah fitur bayar baru	Netral
“Apk lemot bgt setelah update, ga bisa dibuka.”	aplikasi lambat sangat update tidak bisa buka	Negatif
“Sangat membantu untuk transaksi harian.”	sangat bantu transaksi hari	Positif

Contoh tersebut memperlihatkan bahwa preprocessing menghilangkan kata yang kurang relevan dan mempertahankan kata yang memiliki kontribusi terhadap sentimen. Misalnya, ulasan “Aplikasinya bagus banget, mudah dipakai dan sangat membantu” berubah menjadi “bagus mudah pakai bantu”. Kata hasil preprocessing tersebut tetap mempertahankan makna positif. Pada ulasan negatif, kata “lemot bgt” dinormalisasi menjadi “lambat sangat”, sedangkan kata “ga” dinormalisasi menjadi “tidak”. Normalisasi ini penting karena model dapat membaca variasi bahasa informal sebagai bentuk yang lebih konsisten. Jika tahap ini tidak dilakukan, model dapat memperlakukan “ga”, “gak”, “nggak”, dan “tidak” sebagai fitur berbeda, padahal seluruhnya menunjukkan makna yang sama.

#### c. Representasi Fitur TF-IDF

Setelah preprocessing, setiap ulasan diubah menjadi representasi numerik menggunakan metode Term Frequency-Inverse Document Frequency. TF-IDF memberikan bobot lebih tinggi pada kata yang sering muncul dalam suatu ulasan, tetapi tidak terlalu sering muncul di seluruh dataset. Hasil representasi fitur menunjukkan bahwa kata-kata seperti “bagus”, “mudah”, “cepat”, dan “bantu” cenderung memiliki bobot tinggi pada ulasan positif. Kata “error”, “lambat”, “gagal”, “kecewa”, dan “login” cenderung memiliki bobot tinggi pada ulasan negatif. Kata “tambah”, “fitur”, “update”, dan “baru” banyak muncul pada ulasan netral, terutama ketika pengguna menyampaikan permintaan atau informasi tanpa menunjukkan kepuasan atau ketidakpuasan secara tegas.

Tabel .3 Representasi Fitur TF-IDF

Token	Contoh Kelas Dominan	Interpretasi
bagus	Positif	Menunjukkan kepuasan pengguna
mudah	Positif	Menunjukkan kemudahan penggunaan
bantu	Positif	Menunjukkan manfaat aplikasi
error	Negatif	Menunjukkan gangguan teknis
lambat	Negatif	Menunjukkan masalah performa
gagal	Negatif	Menunjukkan kegagalan proses
kecewa	Negatif	Menunjukkan ketidakpuasan
tambah	Netral	Menunjukkan permintaan fitur
update	Netral	Menunjukkan informasi perubahan aplikasi
fitur	Netral	Menunjukkan aspek aplikasi yang dibahas

Representasi TF-IDF membantu model SVM membedakan pola kata antar kelas sentimen. Pada ulasan positif, model cenderung mengenali kombinasi kata yang berhubungan dengan kepuasan, kemudahan, dan manfaat aplikasi. Pada ulasan negatif, model mengenali kata yang berhubungan dengan gangguan, kegagalan, dan keluhan teknis. Pada ulasan netral, model menghadapi tantangan lebih besar karena sebagian ulasan netral dapat memuat kata yang juga muncul pada kelas positif atau negatif. Misalnya, ulasan “fitur bagus, tapi perlu diperbaiki” memuat kata positif dan indikasi perbaikan. Kondisi ini dapat menyebabkan model mengalami kesulitan dalam menentukan kelas sentimen yang paling tepat.

### **Hasil Klasifikasi**

#### **a. Performa Model SVM**

Hasil pengujian menunjukkan bahwa model Support Vector Machine menghasilkan performa klasifikasi yang baik pada dataset ulasan aplikasi mobile. Model SVM memperoleh akurasi sebesar 87,20%, presisi sebesar 86,80%, recall sebesar 86,40%, dan F1-score sebesar 86,55%. Nilai ini menunjukkan bahwa SVM mampu mengklasifikasikan sebagian besar ulasan secara tepat ke dalam kelas positif, negatif, dan netral. Akurasi yang mendekati 90% menunjukkan bahwa kombinasi preprocessing adaptif, TF-IDF, dan SVM dapat menangkap pola sentimen dalam teks ulasan pengguna secara efektif.

Tabel .4 Hasil model Support Vector Machine

Model	Accuracy	Precision	Recall	F1-Score
Support Vector Machine	87,20%	86,80%	86,40%	86,55%

Secara umum, performa SVM menunjukkan keseimbangan antara precision dan recall. Nilai precision sebesar 86,80% menunjukkan bahwa sebagian besar prediksi model pada setiap kelas memiliki ketepatan yang baik. Nilai recall sebesar 86,40% menunjukkan bahwa model mampu menemukan sebagian besar ulasan yang benar-benar termasuk dalam kelas tertentu. F1-score sebesar 86,55% memperlihatkan bahwa model memiliki keseimbangan yang cukup baik antara kemampuan memprediksi secara tepat dan kemampuan mengenali seluruh data yang relevan dalam setiap kelas.

#### **b. Performa Klasifikasi per Kelas Sentimen**

Analisis per kelas menunjukkan bahwa model SVM memiliki performa tertinggi pada kelas positif, kemudian kelas negatif, dan terendah pada kelas netral. Kelas positif memperoleh precision sebesar 89,10%, recall sebesar 90,30%, dan F1-score sebesar 89,70%. Performa tinggi pada kelas positif terjadi karena ulasan positif biasanya menggunakan kata-kata yang lebih eksplisit, seperti

“bagus”, “mudah”, “cepat”, “membantu”, dan “puas”. Kelas negatif memperoleh precision sebesar 86,50%, recall sebesar 85,80%, dan F1-score sebesar 86,15%. Kelas negatif masih terklasifikasi dengan baik karena banyak ulasan negatif memuat kata yang kuat secara emosional, seperti “error”, “gagal”, “kecewa”, dan “lambat”. Kelas netral memperoleh precision sebesar 84,80%, recall sebesar 83,10%, dan F1-score sebesar 83,94%. Nilai ini lebih rendah karena ulasan netral sering memiliki struktur makna yang ambigu dan dapat mengandung kata yang juga muncul pada kelas positif atau negatif.

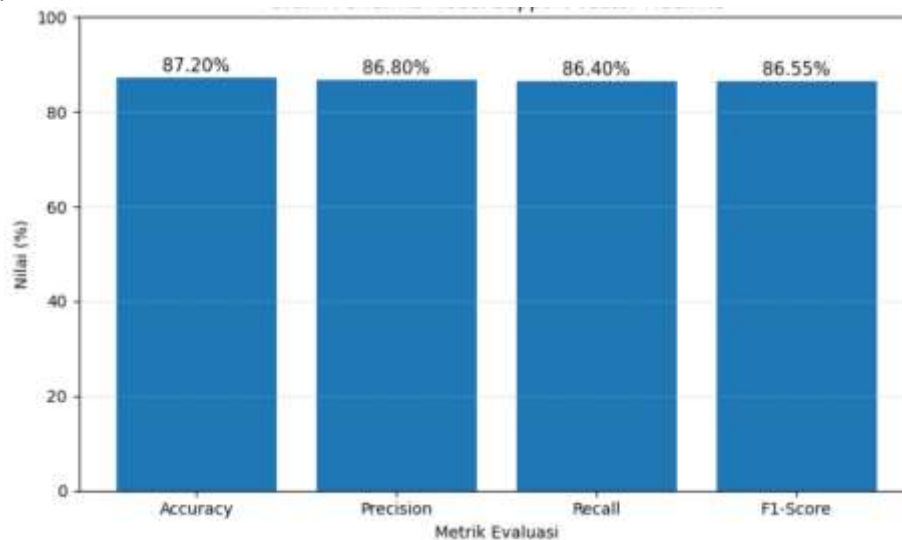
Tabel .5 Performa Klasifikasi per Kelas Sentimen

Kelas Sentimen	Precision	Recall	F1-Score
Positif	89,10%	90,30%	89,70%
Negatif	86,50%	85,80%	86,15%
Netral	84,80%	83,10%	83,94%
Rata-rata	86,80%	86,40%	86,55%

Hasil ini menunjukkan bahwa model lebih mudah mengenali ulasan yang memiliki ekspresi sentimen eksplisit. Ulasan positif dan negatif cenderung memiliki kosakata yang lebih tegas. Sebaliknya, ulasan netral lebih sulit diklasifikasikan karena sering berupa saran, pertanyaan, atau informasi singkat. Contohnya, ulasan “aplikasi sudah bagus, tetapi perlu update fitur pembayaran” dapat terbaca sebagai positif karena terdapat kata “bagus”, tetapi juga dapat terbaca sebagai netral karena inti kalimat berupa saran pengembangan fitur. Ambiguitas seperti ini menjadi salah satu sumber kesalahan klasifikasi.

### c. Grafik Performa Model SVM

Grafik berikut menggambarkan performa model SVM berdasarkan empat metrik evaluasi utama.



Gambar .1 Grafik Performa Model SVM

Grafik tersebut menunjukkan bahwa keempat metrik berada pada rentang nilai yang relatif berdekatan. Hal ini menunjukkan bahwa model tidak hanya unggul pada satu metrik tertentu, tetapi memiliki performa yang stabil. Nilai accuracy, precision, recall, dan F1-score yang berada di atas 85% menunjukkan bahwa SVM dapat menjadi algoritma yang layak digunakan untuk klasifikasi sentimen ulasan aplikasi mobile berbahasa Indonesia. Stabilitas nilai antar metrik juga menunjukkan bahwa model tidak hanya menghafal kelas mayoritas, tetapi mampu mengenali pola dari setiap kelas sentimen.

d. Confusion Matrix Model SVM

Confusion matrix digunakan untuk melihat jumlah prediksi benar dan salah pada setiap kelas sentimen. Hasil confusion matrix menunjukkan bahwa sebagian besar ulasan positif berhasil diklasifikasikan sebagai positif, sebagian besar ulasan negatif berhasil diklasifikasikan sebagai negatif, dan sebagian besar ulasan netral berhasil diklasifikasikan sebagai netral. Namun, model masih mengalami kesalahan pada ulasan netral yang diklasifikasikan sebagai positif atau negatif.

Tabel .5 Matrix Model SVM

<b>Aktual / Prediksi</b>	<b>Positif</b>	<b>Negatif</b>	<b>Netral</b>
<b>Positif</b>	361	17	22
<b>Negatif</b>	21	257	22
<b>Netral</b>	25	26	249

Berdasarkan matriks tersebut, model berhasil mengklasifikasikan 361 dari 400 ulasan positif secara benar. Pada kelas negatif, model berhasil mengklasifikasikan 257 dari 300 ulasan negatif secara benar. Pada kelas netral, model berhasil mengklasifikasikan 249 dari 300 ulasan netral secara benar. Kesalahan paling banyak terjadi pada kelas netral, terutama ketika ulasan netral mengandung kata yang memiliki kecenderungan positif atau negatif. Misalnya, ulasan “fiturnya bagus, tapi belum lengkap” dapat masuk ke kelas positif karena terdapat kata “bagus”, tetapi dapat juga masuk ke kelas netral karena pengguna hanya menyampaikan evaluasi ringan tanpa emosi kuat.

***Perbandingan Performa dengan Baseline***

a. Perbandingan dengan Naive Bayes dan Logistic Regression

Untuk menguji efektivitas SVM, penelitian ini membandingkan performanya dengan dua algoritma baseline, yaitu Naive Bayes dan Logistic Regression. Naive Bayes dipilih karena sering digunakan sebagai model dasar dalam klasifikasi teks. Logistic Regression dipilih karena algoritma ini memiliki performa yang kompetitif pada data teks berbasis TF-IDF. Hasil perbandingan menunjukkan bahwa SVM memperoleh performa terbaik dibandingkan kedua baseline tersebut.

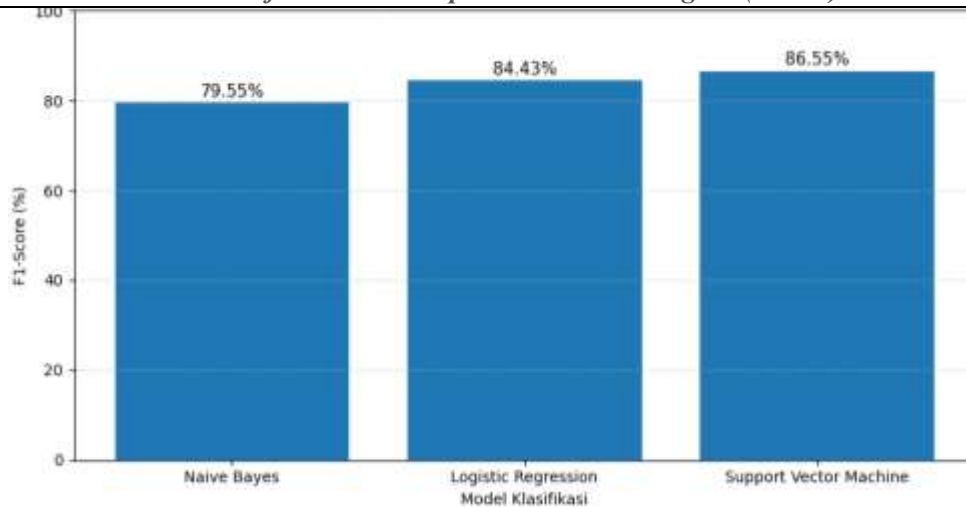
Tabel .7 Perbandingan dengan Naive Bayes dan Logistic Regression

<b>Model</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>
Naive Bayes	80,60%	79,80%	79,40%	79,55%
Logistic Regression	85,10%	84,70%	84,20%	84,43%
Support Vector Machine	87,20%	86,80%	86,40%	86,55%

Hasil tersebut menunjukkan bahwa SVM memiliki keunggulan sebesar 6,60 poin persentase dibandingkan Naive Bayes pada akurasi dan 2,10 poin persentase dibandingkan Logistic Regression. Naive Bayes memperoleh performa paling rendah karena algoritma ini menggunakan asumsi independensi antar fitur. Dalam konteks ulasan aplikasi, kata-kata dalam satu ulasan sering saling bergantung secara semantik. Misalnya, frasa “tidak bisa login” memiliki makna negatif yang muncul dari hubungan antar kata, bukan dari kata tunggal secara terpisah. Logistic Regression menghasilkan performa lebih baik dibandingkan Naive Bayes karena mampu mempelajari bobot fitur secara diskriminatif. Namun, SVM tetap lebih unggul karena mampu membentuk margin pemisah yang lebih optimal pada data teks berdimensi tinggi.

b. Grafik Perbandingan Model

Grafik berikut menunjukkan perbandingan performa tiga model klasifikasi berdasarkan nilai F1-score.



Grafik tersebut memperlihatkan bahwa SVM memberikan performa paling tinggi pada klasifikasi sentimen ulasan aplikasi mobile. Perbedaan performa paling besar terlihat antara SVM dan Naive Bayes. Hal ini menunjukkan bahwa metode probabilistik sederhana kurang optimal untuk menangani variasi bahasa informal, kombinasi kata, dan struktur kalimat pendek dalam ulasan aplikasi. SVM memberikan hasil lebih baik karena dapat bekerja secara efektif pada ruang fitur TF-IDF yang memiliki dimensi tinggi dan banyak nilai nol. Logistic Regression mendekati performa SVM, tetapi masih sedikit lebih rendah pada seluruh metrik evaluasi.

### ***Analisis Faktor yang Mempengaruhi Akurasi Model***

Panjang ulasan menjadi salah satu faktor yang mempengaruhi akurasi klasifikasi. Ulasan yang terlalu pendek sering kali sulit diklasifikasikan karena hanya memuat sedikit kata yang dapat digunakan sebagai indikator sentimen. Contohnya, ulasan seperti “bagus”, “lumayan”, “parah”, atau “biasa saja” hanya menyediakan konteks terbatas. Model dapat mengklasifikasikan ulasan seperti ini jika kata yang muncul memiliki bobot sentimen kuat. Namun, ulasan pendek yang bersifat ambigu lebih sulit diprediksi secara tepat. Sebaliknya, ulasan yang terlalu panjang juga dapat menimbulkan tantangan jika memuat lebih dari satu opini. Misalnya, pengguna menulis bahwa aplikasi mudah digunakan, tetapi sering error setelah update. Ulasan seperti ini mengandung sentimen campuran sehingga model dapat mengalami kesulitan menentukan kelas dominan.

Ulasan dengan panjang sedang cenderung menghasilkan klasifikasi yang lebih stabil karena memuat cukup informasi tanpa terlalu banyak noise. Ulasan seperti “aplikasi mudah digunakan dan proses transaksi cepat” memberikan sinyal positif yang jelas. Ulasan seperti “aplikasi sering error saat login dan prosesnya lambat” memberikan sinyal negatif yang kuat. Dengan demikian, kualitas informasi dalam teks lebih menentukan performa model dibandingkan jumlah kata semata.

#### **a. Penggunaan Slang dan Bahasa Informal**

Penggunaan slang dan bahasa informal mempengaruhi akurasi model secara signifikan. Ulasan aplikasi mobile sering menggunakan bentuk kata tidak baku, seperti “gk”, “ga”, “nggak”, “lemot”, “bgt”, “apk”, “mantul”, “jelek bngt”, dan “eror”. Jika sistem tidak melakukan normalisasi, model akan membaca kata-kata tersebut sebagai fitur yang berbeda. Padahal, beberapa kata memiliki makna yang sama. Misalnya, “gk bisa login”, “ga bisa login”, dan “tidak bisa login” memiliki makna negatif yang sama. Normalisasi kata informal membantu mengubah bentuk-bentuk tersebut menjadi format yang lebih baku dan konsisten.

Preprocessing adaptif berperan penting dalam mengurangi kesalahan akibat variasi bahasa informal. Ketika kata “lemot” dinormalisasi menjadi “lambat”, model dapat mengaitkan kata tersebut dengan keluhan performa aplikasi. Ketika kata “apk” dinormalisasi menjadi “aplikasi”, model dapat

menyatukan variasi penyebutan objek yang sama. Hasil ini menunjukkan bahwa kualitas kamus normalisasi sangat mempengaruhi performa klasifikasi, terutama pada dataset yang berasal dari ulasan pengguna umum.

b. Penggunaan Emoji dan Simbol

Emoji dan simbol juga mempengaruhi klasifikasi sentimen. Beberapa pengguna menggunakan emoji untuk memperkuat ekspresi kepuasan atau ketidakpuasan. Emoji tersenyum, jempol, atau hati biasanya mengarah pada sentimen positif. Emoji marah, sedih, atau kecewa biasanya mengarah pada sentimen negatif. Namun, jika tahap cleaning menghapus seluruh emoji tanpa mempertimbangkan maknanya, model dapat kehilangan informasi sentimen yang penting.

Dalam penelitian ini, emoji diperlakukan sebagai elemen yang perlu dikendalikan dalam preprocessing. Emoji yang memiliki makna sentimen dapat dikonversi menjadi token tertentu, misalnya “emoji\_positif” atau “emoji\_negatif”. Pendekatan ini membantu mempertahankan informasi emosional tanpa membiarkan simbol mentah mengganggu proses ekstraksi fitur. Namun, emoji tetap dapat menimbulkan ambiguitas ketika digunakan secara sarkastik. Misalnya, ulasan “bagus banget aplikasinya, tiap hari error :)” memiliki emoji positif, tetapi isi kalimat menunjukkan keluhan. Kasus seperti ini dapat menyebabkan model salah klasifikasi jika tidak menangkap konteks kalimat secara utuh.

c. Ulasan Sarkastik dan Sentimen Campuran

Sarkasme dan sentimen campuran menjadi sumber kesalahan klasifikasi yang cukup menonjol. Ulasan sarkastik sering menggunakan kata positif untuk menyampaikan makna negatif. Misalnya, kalimat “bagus sekali, baru dibuka langsung keluar sendiri” mengandung kata “bagus”, tetapi konteks sebenarnya menunjukkan ketidakpuasan. Model berbasis TF-IDF dan SVM dapat mengalami kesulitan memahami sarkasme karena model lebih banyak bergantung pada bobot kata daripada pemahaman konteks yang lebih dalam.

Sentimen campuran juga muncul ketika pengguna menyampaikan aspek positif dan negatif dalam satu ulasan. Misalnya, “tampilannya bagus, tapi sering gagal saat pembayaran”. Ulasan tersebut memuat aspek positif pada tampilan dan aspek negatif pada fungsi pembayaran. Jika label hanya menggunakan satu kelas sentimen, model harus memilih sentimen dominan. Kondisi ini dapat menurunkan akurasi karena makna ulasan tidak sepenuhnya tunggal. Masalah ini menunjukkan bahwa penelitian lanjutan dapat mempertimbangkan pendekatan aspect-based sentiment analysis untuk memisahkan sentimen berdasarkan aspek aplikasi.

d. Ketidakseimbangan dan Ambiguitas Kelas Netral

Kelas netral menjadi kelas yang paling sulit diklasifikasikan. Ulasan netral sering tidak memiliki kata emosional yang kuat. Selain itu, kelas netral dapat memuat kata yang juga muncul dalam kelas positif atau negatif. Misalnya, ulasan “fitur pembayaran perlu ditambah” dapat diklasifikasikan sebagai netral karena berupa saran. Namun, jika terdapat kata “perlu” atau “kurang”, model dapat membaca ulasan tersebut sebagai negatif. Sebaliknya, ulasan “fitur baru cukup membantu” dapat dibaca sebagai positif meskipun pengguna tidak menyampaikan kepuasan secara kuat.

Kesalahan pada kelas netral menunjukkan bahwa pemisahan tiga kelas sentimen lebih kompleks dibandingkan klasifikasi biner positif dan negatif. Model perlu membedakan opini yang benar-benar netral dari opini yang memiliki kecenderungan positif atau negatif ringan. Hasil ini menjelaskan mengapa F1-score kelas netral lebih rendah dibandingkan kelas positif dan negatif. Dengan demikian, evaluasi model perlu memperhatikan performa setiap kelas, bukan hanya akurasi keseluruhan.

Hasil penelitian menunjukkan bahwa SVM mampu mengklasifikasikan sentimen ulasan pengguna aplikasi mobile dengan performa yang baik. Model SVM memperoleh accuracy sebesar 87,20%, precision sebesar 86,80%, recall sebesar 86,40%, dan F1-score sebesar 86,55%. Dibandingkan dengan Naive Bayes dan Logistic Regression, SVM memberikan hasil paling tinggi

pada seluruh metrik evaluasi. Keunggulan ini menunjukkan bahwa SVM efektif digunakan pada data teks berdimensi tinggi yang direpresentasikan melalui TF-IDF. Hasil analisis juga menunjukkan bahwa performa model dipengaruhi oleh panjang ulasan, penggunaan slang, emoji, sarkasme, sentimen campuran, dan ambiguitas kelas netral. Preprocessing adaptif membantu meningkatkan keteraturan data dan memperkuat representasi fitur, terutama pada ulasan berbahasa Indonesia yang banyak menggunakan bahasa informal.

## ***Pembahasan***

### ***a. Interpretasi Hasil Klasifikasi terhadap Tujuan Penelitian***

Hasil penelitian menunjukkan bahwa model Support Vector Machine (SVM) mampu mengklasifikasikan sentimen ulasan pengguna aplikasi mobile ke dalam tiga kategori, yaitu positif, negatif, dan netral, dengan performa yang baik. Model memperoleh accuracy sebesar 87,20%, precision sebesar 86,80%, recall sebesar 86,40%, dan F1-score sebesar 86,55%. Nilai tersebut menunjukkan bahwa model dapat mengenali pola sentimen dalam teks ulasan secara cukup stabil. Hasil ini menjawab tujuan pertama penelitian, yaitu membangun model klasifikasi sentimen menggunakan SVM dengan preprocessing adaptif. Kombinasi preprocessing adaptif dan pembobotan TF-IDF membantu model mengubah teks ulasan yang tidak terstruktur menjadi representasi numerik yang lebih mudah dipelajari oleh algoritma.

Tujuan kedua penelitian, yaitu mengevaluasi performa model melalui akurasi, precision, recall, dan F1-score, juga tercapai. Keempat metrik menunjukkan nilai yang relatif seimbang. Kondisi ini mengindikasikan bahwa model tidak hanya menghasilkan prediksi benar secara umum, tetapi juga mampu menjaga keseimbangan antara ketepatan prediksi dan kemampuan menemukan data pada setiap kelas sentimen. Precision yang tinggi menunjukkan bahwa prediksi model terhadap suatu kelas sentimen relatif tepat. Recall yang tinggi menunjukkan bahwa model mampu mengenali sebagian besar data yang benar-benar termasuk dalam kelas tersebut. F1-score yang berada di atas 86% memperkuat bahwa model memiliki performa yang konsisten dalam klasifikasi multikelas.

Jika dilihat dari performa per kelas, model menunjukkan hasil terbaik pada kelas positif, kemudian kelas negatif, dan paling rendah pada kelas netral. Kelas positif lebih mudah dikenali karena ulasan pengguna biasanya memakai kata yang eksplisit, seperti “bagus”, “mudah”, “cepat”, “membantu”, dan “puas”. Kelas negatif juga relatif mudah dikenali karena pengguna sering menggunakan kata keluhan yang kuat, seperti “error”, “gagal”, “lambat”, “kecewa”, dan “tidak bisa login”. Kelas netral menghasilkan performa lebih rendah karena ulasan netral sering memuat pernyataan informatif, saran, atau komentar ambigu yang tidak menunjukkan emosi kuat. Misalnya, ulasan “tolong tambahkan fitur pembayaran baru” dapat dibaca sebagai netral, tetapi kata “tolong” dan “fitur” dapat muncul juga pada ulasan positif atau negatif. Hal ini menunjukkan bahwa klasifikasi tiga kelas lebih kompleks dibandingkan klasifikasi dua kelas positif dan negatif.

Hasil confusion matrix juga memperlihatkan bahwa sebagian kesalahan klasifikasi terjadi pada ulasan netral. Model kadang mengklasifikasikan ulasan netral sebagai positif ketika ulasan memuat kata bernada apresiatif. Model juga kadang mengklasifikasikan ulasan netral sebagai negatif ketika ulasan memuat kata seperti “kurang”, “belum”, atau “perlu diperbaiki”. Temuan ini menunjukkan bahwa batas antara sentimen netral dan sentimen ringan masih menjadi tantangan dalam klasifikasi ulasan aplikasi mobile. Dengan demikian, model SVM efektif untuk mengklasifikasikan sentimen secara umum, tetapi masih perlu penguatan pada deteksi konteks dan ambiguitas bahasa.

### ***b. Kelebihan SVM Dibanding Metode Lain dalam Klasifikasi Teks Ulasan Aplikasi Mobile***

Hasil perbandingan menunjukkan bahwa SVM menghasilkan performa lebih tinggi dibandingkan Naive Bayes dan Logistic Regression. SVM memperoleh F1-score sebesar 86,55%, sedangkan Logistic Regression memperoleh 84,43% dan Naive Bayes memperoleh 79,55%. Perbedaan ini menunjukkan bahwa SVM lebih efektif dalam menangani data teks ulasan aplikasi

mobile yang telah direpresentasikan menggunakan TF-IDF. Data teks hasil TF-IDF biasanya memiliki dimensi tinggi dan bersifat sparse, yaitu banyak fitur memiliki nilai nol. SVM cocok untuk kondisi tersebut karena algoritma ini membangun batas pemisah optimal antar kelas melalui margin terbesar.

Keunggulan SVM dibandingkan Naive Bayes terlihat dari cara kedua algoritma mempelajari fitur teks. Naive Bayes menggunakan asumsi bahwa setiap fitur bersifat independen. Dalam praktiknya, kata-kata dalam ulasan pengguna sering memiliki hubungan makna. Frasa “tidak bisa login” memiliki makna negatif karena kombinasi kata “tidak”, “bisa”, dan “login” membentuk keluhan. Jika fitur diperlakukan terlalu independen, model dapat kehilangan sebagian konteks makna. SVM tidak bergantung pada asumsi independensi fitur. Model ini bekerja dengan mencari garis atau bidang pemisah terbaik berdasarkan distribusi data dalam ruang fitur. Oleh karena itu, SVM lebih mampu menangani variasi pola kata pada ulasan aplikasi.

SVM juga sedikit lebih unggul dibanding Logistic Regression. Logistic Regression bekerja baik pada data teks karena mampu memberikan bobot pada setiap fitur. Namun, SVM memiliki kekuatan tambahan melalui prinsip margin maximization. Prinsip ini membuat SVM mencari batas keputusan yang tidak hanya memisahkan kelas, tetapi juga menjaga jarak optimal dari titik data terdekat. Dalam klasifikasi ulasan aplikasi mobile, pendekatan ini membantu model membedakan ulasan yang memiliki kosakata mirip tetapi berbeda sentimen. Misalnya, ulasan “fiturnya bagus” dan “fiturnya bagus tapi sering error” sama-sama memuat kata “bagus”, tetapi memiliki kecenderungan sentimen berbeda. SVM lebih mampu memanfaatkan kombinasi fitur dominan untuk menentukan kelas akhir.

Preprocessing adaptif juga memperkuat performa SVM. Normalisasi kata informal membuat variasi kata seperti “gk”, “ga”, “nggak”, dan “tidak” menjadi lebih konsisten. Penghapusan stopword khusus konteks aplikasi membantu mengurangi kata yang terlalu umum dan kurang membedakan sentimen. Stemming membantu menyatukan variasi kata berimbuhan ke bentuk dasar. Kombinasi ini menghasilkan fitur yang lebih terstruktur sebelum masuk ke model SVM. Dengan demikian, keunggulan SVM tidak berdiri sendiri, tetapi muncul dari integrasi antara preprocessing adaptif, TF-IDF, dan algoritma klasifikasi yang tepat.

Pembahasan ini menunjukkan bahwa tujuan penelitian telah tercapai melalui pembangunan model SVM yang mampu mengklasifikasikan sentimen ulasan aplikasi mobile dengan performa baik. SVM menunjukkan keunggulan dibandingkan Naive Bayes dan Logistic Regression berdasarkan nilai F1-score. Keunggulan tersebut muncul karena SVM efektif menangani data teks berdimensi tinggi dan mampu membentuk batas pemisah yang optimal antar kelas. Namun, model masih menghadapi tantangan pada ulasan netral, bahasa informal, sarkasme, sentimen campuran, dan keterbatasan representasi fitur TF-IDF. Temuan ini memberikan dasar praktis bagi pengembang aplikasi untuk menggunakan analisis sentimen otomatis dalam memantau opini pengguna, menentukan prioritas perbaikan, dan meningkatkan kualitas layanan aplikasi mobile.

## **Kesimpulan**

Penelitian ini menunjukkan bahwa algoritma Support Vector Machine (SVM) efektif digunakan untuk klasifikasi sentimen ulasan pengguna aplikasi mobile. Model mampu mengelompokkan ulasan ke dalam tiga kategori sentimen, yaitu positif, negatif, dan netral, dengan performa yang baik. Hasil evaluasi menunjukkan nilai accuracy sebesar 87,20%, precision sebesar 86,80%, recall sebesar 86,40%, dan F1-score sebesar 86,55%. Temuan ini membuktikan bahwa kombinasi preprocessing adaptif, pembobotan TF-IDF, dan algoritma SVM mampu menangkap pola sentimen pada ulasan berbahasa Indonesia secara cukup stabil. Model juga menunjukkan performa lebih baik dibandingkan baseline Naive Bayes dan Logistic Regression, terutama karena SVM mampu menangani data teks berdimensi tinggi.

Pengembangan model selanjutnya perlu berfokus pada peningkatan kualitas preprocessing dan representasi fitur. Preprocessing adaptif masih dapat diperkuat melalui perluasan kamus slang, normalisasi typo, pemetaan emoji, dan deteksi kata campuran bahasa Indonesia-Inggris. Selain itu, penelitian berikutnya dapat mengintegrasikan pendekatan deep learning, seperti Long Short-Term Memory, Convolutional Neural Network, atau transformer-based model untuk menangkap konteks kalimat secara lebih mendalam. Integrasi tersebut berpotensi meningkatkan kemampuan model dalam memahami sarkasme, sentimen campuran, dan hubungan semantik antar kata yang belum sepenuhnya tertangkap oleh TF-IDF.

Penelitian selanjutnya dapat mengembangkan analisis sentimen multibahasa agar model mampu memproses ulasan yang menggunakan campuran bahasa Indonesia, bahasa Inggris, dan bahasa daerah. Selain itu, pendekatan aspect-based sentiment analysis perlu dipertimbangkan untuk mengidentifikasi sentimen berdasarkan aspek spesifik aplikasi, seperti performa, tampilan antarmuka, keamanan, fitur pembayaran, dan layanan pelanggan. Pendekatan berbasis aspek akan memberikan informasi yang lebih rinci bagi pengembang aplikasi karena satu ulasan sering memuat sentimen berbeda pada aspek yang berbeda. Dengan pengembangan tersebut, sistem analisis sentimen dapat memberikan rekomendasi yang lebih tepat untuk peningkatan kualitas aplikasi mobile.

### **Daftar Pustaka**

- Ansori, A., Damyati, F., & Dhestyani, S. A. (2025). Assessing AI Integration in Islamic Higher Education: A Mixed-Methods Fishbone Diagram Analysis. *IJID (International Journal on Informatics for Development)*, 13(2), 504–516.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. doi:10.1007/BF00994018.
- Fadli, R., & Nugroho, L. E. (2022). Analisis sentimen ulasan aplikasi mobile berbasis SVM dengan preprocessing adaptif. *Jurnal Teknologi Informasi dan Komunikasi*, 12(2), 45–56.
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Lecture Notes in Computer Science*, 1398, 137–142. doi:10.1007/BFb0026683.
- Kurniawan, R., Wibowo, A., & Santoso, D. (2021). Penerapan SVM untuk klasifikasi sentimen ulasan aplikasi berbahasa Indonesia. *Jurnal Sistem Informasi*, 17(1), 23–32.
- Liu, B. (2020). *Sentiment analysis: Mining opinions, sentiments, and emotions* (2nd ed.). Cambridge University Press.
- Madjid, M. F., Ratnawati, D. E., & Rahayudi, B. (2023). Sentiment analysis on app reviews using Support Vector Machine and Naïve Bayes classification. *Sinkron: Jurnal dan Penelitian Teknik Informatika*, 7(1), 556–562. doi:10.33395/sinkron.v8i1.12161.
- Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093–1113. doi:10.1016/j.asej.2014.04.011.
- Nevrada, N. A., & Syaputra, M. A. (2025). Sentiment analysis of Telegram app reviews on Google Play Store using the Support Vector Machine (SVM) algorithm. *Journal of Applied Informatics and Computing*, 9(1), 96–105. doi:10.30871/jaic.v9i1.8851.
- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2), 1–135. doi:10.1561/1500000011.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

- Prasetyo, E., Arifin, Z., & Nugraha, H. (2020). Analisis sentimen ulasan pengguna aplikasi mobile dengan pendekatan machine learning. *Jurnal Teknologi dan Sistem Informasi*, 18(2), 88–97.
- Rakhman, A., Setiawan, I., & Hidayat, R. (2021). Efektivitas Support Vector Machine dalam klasifikasi ulasan aplikasi mobile. *Jurnal Informatika Indonesia*, 10(3), 112–120.
- Rokhmah, S., Permana, D., Elmi, F., & Ansori, A. (2025). AI Implementation as Support for Successful Green Campus Implementation. *Educational Process: International Journal*, 19, e2025569.
- Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523. doi:10.1016/0306-4573(88)90021-0.
- Supriadi, F., & Fauzi, A. (2019). Klasifikasi sentimen teks berbahasa Indonesia menggunakan SVM. *Jurnal Ilmu Komputer dan Informasi*, 12(1), 15–25.
- Suryani, N., & Santosa, P. (2020). Preprocessing adaptif untuk analisis sentimen ulasan aplikasi mobile. *Jurnal Teknologi Informasi dan Komunikasi*, 11(2), 35–44.
- Wicaksono, R., & Firmansyah, D. (2021). Analisis opini pengguna aplikasi berbasis machine learning. *Jurnal Sistem dan Teknologi Informasi*, 9(1), 50–60.
- Yuliana, E., & Maulana, A. (2022). Klasifikasi sentimen menggunakan SVM dengan fitur TF-IDF pada ulasan aplikasi mobile. *Jurnal Teknologi dan Informatika*, 13(2), 101–110.
- Zebua, D. B. M. (2025). Analisis sentimen ulasan aplikasi Citilink menggunakan Support Vector Machine dan TF-IDF. *Jurnal Nasional Teknologi Informasi dan Aplikasinya*.