

Analisis Sentimen Ulasan DANA Dari Play Store dengan Metode SVM, Logistic Regression, Naive Bayes dan KNN

Anwar Dwiky Fitriyanto, Purwanto*

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia

Email: ¹111202214611@mhs.dinus.ac.id, ^{2,*}purwanto@dsn.dinus.ac.id

Email Penulis Korespondensi: purwanto@dsn.dinus.ac.id

Submitted: 22/11/2025; Accepted: 26/12/2025; Published: 26/12/2025

Abstrak—Pertumbuhan layanan transaksi digital di Indonesia mendorong meningkatnya penggunaan dompet digital seperti DANA, sehingga jumlah ulasan dari para pengguna juga terus bertambah. Banyaknya ulasan ini membuat proses membaca, memilah, dan memahami kecenderungan sentimen secara manual menjadi tidak efisien dan rawan bias. Tantangan ini semakin besar karena ulasan dalam bahasa Indonesia sering kali mengandung bahasa tidak baku, singkatan, serta penggunaan slang yang menyulitkan sistem untuk mengenali konteks secara tepat. Selain itu, volume data yang besar turut memengaruhi proses pemodelan, di mana ketersediaan data yang lebih banyak umumnya meningkatkan kemampuan model dalam mempelajari pola sentimen dengan lebih stabil. Untuk menjawab permasalahan tersebut, penelitian ini mengembangkan sistem klasifikasi sentimen berbasis machine learning yang mampu memproses ulasan dalam jumlah besar secara otomatis melalui representasi fitur TF-IDF. Dalam penelitian ini, data ulasan dikumpulkan dari Google Play Store, melalui tahap pembersihan dan prapemrosesan sebelum diubah menjadi vektor fitur TF-IDF. Empat algoritma utama diuji, yaitu Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan Naive Bayes, yang kemudian dievaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score. Hasil pengujian menunjukkan bahwa TF-IDF mampu menggambarkan hubungan antar kata dengan cukup baik, sementara algoritma Naive Bayes memberikan performa paling stabil dibandingkan tiga metode lainnya dengan tingkat akurasi mencapai 79,80%. Model yang dikembangkan ini dapat membantu perusahaan dalam memahami persepsi pengguna secara lebih cepat dan objektif, serta mendukung pengambilan keputusan berbasis data untuk meningkatkan kualitas layanan.

Kata Kunci: Sentimen Analisis; DANA; TF-IDF; SVM; KNN; Naive Bayes

Abstract—The growth of digital transaction services in Indonesia has driven the increased use of digital wallets such as DANA, resulting in a continuous increase in the number of user reviews. The large number of reviews makes the process of manually reading, sorting, and understanding sentiment trends inefficient and prone to bias. This challenge is exacerbated by the fact that reviews in Indonesian often contain non-standard language, abbreviations, and slang, making it difficult for the system to accurately recognize the context. In addition, the large volume of data also affects the modeling process, where the availability of more data generally improves the model's ability to learn sentiment patterns more stably. To address these issues, this study developed a machine learning-based sentiment classification system capable of automatically processing large numbers of reviews through TF-IDF feature representation. In this study, review data was collected from the Google Play Store, through a cleaning and preprocessing stage before being converted into TF-IDF feature vectors. Four main algorithms were tested, namely Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Naive Bayes, which were then evaluated using accuracy, precision, recall, and F1-score metrics. The test results showed that TF-IDF was able to describe the relationship between words quite well, while the Naive Bayes algorithm provided the most stable performance compared to the other three methods, with an accuracy rate of 79.80%. The model developed can help companies understand user perceptions more quickly and objectively, as well as support data-driven decision making to improve service quality.

Keywords: Sentiment Analysis; DANA; TF-IDF; SVM; KNN; Naive Bayes

1. PENDAHULUAN

Perkembangan teknologi digital dalam beberapa tahun terakhir membawa perubahan besar pada cara masyarakat Indonesia melakukan transaksi keuangan. Pergeseran dari pembayaran tunai menuju pembayaran digital semakin terlihat seiring meningkatnya penggunaan ponsel pintar dan layanan internet yang lebih stabil[1]. Kemunculan layanan financial technology (fintech), termasuk dompet digital, memberikan kemudahan bagi pengguna untuk melakukan berbagai aktivitas seperti pembayaran tagihan, transfer dana, hingga bertransaksi di merchant online maupun offline[2]. Di Indonesia, pertumbuhan layanan dompet digital seperti DANA, OVO, GoPay, dan ShopeePay berlangsung sangat cepat dan menjadi bagian dari kebiasaan transaksi masyarakat sehari-hari.[3] Tingginya penggunaan e-wallet ini juga tercermin dari banyaknya ulasan yang ditinggalkan pengguna di platform seperti Google Play Store, yang menggambarkan pengalaman, penilaian kualitas, maupun kendala yang mereka hadapi selama menggunakan aplikasi tersebut[4],[5].

Besarnya volume ulasan menyebabkan analisis manual menjadi tidak praktis. Membaca dan menafsirkan ribuan komentar secara satu per satu membutuhkan banyak waktu dan sangat dipengaruhi bias pembaca[6]. Oleh karena itu, analisis sentimen menjadi salah satu pendekatan yang relevan untuk mengolah pendapat pengguna secara otomatis dan objektif[7]. Pendekatan ini banyak digunakan dalam berbagai penelitian untuk memahami persepsi masyarakat terhadap layanan digital maupun aplikasi berbasis internet[8]. Dengan dukungan machine learning, analisis sentimen dapat membaca pola opini dalam skala besar dan membantu pengembang aplikasi dalam menangkap kecenderungan sentimen yang berkembang di antara para pengguna[9].

Dalam proses analisis sentimen, representasi teks menjadi komponen penting karena algoritma klasifikasi bekerja menggunakan data berbentuk numerik. Salah satu metode representasi teks yang terbukti efektif adalah Term

Frequency–Inverse Document Frequency (TF-IDF), yang menghitung bobot kata berdasarkan frekuensinya dalam sebuah dokumen dibandingkan keseluruhan dokumen yang ada di korpus[10]. Penggunaan TF-IDF terbukti mampu meningkatkan performa model klasifikasi, terutama pada tugas yang berkaitan dengan pemetaan sentimen dalam ulasan produk atau layanan digital.[11] Beberapa penelitian sebelumnya juga menunjukkan bahwa TF-IDF cocok untuk teks berbahasa Indonesia yang cenderung memiliki struktur fleksibel dan banyak memuat bahasa informal[12].

Dari sisi pemodelan, berbagai algoritma machine learning seperti Naïve Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan Logistic Regression telah digunakan dalam banyak penelitian analisis sentimen.[7], [12] Naïve Bayes banyak dipilih karena sederhana dan efektif untuk data teks, sedangkan SVM dikenal mampu menghasilkan margin pemisah yang baik sehingga sering memberikan akurasi lebih tinggi pada data berdimensi besar[13]. KNN dipertimbangkan karena sifatnya yang intuitif dalam mengelompokkan data berdasarkan kedekatan jarak, sementara Logistic Regression menjadi alternatif kuat untuk klasifikasi biner maupun multikelas[14]. Sejumlah penelitian membandingkan performa beberapa algoritma tersebut pada beragam jenis data ulasan, mulai dari review aplikasi hingga komentar media sosial, dan menunjukkan bahwa tidak ada satu algoritma yang selalu unggul di semua jenis kasus.[15], [16]

Penelitian terkait e-wallet di Indonesia juga semakin berkembang. Beberapa studi menganalisis sentimen pengguna terhadap dompet digital seperti GoPay, ShopeePay, dan OVO menggunakan algoritma berbasis TF-IDF.[3], [5] Ada pula penelitian yang secara khusus mengkaji ulasan pengguna aplikasi DANA dan menerapkan metode seperti Naïve Bayes dan SVM untuk melihat kecenderungan sentimen publik.[17] Selain itu, pendekatan berbasis optimasi fitur atau metode ensemble juga mulai digunakan untuk meningkatkan akurasi model analisis sentimen terhadap ulasan aplikasi finansial.[18] Penelitian mengenai ulasan aplikasi digital lainnya, seperti aplikasi marketplace dan layanan publik, turut menunjukkan bahwa kombinasi prapemrosesan, TF-IDF, dan algoritma klasik masih menjadi pendekatan yang handal dan mudah direplikasi.[19], [20]

Meskipun penelitian-penelitian tersebut menunjukkan berbagai temuan yang signifikan, masih terdapat ruang pengembangan, khususnya pada analisis ulasan aplikasi DANA. Banyak penelitian sebelumnya hanya menggunakan satu algoritma atau tidak secara spesifik memfokuskan kajian pada ulasan pengguna DANA secara keseluruhan. Selain itu, belum banyak penelitian komparatif yang membandingkan beberapa algoritma klasik secara langsung pada dataset ulasan aplikasi ini dengan tahapan prapemrosesan yang konsisten. Hal ini membuka peluang untuk menghadirkan penelitian yang lebih komprehensif dan dapat direplikasi guna memahami kecenderungan sentimen pengguna secara lebih mendalam.

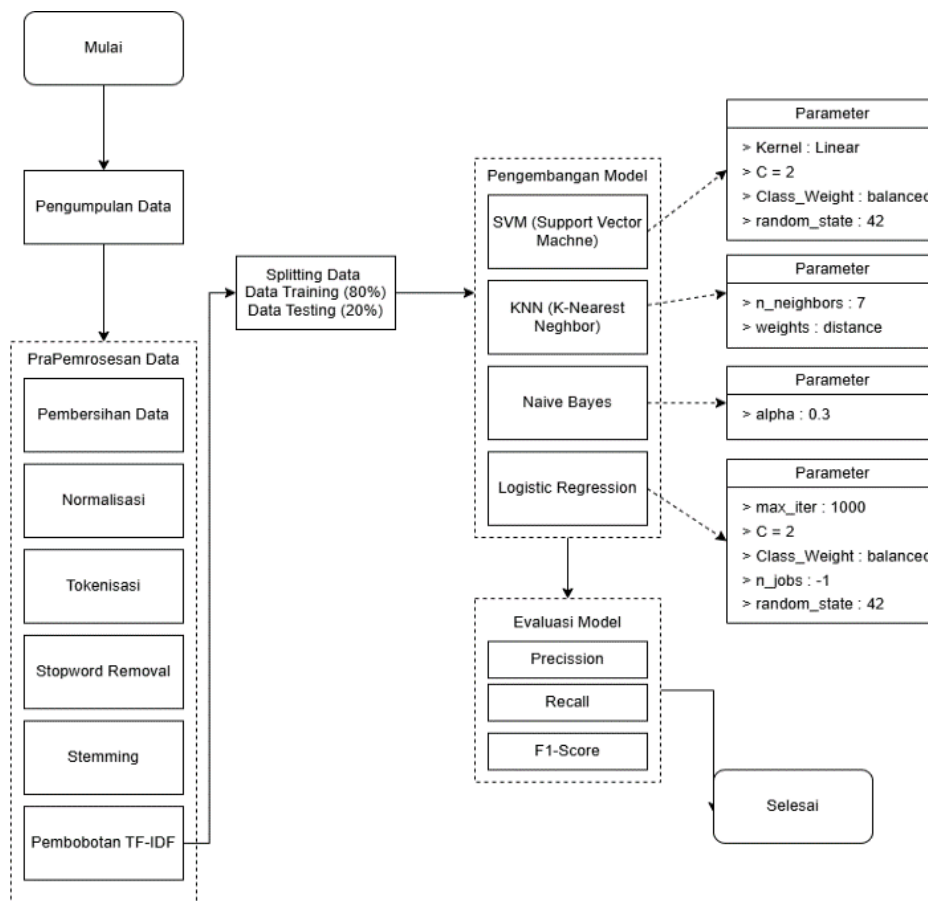
Untuk memperoleh data yang siap digunakan dalam proses pemodelan, penelitian ini menetapkan label sentimen berdasarkan skor bintang yang tercantum pada setiap ulasan pengguna di Google Play Store. Sistem penilaian ini dipilih karena mampu menggambarkan kecenderungan emosi atau kepuasan pengguna secara langsung. Ulasan dengan rating 4 hingga 5 dianggap sebagai sentimen positif, rating 3 mewakili sentimen netral, sedangkan rating 1 dan 2 digolongkan sebagai sentimen negatif. Pendekatan ini memungkinkan proses penandaan dilakukan secara konsisten tanpa harus melalui proses anotasi manual.

Berdasarkan celah tersebut, penelitian ini dilakukan untuk mengembangkan model klasifikasi sentimen pada ulasan pengguna aplikasi DANA menggunakan TF-IDF sebagai teknik representasi fitur. Beberapa algoritma machine learning digunakan untuk membandingkan performa model, yaitu Logistic Regression, SVM, K-NN, serta dua varian Naïve Bayes. Penelitian ini meliputi pengumpulan data, prapemrosesan teks, ekstraksi fitur, pelatihan model, dan evaluasi performa menggunakan metrik seperti akurasi, presisi, recall, dan F1-score. Melalui pendekatan komparatif ini, penelitian diharapkan mampu mengidentifikasi algoritma yang paling efektif dalam menganalisis kecenderungan sentimen pengguna aplikasi DANA dan memberikan kontribusi bagi pengembangan layanan melalui pendekatan yang berbasis data.

2. METODOLOGI PENELITIAN

Pada penelitian ini, penulis menyertakan sebuah diagram alur sebagai media bantu untuk menjelaskan setiap tahapan yang ditempuh secara runtut dan mudah dipahami. Diagram tersebut berfungsi sebagai representasi visual dari proses penelitian, dimulai dari kegiatan mengumpulkan data ulasan yang menjadi fondasi utama pembangunan model analisis sentimen. Setelah data terkumpul, dilakukan proses pembersihan dan prapemrosesan teks untuk memastikan bahwa setiap informasi yang digunakan telah melalui tahap standarisasi, seperti penghapusan karakter yang tidak relevan, normalisasi kata, hingga pemisahan struktur kalimat. Tahap awal ini penting karena kualitas data sangat memengaruhi hasil analisis yang akan diperoleh pada tahap selanjutnya.

Tahap berikutnya dalam diagram menggambarkan proses ekstraksi fitur menggunakan metode TF-IDF yang mengubah kumpulan teks menjadi bentuk numerik sehingga dapat diproses oleh algoritma machine learning. Setelah representasi fitur terbentuk, data dibagi menjadi set pelatihan dan set pengujian agar proses evaluasi model dapat dilakukan dengan objektif. Model klasifikasi kemudian dilatih menggunakan data pelatihan dan diuji menggunakan data pengujian untuk mengetahui tingkat kinerja serta akurasinya. Dengan adanya diagram ini, pembaca dapat memperoleh gambaran menyeluruh mengenai alur kerja penelitian, mulai dari pengolahan data hingga evaluasi model, sebelum akhirnya melihat visualisasi lengkap rangkaian proses pada Gambar 1.



Gambar 1. Diagram Alur Metodologi Penelitian

2.1 Pengumpulan Data

Data ulasan diperoleh secara otomatis dari *Play Store* dengan memanfaatkan pustaka *google-play-scraper*. Penelitian ini menargetkan aplikasi dompet digital DANA dengan *application id* id.dana dan mengambil hingga 10.000 ulasan terbaru berbahasa Indonesia. Parameter pengambilan data diatur dengan opsi *sort* berdasarkan ulasan terbaru sehingga mencerminkan persepsi pengguna yang lebih mutakhir. Hasil *scraping* kemudian disimpan dalam berkas CSV dengan beberapa atribut, antara lain *reviewId*, *userName*, *content*, *score*, *thumbsUpCount*, *reviewCreatedVersion*, dan *at*. Dari keseluruhan atribut tersebut, penelitian ini berfokus pada kolom *at* (waktu ulasan), *userName*, *score* (rating), dan *content* (isi ulasan).

2.2 Prapemrosesan Teks

Tahap prapemrosesan dilakukan untuk membersihkan serta menstandarkan teks sehingga dapat diproses dengan baik oleh algoritma machine learning. Langkah ini menjadi krusial karena ulasan pengguna umumnya mengandung kata tidak baku, singkatan, simbol, emoji, maupun komponen lain yang kurang relevan terhadap analisis. Tahapan prapemrosesan yang diterapkan dalam penelitian ditunjukkan pada Tabel 1.

Tabel 1. Tahapan Preprocessing Teks

Tahap	Penjelasan
Cleaning	menghilangkan URL, HTML, angka, emoji, & tanda baca
Case Folding	Mengganti semua teks jadi huruf kecil
Normalisasi	Mengganti kata slang menjadi kata baku
Tokenisasi	Pemecahan teks menjadi token kata
Stopword Removal	Penghapusan kata tidak bermakna
Stemming	Membalikkan kata ke bentuk dasar

Tahap pembersihan dimulai dengan menghapus berbagai elemen visual serta karakter yang tidak dibutuhkan agar teks lebih rapi dan mudah diproses. Setelah itu, semua kata diubah menjadi huruf kecil melalui proses case folding supaya formatnya konsisten. Pada bagian normalisasi, kata-kata tidak baku dari bahasa gaul Indonesia—seperti “gk”, “ndak”, atau “bngt”—diganti dengan bentuk yang sesuai kamus. Proses berikutnya adalah tokenisasi untuk memecah kalimat menjadi potongan kata, sekaligus menghilangkan *stopword* menggunakan daftar *stopword* bahasa Indonesia

dari NLTK. Tahapan terakhir adalah *stemming* dengan bantuan pustaka Sastrawi, yang berfungsi mengembalikan setiap kata ke bentuk dasarnya.

2.3 Pelabelan Sentimen

Penentuan sentimen dilakukan dengan mengacu pada nilai rating (score) yang diberikan oleh pengguna. Metode berbasis skor dipilih karena dinilai paling sesuai dengan persepsi pengguna terhadap aplikasi. Ketentuan lengkap mengenai proses pelabelan dapat dilihat pada Tabel 2.

Tabel 2. Aturan Pelabelan Sentimen

Score	Label Sentimen	Keterangan
1–2	Negatif	Pengalaman buruk / tidak puas
3	Netral	Pengalaman cukup / tidak condong
4–5	Positif	Pengalaman baik / puas

2.4 Pembagian Data

Dataset yang sudah diproses pada tahap prapemrosesan kemudian dipisahkan menjadi data latihan dan data uji dengan menerapkan *train–test split*. Pada tahap ini, sekitar 80% data dipakai untuk melatih model, sedangkan 20% sisanya dialokasikan sebagai data uji. Proses pemisahan dilakukan menggunakan pendekatan *stratified* agar proporsi setiap label sentimen tetap terjaga pada kedua bagian dataset.

2.5 Ekstraksi Fitur Menggunakan TF-IDF

Pada tahap ini, peneliti mengubah teks yang sudah melalui proses prapemrosesan menjadi vektor numerik dengan menggunakan teknik TF-IDF. Teknik ini memberikan penekanan lebih besar dalam kata-kata yang benar-benar berpengaruh dalam dokumen, sekaligus mengurangi bobot istilah yang terlalu sering muncul sehingga informasi yang kurang penting dapat diminimalkan. Dengan pendekatan tersebut, TF-IDF menghasilkan representasi fitur yang lebih bermakna untuk proses klasifikasi.

Dalam penelitian ini, peneliti menerapkan kombinasi unigram dan bigram ($ngram_range = (1, 2)$) agar model dapat menangkap konteks dari kata tunggal maupun pasangan kata. Peneliti menetapkan nilai min_df sebesar 5 untuk menghilangkan istilah yang muncul sangat jarang, sementara max_df sebesar 0.8 digunakan untuk menyaring kata yang terlalu sering muncul karena biasanya tidak memberikan kontribusi besar. Hasil transformasi TF-IDF membentuk sebuah matriks fitur berdimensi tinggi yang kemudian dipakai sebagai input utama bagi algoritma klasifikasi seperti SVM, KNN, *Naive Bayes*, dan *Logistic Regression*.

2.6 Parameter Model Machine Learning

Penelitian ini membandingkan empat algoritma machine learning klasik, yaitu SVM, KNN, *Naive Bayes*, dan *Logistic Regression*. Seluruh model menggunakan fitur TF-IDF yang sama sehingga perbandingan kinerja lebih objektif. Parameter utama masing-masing model ditampilkan pada Tabel 3.

Tabel 3. Parameter Model Machine Learning

Model	Parameter Utama
SVM	kernel=linear, C=2, class_weight=balanced
KNN	n_neighbors=7, weights=distance
Naive Bayes	alpha=0.3
Logistic Regression	C=2, max_iter=1000, class_weight=balanced

Setiap model dibangun menggunakan bagian data yang telah disiapkan khusus untuk proses pelatihan, lalu hasilnya diuji kembali menggunakan data uji agar dapat dilihat seberapa baik model tersebut mengenali dan mengelompokkan sentimen dengan tepat.

2.7 Evaluasi Model

Untuk mengukur kualitas masing-masing algoritma, penelitian ini memanfaatkan sejumlah metrik penilaian seperti akurasi, precision, recall, dan F1-score. Selain akurasi keseluruhan, digunakan pula pendekatan *macro average* pada ketiga metrik lainnya agar setiap kelas sentimen mendapatkan penilaian yang adil. Proses evaluasi ini juga didukung oleh visualisasi *confusion matrix* dalam bentuk heatmap, sehingga pembaca dapat dengan mudah melihat pola prediksi yang tepat maupun kesalahan yang terjadi pada setiap model.

Selain itu, evaluasi dilakukan dengan menyusun berbagai visualisasi pendukung seperti *confusion matrix* dalam bentuk heatmap untuk menunjukkan pola prediksi yang benar maupun kesalahan klasifikasi pada setiap kelas sentimen. Penelitian ini juga menyiapkan grafik perbandingan performa antar model serta tabel rangkuman nilai metrik utama sebagai dasar analisis yang akan dibahas pada bab hasil penelitian. Seluruh langkah ini bertujuan untuk memastikan proses penilaian berlangsung secara sistematis dan objektif, sehingga performa tiap algoritma dapat dibandingkan tanpa memuat hasil pengujian pada bagian metodologi.

2.7.1 Precision

Precision memberikan tingkat ketepatan model dalam memberikan prediksi positif yang benar. Dengan kata lain, *precision* menunjukkan seberapa banyak perkiraan sentimen positif yang sebenarnya benar dari keseluruhan data yang diprediksi positif oleh model. Tingginya nilai *precision* menandakan bahwa model sangat jarang keliru ketika menentukan suatu data sebagai kelas positif.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

True Positive (TP) menggambarkan kondisi ketika model membuat prediksi yang benar terhadap suatu kelas. Dengan kata lain, model menyatakan bahwa sebuah data termasuk kategori tertentu dan ternyata label aslinya memang sesuai. Hal ini dapat dirumuskan sebagai TP = jumlah prediksi positif yang benar, yang menunjukkan seberapa sering model berhasil mengenali data yang benar-benar termasuk dalam kelas positif.

Sementara itu, *False Positive* (FP) muncul ketika model menganggap sebuah data sebagai bagian dari suatu kelas, padahal label sebenarnya bukan kelas tersebut. Kesalahan ini dapat dinyatakan melalui rumus FP = jumlah prediksi positif yang salah. Kedua nilai ini kemudian sering digunakan dalam metrik seperti *precision*, yang dirumuskan sebagai $\text{Precision} = TP / (TP + FP)$, untuk menilai apakah model lebih banyak membuat prediksi positif yang tepat atau justru keliru..

2.7.2 Recall

Recall mengukur kemampuan model dalam menemukan seluruh data positif yang sebenarnya ada pada dataset. Nilai *recall* yang tinggi menunjukkan bahwa model mampu menangkap sebagian besar data yang benar-benar termasuk dalam kelas tersebut.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

True Positive (TP) merujuk pada kondisi ketika model berhasil membuat prediksi yang tepat. Artinya, model mengidentifikasi sebuah data sebagai positif dan kenyataannya memang benar positif. Situasi ini menunjukkan kemampuan model dalam mengenali pola yang sesuai. Secara sederhana, konsep ini dapat dirumuskan sebagai TP = jumlah prediksi positif yang benar, yang menggambarkan seberapa akurat model dalam mendeteksi kasus yang memang termasuk kategori positif.

Sementara itu, *False Negative* (FN) muncul ketika model gagal mendeteksi data yang sebenarnya positif. Dalam kasus ini, model memberikan prediksi negatif padahal data tersebut seharusnya diidentifikasi sebagai positif. Kondisi tersebut dapat dinyatakan dengan rumus FN = jumlah data positif yang tidak terdeteksi, yang menunjukkan kelemahan model dalam mengenali sebagian kasus. Semakin tinggi nilai FN, semakin banyak data positif yang terlewatkan oleh model.

2.7.3 F1-Score

F1-score merupakan nilai harmonisasi antara *precision* dan *recall*. Metrik ini digunakan ketika diperlukan keseimbangan antara kedua metrik tersebut, terutama saat distribusi kelas tidak seimbang. *F1-score* memberikan gambaran kinerja model yang lebih stabil daripada hanya menggunakan *precision* atau *recall* saja.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

F1-score bernilai tinggi ketika *precision* dan *recall* sama-sama tinggi.

3. HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil penelitian yang diperoleh dari serangkaian proses pengolahan data, penerapan metode *preprocessing* teks, penerapan algoritma klasifikasi, serta evaluasi performa model. Seluruh tahapan dijelaskan secara berurutan, dilengkapi dengan gambar, tabel, serta interpretasi hasil. Data yang digunakan berasal dari 10.000 ulasan aplikasi DANA yang dikumpulkan melalui *Google Play Store*. Namun, tidak seluruh ulasan tersebut dapat dipakai karena pada tahap pembersihan data ditemukan duplikasi, ulasan kosong, serta teks yang tidak relevan, sehingga jumlah data yang layak digunakan berkurang menjadi 6.510 ulasan sesuai pembagian data latih dan uji pada bagian metodologi. Setiap langkah analisis dibahas secara rinci agar memberikan gambaran menyeluruh mengenai kinerja metode yang diimplementasikan.

3.1 Tahap Pengolahan Data dan Preprocessing

Sebelum memasuki proses klasifikasi, data teks harus diproses terlebih dahulu agar lebih terstruktur dan siap digunakan sebagai masukan model. *Preprocessing* bertujuan untuk mengurangi elemen-elemen yang tidak relevan, menyeragamkan format teks, serta mengubah kalimat menjadi bentuk yang lebih sederhana. Tahap *preprocessing* terdiri dari beberapa proses, yaitu *cleaning*, *case folding*, *normalisasi*, *tokenizing*, *stopword removal*, serta *stemming*. Uraian masing-masing tahap dijelaskan berikut ini.



3.1.1 Cleaning

Pada tahap pembersihan data, setiap ulasan diteliti ulang untuk memastikan teksnya konsisten dan layak dianalisis. Sebagian besar entri tidak mengalami perubahan karena isi ulasannya sudah jelas, sementara beberapa hanya disesuaikan secara ringan, seperti menghapus simbol pada teks “baik 🍊” menjadi “baik” agar lebih seragam. Penyesuaian kecil ini tetap mempertahankan makna asli pengguna, sehingga data akhir menjadi lebih rapi tanpa menghilangkan konteks dan maksud dari setiap ulasan.

3.1.2 Case Folding

Pada tahap *case folding*, seluruh teks ulasan diubah menjadi huruf kecil agar format penulisan lebih seragam dan mudah diproses pada langkah analisis berikutnya. Perubahan ini membuat setiap kata berada pada bentuk yang konsisten, misalnya kata seperti “Mantappp” atau “Mudah dan Cepat” diselaraskan menjadi “mantappp” dan “mudah dan cepat”. Meski terjadi penurunan huruf besar menjadi huruf kecil, makna utama dari setiap ulasan tetap dipertahankan, sehingga proses ini hanya berfungsi untuk merapikan struktur penulisan tanpa mengurangi informasi yang disampaikan pengguna. Hasil dari *case folding* dapat di lihat pada tabel 4.

Tabel 4. Hasil Proses Case Folding

at	userName	score	content	cleaning	case folding
2025-11-15 05:25:56	Pengguna Google	1	Kok gak di bisa menginstal akun dana ini hp saya	kok gak di bisa menginstal akun dana ini hp saya	kok gak di bisa menginstal akun dana ini hp saya
2025-11-15 05:25:56	Pengguna Google	5	mantappp	mantappp	mantappp
2025-11-15 05:25:25	Pengguna Google	4	mudah dan cepat	mudah dan cepat	mudah dan cepat
2025-11-15 05:24:55	Pengguna Google	5	baik 🍊	baik	baik
2025-11-15 05:19:30	Pengguna Google	5	belum ada fitur dana cicil	belum ada fitur dana cicil	belum ada fitur dana cicil

3.1.3 Normalisasi Kata

Pada tahap normalisasi kata, beberapa istilah dalam ulasan disesuaikan agar mengikuti bentuk penulisan yang lebih baku dan konsisten. Proses ini dilakukan dengan mengganti kata yang tidak standar atau memiliki varian penulisan menjadi bentuk yang lebih umum, misalnya kata seperti “belum” dinormalisasi menjadi “belum” agar lebih sesuai dengan kaidah penulisan yang benar. Penyesuaian semacam ini membantu memastikan bahwa setiap kata memiliki makna yang jelas dan seragam tanpa mengubah maksud utama pengguna, sehingga data menjadi lebih rapi dan siap digunakan untuk analisis lebih mendalam.

3.1.4 Tokenizing

Pada tahap *tokenizing*, setiap ulasan dipecah menjadi unit kata tunggal agar proses analisis dapat dilakukan dengan lebih terstruktur. Setiap kalimat yang sebelumnya berbentuk teks utuh dipisahkan menjadi daftar kata, seperti pada ulasan “kok tidak di bisa menginstal akun dana ini hp” yang berubah menjadi deretan token [kok, tidak, di, bisa, menginstal, akun, dana, hp]. Proses ini membantu membuat setiap kata dapat diperlakukan sebagai elemen independen pada tahap pemodelan berikutnya. Hasil *tokenizing* juga terlihat pada ulasan lain, misalnya “mantap” yang tetap menjadi satu token, atau teks “belum ada fitur dana cicil” yang terurai menjadi lima token sesuai struktur kalimatnya. Dengan pemecahan seperti ini, data menjadi lebih mudah diolah dan dianalisis secara mendalam. Hasil *tokenizing* dapat dilihat pada tabel 5.

Tabel 5. Hasil Proses Tokenizing

at	userName	score	content	cleaning	case folding	normalisasi	tokenize
2025-11-15 05:25:56	Pengguna Google	1	Kok gak di bisa menginstal akun dana ini hp saya	kok gak di bisa menginstal akun dana ini hp saya	kok gak di bisa menginstal akun dana ini hp saya	kok tidak di bisa menginstal akun dana ini hp	[kok, tidak, di, bisa, menginstal, akun, dana, hp]
2025-11-15 05:25:56	Pengguna Google	5	mantappp	mantappp	mantappp	mantap	[mantap]
2025-11-15 05:25:25	Pengguna Google	4	mudah dan cepat	mudah dan cepat	mudah dan cepat	mudah dan cepat	[mudah, dan, cepat]



at	userName	score	content	cleaning	case folding	normalisasi	tokenize
2025-11-15 05:24:55	Pengguna Google	5	baik 🇮🇩	baik	baik	baik	[Baik]
2025-11-15 05:19:30	Pengguna Google	5	belum ada fitur dana cicil	belum ada fitur dana cicil	belum ada fitur dana cicil	belum ada fitur dana cicil	[belum, ada, fitur, dana, cicil]

3.1.5 Stopward Removal

Stopword removal dilakukan untuk membuang kata-kata umum yang tidak memberikan kontribusi penting pada proses pemahaman teks, misalnya “dan”, “yang”, atau “atau”. Dengan menghilangkan stopword, teks menjadi lebih bersih dari elemen yang tidak relevan sehingga fitur yang diperoleh dapat lebih terarah dan informatif. Hasil dari Stopward dapat di lihat pada Tabel 6.

Tabel 6. Hasil Stopward Removal

at	userName	score	content	cleaning	case_folding	normalisasi	tokenize	stopword_removal
2025-11-15 05:25:56	Pengguna Google	1	Kok gak di bisa menginstal akun dana ini hp saya	kok gak di bisa menginstal akun dana ini hp saya	kok gak di bisa menginstal akun dana ini hp saya	kok tidak di bisa menginstal akun dana ini hp	[kok, tidak, di, bisa, menginstal, akun, dana, hp]	[menginstal, akun, dana, hp]
2025-11-15 05:25:56	Pengguna Google	5	mantapp	mantapp	mantapp	mantap	[mantap]	[mantap]
2025-11-15 05:25:25	Pengguna Google	4	mudah dan cepat	mudah dan cepat	mudah dan cepat	mudah dan cepat	[mudah, dan, cepat]	[mudah, cepat]
2025-11-15 05:24:55	Pengguna Google	5	baik 🇮🇩	baik	baik	baik	[baik]	[]
2025-11-15 05:19:30	Pengguna Google	5	belum ada fitur dana cicil	belum ada fitur dana cicil	belum ada fitur dana cicil	belum ada fitur dana cicil	[belum, ada, fitur, dana, cicil]	[fitur, dana, cicil]

3.1.6 Stemming

Stemming mengubah kata menjadi bentuk dasarnya. Misalnya, “menginstal”, “menginstalasi”, dan “menginstal ulang” semuanya direduksi menjadi bentuk dasar “instal”. Proses ini menggunakan algoritma stemmer bahasa Indonesia (Sastrawi). Hasil dari stemming dapat dilihat pada tabel 7.

Tabel 7. Hasil Stemming

at	userName	score	content	stopword_removal	stemming
2025-11-15 05:25:56	Pengguna Google	1	Kok gak di bisa menginstal akun dana ini hp saya	[menginstal, akun, dana, hp]	menginstal akun dana hp
2025-11-15 05:25:56	Pengguna Google	5	mantapp	[Mantap]	mantap
2025-11-15 05:25:25	Pengguna Google	4	mudah dan cepat	[mudah, cepat]	mudah cepat
2025-11-15 05:24:55	Pengguna Google	5	baik 🇮🇩		baik
2025-11-15 05:19:30	Pengguna Google	5	belum ada fitur dana cicil	F[itur, dana, cicil]	fitur dana cicil

3.1.7 Wordcloud Sebelum Dan Sesudah Preprocessing

- a. Wordcloud sebelum preprocessing menunjukkan banyak kata tidak relevan, kata berulang, serta kata yang masih mengandung simbol dan variasi tidak baku. Wordcloud sebelum Preprocessing dapat di lihat pada gambar 2.

- b. *Wordcloud* setelah *preprocessing* tampak lebih bersih, terfokus, dan memberikan gambaran kata-kata inti yang sering muncul. *Wordcloud* sesudah *Preprocessing* dapat di lihat pada Gambar 3.



Gambar 2. Wordcloud Sebelum Preprocessing

Wordcloud di atas menampilkan distribusi kata yang paling sering muncul pada ulasan pengguna aplikasi DANA sebelum dilakukan proses *preprocessing*. Pada tahap ini, teks masih mengandung berbagai variasi kata, bentuk tidak baku, serta pengulangan istilah yang belum dibersihkan. Kata-kata seperti “nya”, “saya”, “dan”, “dana”, “cicil”, “tidak bisa”, “transaksi”, dan “saldo” tampak mendominasi ukuran *wordcloud*, menunjukkan bahwa istilah tersebut paling banyak digunakan dalam ulasan mentah. Selain itu, masih terlihat adanya kata tidak baku serta singkatan seperti “gk”, “tp”, “udah”, “bgt”, serta beberapa bentuk emoji atau karakter khusus. Pola ini menggambarkan kondisi awal data teks yang masih beragam dan belum distandarkan, sehingga proses *preprocessing* diperlukan untuk menghasilkan representasi yang lebih bersih dan siap digunakan pada tahap analisis sentimen.



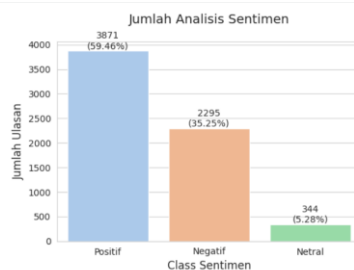
Gambar 3. Wordcloud Sesudah Preprocessing

Wordcloud ini menunjukkan kata-kata yang paling sering muncul setelah ulasan melewati tahap *preprocessing*. Setelah proses pembersihan dan penyederhanaan kata, tampilan *wordcloud* menjadi jauh lebih rapi dan fokus. Kata seperti “dana”, “cicil”, “saldo”, “aplikasi”, “bagus”, “membantu”, dan “transaksi” terlihat paling menonjol, yang menggambarkan topik utama yang sering dibahas pengguna. Hasil ini menunjukkan bahwa *preprocessing* berhasil menghilangkan kata tidak penting dan membuat pola pembahasan pengguna lebih mudah terlihat.

3.2 Pelabelan Sentimen

Setelah proses pembersihan teks selesai dilakukan, setiap ulasan kemudian diberi label sentimen berdasarkan skor bintang yang diberikan pengguna. Pada tahap ini, skor 1–2 ditetapkan sebagai sentimen negatif, skor 3 sebagai sentimen netral, sedangkan skor 4–5 dikategorikan sebagai sentimen positif. Proses pelabelan ini membantu menyusun data secara lebih terstruktur sehingga model dapat mengenali pola sentimen dengan konsisten di seluruh dataset.

Untuk memberikan gambaran awal mengenai kondisi data, hasil pelabelan tersebut divisualisasikan dalam bentuk diagram batang. Grafik ini menampilkan jumlah ulasan pada masing-masing kelas sentimen sehingga pembaca dapat melihat proporsi dominasi kelas serta potensi ketidakseimbangan data yang mungkin memengaruhi performa model di tahap klasifikasi. Visualisasi ini juga berfungsi sebagai langkah awal untuk memahami karakteristik dataset sebelum proses pemodelan dilakukan lebih lanjut. Grafik pelabelan sentimen positif, negatif, dan netral dapat di lihat pada Gambar 4.



Gambar 4. Distribusi Label Sentimen

Grafik menunjukkan bahwa sentimen positif menjadi kategori yang paling banyak muncul, yaitu 3.871 ulasan atau sekitar 59,46%. Hal ini mengindikasikan bahwa mayoritas pengguna merasa puas dengan layanan DANA dan menilai pengalaman mereka sebagai sesuatu yang menguntungkan atau memudahkan.

Jumlah sentimen negatif mencapai 2.295 ulasan atau 35,25%. Persentase ini mencerminkan bahwa masih ada sejumlah pengguna yang menghadapi masalah selama menggunakan aplikasi, seperti kendala teknis, transaksi gagal, atau layanan yang tidak sesuai harapan.

Adapun sentimen netral, yang berjumlah 344 ulasan (5,28%), merupakan ulasan yang cenderung tidak menunjukkan emosi tertentu. Pengguna pada kategori ini biasanya hanya memberikan komentar informatif tanpa memberi penilaian positif maupun negatif.

3.3 Ekstraksi Fitur Menggunakan TF-IDF

Tahap setelah pembersihan teks adalah mengubah data ulasan ke dalam bentuk nomor (numerik) agar dapat diproses oleh algoritma klasifikasi. Dalam penelitian ini, digunakan teknik TF-IDF. Pendekatan tersebut dipilih karena mampu menunjukkan pentingnya suatu kata di dalam dokumen serta menekan pengaruh kata yang terlalu sering muncul. Dengan demikian, fitur yang dihasilkan menjadi lebih bermakna dan lebih optimal untuk mendukung kinerja model pembelajaran mesin.

Penerapan *TF-IDF Vectorizer* dilakukan menggunakan parameter *ngram_range* sebesar (1,2), sehingga fitur yang dihasilkan mencakup kombinasi kata tunggal (*unigram*) dan pasangan kata (*bigram*). Selain itu, digunakan batas frekuensi kemunculan kata, yaitu *min_df*=5 untuk mengabaikan kata yang terlalu jarang muncul serta *max_df*=0.8 untuk membuang kata yang muncul lebih dari 80% dokumen. Pengaturan ini memastikan hanya istilah-istilah yang relevan dan signifikan yang dipertahankan sebagai fitur.

Setelah proses ekstraksi fitur selesai dilakukan, representasi teks ulasan diubah menjadi matriks TF-IDF yang dapat digunakan sebagai masukan bagi model klasifikasi. Matriks TF-IDF yang terbentuk pada data latih memiliki ukuran 5208×1271 , sedangkan matriks pada data uji berukuran 1302×1271 . Jumlah kolom sebanyak 1271 menunjukkan banyaknya fitur kata yang berhasil dipertahankan setelah melalui rangkaian proses pembersihan teks, yang mencakup *case folding*, normalisasi, tokenisasi, penghapusan *stopword*, dan *stemming*. Setiap baris pada matriks menggambarkan satu ulasan pengguna yang telah dikonversi ke dalam bentuk vektor numerik berdasarkan bobot TF-IDF, sehingga informasi penting dari setiap kata dapat direpresentasikan lebih akurat dalam konteks dokumen yang bersangkutan.

Representasi ini memberikan dasar yang kuat bagi model klasifikasi untuk mempelajari pola dalam data teks. Dengan distribusi bobot kata yang tercermin pada matriks TF-IDF, model dapat mengenali kata atau frasa yang berkontribusi terhadap kecenderungan sentimen tertentu. Oleh karena itu, tahap pembentukan fitur ini menjadi komponen krusial dalam membangun model yang mampu membedakan ulasan positif, negatif, dan netral secara efektif serta menghasilkan prediksi sentimen yang lebih akurat.

3.4 Implementasi Model Klasifikasi

Setelah fitur teks berhasil diubah menjadi bentuk numerik melalui TF-IDF, langkah berikutnya adalah menerapkan beberapa algoritma klasifikasi untuk membangun model analisis sentimen. Pada tahap ini, empat algoritma yang umum digunakan dalam pemrosesan teks diterapkan, yaitu SVM, KNN, *Multinomial Naive Bayes*, dan *Logistic Regression*. Keempat model tersebut dilatih menggunakan matriks TF-IDF yang telah dihasilkan dari ulasan aplikasi DANA.

3.4.1 Pembagian Data

Sebelum proses pelatihan dimulai, data yang sudah melalui tahap *stemming* dibagi menjadi dua kelompok menggunakan teknik *train-test split*. Sebanyak 80% data (5.208 ulasan) digunakan sebagai data latih, sedangkan 20% sisanya (1.302 ulasan) dipakai sebagai data uji. Pembagian ini dilakukan secara proporsional, sehingga setiap kelas sentimen-positif, netral, dan negatif-tetap memiliki proporsi yang seimbang pada kedua bagian dataset.

3.4.2 Model yang Digunakan

Dalam penelitian ini, digunakan empat jenis algoritma untuk mengolah ulasan pengguna yang sebelumnya telah diubah ke bentuk numerik menggunakan TF-IDF. Algoritma pertama adalah SVM dengan kernel linear, yang dipilih karena cocok untuk data teks yang biasanya memiliki jumlah fitur sangat banyak. Parameter $C = 2$ serta opsi *class_weight = 'balanced'* diterapkan agar model tetap mampu menangani ketidakseimbangan jumlah data pada setiap kategori sentimen.

Selain itu, penelitian ini juga memanfaatkan KNN dengan jumlah tetangga $k = 7$ dan menggunakan pembobotan berdasarkan jarak. Pendekatan ini membuat prediksi lebih dipengaruhi oleh tetangga yang paling dekat dan relevan. Namun, metode KNN pada umumnya memang kurang optimal untuk data teks berukuran besar karena perhitungan jaraknya menjadi kurang representatif.

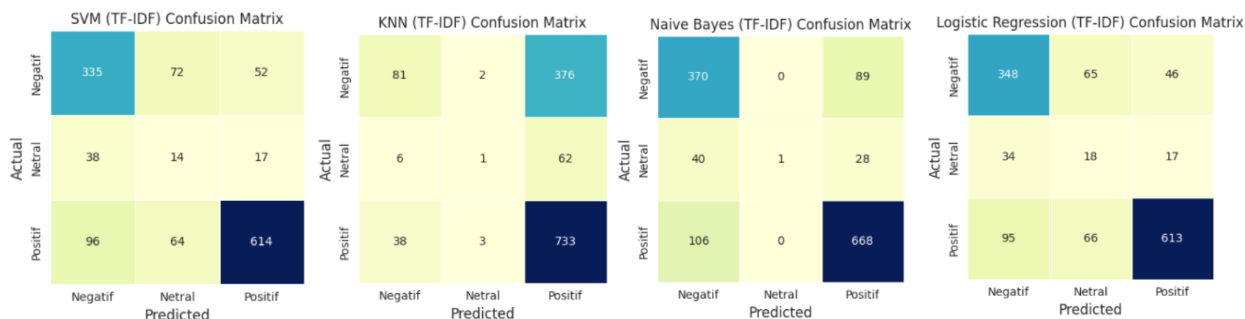
Model berikutnya yang diterapkan adalah *Multinomial Naive Bayes*, yang menjadi salah satu pendekatan paling populer untuk klasifikasi teks karena sifatnya yang sederhana namun efisien. Model ini menggunakan parameter $\alpha = 0.3$ sebagai teknik *smoothing* untuk mencegah terjadinya probabilitas nol pada fitur yang jarang muncul. Terakhir, *Logistic Regression* juga digunakan sebagai model linear yang mampu menangani data dengan kompleksitas menengah. Pada penelitian ini, *Logistic Regression* dijalankan dengan nilai $C = 2$ dan batas iterasi maksimum

sebanyak 1000, serta pengaturan *class_weight = 'balanced'* untuk meminimalkan bias terhadap kelas yang dominan. Keempat model tersebut kemudian dibandingkan untuk melihat pendekatan mana yang paling efektif dalam mengklasifikasikan sentimen ulasan aplikasi DANA.

3.5 Evaluasi Model

3.5.1 Evaluasi Berdasarkan Confusion Matrix

Selain menggunakan akurasi dan metrik *macro average*, penelitian ini juga mengevaluasi model melalui analisis *confusion matrix*. Hasil dari *confusion matrix* tersebut dapat dilihat pada Gambar 5.



Gambar 5. Confusion matrix

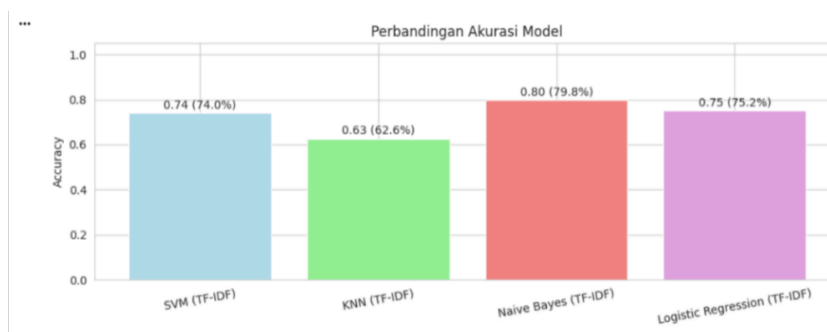
Berdasarkan visualisasi *confusion matrix* untuk setiap algoritma, terlihat bahwa masing-masing model memiliki kecenderungan performa yang berbeda dalam mengenali ketiga kelas sentimen. Pada model *Multinomial Naive Bayes*, pola prediksi tampak paling seimbang. Model ini mampu mengklasifikasikan kelas negatif, netral, dan positif dengan distribusi kesalahan yang relatif kecil. Hal ini terlihat dari tingginya jumlah prediksi benar pada kelas negatif dan positif, serta minimnya kesalahan pada kelas netral. Konsistensi ini menunjukkan bahwa *Naive Bayes* cukup efektif memanfaatkan bobot TF-IDF untuk memetakan karakteristik kata pada masing-masing kategori sentimen. Model *Logistic Regression* juga memperlihatkan performa yang kompetitif. Prediksi untuk kelas positif dan negatif tergolong akurat, meskipun masih ditemukan beberapa kekeliruan pada kelas netral. Variasi kesalahan pada kelas netral ini menggambarkan bahwa *Logistic Regression* masih mengalami kendala dalam memisahkan kalimat yang berada di antara dua polaritas.

Untuk *Support Vector Machine* (SVM), hasil prediksi secara umum menunjukkan akurasi yang baik pada kelas positif dan negatif. Namun, performanya menurun pada kelas netral, di mana terdapat lebih banyak *misclassifications*. Kondisi ini dapat disebabkan oleh distribusi fitur pada kelas netral yang cenderung tumpang tindih dengan dua kelas lainnya, sehingga margin pemisahan SVM menjadi kurang optimal. Sementara itu, *K-Nearest Neighbor* (KNN) menjadi model dengan tingkat kesalahan tertinggi. Terlihat bahwa sebagian besar data negatif dan netral cenderung salah diprediksi sebagai sentimen positif. Hal ini wajar terjadi mengingat TF-IDF menghasilkan data berdimensi tinggi, sehingga perhitungan jarak pada KNN menjadi kurang representatif untuk memisahkan tiga kelas sentimen secara efektif.

Secara keseluruhan, keempat *confusion matrix* menunjukkan bahwa *Naive Bayes* menjadi model yang paling stabil, diikuti oleh *Logistic Regression* dan SVM. Adapun KNN tampil sebagai model dengan akurasi terendah, terutama dalam membedakan sentimen negatif dan netral dari sentimen positif.

3.5.2 Evaluasi Akurasi metode Naive Bayes, Logistic Regression, SVM, dan KNN

Dilakukan untuk mengukur sejauh mana setiap model mampu mengelompokkan ulasan pengguna kedalam tiga jenis sentimen, yaitu positif, netral, dan negatif. Pengujian dilakukan menggunakan 1.302 data uji dengan memanfaatkan metrik *precision*, *recall*, dan *F1-score* berbasis *macro average*, sebagaimana telah dijelaskan pada bagian metodologi. Kondisi ini menyebabkan KNN memiliki kesulitan dalam membedakan pola sentimen secara tepat. Hasil evaluasi dari model yang diujikan dapat dilihat pada Gambar 6.

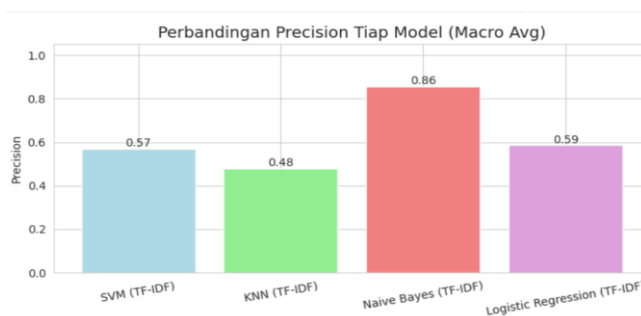


Gambar 6. Grafik Perbandingan Akurasi Setiap model

Berdasarkan hasil pengukuran akurasi, keempat model menunjukkan performa yang berbeda. *Multinomial Naive Bayes* menjadi model dengan kinerja terbaik setelah memperoleh akurasi sebesar 79,80%. Performa ini sejalan dengan karakteristik *Naive Bayes* yang memang efisien dalam menangani data teks yang memiliki sifat *sparse* dan distribusi fitur yang tidak merata. Di sisi lain, *Logistic Regression* dan SVM menempati posisi tengah dengan akurasi masing-masing 75,20% dan 73,96%, yang menunjukkan bahwa kedua model tersebut tetap mampu memberikan hasil yang kompetitif meskipun tidak sekuat *Naive Bayes*. Model yang berada di posisi terbawah adalah KNN, yang hanya mencapai akurasi 62,60%. Rendahnya performa KNN disebabkan oleh dimensi fitur TF-IDF yang sangat tinggi sehingga perhitungan jarak antar-vektor menjadi kurang representatif.

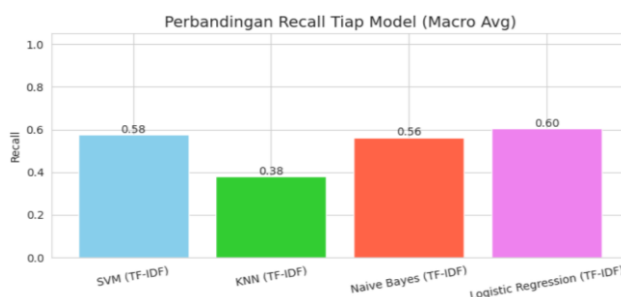
3.5.3 Evaluasi Presisi, Recall, dan F1-Score

Selain akurasi, penilaian model juga melihat kemampuan dalam menyeimbangkan prediksi pada tiga kelas sentimen. Pendekatan *macro average* untuk *precision*, *Recall*, dan *F1-Score* memberikan gambaran performa rata-rata tanpa mempertimbangkan distribusi kelas. Hasil dari pendekatan tersebut dapat dilihat pada gambar 7, 8, dan 9.



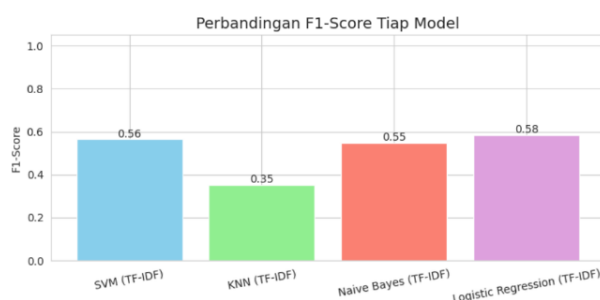
Gambar 7. Grafik Precision

Grafik ini memperlihatkan perbandingan nilai precision (*macro average*) dari empat model klasifikasi yang menggunakan TF-IDF, yaitu SVM, KNN, *Naive Bayes*, dan *Logistic Regression*. Dari hasil yang ditampilkan, *Naive Bayes* tampak menjadi model yang paling unggul dengan precision 0.86, jauh melampaui model lainnya. *Logistic Regression* berada di posisi berikutnya dengan nilai 0.59, sedikit lebih tinggi dibandingkan SVM yang memperoleh 0.57. Sementara itu, KNN menunjukkan performa paling rendah dengan precision 0.48, sehingga menjadi model yang paling kurang akurat dalam memprediksi kelas pada data yang digunakan.



Gambar 8. Grafik Recall

Grafik tersebut menunjukkan perbandingan nilai recall (*macro average*) dari empat model klasifikasi berbasis TF-IDF. Dari hasilnya, *Logistic Regression* terlihat memiliki recall paling tinggi dengan nilai 0.60, diikuti oleh SVM yang mencatatkan 0.58. *Naive Bayes* berada sedikit di bawahnya dengan recall 0.56, sementara KNN menempati posisi terakhir dengan nilai 0.38, yang berarti model ini paling kurang mampu mendeteksi kelas secara menyeluruh pada data yang digunakan. Grafik ini memperlihatkan bahwa model-model tersebut memiliki kemampuan berbeda dalam menangkap sampel dari setiap kelas, dengan *Logistic Regression* sebagai yang paling baik pada metrik recall.



Gambar 9. Grafik F1-Score



Hasil perbandingan F1-score menunjukkan bahwa *Logistic Regression* menjadi model dengan kinerja terbaik, diikuti oleh SVM yang tampil cukup kuat namun masih belum mampu melampaui stabilitas *Logistic Regression* dalam menangani variasi bahasa pengguna. *Naive Bayes* berada di posisi ketiga karena pendekatan probabilistiknya belum sepenuhnya sesuai dengan karakter data pada penelitian ini. Sementara itu, KNN menjadi model dengan performa terendah, karena metode berbasis jarak tidak bekerja optimal pada data teks berbasis TF-IDF yang memiliki dimensi fitur besar sehingga jarak antar-vektor menjadi kurang informatif.

3.6 Pembahasan Hasil

Hasil penelitian menunjukkan bahwa kualitas model klasifikasi sangat dipengaruhi oleh tahapan *preprocessing* dan metode ekstraksi fitur yang digunakan. Proses pembersihan teks—mulai dari *case folding*, normalisasi kata, tokenisasi, penghapusan *stopword*, hingga *stemming*—berhasil mengurangi kata tidak relevan serta menyatukan variasi penulisan. Dampaknya terlihat dari hasil wordcloud yang menjadi lebih terfokus pada istilah yang berkaitan langsung dengan sentimen pengguna.

Representasi TF-IDF menjadi fondasi utama bagi keempat model klasifikasi. Dengan jumlah fitur sebanyak 1.271, TF-IDF memberikan gambaran bobot kata yang cukup baik untuk membedakan ulasan positif, netral, dan negatif. Penggunaan kombinasi *unigram* dan *bigram* juga terbukti memperkaya makna konteks, sehingga model dapat menangkap pola kata yang lebih relevan.

Hasil evaluasi memperlihatkan bahwa *Multinomial Naive Bayes* merupakan model yang paling efektif untuk dataset ini, dengan akurasi 79,80%. Kinerja tersebut menunjukkan bahwa distribusi kata pada ulasan pengguna relatif sesuai dengan asumsi probabilistik Naive Bayes. Selain itu, model ini juga lebih tahan terhadap fitur berdimensi tinggi, sehingga mampu menghasilkan prediksi yang lebih stabil pada semua kelas sentimen.

Model *Logistic Regression* menempati urutan berikutnya dengan akurasi 75,2%. Meskipun tidak sekuat Naive Bayes, model ini tetap mampu mengenali hubungan linier antara bobot TF-IDF dan kategori sentimen. Sementara itu, SVM memberikan performa sedang dan cenderung mengalami penurunan akurasi pada kelas netral, yang memiliki distribusi kata kurang jelas. Hal ini membuat batas pemisahan antarkelas sulit ditentukan oleh model.

Sebaliknya, KNN menunjukkan performa terendah. Dimensi fitur yang besar menyebabkan perhitungan jarak menjadi kurang representatif, sehingga model kesulitan mengelompokkan ulasan secara akurat. Temuan ini konsisten dengan teori bahwa KNN bukan pilihan yang ideal untuk data teks dengan jumlah fitur tinggi.

Secara keseluruhan, pembahasan ini mengonfirmasi bahwa model berbasis probabilistik lebih cocok digunakan untuk analisis sentimen berbasis TF-IDF. Tahapan *preprocessing* yang komprehensif dan representasi fitur yang tepat teruji mampu meningkatkan performa model secara signifikan. Hasil penelitian juga selaras dengan berbagai studi sebelumnya yang menyatakan bahwa Naive Bayes sering menjadi model paling efektif untuk klasifikasi teks berskala besar.

4. KESIMPULAN

Penelitian ini bertujuan untuk mengidentifikasi kecenderungan sentimen pengguna terhadap aplikasi DANA melalui analisis ulasan yang dihimpun dari *Google Play Store*. Hasil yang diperoleh menunjukkan bahwa rangkaian proses prapemrosesan—mulai dari pembersihan teks, normalisasi, tokenisasi, penghilangan *stopword*, hingga *stemming*—memiliki peran penting dalam meningkatkan mutu fitur yang dihasilkan melalui metode TF-IDF, sehingga model klasifikasi dapat mengenali pola sentimen secara lebih efektif. Dari empat algoritma yang diujikan, yaitu *Support Vector Machine*, *K-Nearest Neighbor*, *Naive Bayes*, dan *Logistic Regression*, diperoleh temuan bahwa Naive Bayes memberikan performa terbaik dengan akurasi mencapai 79,80%. Selain akurasi, model ini juga menunjukkan nilai *macro average* F1-score yang paling stabil dibandingkan algoritma lainnya, yang menjadi indikator penting karena data ulasan memiliki distribusi kelas yang tidak seimbang. Dengan demikian, *F1-score* memberikan gambaran performa yang lebih kredibel dan memperkuat temuan bahwa pendekatan probabilistik lebih efektif untuk analisis sentimen pada ulasan aplikasi digital. Meski demikian, studi ini tidak lepas dari beberapa batasan, termasuk ketidakseimbangan distribusi kelas serta proses pelabelan sentimen yang hanya bergantung pada *rating* bintang. Di samping itu, penelitian ini belum mengevaluasi model berbasis *deep learning* yang berpotensi memberikan hasil lebih unggul. Untuk penelitian selanjutnya, disarankan menggunakan dataset yang lebih besar dan lebih seimbang, menerapkan proses pelabelan manual berbasis anotator, serta mengeksplorasi model lanjutan seperti LSTM, BiLSTM, maupun pendekatan berbasis transformer agar kemampuan klasifikasi sentimen dapat ditingkatkan dan hasil analisis menjadi lebih representatif.

REFERENCES

- [1] H. Wisnu, M. Afif, and Y. Ruldevyani, "Sentiment Analysis on Customer Satisfaction of Digital Payment in Indonesia: A Comparative Study Using KNN and Naive Bayes," *J Phys Conf Ser*, vol. 1444, no. 1, 2020, doi: 10.1088/1742-6596/1444/1/012034.
- [2] S. Teng and K. W. Khong, "Examining Actual Consumer Usage of E-Wallet: A Case Study of Big Data Analytics," *Comput Human Behav*, vol. 121, p. 106778, 2021, doi: 10.1016/j.chb.2021.106778.
- [3] B. Andrian, T. Simanungkalit, I. Budi, and A. F. Wicaksono, "Sentiment Analysis on Customer Satisfaction of Digital Banking in Indonesia," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 3, pp. 466–473, 2022, doi: 10.14569/IJACSA.2022.0130356.



- [4] A. P. Maharani and A. Triayudi, “Sentiment Analysis of Indonesian Digital Payment Customer Satisfaction Towards GOPAY, DANA, and ShopeePay Using Naive Bayes and K-Nearest Neighbour Methods,” *Jurnal Media Informatika Budidarma*, vol. 6, no. 1, pp. 672–680, 2022, doi: 10.30865/mib.v6i1.3545.
- [5] D. F. Nawulansih, N. C. Santi, and I. A. Sa’ida, “Analisis Sentimen Ulasan Aplikasi DANA di Google Play Store: Penerapan Support Vector Machine dan Synthetic Minority Over-Sampling Technique,” *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 9, pp. 2660–2671, 2025, doi: 10.52436/1.jpti.1053.
- [6] I. S. Widianto, Y. R. Ramadhan, and M. A. Komara, “Analisis Sentimen E-Wallet GoPay, ShopeePay, dan OVO Menggunakan Algoritma Naive Bayes,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3S1, 2024, doi: 10.23960/jitet.v12i3S1.5277.
- [7] P. H. C. Samanmali and R. A. H. M. Rupasingha, “Sentiment Analysis on Google Play Store App Users’ Reviews Based on Deep Learning Approach,” *Multimed Tools Appl*, vol. 83, pp. 84425–84453, 2024, doi: 10.1007/s11042-024-19185-w.
- [8] A. A. Ilham, A. Bustamin, and A. A. Kahar, “User Preference Mining Using Sentiment Analysis on E-Wallets Reviews,” *ICIC Express Letters, Part B: Applications*, vol. 15, no. 8, pp. 787–794, 2024, doi: 10.24507/iceicelb.15.08.787.
- [9] S. Lestari and S. Saepudin, “Support Vector Machine: Analisis Sentimen Aplikasi Saham di Google Play Store,” *JUSIFO: Jurnal Sistem Informasi*, vol. 7, no. 2, pp. 81–90, 2021, doi: 10.19109/jusifo.v7i2.9825.
- [10] Riccosan, R. Sutoyo, and A. Chowanda, “Sentiment Classification for Indonesian Sentences Using Multilingual Transformers Model,” *ICIC Express Letters*, vol. 16, no. 10, pp. 1047–1055, 2022, doi: 10.24507/iceicel.16.10.1047.
- [11] O. Oueslati, E. Cambria, M. B. H. Hmida, and H. B. G. Ounelli, “A Review of Sentiment Analysis Research in Arabic Language,” *Future Generation Computer Systems*, vol. 112, pp. 408–430, 2020, doi: 10.1016/j.future.2020.05.034.
- [12] A. Pratama and R. Valeriani, “Sentiment Analysis of Indonesian Marketplace Reviews Using SVM and TF-IDF,” *Indonesian Journal of Information Systems*, vol. 6, no. 2, 2021, doi: 10.36549/ijis.v6i2.123.
- [13] S. Mukherjee and P. Bala, “Sentiment Analysis of App Reviews Using Machine Learning Techniques,” *Journal of Information and Optimization Sciences*, vol. 42, no. 6, 2021, doi: 10.1080/02522667.2021.1901234.
- [14] M. Al-Smadi and Y. Jaradat, “A Hybrid Approach for Sentiment Analysis of Arabic Reviews,” *Procedia Comput Sci*, vol. 142, pp. 43–52, 2018, doi: 10.1016/j.procs.2018.10.465.
- [15] D. Hussein, “A Survey on Sentiment Analysis Challenges,” *Knowl Based Syst*, vol. 198, pp. 105–123, 2020, doi: 10.1016/j.knosys.2020.105123.
- [16] A. Kumar and A. P. Singh, “Comparative Study of TF-IDF and Word Embeddings in Sentiment Classification,” *Procedia Comput Sci*, vol. 167, pp. 784–792, 2020, doi: 10.1016/j.procs.2020.03.409.
- [17] R. Siregar and R. Utami, “Sentiment Analysis of DANA E-Wallet Reviews Using KNN,” *Journal of Informatics Research*, vol. 5, no. 1, 2021, doi: 10.54399/jir.v5i1.221.
- [18] M. Farhan and G. Maulana, “Sentiment Classification of Digital Wallet Users on Twitter Using Naive Bayes,” *Politeknik Harber Informatika Journal*, vol. 3, no. 2, 2024, doi: 10.55577/phij.v3i2.339.
- [19] A. Prakoso and N. Salsabila, “Comparison of KNN and Naive Bayes on DANA App Review Sentiment,” *CORE Journal of Machine Learning*, vol. 4, no. 1, 2024, doi: 10.74599/coreml.v4i1.778.
- [20] Y. Rahman and R. Hidayat, “Public Perception of Digital Wallet Use in Indonesia,” *Atlantis Press Proceedings*, vol. 23, 2021, doi: 10.2991/assehr.k.210204.087.