



Klasifikasi Sentimen Terhadap Ulasan Pengguna Aplikasi DANA Menggunakan Metode *Support Vector Machine*

Dentar Prapasta^{1*}, Mohammad Zainuddin²

^{1,2} Teknik Informatika, Fakultas Teknologi dan Desain, Institut Teknologi dan Bisnis Asia Malang, Indonesia

Info Artikel

Riwayat Artikel:

Diterima : **28 November 2025**

Direvisi : **22 Desember 2025**

Diterbitkan : **01 Juni 2026**

Kata Kunci:

DANA,
Ulasan Pengguna,
Analisis Sentimen,
TF-IDF,
Support Vector Machine.

Keywords:

DANA,
User Reviews,
Sentiment Analysis,
TF-IDF,
Support Vector Machine.

ABSTRAK

DANA merupakan salah satu platform *fintech* asal Indonesia. Banyaknya jumlah ulasan aplikasi DANA sebagai *e-wallet* populer menyebabkan proses pemantauan kepuasan pengguna secara manual menjadi tidak praktis. Penelitian ini berfokus pada analisis sentimen untuk menggali persepsi masyarakat terhadap aplikasi DANA. Metode yang digunakan adalah algoritma *Support Vector Machine (SVM)* yang dikombinasikan dengan pembobotan *Term Frequency-Inverse Document Frequency (TF-IDF)*. Sebanyak 50.000 ulasan diproses melalui sejumlah tahapan *preprocessing*, mulai dari *cleaning*, *case folding*, *normalization*, *stopword removal*, hingga *stemming*. Hasil pengujian menunjukkan bahwa model *SVM* memberikan performa yang baik dengan akurasi sebesar 83%. Pada sentimen positif, model mencapai *precision* 89%, *recall* 92%, dan *F1-score* 90%. Selain itu, analisis frekuensi kata memperlihatkan bahwa kata kunci "cepat" dan "mudah" muncul lebih dominan dibandingkan "lambat" maupun "susah", yang menggambarkan kepuasan pengguna terhadap performa aplikasi. Dengan demikian, metode *SVM* dapat dikatakan efektif dalam melakukan analisis sentimen pada data ulasan berjumlah besar.

ABSTRACT

DANA is an Indonesian financial technology (*fintech*) platform. The large volume of user reviews on the DANA application, one of the country's major digital wallets, makes manual assessment of user satisfaction impractical. This research focuses on sentiment analysis to gain insights into public perception of the DANA app. The study employs a *Support Vector Machine (SVM)* algorithm combined with *Term Frequency-Inverse Document Frequency (TF-IDF)* weighting. A dataset of 50,000 reviews underwent several preprocessing steps, including *cleaning*, *case folding*, *normalization*, *stopword removal*, and *stemming*. Experimental findings indicate that the *SVM* model achieved solid performance with an accuracy of 83%. For the positive sentiment class, the model obtained a *precision* of 89%, *recall* of 92%, and *F1-score* of 90%. Moreover, the word frequency analysis revealed that the terms "fast" and "easy" appeared more frequently than "slow" or "difficult," indicating users' satisfaction with the app's performance. Thus, the *SVM* approach is proven to be effective for sentiment analysis on large-scale review datasets.

Penulis Korespondensi:

Dentar Prapasta,
Program Studi Teknik Informatika,
Institut Teknologi dan Bisnis Asia Malang
Email: dentarprapasta19@gmail.com

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



1. PENDAHULUAN

Kemajuan di bidang teknologi informasi dan komunikasi telah menjadi katalis bagi munculnya berbagai inovasi dalam sektor keuangan digital. Di Indonesia, fenomena ini terlihat jelas dari pertumbuhan dompet

digital (*e-wallet*) yang sangat signifikan. Tingginya tingkat adopsi teknologi finansial (*fintech*) oleh masyarakat turut memicu persaingan yang semakin kompetitif di antara penyedia layanan [1]. Teknologi ini menawarkan keunggulan berupa efisiensi, kecepatan, dan keamanan dalam bertransaksi. Dalam ekosistem ini, aplikasi DANA hadir sebagai salah satu platform dompet digital yang populer dan diandalkan masyarakat untuk berbagai keperluan, mulai dari pembelian produk digital, pembayaran tagihan, hingga transfer uang [2]. Semakin tingginya tingkat penggunaan aplikasi DANA menunjukkan pentingnya pemantauan terhadap kepuasan dan persepsi pengguna, yang umumnya terekam dalam bentuk ulasan atau review di Google Play Store. Ulasan pengguna dalam platform digital seperti Google Play Store tidak hanya berfungsi sebagai umpan balik, tetapi juga menjadi sumber data tidak terstruktur yang kaya akan informasi mengenai pengalaman dan opini pengguna. Ulasan mengandung opini, keluhan, hingga pujian yang sangat berharga jika dianalisis dengan benar. Namun, volume data yang besar serta keragaman gaya bahasa dalam ulasan membuat proses analisis secara manual menjadi tidak efisien dan tidak praktis [3]. Oleh karena itu, analisis sentimen berbasis klasifikasi teks diperlukan untuk mengekstraksi makna dari ulasan-ulasan tersebut secara sistematis dan objektif.

Dalam ranah analisis sentimen, salah satu pendekatan yang paling sering diterapkan adalah *Support Vector Machine (SVM)*. Sebagai algoritma berbasis *supervised learning*, SVM terbukti mumpuni dalam mengolah data teks yang memiliki dimensi tinggi [4]. Keunggulan kompetitif metode ini adalah kemampuannya mencari garis pemisah (*hyperplane*) terbaik, sehingga pemisahan kategori data dapat dilakukan secara optimal. SVM dapat dimanfaatkan untuk mengelompokkan ulasan pengguna berdasarkan polaritas atau kecenderungan sentimennya, dengan bantuan fitur representasi teks seperti *Term Frequency-Inverse Document Frequency (TF-IDF)* [5]. Penelitian terdahulu memperlihatkan bahwa metode SVM menghasilkan performa tinggi dalam analisis sentimen. Misalnya, analisis ulasan aplikasi DeepL di Google Play berhasil mencapai akurasi sebesar 91%, dengan presisi tinggi pada kelas positif yaitu 94% [6]. Temuan lain juga memperlihatkan bahwa kombinasi *TF-IDF* dan SVM mampu mengklasifikasikan sentimen publik dengan akurasi 92,8% [7]. Berdasarkan hasil-hasil tersebut, algoritma SVM dinilai efektif untuk menganalisis ulasan pengguna aplikasi DANA dalam penelitian ini.

Selain itu, hasil analisis sentimen bisa menjadi dasar untuk mengukur tingkat kepercayaan pengguna pada aplikasi DANA. Sentimen negatif yang konsisten dapat menjadi indikator adanya masalah dalam layanan, sedangkan sentimen positif yang dominan dapat memperkuat citra aplikasi sebagai layanan yang dapat dipercaya. Kepercayaan pengguna merupakan modal penting dalam mempertahankan loyalitas terhadap platform *fintech* [8]. Oleh sebab itu, analisis sentimen berbasis SVM dapat dijadikan alat bantu strategis untuk mendukung proses pengambilan keputusan yang berkaitan dengan pengembangan layanan keuangan digital di tengah kompetisi industri *fintech* yang semakin ketat.

2. METODE PENELITIAN

Penelitian ini dilaksanakan melalui kerangka kerja sistematis yang terdiri dari lima tahapan utama untuk menjamin validitas hasil klasifikasi sentimen. Proses diawali dengan tahap pengumpulan data (*data collection*) dari ulasan pengguna, yang kemudian dilanjutkan dengan data *preprocessing* guna mereduksi *noise* dan menstandarisasi teks melalui teknik *cleaning*, *case folding*, *normalization*, *stopword removal*, serta *stemming*. Data tekstual yang telah terstruktur selanjutnya ditransformasi menjadi representasi vektor numerik menggunakan metode pembobotan *Term Frequency-Inverse Document Frequency (TF-IDF)* agar dapat diproses secara komputasional. Representasi fitur tersebut kemudian digunakan sebagai input pada algoritma *Support Vector Machine (SVM)* untuk mengklasifikasikan ulasan ke dalam kategori sentimen. Tahap akhir penelitian melibatkan evaluasi kinerja model secara komprehensif menggunakan *Confusion Matrix* dan pengukuran metrik *accuracy*, *precision*, *recall*, serta *F1-score* untuk mengukur efektivitas metode yang diusulkan.

2.1. Data Collection

Pada tahap *data collection*, dilakukan pengambilan dataset dari berbagai sumber sebagai komponen utama dalam analisis *data mining*. Dalam penelitian ini, dataset diperoleh dari Kaggle, <https://www.kaggle.com/datasets/alexmariosimanjuntak/dana-app-sentiment-review-on-playstore-indonesia>. Data berisi kumpulan ulasan pengguna aplikasi DANA di Google Play Store. Data tersebut mencakup 50.000 ulasan berbahasa Indonesia yang merepresentasikan berbagai opini pengguna terhadap kinerja, fitur, serta pengalaman penggunaan aplikasi DANA. Setelah data diperoleh, selanjutnya akan dilakukan tahap

preprocessing dan analisis sentimen. Sebagai gambaran dataset, ditampilkan sebanyak 10 data pada Gambar 1.

userName	score	at	content	sentimen
Elisya Kasni	5	2024-02-15 11:24:56	Bagus	POSITIVE
Rusman Man	2	2024-02-15 11:24:03	Dana mmg keren mantap.	POSITIVE
Qiliw Sadega	1	2024-02-15 11:23:34	Saya ngajuin upgrade dana premium krna ktp saya buram jdi ga bisa verifikasi..trus coba lewat email di suruh nunggu 3 hri..ini udh 3 hri lebih msih gitu ² doang saldo sya d dana jdi ga bisa d apa'in.. gmn ini solusi nya	NEGATIVE
Kjutjrv2 Kjut	3	2024-02-15 11:22:46	Kocak mana diskon nya ml malah eror segala kaga ikhlas ngasih diskon nya	NEGATIVE
Fifi Alfiah	1	2024-02-15 11:21:34	Saldo hilang karena no lama Hilang ganti no saldonya gk ada sama sekali dana tidak bertanggung jawab Buktinya saldo saya gk ada !!! Saldo lama saya 1800.000	NEGATIVE
Kiki57	3	2024-02-15 11:21:13	mayan	POSITIVE
Parhan Parhan	1	2024-02-15 11:19:32	Udah gua hapus dana ya.. ilang ya udah 1 juta lebih duwit gua.mf dana sayah buka orang kaya itu uang keringat sayah uang halal	NEGATIVE
Dewi Anggreni	5	2024-02-15 11:19:22	baik	POSITIVE
Bang Ewok13	1	2024-02-15 11:18:01	TOLONG UNTUK SISTEM KEAMANAN DI PERBAIKI. KALAU 1 ATAU 2 ORNG YG KE HACK WAJAR, MUNGKIN KETELEDORAN PENGGUNA. TAPI KALAU SUDAH BANYAK YG TERKENA HACK. BERARTI ADA YG SALAH DENGAN SISTEM KEAMANAN DANA	NEUTRAL
M Alifian	5	2024-02-15 11:17:58	mempermudah transfer	POSITIVE

Gambar 1. Dataset

Gambar 1 menyajikan representasi sampel dataset ulasan pengguna aplikasi DANA yang diperoleh dari Kaggle, yang terdiri dari atribut identitas pengguna (*userName*), peringkat penilaian (*score*), waktu pengiriman ulasan (*at*), isi teks ulasan (*content*), serta label kelas sentimen (*sentiment*). Sampel data tersebut memperlihatkan variasi opini pengguna yang mencakup kategori sentimen positif, negatif, dan netral dalam format teks mentah yang belum melalui tahapan *preprocessing*, sehingga masih mengandung elemen *noise* dan gaya bahasa tidak baku.

2.2. Data Preprocessing

Tahap *data preprocessing* merupakan proses penting dalam pelatihan data yang bertujuan mengubah data mentah menjadi terstruktur agar lebih mudah diproses oleh algoritma [9]. Ulasan pengguna di Google Play Store sering mengandung bahasa tidak baku, singkatan, atau kata tidak dikenal, sehingga diperlukan *preprocessing* untuk mengurangi *noise* dan meningkatkan kualitas data.

Preprocessing terbagi dalam lima tahap, yaitu sebagai berikut :

- 1) *Cleaning*: Menghapus komponen yang dianggap sebagai *noise* atau tidak bernilai informasi, seperti simbol, emoji, dan tanda baca yang tidak relevan, guna mempermudah komputasi data;
- 2) *Case Folding*: Penyeragaman format huruf dalam dokumen teks. Tahap ini bertujuan menstandarisasi teks yang seringkali memiliki penggunaan huruf kapital yang tidak konsisten;
- 3) *Normalization*: Perbaikan kata-kata non-baku, singkatan, atau istilah gaul (slang) agar dikonversi menjadi bentuk standar atau kata baku;
- 4) *Stopword Removal*: Menghilangkan kata-kata umum seperti kata hubung yang frekuensinya tinggi namun minim makna, sehingga bobot analisis dapat difokuskan pada kata-kata yang lebih berbobot atau substantif;
- 5) *Stemming*: Mengubah kata kembali ke bentuk dasarnya (*root word*) untuk mereduksi kata-kata yang bervariasi sehingga dapat menyederhanakan proses analisis.

2.3. Term Frequency-Inverse Document Frequency (TF-IDF)

Setelah rangkaian tahapan *preprocessing* selesai dilakukan, data ulasan yang semula berupa teks mentah belum siap untuk langsung diolah oleh komputer. Agar dapat diproses secara komputasional oleh algoritma *machine learning*, data tersebut harus terlebih dahulu dikonversi menjadi representasi numerik atau vektor. Penelitian menerapkan metode ekstraksi fitur *Term Frequency-Inverse Document Frequency (TF-IDF)*. Metode

ini dipilih karena keandalannya dalam mengevaluasi seberapa penting sebuah kata dan memberikan bobot pada kata dalam suatu dokumen maupun korpus data [10].

Term Frequency (TF) menyatakan seberapa sering suatu kata muncul dalam sebuah dokumen.

$$TF_t = (t, d) \quad (1)$$

Inverse Document Frequency (IDF) mengukur tingkat kepentingan suatu term dalam seluruh koleksi dokumen.

$$IDF_t = \log \frac{n}{df(t)} \quad (2)$$

Bobot *TF-IDF* (W_t) untuk sebuah term dalam dokumen dirumuskan sebagai:

$$W_t = TF_t \times IDF_t \quad (3)$$

Secara prinsip, *TF-IDF* bekerja dengan memadukan dua proses perhitungan utama. Pertama, *Term Frequency (TF)* digunakan untuk menilai frekuensi kemunculan suatu kata dalam sebuah dokumen. Kedua, *Inverse Document Frequency (IDF)* dimanfaatkan untuk menentukan seberapa jarang kata tersebut muncul pada keseluruhan dokumen. Melalui kombinasi keduanya, kata-kata yang lebih khas dan relevan terhadap sebuah sentimen akan memperoleh bobot yang lebih tinggi dibandingkan istilah umum. Hasil pembobotan tersebut kemudian direpresentasikan dalam bentuk vektor numerik yang menjadi masukan bagi proses pelatihan maupun pengujian model klasifikasi dengan *Support Vector Machine (SVM)*.

2.4. Support Vector Machine (SVM)

Salah satu algoritma populer dalam ranah *supervised learning* yang memiliki fleksibilitas tinggi adalah *Support Vector Machine (SVM)*. Meskipun algoritma ini dirancang untuk dapat menyelesaikan dua jenis tugas sekaligus, yaitu klasifikasi dan regresi, penerapannya jauh lebih dominan pada kasus klasifikasi. Hal ini dikarenakan *SVM* memiliki keunggulan kompetitif dalam mengelola data yang memiliki dimensi tinggi dan kompleks [11]. Prinsip utama algoritma ini terletak pada penentuan *hyperplane* terbaik. *Hyperplane* berfungsi sebagai pemisah struktural antara dua kelas data, baik berupa garis lurus dalam grafik dua dimensi maupun bidang dalam ruang dimensi banyak, dengan target utama memaksimalkan *margin* pemisahan [12].

Proses pelatihan dilakukan dengan menyelesaikan fungsi optimasi berikut:

$$\min_{w, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (4)$$

Untuk mendapatkan pemisahan yang optimal (memaksimalkan *margin*), *SVM* meminimalkan fungsi tujuan berikut dengan batasan tertentu:

$$\begin{aligned} \text{s. t. } & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \quad (5)$$

di mana:

w : Vektor bobot (*weight vector*) yang menentukan orientasi *hyperplane*.

x : Vektor *input* (data ulasan hasil *TF-IDF*).

b : Bias atau parameter posisi *hyperplane*.

y_i : Label kelas sentimen yang sebenarnya.

Kualitas pemisahan ini sangat bergantung pada keberadaan *support vectors*, yaitu sekumpulan data yang letaknya paling berdekatan dengan garis batas. Posisi *support vectors* inilah yang menjadi referensi utama kalkulasi algoritma untuk membentuk *hyperplane* yang paling optimal, sehingga akurasi klasifikasi dapat dimaksimalkan [13].

2.5. Evaluasi

Proses evaluasi hasil bertujuan untuk mengukur efektivitas algoritma *SVM* dalam menyelesaikan tugas klasifikasi. Untuk mendapatkan gambaran performa yang komprehensif, analisis dilakukan dengan memanfaatkan *Confusion Matrix* serta serangkaian metrik evaluasi, meliputi akurasi (*accuracy*), presisi (*precision*), *recall*, dan *F1-score* [14].

Confusion Matrix merupakan alat visualisasi berupa tabel dua dimensi yang digunakan untuk menggambarkan perbandingan antara label yang diprediksi model dan label sebenarnya dari data. Tabel ini memuat empat elemen penting yang menjadi dasar evaluasi kinerja model, yaitu:

- 1) *True Positive (TP)*: Merepresentasikan kondisi di mana model memprediksi data kelas positif dengan tepat sesuai dengan label aslinya.
- 2) *True Negative (TN)*: Menggambarkan keberhasilan model dalam mengenali data kelas negatif secara benar sebagai negatif.
- 3) *False Positive (FP)*: Merupakan jenis kesalahan prediksi, di mana data yang hakikatnya negatif justru salah diklasifikasikan oleh model sebagai positif.
- 4) *False Negative (FN)*: Merupakan kesalahan prediksi lainnya, terjadi ketika data yang seharusnya positif justru salah diprediksi sebagai negatif.

Berdasarkan keempat komponen tersebut, dapat diturunkan beberapa metrik evaluasi utama yang dipakai untuk mengukur kinerja model, antara lain sebagai berikut:

- 1) *Accuracy* digunakan untuk mengetahui tingkat ketepatan keseluruhan model dalam melakukan prediksi, *accuracy* diperoleh dengan membandingkan jumlah prediksi yang benar dengan jumlah total data uji.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

- 2) *Precision* menggambarkan tingkat keberhasilan model memprediksi kelas positif. Nilai *precision* yang tinggi menunjukkan bahwa kesalahan dalam mengklasifikasikan data negatif sebagai positif relatif kecil.

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

- 3) *Recall* merupakan metrik yang menunjukkan seberapa banyak data berlabel positif yang berhasil dikenali oleh model. Semakin tinggi nilai *recall* mengindikasikan bahwa model mampu mengidentifikasi sebagian besar data positif dengan baik.

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

- 4) *F1-Score* merupakan rata-rata harmonik antara *precision* dan *recall*. Tujuannya adalah menyeimbangkan kedua metrik tersebut agar penilaian tetap objektif, terutama pada kondisi data yang tidak seimbang.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

Secara keseluruhan, rangkaian metrik evaluasi ini digunakan untuk mengukur performa model secara komprehensif dan objektif. Hasil pengukuran tersebut menjadi landasan validasi utama efektivitas algoritma SVM.

2.6. Visualisasi

WordCloud dimanfaatkan sebagai media visual untuk memberikan gambaran awal mengenai kata-kata yang paling sering muncul dalam data sentimen setelah melalui *preprocessing*. Visualisasi ini menampilkan frekuensi kemunculan suatu kata melalui variasi ukuran pada setiap kata [15]. Selain itu, diagram batang juga digunakan untuk menampilkan perbandingan atau hasil analisis yang telah dilakukan.

3. HASIL DAN ANALISIS

3.1. Data Preprocessing

Tahap *data preprocessing* dilakukan untuk mengubah data mentah yang masih mengandung berbagai elemen tidak relevan menjadi data yang bersih, terstruktur dan dapat diolah secara komputasional. Adapun tahapan yang dilakukan meliputi *cleaning*, *case folding*, *normalization*, *stopword removal*, dan *stemming*.

a. Cleaning

Langkah pertama dalam *preprocessing* adalah *cleaning*, yaitu proses untuk menghapus karakter atau komponen yang tidak bernilai informasi seperti angka, tanda baca, simbol, atau emoji yang tidak memiliki makna bagi analisis. Tujuannya adalah agar data menjadi lebih bersih dan seragam sebelum melalui tahap selanjutnya. Sebagaimana terlihat pada Tabel 1, teks sebelum *cleaning* masih mengandung tanda baca seperti titik dan huruf kapital berlebih, sementara setelah *cleaning*, teks telah dibersihkan dan tampil lebih rapi tanpa karakter yang tidak diperlukan.

Tabel 1. *Cleaning*

Sebelum <i>Cleaning</i>	Setelah <i>Cleaning</i>
TOLONG UNTUK SISTEM KEAMANAN DI PERBAIKI. KALAU 1 ATAU 2 ORNG YG KE HACK WAJAR, MUNGKIN KETELEDORAN PENGGUNA. TAPI KALAU SUDAH BANYAK YG TERKENA HACK. BERARTI ADA YG SALAH DENGAN SISTEM KEAMANAN DANA	TOLONG UNTUK SISTEM KEAMANAN DI PERBAIKI KALAU ATAU ORNG YG KE HACK WAJAR MUNGKIN KETELEDORAN PENGGUNA TAPI KALAU SUDAH BANYAK YG TERKENA HACK BERARTI ADA YG SALAH DENGAN SISTEM KEAMANAN DANA

b. *Case Folding*

Tahap berikutnya adalah *case folding*, yaitu proses mengonversi seluruh karakter huruf menjadi bentuk huruf kecil (*lowercase*). Langkah ini bertujuan agar sistem tidak memperlakukan perbedaan pada huruf besar dan huruf kecil, sehingga analisis teks dapat dilakukan secara lebih konsisten. Pada Tabel 2, hasil *case folding* menunjukkan bahwa seluruh huruf kapital pada teks sebelumnya telah diubah menjadi huruf kecil, tanpa mengubah makna dari kalimat aslinya.

Tabel 2. *Case Folding*

Sebelum <i>Case Folding</i>	Setelah <i>Case Folding</i>
TOLONG UNTUK SISTEM KEAMANAN DI PERBAIKI KALAU ATAU ORNG YG KE HACK WAJAR MUNGKIN KETELEDORAN PENGGUNA TAPI KALAU SUDAH BANYAK YG TERKENA HACK BERARTI ADA YG SALAH DENGAN SISTEM KEAMANAN DANA	tolong untuk sistem keamanan di perbaiki kalau atau orng yg ke hack wajar mungkin keteledoran pengguna tapi kalau sudah banyak yg terkena hack berarti ada yg salah dengan sistem keamanan dana

c. *Normalization*

Selanjutnya, data melewati tahap *normalization* yang bertujuan untuk menstandarisasi penulisan kata, khususnya mengubah singkatan dan bahasa gaul menjadi kosa kata baku bahasa Indonesia. Sebagai contoh penerapannya, kata singkatan seperti "yg" akan dikonversi secara otomatis menjadi bentuk lengkapnya, yaitu "yang". Proses ini menjadi kunci penting dalam memastikan analisis sentimen dapat berjalan dengan akurasi yang tinggi. Tabel 3 menyajikan gambaran yang lebih rinci mengenai perbandingan antara kondisi data sebelum dan sesudah melalui proses *normalization*.

Tabel 3. *Normalization*

Sebelum <i>Normalization</i>	Setelah <i>Normalization</i>
tolong untuk sistem keamanan di perbaiki kalau atau orng yg ke hack wajar mungkin keteledoran pengguna tapi kalau sudah banyak yg terkena hack berarti ada yg salah dengan sistem keamanan dana	tolong untuk sistem keamanan di perbaiki kalau atau orang yang ke hack wajar mungkin keteledoran pengguna tapi kalau sudah banyak yang terkena hack berarti ada yang salah dengan sistem keamanan dana

d. *Stopword Removal*

Tahap berikutnya adalah *stopword removal*, yaitu proses menghilangkan kata-kata umum seperti "dan", "yang", "di", atau "pada" yang tidak memiliki kontribusi signifikan terhadap konteks analisis. Dengan menghilangkan kata-kata tersebut, model dapat lebih menekankan pada istilah-istilah yang memiliki bobot informasi lebih tinggi. Hasil tahap ini dapat dilihat pada Tabel 4, di mana sejumlah kata umum telah dihilangkan sehingga teks menjadi lebih padat namun tetap mewakili makna utama kalimat.

Tabel 4. *Stopword Removal*

Sebelum <i>Stopword Removal</i>	Setelah <i>Stopword Removal</i>
---------------------------------	---------------------------------

tolong untuk sistem keamanan di perbaiki kalau atau orang yang ke hack wajar mungkin keteledoran pengguna tapi kalau sudah banyak yang terkena hack berarti ada yang salah dengan sistem keamanan dana	tolong sistem keamanan perbaiki kalau orang hack wajar mungkin keteledoran pengguna kalau banyak terkena hack berarti ada salah sistem keamanan dana
--	---

e. Stemming

Tahap akhir adalah *stemming*, yaitu proses mengubah kata-kata menjadi bentuk dasarnya (*root word*). Seperti kata "perbaiki" akan dikonversi menjadi "baik" dan "keamanan" menjadi "aman". Tujuan dari proses ini adalah menyatukan berbagai variasi kata yang memiliki akar makna serupa sehingga analisis dapat dilakukan lebih efisien. Perbandingan hasil sebelum dan sesudah *stemming* ditunjukkan pada Tabel 5, di mana teks tampak lebih sederhana karena hanya menyisakan bentuk dasar dari tiap kata.

Tabel 5. Stemming

Sebelum Stemming	Setelah Stemming
tolong sistem keamanan perbaiki kalau orang hack wajar mungkin keteledoran pengguna kalau banyak terkena hack berarti ada salah sistem keamanan dana	tolong sistem aman baik kalau orang hack wajar mungkin teledor guna kalau banyak kena hack arti ada salah sistem aman dana

3.2. Klasifikasi Menggunakan SVM dengan TF-IDF Vectorizer

Setelah melewati tahap *preprocessing*, data ulasan dikonversi ke dalam bentuk numerik menggunakan *TF-IDF Vectorizer* untuk menentukan bobot penting setiap kata dalam korpus. Hasil representasi tersebut kemudian digunakan sebagai *input* bagi model *SVM*. Model ini mengklasifikasikan ulasan ke dalam tiga sentimen, yaitu *negative*, *neutral*, dan *positive*, dengan pembagian data sebesar 80% untuk *training* dan 20% untuk *testing*. Hasil evaluasi ditampilkan pada Gambar 2.

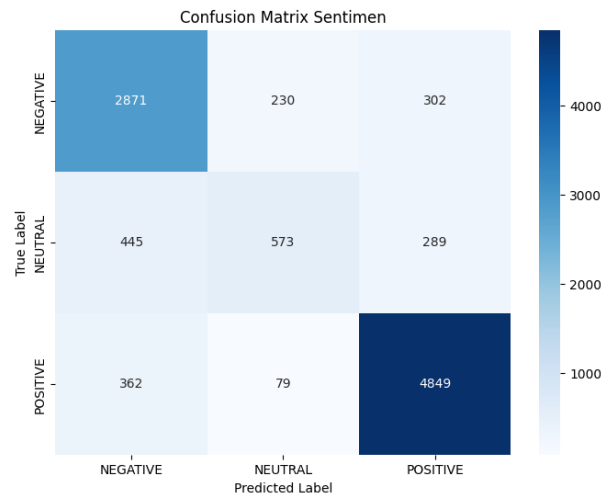
	precision	recall	f1-score	support
NEGATIVE	0.78	0.84	0.81	3403
NEUTRAL	0.65	0.44	0.52	1307
POSITIVE	0.89	0.92	0.90	5290
accuracy			0.83	10000
macro avg	0.77	0.73	0.75	10000
weighted avg	0.82	0.83	0.82	10000

Gambar 2. Metrik Evaluasi

Pengujian terhadap performa model memperlihatkan bahwa kombinasi metode *SVM* dan *TF-IDF Vectorizer* cukup efektif dalam menangani analisis sentimen ini, dibuktikan dengan tingkat akurasi keseluruhan yang menyentuh angka 83%. Model menunjukkan dominasi kinerja terbaik pada kelas sentimen positif, di mana nilai *precision* mencapai 89%, *recall* 92%, dan *f1-score* di angka 90% yang berarti model sangat jarang melakukan kesalahan dalam memprediksi ulasan positif. Di sisi lain, kemampuan model dalam mengidentifikasi sentimen negatif juga tergolong efektif, ditandai dengan perolehan *precision* 78%, *recall* 84%, dan *f1-score* 81%. Akan tetapi, performa model mengalami penurunan yang cukup tajam saat berhadapan dengan kelas netral. Kategori ini mencatatkan performa paling rendah dibandingkan lainnya, dengan *precision* 65%, *recall* 44%, dan *f1-score* hanya 52%. Dalam ekosistem *fintech*, metrik *recall* lebih diprioritaskan, khususnya untuk sentimen negatif, guna meminimalkan risiko terlewatnya keluhan krusial pengguna (*False Negative*). Hasil penelitian menunjukkan model telah memenuhi kriteria ini dengan capaian *recall* pada kelas negatif sebesar 84%. Hal ini membuktikan bahwa model efektif bekerja sebagai alat deteksi masalah layanan secara menyeluruh.

3.3. Confusion Matrix

Untuk memahami lebih dalam kinerja model dalam mengklasifikasikan sentimen, dilakukan analisis menggunakan *confusion matrix* sebagaimana ditunjukkan pada Gambar 3.

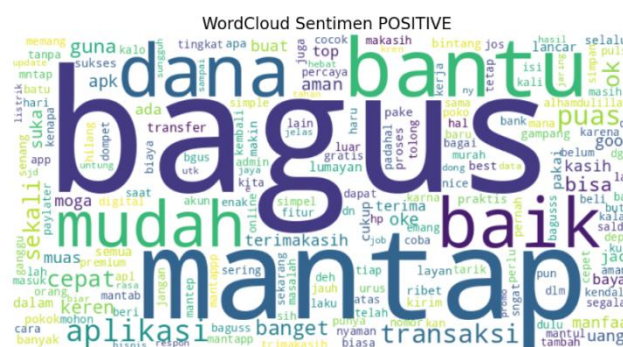


Gambar 3. *Confusion Matrix*

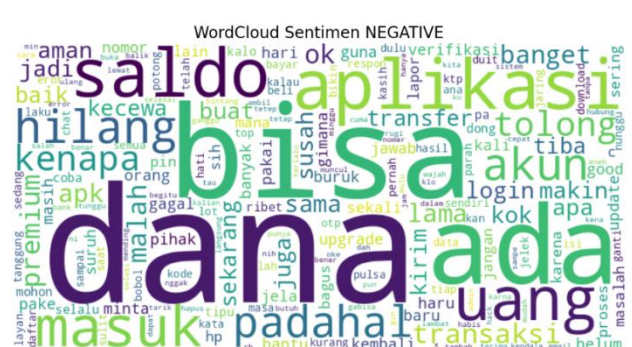
Matriks tersebut memperlihatkan distribusi hasil prediksi model terhadap label sebenarnya pada tiga kelas sentimen, yaitu *negative*, *neutral*, dan *positive*. Dari hasil yang ditampilkan, model menunjukkan performa terbaik pada kelas *positive*, terdapat 4.849 data yang sudah terklasifikasi dengan benar dan hanya sebagian kecil salah diprediksi sebagai *negative* atau *neutral*. Model juga menunjukkan hasil yang cukup baik pada kelas *negative* dengan 2.871 prediksi benar, meskipun terdapat beberapa kesalahan klasifikasi ke kelas lain. Sementara itu, kelas *neutral* memiliki jumlah kesalahan tertinggi, dengan banyak data netral yang justru diklasifikasikan sebagai *negative* atau *positive*. Dari 1707 prediksi yang salah, sebagai contoh ulasan "Sudah lama menunggu instalnya" yang memiliki label *negative* justru diprediksi *positive*. Model kesulitan menangkap nuansa negatif dari frasa "lama menunggu" karena kata "sudah" dihapus sebagai *stopword*, dan kata-kata yang tersisa "lama tunggu instal" tidak memiliki bobot negatif yang cukup tinggi untuk mengalahkan potensi asosiasi positif atau netral dari kata "instal" atau bahkan "tunggu" dalam data *training*.

3.4. Visualisasi WordCloud

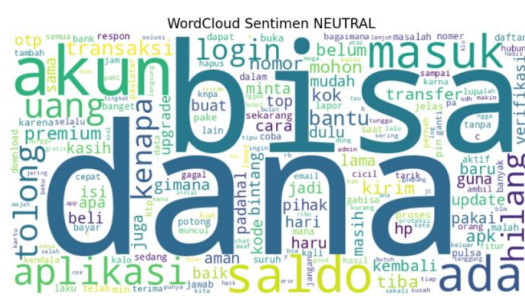
Untuk menampilkan kata-kata yang paling dominan dalam kumpulan ulasan, dilakukan visualisasi menggunakan *WordCloud* sebagaimana terlihat pada Gambar 4 sampai Gambar 6.



Gambar 4. WordCloud sentimen POSITIVE



Gambar 5. WordCloud sentimen *NEGATIVE*

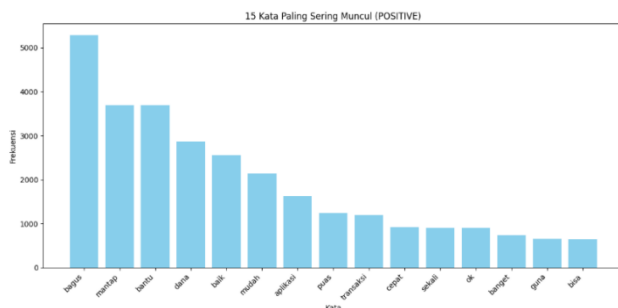


Gambar 6. WordCloud sentimen *NEUTRAL*

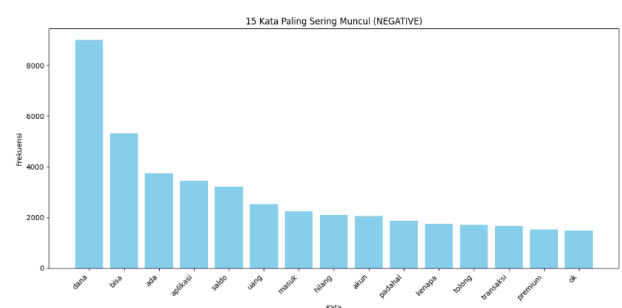
WordCloud menampilkan frekuensi kemunculan kata berdasarkan ukuran, kata yang lebih besar merepresentasikan tingkat kemunculan yang lebih sering. Dari hasil visualisasi, terlihat bahwa kata seperti "dana", "bagus" dan "bisa" muncul paling dominan, menunjukkan topik yang paling sering dibahas oleh pengguna. Kata-kata tersebut mencerminkan fokus utama ulasan terhadap kinerja aplikasi dan pengalaman pengguna secara keseluruhan.

3.5. Analisis Frekuensi Kata

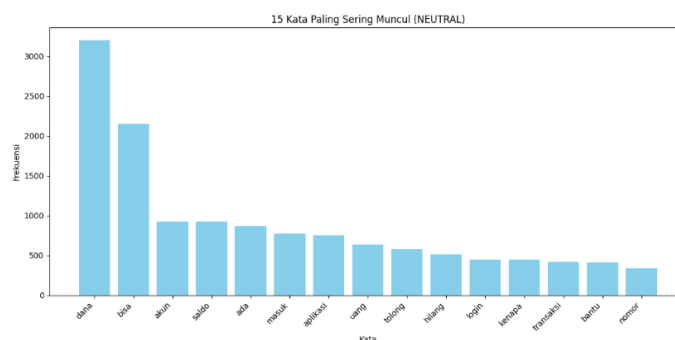
Untuk melengkapi hasil visualisasi *WordCloud*, dilakukan analisis kuantitatif terhadap 15 kata yang memiliki frekuensi kemunculan tertinggi, sebagaimana ditunjukkan pada Gambar 7 sampai Gambar 9.



Gambar 7. Kata Dengan Frekuensi Kemunculan Tertinggi Pada Sentimen *POSITIVE*



Gambar 8. Kata Dengan Frekuensi Kemunculan Tertinggi Pada Sentimen *NEGATIVE*

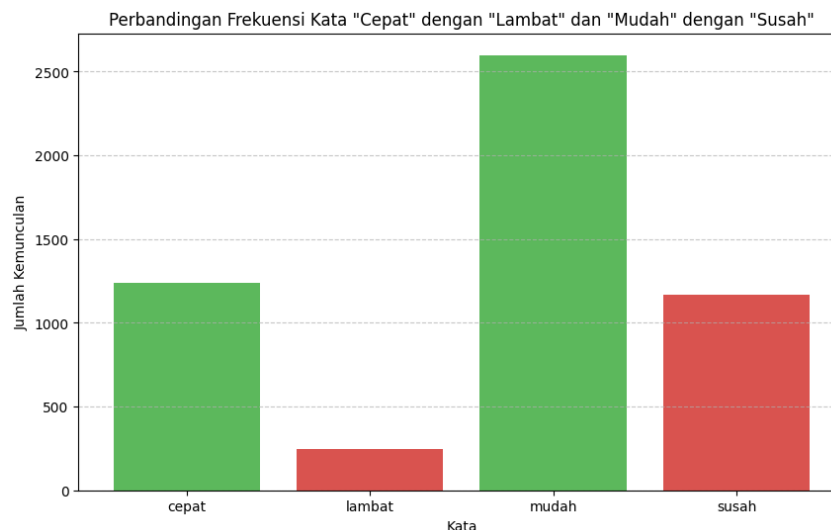


Gambar 9. Kata Dengan Frekuensi Kemunculan Tertinggi Pada Sentimen *NEUTRAL*

Analisis visualisasi 15 kata yang paling sering muncul pada ketiga kelas sentimen menunjukkan pola pengalaman pengguna yang distingtif, di mana sentimen positif didominasi oleh kata sifat apresiatif seperti "bagus", "mantap", "mudah", dan "cepat" yang mencerminkan kepuasan tinggi terhadap performa dan kegunaan sistem, sedangkan sentimen negatif justru menyoroiti isu krusial terkait keamanan dan finansial melalui frekuensi tinggi kata "saldo", "uang", "hilang", dan "padahal" yang mengindikasikan masalah kegagalan transaksi atau kehilangan uang. Di sisi lain, sentimen netral cenderung memuat kata-kata operasional dan interogatif seperti "akun", "login", "tolong", dan "kenapa", yang merepresentasikan pengguna yang sedang mengalami hambatan akses atau membutuhkan layanan bantuan teknis tanpa muatan emosi yang ekstrem.

3.6. Perbandingan Frekuensi Kata

Untuk memperdalam analisis terhadap persepsi pengguna, dilakukan perbandingan antara kata yang memiliki makna berlawanan, yaitu "cepat" dan "lambat", serta "mudah" dan "susah", sebagaimana ditunjukkan pada Gambar 10.



Gambar 10. Perbandingan Frekuensi Kata

Hasil analisis menunjukkan bahwa kata "cepat" muncul sebanyak 1.237 kali, sedangkan "lambat" hanya 247 kali, yang menandakan bahwa mayoritas pengguna menilai kinerja aplikasi cukup responsif dan efisien. Selain itu, kata "mudah" memiliki frekuensi kemunculan 2.599 kali, lebih tinggi dibandingkan "susah" yang muncul 1.167 kali. Perbandingan frekuensi kata menyajikan akumulasi sentimen dari keseluruhan periode data, namun tidak merepresentasikan stabilitas performa antar versi aplikasi. Terdapat kemungkinan bahwa mayoritas ulasan "mudah" berasal dari pengguna versi aplikasi terdahulu yang stabil, sementara kemunculan kata "susah" justru terkonsentrasi secara spesifik setelah peluncuran pembaruan (*update*) versi terbaru. Dalam skenario tersebut, data agregat dapat menyamarkan penurunan kualitas layanan yang sebenarnya sedang terjadi pada versi mutakhir (terkini). Oleh karena itu, dominasi kuantitatif kata positif dalam dataset historis tidak serta-merta menjamin kepuasan pengguna pada kondisi teknis saat ini (*real-time*).

4. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa penerapan metode *Support Vector Machine (SVM)* yang dipadukan dengan ekstraksi fitur *TF-IDF* mampu memberikan hasil yang efektif dalam melakukan analisis sentimen pengguna aplikasi DANA. Dengan memanfaatkan dataset berskala besar sebanyak 50.000 ulasan dari Google Play Store, model yang dibangun berhasil mencapai tingkat akurasi sebesar 83%. Jika ditinjau lebih mendalam, model tampak sangat unggul dalam mengenali kelas *positive*, yang terlihat dari capaian *precision* 89%, *recall* 92%, dan *F1-score* 90%. Namun, model kinerja menangani masalah khusus dalam klasifikasi kelas *neutral*, di mana capaian skornya masih relatif rendah, yaitu nilai *precision* sebesar 65%, *recall* 44%, dan *F1-score* 52%. Untuk mengatasi rendahnya akurasi pada kelas *neutral* akibat ambiguitas bahasa, penelitian selanjutnya disarankan menerapkan ekstraksi fitur N-gram atau model berbasis *Deep Learning* (seperti IndoBERT) guna menangkap konteks semantik yang lebih utuh. Selain itu, penerapan teknik penyeimbang data (*resampling*) seperti SMOTE sangat direkomendasikan untuk menanggulangi ketimpangan jumlah data latih yang menyebabkan model kurang optimal dalam mengenali pola sentimen *neutral*.

Selain itu, analisis frekuensi kata memperlihatkan bahwa kata "cepat" muncul sebanyak 1.237 kali dibandingkan "lambat" sebanyak 247 kali, serta kata "mudah" muncul sebanyak 2.599 kali dibandingkan "susah" sebanyak 1.167 kali. Hasil ini mengindikasikan bahwa mayoritas pengguna memberikan ulasan yang bernada positif terhadap kinerja dan kemudahan penggunaan aplikasi DANA. Dengan demikian, pengalaman pengguna dengan aplikasi DANA secara umum cenderung positif, terutama jika dilihat dari kecepatan layanan dan kemudahan penggunaan.

REFERENSI

- [1] S. A. Ghaffar and W. C. Setiawan, "Comparative Sentiment Analysis of Digital Wallet Applications in Indonesia Using Naïve Bayes," *Int. J. Informatics Inf. Syst.*, vol. 8, no. 2, pp. 55–66, 2025, doi: 10.47738/ijis.v8i2.251.

- [2] R. M. Simamora and M. Y. Fathoni, "Sentiment Analysis of Customer Reviews on DANA Application Service using Multinomial Naïve Bayes Classifier Algorithm," *COMNETSAT 2024 - IEEE Int. Conf. Commun. Networks Satell.*, pp. 144–151, 2024, doi: 10.1109/COMNETSAT63286.2024.10862974.
- [3] H. Atsqalani, N. Hayatin, and C. S. K. Aditya, "Sentiment Analysis from Indonesian Twitter Data Using Support Vector Machine And Query Expansion Ranking," *J. Online Inform.*, vol. 7, no. 1, pp. 116–122, 2022, doi: 10.15575/join.v7i1.669.
- [4] R. A. Pranata, Rudiman, and N. A. Verdikha, "Metode Pembobotan TF-IDF Untuk Klasifikasi Teks Quick Count Pemilihan Wakil Presiden Indonesia 2024 Pada X Twitter dengan Metode SVM," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 18, no. 2, pp. 126–138, 2024, doi: 10.54914/jit.v10i2.1279.
- [5] N. Ananda, M. Fikry, Yusra, L. Handayani, and I. Iskandar, "Klasifikasi Sentimen Tweet Masyarakat Terhadap Kendaraan Listrik Menggunakan Support Vector Machine," *J. Inform. Univ. Pamulang*, vol. 8, no. 4, pp. 568–577, 2023, doi: 10.32493/informatika.v8i4.36754.
- [6] S. P. Azzahra, Y. A. Apriyanto, and A. Wijaya, "Analisis Sentimen Ulasan Aplikasi Deepl Pada Google Play Dengan Metode Support Vector Machine (SVM)," *J. Sist. Inf.*, vol. 4, no. 2, pp. 59–66, 2023, doi: 10.32546/jusin.v4i2.2368.
- [7] U. I. Arsyah, M. Pratiwi, and A. Muhammad, "Twitter Sentiment Analysis of Public Space Opinions using SVM and TF-IDF Methods," *Indones. J. Comput. Sci.*, vol. 13, no. 1, pp. 387–394, 2024, doi: 10.33022/ijcs.v13i1.3594.
- [8] M. Arman, Y. A. Ameliana, and A. Arfah, "Pengaruh Kepercayaan Konsumen Pada Loyalitas Pengguna Fintech (Layanan Transaksi Online)," *SEIKO J. Manag. Bus.*, vol. 6, no. 2, pp. 384–390, 2023, doi: 10.37531/sejaman.v6i2.4944.
- [9] M. A. Palomino and F. Aider, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis," *Appl. Sci.*, vol. 12, no. 17, pp. 1–21, 2022, doi: 10.3390/app12178765.
- [10] B. A. Maulana, M. J. Fahmi, A. M. Imran, and N. Hidayati, "Analisis Sentimen Terhadap Aplikasi Pluang Menggunakan Algoritma Naive Bayes dan Support Vector Machine (SVM)," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 375–384, 2024, doi: 10.57152/malcom.v4i2.1206.
- [11] B. Harnadi and A. D. Widianoro, "Evaluating the Performance and Accuracy of Supervised Learning Models on Sentiment Analysis of E-Wallet," in *7th International Conference on Information Technology, InCIT 2023*, 2023, pp. 175–180. doi: 10.1109/InCIT60207.2023.10413111.
- [12] J. Setyanto and T. B. Sasongko, "Sentiment Analysis of Sirekap Application Users Using the Support Vector Machine Algorithm," *J. Appl. Informatics Comput.*, vol. 8, no. 1, pp. 71–76, 2024, doi: 10.30871/jaic.v8i1.7772.
- [13] A. Gosal and R. Septifani, "Implementation of Machine Learning in the Classification of Text Created by Humans or Artificial Intelligence," *Res. Innov. Inf. Eng. Technol.*, vol. 1, pp. 45–52, 2025, doi: 10.5220/0014269100004928.
- [14] D. Prismala and I. K. D. Nuryana, "Analisis Opinion Mining Pada Topik Chatgpt di Aplikasi X Dengan Pendekatan Algoritma SVM Berbasis Lexicon," *J. Emerg. Inf. Syst. Bus. Intell.*, vol. 5, no. 03, pp. 62–75, 2024, doi: 10.26740/jeisbi.v5i3.61575.
- [15] J. A. Wibowo, V. C. Mawardi, and T. Sutrisno, "Visualisasi Word Cloud Hasil Analisis Sentimen Berbasis Fitur Layanan Aplikasi Gojek Dengan Support Vector Machine," *J. Serina Sains, Tek. dan Kedokt.*, vol. 2, no. 1, pp. 61–70, 2024, doi: 10.24912/jsstk.v2i1.32058.