

## Perbandingan Metode *Naïve Bayes* dan *Support Vector Machine* untuk Analisis Sentimen pada Ulasan Pengguna Aplikasi X

Meutia Raissa Emiliana<sup>1</sup>, Muhammad Riza Indra Fata<sup>2</sup>, Muhammad Ghaly Adam<sup>3</sup>  
Teknologi Informasi, Teknik dan Informatika, Universitas Bina Sarana Informatika, Indonesia  
<sup>1</sup>meutiaemiliana5@gmail.com, <sup>2</sup>rizaindra2609@gmail.com, <sup>3</sup>ghalymuhammad668@gmail.com

### Abstract

*Advances in digital technology have driven the increasing use of mobile applications as a medium for obtaining information, communicating, and supporting various user activities. Application X is one such widely used platform, resulting in a variety of user reviews representing their experience and level of satisfaction with the available services. This information is crucial for analyzing to comprehensively identify user sentiment trends. This study compares the performance of two machine learning algorithms, Naïve Bayes and Support Vector Machine, in conducting sentiment analysis on user reviews collected through web scraping from the Google Play Store. The retrieved data was then processed through text preprocessing stages including data cleaning, case folding, tokenizing, stopword removal, and stemming. The evaluation results showed that Support Vector Machine performed superiorly with an accuracy of 0.984 and an F1-score of 0.840, while Naïve Bayes produced an accuracy of 0.849 and an F1-score of 0.550. Overall, Support Vector Machine was deemed more effective in classifying user sentiment and can be recommended as a more optimal approach for review analysis in similar mobile applications.*

**Keywords:** *application X, mobile application, naïve bayes, support vector machine, sentiment analysis*

### Abstrak

Kemajuan teknologi digital telah mendorong meningkatnya penggunaan aplikasi mobile sebagai media untuk memperoleh informasi, melakukan komunikasi, serta mendukung berbagai aktivitas pengguna. Aplikasi X merupakan salah satu platform yang banyak dimanfaatkan, sehingga beragam ulasan yang diberikan oleh pengguna muncul sebagai representasi persepsi pengguna mengenai kualitas layanan yang diterima serta tingkat kenyamanan yang mereka rasakan saat memanfaatkannya. Informasi tersebut penting untuk dianalisis guna mengidentifikasi kecenderungan sentimen pengguna secara komprehensif. Penelitian ini mengkaji perbedaan performa dua algoritma machine learning, yakni Naïve Bayes dan Support Vector Machine, dalam mengklasifikasikan sentimen pada ulasan pengguna yang diperoleh melalui teknik web scraping dari Google Play Store. Dataset yang diambil selanjutnya diproses melalui serangkaian tahap text preprocessing, meliputi pembersihan data, normalisasi huruf, tokenisasi, eliminasi stopword, serta proses stemming. Berdasarkan hasil pengujian, Support Vector Machine menunjukkan kinerja yang lebih unggul dengan akurasi mencapai 0.984 dan F1-Score sebesar 0.840, sementara Naïve Bayes menghasilkan akurasi 0.849 dan F1-Score 0.550. Dengan demikian, Support Vector Machine dinilai lebih efektif dalam mengidentifikasi sentimen pengguna dan direkomendasikan sebagai metode yang lebih optimal untuk analisis ulasan pada aplikasi mobile yang memiliki karakteristik serupa.

**Kata kunci:** aplikasi X, aplikasi mobile, naïve bayes, support vector machine, analisis sentimen

## 1. Pendahuluan

Di tengah pesatnya perkembangan teknologi digital, aplikasi mobile telah menjadi elemen penting dalam aktivitas sehari-hari masyarakat, mulai dari akses informasi hingga hiburan dan interaksi sosial. Aplikasi X merupakan salah satu platform yang memiliki tingkat pemanfaatan yang tinggi oleh pengguna, yaitu platform yang menyediakan fasilitas berbagi informasi, komunikasi, serta perluasan jaringan profesional. Informasi dari Google Play Store menunjukkan bahwa aplikasi ini berkontribusi dalam membantu pengguna menemukan peluang kerja, membangun koneksi, dan meningkatkan keterampilan. Namun demikian, sebagaimana aplikasi digital pada umumnya, Aplikasi X juga menerima beragam ulasan yang merefleksikan pengalaman pengguna, baik positif maupun negatif, terhadap kualitas layanan yang diberikan [1]. Ulasan pengguna atau online customer reviews merupakan bentuk evaluasi konsumen terhadap suatu produk atau layanan dengan menyoroti aspek seperti fitur, kinerja, dan pengalaman penggunaan [2]. Pada Aplikasi X, sebagian ulasan mengungkapkan kendala teknis seperti kesulitan login, gangguan saat mengunggah data, serta kebutuhan dukungan bahasa lokal. Di sisi lain, sejumlah ulasan positif menyoroti kemudahan pencarian lowongan dan kecepatan notifikasi. Tingginya jumlah ulasan yang disajikan dalam format tidak terorganisasi menyebabkan proses analisis secara manual menjadi semakin kompleks dan sulit dilakukan, oleh karena itu diperlukan metode komputasional, seperti analisis sentimen, guna memetakan kecenderungan opini pengguna secara lebih terstruktur dan sistematis.

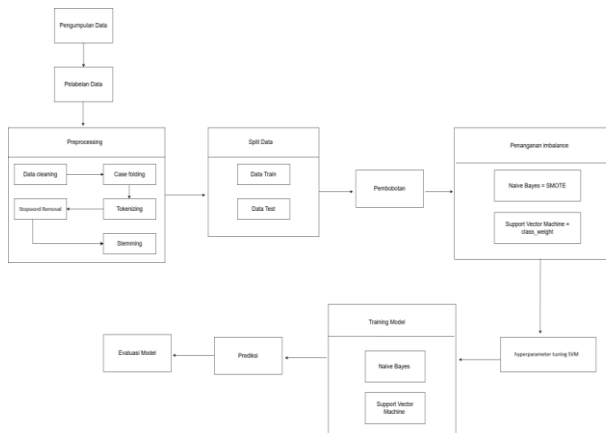
Pada implementasi analisis sentimen, salah satu isu yang kerap ditemui adalah distribusi data antar kelas yang tidak seimbang. Dalam sejumlah kondisi, proporsi ulasan positif dapat jauh mendominasi atau bahkan lebih sedikit dibandingkan ulasan negatif, yang pada akhirnya membuat model cenderung memberikan prediksi berpihak kepada kelas dengan jumlah data terbanyak. Situasi tersebut berpotensi mengurangi efektivitas model dalam mengidentifikasi kelas yang jumlah datanya lebih sedikit. Sebagaimana dijelaskan oleh [3], ketidakseimbangan data dapat menghasilkan akurasi yang tampak tinggi, tetapi sebenarnya mengurangi performa pada kelas dengan jumlah lebih sedikit. Pada penelitian ini, dataset ulasan Aplikasi X terdiri atas dua kelas, yaitu positif dan negatif, sehingga pemeriksaan distribusi data menjadi penting untuk memastikan bahwa proses klasifikasi tidak bersifat bias. Algoritma Naïve Bayes merupakan metode klasifikasi yang berlandaskan pendekatan probabilistik, di mana perhitungannya mengacu pada Teorema Bayes dengan mengasumsikan bahwa setiap fitur bersifat saling independen [4]. Algoritma ini dikenal sederhana dan efisien, meskipun performanya dapat menurun ketika fitur memiliki keterkaitan. Di

sisi lain, Support Vector Machine (SVM) memiliki kemampuan yang kuat dalam melakukan pemisahan data pada ruang fitur berdimensi tinggi serta menunjukkan ketahanan terhadap overfitting, meskipun metode ini umumnya memerlukan proses komputasi yang lebih intensif [5]. Kedua algoritma tersebut termasuk ke dalam supervised learning dan banyak diterapkan pada klasifikasi sentimen biner.

Selain pemilihan algoritma, representasi fitur teks juga memegang peranan penting dalam analisis sentimen. Penelitian ini menerapkan teknik pembobotan Term Frequency–Inverse Document Frequency (TF-IDF) karena metode tersebut dapat menentukan nilai bobot setiap kata berdasarkan tingkat keterkaitannya dalam suatu dokumen maupun keseluruhan korpus. Teknik ini membantu menonjolkan term yang informatif sekaligus mengurangi pengaruh kata yang kurang penting. Temuan sebelumnya juga menunjukkan bahwa TF-IDF dapat mengatasi keterbatasan vectorizer tradisional dengan memberikan bobot yang lebih proporsional pada setiap term, sehingga meningkatkan efektivitas proses klasifikasi [6]. Selain itu, TF-IDF bersifat ringan, mudah diterapkan, dan sesuai untuk dataset berskala kecil hingga menengah. Beberapa studi sebelumnya menyatakan bahwa Naïve Bayes dan Support Vector Machine merupakan algoritma yang sering diterapkan dalam analisis sentimen, di mana efektivitas keduanya sangat bergantung pada karakteristik data serta tingkat keseimbangannya [6]. Studi lain juga mencatat bahwa algoritma Naïve Bayes mampu mencapai tingkat akurasi sebesar 88%, sementara Support Vector Machine memperoleh akurasi hingga 90% dalam penerapan analisis sentimen terhadap ulasan pengguna LinkedIn [7]. Berdasarkan landasan tersebut, penelitian ini difokuskan pada analisis perbandingan performa kedua algoritma tersebut dalam mengklasifikasikan sentimen ulasan pengguna Aplikasi X untuk menentukan metode yang paling akurat dan relevan dalam mendukung peningkatan kualitas layanan aplikasi.

## 2. Metode Penelitian

Tahapan penelitian ini mencakup beberapa langkah utama, dimulai dengan proses pengumpulan serta pelabelan data. Setelah itu, dilakukan tahap pra-proses yang mencakup data cleaning, case folding, tokenizing, stopword removal, dan stemming. Data yang telah diproses kemudian dipisahkan menjadi dua bagian untuk kebutuhan pelatihan dan pengujian, diikuti oleh proses pembobotan fitur dan penanganan ketidakseimbangan data. Selanjutnya, dilakukan penyesuaian hyperparameter pada algoritma SVM, kemudian model dilatih, digunakan untuk melakukan prediksi, dan dievaluasi performanya. Rangkaian lengkap tahapan tersebut ditunjukkan pada Gambar 1.



Gambar 1. Tahap Penelitian

## 2.1 Pengumpulan Data

Tahap ini bertujuan memperoleh data ulasan pengguna dari Google Play Store dengan memanfaatkan library Python google-play-scraper pada lingkungan Jupyter Notebook. Sebanyak 5000 ulasan Aplikasi X berhasil dikumpulkan, yang mencakup teks ulasan dan rating pengguna, selanjutnya data tersebut disimpan dalam format CSV guna mendukung proses prapemrosesan. pada tahap berikutnya. Pemanfaatan Jupyter Notebook dianggap penting karena platform ini memungkinkan integrasi kode, penjelasan teks, dan keluaran analisis secara interaktif, sehingga meningkatkan efisiensi serta reproduktibilitas proses analisis [8].

## 2.2 Pelabelan Data

Pelabelan merupakan tahap krusial dalam analisis sentimen karena menentukan dasar klasifikasi sesuai dengan ekspresi emosional maupun opini yang muncul pada teks. Pada tahap ini, setiap ulasan dikelompokkan ke dalam dua jenis sentimen, yakni positif dan negatif. Proses pelabelan dilakukan secara otomatis berdasarkan nilai rating pengguna yang terdapat pada data hasil scraping, sehingga mempercepat proses pengelompokan data [9].

## 2.3 Preprocessing

Tahap ini dilakukan sebagai proses awal untuk mempersiapkan data teks sebelum dianalisis dengan cara membersihkan serta menstandarkan bentuk kata agar data menjadi lebih konsisten dan terstruktur. Proses pra-pemrosesan yang diterapkan mencakup data cleaning, case folding, tokenizing, stopwords removal, dan stemming. Penerapan rangkaian preprocessing tersebut memegang peranan penting dalam meningkatkan kualitas data serta menunjang ketepatan hasil analisis sentimen. [10],[11].

### 2.3.1 Data Cleaning

Tahap data cleaning atau pembersihan data dilakukan untuk memastikan bahwa data berada dalam kondisi yang bersih, konsisten, dan layak diolah pada tahap

analisis selanjutnya. Proses ini mencakup penghapusan berbagai elemen teks yang tidak diperlukan seperti duplikasi data, karakter numerik, tanda baca, maupun emoji, sehingga hanya tersisa informasi yang bermakna untuk proses analisis sentimen [12].

### 2.3.2 Case Folding

Case folding merupakan tahapan awal dalam proses text preprocessing yang berfungsi menyeragamkan teks dengan mengubah seluruh karakter huruf menjadi huruf kecil, sehingga data memiliki konsistensi yang lebih baik dan mempermudah pemrosesan pada tahap analisis berikutnya [13].

### 2.3.3 Tokenizing

Tokenizing adalah proses memisahkan suatu kalimat menjadi unit-unit kata (token) sehingga teks dapat dianalisis secara terstruktur dan pola linguistik di dalamnya dapat dikenali dengan lebih akurat [14].

### 2.3.4 Stopword Removal

Stopword removal merupakan proses seleksi kata dalam dataset dengan menghilangkan kata-kata yang bersifat umum, seperti konjungsi atau kata yang tidak memberikan makna signifikan, sehingga teks yang tersisa hanya mencakup kata-kata yang lebih relevan untuk analisis.

### 2.3.5 Stemming

Stemming merupakan proses mengonversi kata berimbuhan menjadi bentuk dasar sehingga diperoleh keseragaman bentuk kata dan mempermudah sistem dalam melakukan analisis teks [15].

## 2.4 Split Data

Tahap ini dilakukan untuk memisahkan dataset menjadi dua bagian utama agar proses pelatihan dan pengujian model dapat berlangsung secara objektif, yaitu data pelatihan (training data) dan data pengujian (testing data). Data pelatihan digunakan untuk membangun dan menyesuaikan algoritma supaya mampu mempelajari pola serta karakteristik yang terkandung dalam dataset. Sebaliknya, data pengujian berfungsi untuk menilai kinerja serta mengukur tingkat akurasi model setelah proses pelatihan selesai dilakukan. Melalui pemisahan ini, evaluasi model menjadi lebih valid karena penilaiannya tidak dipengaruhi oleh data yang sebelumnya terlibat dalam proses pembelajaran [16].

## 2.5 Pembobotan

Setelah proses pra-pemrosesan diselesaikan, tahap selanjutnya adalah menerapkan teknik pembobotan TF-IDF (Term Frequency–Inverse Document Frequency) untuk mengukur tingkat kepentingan setiap istilah dalam kumpulan ulasan. Pendekatan ini berfungsi mengidentifikasi kata yang memiliki nilai informatif tinggi, yaitu istilah yang muncul secara dominan pada satu dokumen namun jarang ditemukan

pada dokumen lain. Secara umum, TF-IDF bertujuan menilai sejauh mana suatu kata relevan baik pada dokumen individual maupun pada keseluruhan korpus, sehingga kata yang paling bermakna dapat diprioritaskan untuk proses analisis berikutnya [17].

Pada tahap ini, setiap istilah memperoleh bobot melalui kombinasi perhitungan Term Frequency (TF), yang mencerminkan intensitas kemunculan kata dalam suatu dokumen, dan Inverse Document Frequency (IDF), yang menggambarkan tingkat kekhasan kata di seluruh dokumen. Dengan penggunaan TF-IDF, sistem mampu menyoroti kata-kata yang memiliki kontribusi paling besar terhadap konteks ulasan serta memfasilitasi identifikasi kecenderungan sentimen positif, negatif, maupun netral. Adapun perhitungan nilai term frequency dalam suatu dokumen dapat ditentukan menggunakan persamaan (1) berikut [12]:

$$tf_{t,d} = \frac{f_{t,d}}{\sum_{i=1}^n f_{i,d}} \quad (1)$$

Nilai Inverse Document Frequency (IDF) diperoleh melalui proses perhitungan yang merujuk pada persamaan (2) [18]

$$idf_t = \log \log \left( \frac{N}{df_t} \right) \quad (2)$$

Nilai N merepresentasikan jumlah total dokumen yang terdapat dalam suatu himpunan data. Setelah memperoleh nilai TF dan IDF, langkah selanjutnya adalah menentukan bobot TF-IDF yang dihitung melalui perkalian kedua nilai tersebut. Mengacu pada kedua persamaan sebelumnya, proses perhitungan bobot TF-IDF dapat dilakukan menggunakan persamaan (3) berikut [18].

$$tfidf_{t,d} = tf_{t,d} \times idf_t \quad (3)$$

## 2.6 Penanganan Imbalance

Upaya mengatasi ketidakseimbangan pada dataset dilakukan untuk mencegah model terlalu condong mengenali kelas mayoritas, sehingga proses klasifikasi dapat berlangsung lebih objektif dan menghasilkan prediksi yang lebih akurat. Salah satu metode yang digunakan adalah SMOTE, yakni strategi oversampling yang membentuk data sintesis pada kelas minoritas melalui proses interpolasi linier antara satu sampel minoritas dengan beberapa sampel tetangga terdekatnya. Teknik ini secara efektif menambah jumlah contoh pada kelas yang kurang terwakili sehingga tingkat ketidakseimbangan dapat diminimalkan [19]. Selain itu, pendekatan class weight turut dimanfaatkan dengan memberikan bobot yang lebih tinggi pada kelas yang memiliki jumlah data lebih sedikit, sehingga algoritma menjadi lebih peka terhadap kesalahan prediksi pada kelas

minoritas dan tidak lagi mendominasi bobot pembelajaran pada kelas mayoritas [20].

## 2.7 Hyperparameter Tuning pada Support Vector Machine

Hyperparameter Tuning pada Support Vector Machine merupakan proses pengaturan parameter kunci untuk meningkatkan performa klasifikasi. Dalam penelitian yang menggunakan kernel Radial Basis Function (RBF), penyesuaian dilakukan terhadap hyperparameter penting seperti C dan gamma yang menentukan fleksibilitas model serta sensitivitasnya terhadap pola non-linear. Untuk memperoleh kombinasi parameter yang optimal, penelitian tersebut memanfaatkan Random Grid Search yang memungkinkan evaluasi berbagai konfigurasi secara lebih efisien. Melalui pendekatan ini, nilai C dan gamma terbaik berhasil diperoleh dan terbukti meningkatkan akurasi, presisi, spesifisitas, serta F1-score model SVM secara signifikan [21].

## 2.8 Training Model

Tahap ini merupakan bagian utama dari penelitian, yang bertujuan membangun model klasifikasi sentimen menggunakan dua algoritma pembelajaran mesin. Kedua metode diterapkan untuk memperoleh pemahaman yang lebih komprehensif mengenai performa model serta memungkinkan dilakukan perbandingan hasil klasifikasi berdasarkan karakteristik masing-masing algoritma.

### 2.8.1 Naïve Bayes

Naïve Bayes Classifier (NBC) adalah metode klasifikasi yang berlandaskan pendekatan probabilistik dengan menerapkan Teorema Bayes untuk memperkirakan peluang suatu instance masuk ke dalam kelas tertentu. Algoritma ini bekerja dengan asumsi bahwa setiap fitur atau atribut pada data bersifat independen satu sama lain dalam memberikan kontribusi terhadap proses penentuan kelas. Penggunaan NBC ditujukan untuk menghasilkan proses klasifikasi yang cepat dan efisien melalui perhitungan probabilitas pada setiap fitur, sehingga penetapan kelas dapat dilakukan secara sederhana namun tetap menghasilkan tingkat akurasi yang kompetitif, terutama ketika diterapkan pada dataset dalam skala besar. Formulasi utama yang digunakan dalam metode Naïve Bayes tercantum pada Persamaan (4).

$$P(X | Y) = \frac{P(Y | X) \times P(X)}{P(Y)}$$

(4)

Notasi  $P(X|Y)$  merepresentasikan probabilitas terjadinya peristiwa X dengan asumsi bahwa peristiwa Y telah terjadi. Sementara itu,  $P(Y|X)P(X)$  menunjukkan peluang gabungan dari terjadinya Y ketika X diketahui, kemudian dikalikan dengan

probabilitas awal dari  $X$ . Adapun  $P(Y)$  menggambarkan probabilitas total dari peristiwa  $Y$  tanpa mempertimbangkan apakah  $X$  terjadi atau tidak. Melalui persamaan ini, estimasi probabilitas suatu peristiwa dapat diperbarui berdasarkan informasi baru yang diperoleh. Konsep tersebut memiliki peran penting dalam pengambilan keputusan berbasis data dan banyak diterapkan dalam analisis statistik serta proses klasifikasi menggunakan algoritma Naïve Bayes.

### 2.8.2 Support Vector Machine

Support Vector Machine (SVM) merupakan metode pembelajaran mesin yang bekerja berdasarkan konsep Structural Risk Minimization (SRM) untuk menentukan sebuah hyperplane yang paling optimal dalam memisahkan data ke dalam kelas yang berbeda secara maksimal [23]. Pada proses pelatihannya, SVM diarahkan untuk membentuk batas keputusan yang mampu mengklasifikasikan data secara efektif, sehingga pola-pola fitur dapat dikenali dengan tingkat akurasi tinggi. Pendekatan ini menghasilkan performa analisis yang konsisten, efisien, dan stabil dalam berbagai kondisi data [24]. Rumus dasar SVM ditampilkan pada Persamaan (5).

$$f(x) = \text{sign} \left( \sum_{i=1}^n \alpha_i y_i K(x_i, x) + b \right)$$

(5)

Fungsi tersebut menggambarkan mekanisme decision function pada algoritma Support Vector Machine (SVM), yang berperan dalam menentukan kelas suatu data baru berdasarkan pola yang telah dipelajari selama proses pelatihan. Dalam formula tersebut, variabel  $x_i$  merupakan data latih yang memiliki label kelas  $y_i$ , sementara nilai  $\alpha_i$  menunjukkan parameter hasil optimasi selama pelatihan. Kernel  $K(x_i, x)$  digunakan untuk menilai tingkat kesesuaian atau kemiripan antara data yang akan diprediksi dengan setiap data latih, sedangkan komponen bias  $b$  berfungsi mengatur posisi batas keputusan agar pemisahan antar kelas dapat tercapai secara maksimal.

### 2.9 Prediksi

Prediksi memanfaatkan model yang telah dilatih untuk memperkirakan data baru, sehingga keputusan yang dihasilkan bersifat lebih objektif. Pendekatan ini dapat diterapkan di berbagai bidang dengan menggunakan algoritma klasifikasi untuk mengenali pola atau tren dalam data [16].

### 2.10 Model Evaluation

Menilai performa model machine learning merupakan tahap penting untuk mengetahui sejauh mana model tersebut bekerja dengan baik. Salah satu pendekatan evaluasi yang paling umum digunakan

adalah confusion matrix, yaitu alat analisis yang membandingkan keluaran prediksi model dengan data sebenarnya. Matriks ini menyajikan jumlah prediksi yang tepat maupun yang keliru pada setiap kelas, seperti kategori positif dan negatif. Melalui penyajian tersebut, peneliti dapat menilai kemampuan model dalam mengenali serta membedakan tiap label sentimen secara akurat. Selain itu, confusion matrix berperan membantu mengidentifikasi jenis kesalahan klasifikasi yang muncul, sehingga memberikan gambaran mengenai keterbatasan model dalam mempelajari pola data secara menyeluruh.

## 3. Hasil dan Pembahasan

### 3.1 Pengumpulan Data

Proses penelitian diawali dengan melakukan scraping terhadap ulasan pengguna Aplikasi  $X$  yang tersedia pada platform Google Play Store. Pengumpulan data dilakukan secara otomatis menggunakan Jupyter Notebook dengan bantuan library Google Play Scraper pada bahasa pemrograman Python. Melalui prosedur ini, terkumpul sebanyak 3.087 ulasan berbahasa Indonesia yang berasal dari pengguna di wilayah Indonesia. Seluruh hasil ekstraksi kemudian disimpan dalam berkas berformat review\_sentimen\_binary.csv sebagai basis analisis lanjutan. Ringkasan data yang berhasil dihimpun pada tahap ini ditampilkan pada Tabel 1.

Tabel 1. Hasil Scraping Data Ulasan

Nama Pengguna	Rating	Review	Sentimen
Ahmad Junadi	5	Sangat cocok buat berita	Positif
Kania Azura	1	Makin ga jelas aja aplikasinya, tolong diperbaiki ya saya mau login kenapa ga bisa terus dan saya bikin akun baru tiba – tiba ditanggukan terus	Negatif

### 3.2 Labeling

Setelah proses pengumpulan data selesai dilakukan, langkah berikutnya adalah memberikan label sentimen pada setiap ulasan dengan cara mengelompokkan teks ke dalam dua kelas utama, yakni sentimen positif dan sentimen negatif. Tahapan pelabelan ini berfungsi untuk mengungkap kecenderungan opini pengguna terhadap Aplikasi  $X$ , sehingga hasilnya dapat digunakan sebagai fondasi dalam proses analisis sentimen serta pengembangan model machine learning pada tahap selanjutnya. Ringkasan distribusi label yang diperoleh disajikan pada Tabel 2.

Tabel 2. Hasil Pelabelan Data

No	Hasil Ulasan	Sentimen	Label
1	Aplikasi yang sungguh luar biasa	Positif	1
2	X sekarang tambah ga jelas & buat ga nyaman. Baru buka, blom scroll tiba² nge-lag, tiba² keluar sendiri. Padahal jaringan bagus, gaada gangguan samsek. Please perbaiki lagi dong	Negatif	0

### 3.3 Hasil Preprocessing

Data ulasan pada Aplikasi X yang digunakan dalam penelitian ini merupakan bentuk teks yang tidak terstruktur, sehingga diperlukan rangkaian tahapan text pre-processing sebelum dapat digunakan dalam analisis sentimen. Tahapan tersebut meliputi data cleaning, case folding, stopword removal, tokenizing, serta stemming, yang secara keseluruhan bertujuan menghasilkan teks yang lebih bersih, seragam, dan relevan untuk proses analisis lanjutan. Setelah seluruh prosedur pra-pemrosesan tersebut diterapkan, data ulasan menjadi siap diolah dalam tahap pemodelan machine learning. Ringkasan keseluruhan hasil pre-processing ditampilkan pada Tabel 3.

Tabel 3. Hasil Preprocessing

No	Proses	Hasil Preprocessing
1	Pengumpulan Data	kenapa ya padahal dah bisa masuk terus ketika mau komentar ko tiba² jadi kembali ke log in padahal kn dah masuk sungguh tidak membantu sekarang jadi jelek aplikasi ini
2	Pelabelan Data	0
3	Data cleaning	dah masuk komentar ko tiba log in kn dah masuk sungguh bantu jelek aplikasi
4	Case folding	dah masuk komentar ko tiba log in kn dah masuk sungguh bantu jelek aplikasi
5	Tokenizing	['dah', 'masuk', 'komentar', 'ko', 'tiba', 'log', 'in', 'kn', 'dah', 'masuk', 'sungguh', 'bantu', 'jelek', 'aplikasi']
6	Stopword removal	dah masuk komentar ko log in kn dah masuk sungguh bantu jelek aplikasi
7	Stemming	dah masuk komentar ko log in kn dah masuk sungguh bantu jelek aplikasi

### 3.4 Split data

Dataset pada penelitian ini dipisahkan ke dalam dua kelompok utama, yaitu data pelatihan dan data pengujian, dengan komposisi sekitar 80% sebagai data latih dan 20% sebagai data uji. Melalui proses pemisahan tersebut, diperoleh 2.470 data untuk melatih model dan 617 data lainnya untuk menguji kinerjanya. Kedua subset ini kemudian dimanfaatkan dalam tahap pembangunan model serta penilaian

performa untuk memastikan akurasi analisis sentimen yang dihasilkan.

### 3.5 Pembobotan Data

Setiap kata dalam dataset diubah menjadi representasi numerik menggunakan metode TF-IDF (Term Frequency–Inverse Document Frequency). Metode ini memberikan bobot pada setiap kata berdasarkan tingkat relevansinya dalam suatu dokumen. Pada hasil perhitungan TF-IDF, setiap kolom merepresentasikan suatu istilah (term) seperti akun, aplikasi, bagus, dan lainnya, sedangkan setiap baris menunjukkan nilai TF-IDF dari istilah-istilah tersebut pada masing-masing dokumen. Nilai TF-IDF yang lebih tinggi menunjukkan bahwa kata tersebut memiliki kontribusi yang lebih signifikan terhadap dokumen terkait. Tahap ini menghasilkan sebuah matriks representasi numerik yang menjadi dasar untuk proses analisis sentimen berikutnya. Ringkasan hasil perhitungan TF-IDF ditunjukkan pada Tabel 3.

Tabel 3. Hasil TF-IDF

Term	Total TF	IDF
akun	121.321252	2.255600
aplikasi	98.378243	2.607820
tangguh	92.050031	2.739334
login	63.447100	3.142497
gak	57.995015	3.188361
bagus	57.221463	3.712512
akun tangguh	56.630258	3.437100
masuk	51.594402	3.529174
nya	43.059692	3.423306
susah	40.977225	3.835644

### 3.6 Penanganan Imbalance

Karena dataset memiliki distribusi kelas yang tidak seimbang, dilakukan penanganan imbalance untuk meningkatkan kinerja model. Pada tahap ini, metode SMOTE diterapkan pada algoritma Naïve Bayes, sedangkan pendekatan class\_weight digunakan pada Support Vector Machine (SVM). SMOTE (Synthetic Minority Over-sampling Technique) bekerja dengan menghasilkan sampel sintesis pada kelas minoritas sehingga proporsi kedua kelas menjadi lebih seimbang. Dengan demikian, model dapat mempelajari pola pada kelas minoritas secara lebih optimal dan menghasilkan performa klasifikasi yang lebih baik. Sebagai gambaran, distribusi data latih sebelum penerapan SMOTE adalah kelas 0: 2.394 dan kelas 1: 76, sedangkan setelah SMOTE berubah menjadi kelas 0: 2.394 dan kelas 1: 2.394.

Sementara itu, penerapan class\_weight = 'balanced' pada algoritma SVM memberikan bobot lebih tinggi pada kelas minoritas selama proses pelatihan. Berbeda dengan SMOTE, teknik ini tidak mengubah jumlah sampel asli karena class\_weight hanya menyesuaikan kontribusi setiap data terhadap fungsi kerugian (loss function). Dengan pendekatan ini, model menjadi lebih peka terhadap kesalahan pada kelas minoritas sehingga kecenderungan bias

terhadap kelas mayoritas dapat diminimalkan, namun struktur asli dataset tetap terjaga.

### 3.7 Hyperparameter Tuning Support Vector Machine

Pada tahap ini, dilakukan Hyperparameter Tuning terhadap algoritma Support Vector Machine (SVM) setelah sebelumnya model menggunakan pengaturan `class_weight = 'balanced'`. Proses ini bertujuan menemukan kombinasi parameter yang dapat memaksimalkan kinerja klasifikasi, sehingga model mampu membedakan kelas positif dan negatif secara lebih akurat, termasuk pada kelas yang kurang terwakili (minoritas). SVM yang diterapkan menggunakan kernel RBF (Radial Basis Function), yang efektif dalam menangani pola non-linear dengan memproyeksikan data ke ruang fitur berdimensi tinggi. Parameter `C` berfungsi mengatur keseimbangan antara lebar margin dan tingkat kesalahan klasifikasi pada data pelatihan, sedangkan `gamma` menentukan sejauh mana pengaruh setiap sampel terhadap pembentukan decision boundary. Dengan demikian, penyesuaian kernel RBF, `C`, dan `gamma` menjadi aspek kunci dalam meningkatkan kemampuan model untuk memisahkan kelas secara optimal.

Untuk memperoleh parameter terbaik, digunakan GridSearchCV dengan teknik cross-validation. Cross-validation membagi data latih ke dalam beberapa subset untuk menghasilkan evaluasi model yang lebih stabil serta meminimalkan potensi overfitting. Sementara itu, GridSearchCV melakukan pencarian sistematis terhadap kombinasi nilai `C` dan `gamma` yang telah ditentukan sebelumnya. Melalui pendekatan ini, diperoleh parameter optimal yaitu: `{'C': 10, 'class_weight': 'balanced', 'gamma': 0.01, 'kernel': 'rbf'}`, dengan nilai best F1-score sebesar 0,476 dari hasil cross-validation. Temuan ini menunjukkan bahwa proses Hyperparameter Tuning memungkinkan SVM menyesuaikan margin, pengaruh sampel, serta pembobotan kelas secara lebih efektif, sehingga menghasilkan prediksi yang lebih akurat dan lebih sensitif terhadap kelas minoritas—suatu aspek yang krusial pada dataset dengan distribusi yang tidak seimbang.

### 3.8 Training Model

Hasil pelatihan model menunjukkan bahwa algoritma Naïve Bayes, yang dikembangkan menggunakan data yang telah diseimbangkan melalui SMOTE, mencapai tingkat akurasi sebesar 85,90%. Sebaliknya, model Support Vector Machine (SVM) dengan konfigurasi parameter optimal hasil Hyperparameter Tuning—menggunakan kernel RBF, `C=10`, `gamma=0,01`, dan `class_weight='balanced'`—menunjukkan performa yang lebih unggul dengan akurasi mencapai 97,57%. Perbedaan ini menegaskan bahwa SVM memiliki kemampuan klasifikasi yang lebih tinggi dibandingkan Naïve Bayes dalam mengelompokkan sentimen pada dataset yang digunakan dalam penelitian ini.

Kinerja SVM yang lebih baik dipengaruhi oleh beberapa aspek. Pertama, kernel RBF memungkinkan SVM mengenali pola non-linear dalam data teks, sedangkan Naïve Bayes bergantung pada asumsi independensi antar fitur yang sering tidak terpenuhi pada data ulasan pengguna. Kedua, proses Hyperparameter Tuning melalui pengaturan `C` dan `gamma`, yang dikombinasikan dengan cross-validation dan GridSearchCV, memberikan kemampuan adaptasi yang lebih optimal bagi SVM dalam mengatur margin keputusan serta meningkatkan sensitivitas terhadap kesalahan pada kelas minoritas.

Dengan demikian, kombinasi kernel RBF, penyesuaian bobot kelas, dan optimasi parameter menjadikan SVM lebih responsif terhadap kompleksitas dan distribusi data yang tidak seimbang, sehingga menghasilkan akurasi yang lebih tinggi dibandingkan Naïve Bayes. Ringkasan hasil performa model ditampilkan pada Tabel 4.

Tabel 4. Hasil Accuracy

Model	Accuracy
Naïve Bayes	85.90%
Support Vector Machine	97.57%

Tabel 5. Hasil Klasifikasi Naïve Bayes

Naïve Bayes	precision	recall	f1-score	support
0	0.98	0.86	0.92	599
1	0.11	0.56	0.18	18

Tabel 6. Hasil Klasifikasi Support Vector Machine

Support Vector Machine	precision	recall	f1-score	support
0	0.99	0.99	0.99	599
1	0.79	0.61	0.69	18

Hasil pelatihan model yang ditampilkan pada Tabel 5 dan Tabel 6 menunjukkan bahwa algoritma Naïve Bayes menunjukkan performa yang baik pada kelas mayoritas (kelas 0), dengan nilai precision sebesar 0,98, recall 0,86, dan F1-score 0,92 dari 599 sampel. Hal ini mengindikasikan bahwa model mampu mengklasifikasikan sebagian besar sampel mayoritas secara akurat dan menangkap pola utama dengan efektif. Namun, pada kelas minoritas (kelas 1), kinerja Naïve Bayes masih terbatas, ditunjukkan oleh precision 0,11, recall 0,56, dan F1-score 0,18 dari 18 sampel, yang mengindikasikan kesulitan model dalam membedakan sampel minoritas, meskipun sebagian kecil berhasil diidentifikasi.

Sebaliknya, model Support Vector Machine (SVM) memperlihatkan kinerja yang lebih seimbang antara

kedua kelas. Pada kelas mayoritas, SVM mencapai precision 0,99, recall 0,99, dan F1-score 0,99, sedangkan pada kelas minoritas, diperoleh precision 0,58, recall 0,61, dan F1-score 0,59. Temuan ini menegaskan bahwa SVM lebih efektif dalam mengenali kelas minoritas tanpa mengurangi performa tinggi pada kelas mayoritas, sehingga menunjukkan kemampuan klasifikasi yang lebih stabil dan andal dibandingkan Naïve Bayes.

Perbandingan kedua model menunjukkan bahwa Naïve Bayes unggul dalam klasifikasi kelas mayoritas tetapi kurang optimal pada kelas minoritas. Sementara itu, SVM mampu menangani kedua kelas secara lebih seimbang, khususnya kelas minoritas, sehingga menghasilkan F1-score keseluruhan yang lebih tinggi. Meski demikian, SVM masih memiliki ruang untuk perbaikan pada kelas minoritas karena nilai precision dan recall pada kelas ini belum mencapai tingkat maksimum.

### 3.9 Prediksi

Hasil prediksi dari kedua model menunjukkan perbedaan kinerja yang cukup jelas. Naïve Bayes yang dilatih menggunakan data hasil SMOTE mencapai akurasi 84,92%. Model ini bekerja baik pada kelas mayoritas, yang ditunjukkan oleh precision 0,98 dan F1-score 0,92. Namun, performanya terhadap kelas minoritas masih lemah, dengan precision 0,11 dan F1-score 0,18. Artinya, meskipun sebagian sampel minoritas dapat terdeteksi, model masih sering salah dalam membedakannya dari kelas mayoritas.

Sebaliknya, Support Vector Machine (SVM) menunjukkan kinerja yang lebih superior dan konsisten. Melalui proses Hyperparameter Tuning menggunakan GridSearchCV yang dipadukan dengan cross-validation, model berhasil memperoleh parameter optimal (kernel RBF, C=10, gamma=0.01, class\_weight='balanced'). Optimalisasi ini memungkinkan SVM mencapai akurasi sebesar 98,37% sekaligus meningkatkan kemampuan dalam mengenali kelas minoritas, dengan precision 0,79 dan F1-score 0,69. Selain itu, SVM tetap mempertahankan performa yang sangat tinggi pada kelas mayoritas.

Secara keseluruhan, Naïve Bayes kuat pada kelas mayoritas tetapi kurang efektif menangani kelas minoritas meskipun sudah dibantu SMOTE. Sebaliknya, SVM dapat mengelola kedua kelas dengan lebih seimbang berkat penggunaan kernel non-linear dan optimasi parameter. Sehingga memungkinkan model menghasilkan prediksi dengan tingkat akurasi dan kestabilan yang lebih tinggi.

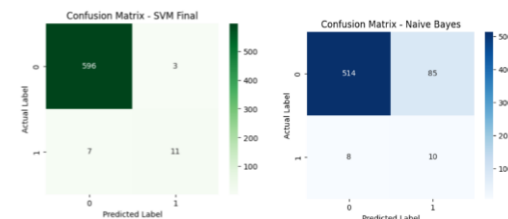
### 3.10 Evaluasi

Selanjutnya, dilakukan evaluasi untuk menilai performa model Naïve Bayes dan Support Vector Machine (SVM) dengan menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*.

Perbandingan kinerja kedua model tersebut disajikan pada Tabel 7

Tabel 7. Hasil Evaluasi Model

Model	Akurasi	Presisi	Recall	F1-score
Naïve Bayes	0.849	0.540	0.710	0.550
Support Vector Machine	0.984	0.890	0.800	0.840



Gambar 2. Confusion Matrix

Hasil evaluasi model yang disajikan pada Tabel 7 menunjukkan adanya perbedaan kinerja yang signifikan antara algoritma Naïve Bayes dan Support Vector Machine (SVM). Model Naïve Bayes mencapai akurasi sebesar 0,849 dengan *precision* 0,540, *recall* 0,710, dan F1-score 0,550. Kombinasi nilai tersebut mengindikasikan bahwa meskipun Naïve Bayes relatif efektif dalam mendeteksi sampel positif—ditunjukkan oleh *recall* yang cukup tinggi—model ini masih menghasilkan jumlah prediksi positif yang salah cukup besar, sehingga menurunkan nilai *precision* dan F1-score.

Sebaliknya, model SVM menunjukkan performa yang lebih tinggi dan konsisten, dengan akurasi 0,984, *precision* 0,890, *recall* 0,800, dan F1-score 0,840. Hasil ini mencerminkan kemampuan SVM dalam menyeimbangkan identifikasi antara kelas positif dan negatif, sehingga prediksi menjadi lebih stabil dan akurat. Analisis *confusion matrix* mendukung temuan ini, di mana Naïve Bayes menghasilkan 85 *false positive* dan 8 *false negative*, menunjukkan kecenderungan salah mengklasifikasikan sampel negatif sebagai positif. Sebaliknya, *confusion matrix* SVM hanya mencatat 3 *false positive* dan 7 *false negative*, menegaskan kemampuan model dalam menjaga tingkat kesalahan seminimal mungkin.

Secara keseluruhan, evaluasi metrik performa dan distribusi kesalahan pada *confusion matrix* memperlihatkan bahwa SVM memiliki kemampuan klasifikasi yang lebih baik dan andal dibandingkan Naïve Bayes pada dataset penelitian ini. Temuan ini menunjukkan bahwa SVM lebih cocok digunakan untuk analisis sentimen pada data serupa, dengan tingkat akurasi dan kestabilan yang lebih tinggi.

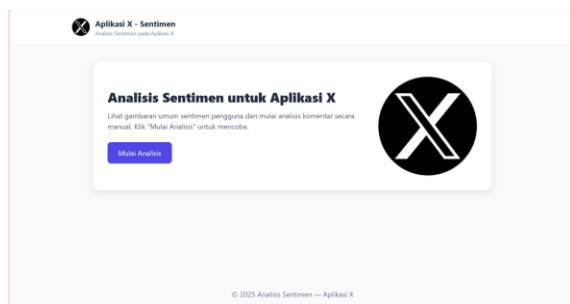


### 3.11 implementasi sistem berbasis web

Setelah model berhasil dibangun, tahap selanjutnya adalah mengintegrasikannya ke dalam aplikasi berbasis web. Bagian backend dikembangkan menggunakan framework Laravel untuk menangani pemrosesan data dan komunikasi dengan model, sedangkan bagian frontend dibuat dengan HTML, CSS, dan JavaScript untuk menyajikan antarmuka yang mudah digunakan. Integrasi antara frontend dan backend ini memungkinkan pengguna melakukan analisis sentimen secara praktis, karena setiap ulasan yang dimasukkan akan diproses otomatis oleh sistem dan hasilnya dapat ditampilkan secara cepat dan konsisten melalui aplikasi.

#### 3.11.1 Tampilan halaman Mulai Analisis

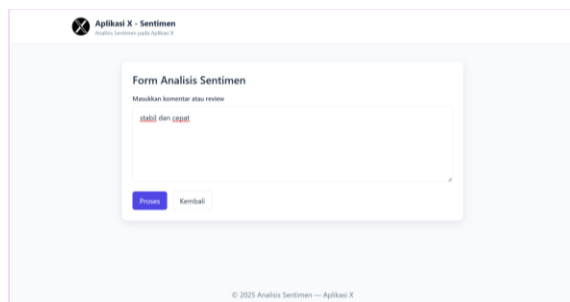
Halaman utama menampilkan ringkasan sistem analisis sentimen yang telah dibangun. Pada halaman ini terdapat menu navigasi menuju halaman analisis, halaman input data, serta halaman hasil pengolahan. Tampilan dashboard dirancang sederhana agar pengguna baru dapat memahami tujuan sistem, cara penggunaan, dan informasi ringkas hasil analisis.



Gambar 3. Tampilan Halaman Mulai Analisis

#### 3.11.2 Tampilan halaman form analisis

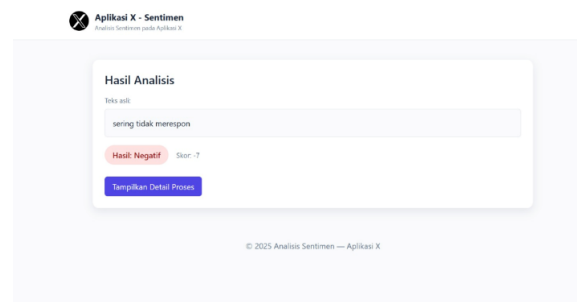
Halaman ini digunakan sebagai tempat pengguna memasukkan ulasan atau teks yang ingin dianalisis. Pengguna hanya perlu mengetikkan kalimat pada kotak input yang tersedia kemudian menekan tombol proses. Setelah tombol ditekan, sistem secara otomatis menjalankan tahapan preprocessing dan klasifikasi menggunakan model yang telah diimplementasikan.



Gambar 4. Tampilan Halaman Form Analisis

#### 3.11.3 Tampilan halaman hasil analisis

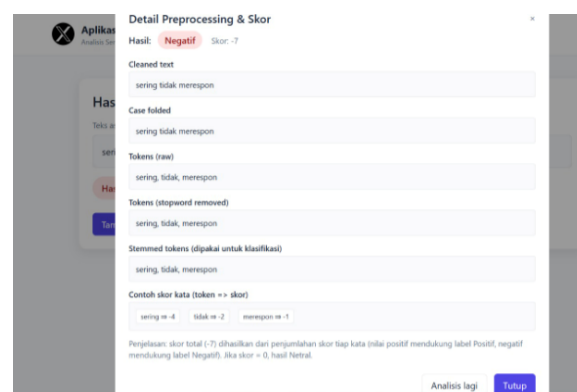
Setelah seluruh proses analisis selesai, sistem menampilkan halaman hasil klasifikasi yang memperlihatkan kategori sentimen dari teks yang dimasukkan, berupa positif dan negatif. Selain menampilkan label sentimen, halaman tersebut juga menyajikan skor probabilitas yang menggambarkan tingkat keyakinan model terhadap prediksi yang diberikan.



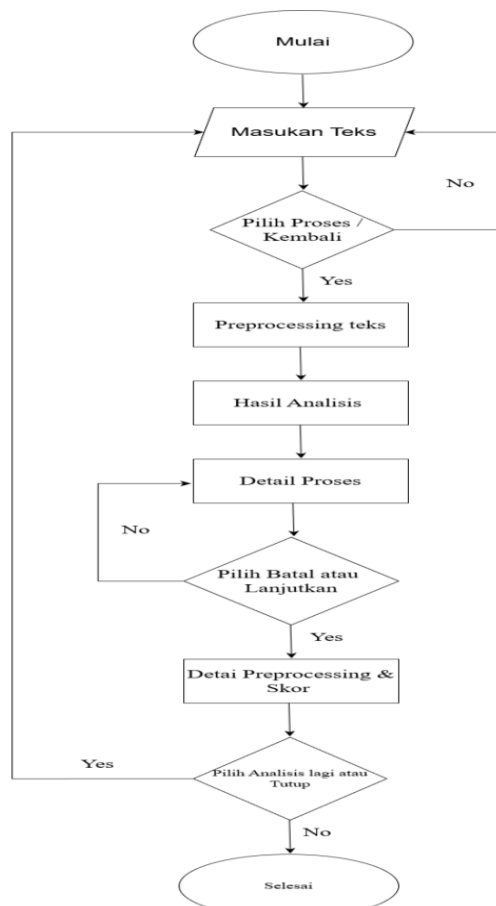
Gambar 5. Tampilan Halaman Hasil Analisis

#### 3.11.4 Tampilan hasil data processing dan skor

Halaman ini menampilkan ringkasan proses preprocessing data teks sebelum dianalisis oleh sistem. Pada halaman ini pengguna dapat melihat hasil tahapan pengolahan teks seperti pembersihan karakter, tokenisasi, penghapusan stopword, serta stemming. Hasil preprocessing disajikan dalam bentuk tabel agar pengguna dapat mengetahui perubahan teks dari bentuk awal hingga menjadi teks yang siap digunakan pada proses analisis sentimen selanjutnya.



Gambar 6. Tampilan Halaman Hasil Data Processing dan Skor



Gambar 7. Tampilan Flowchart Sistem

Alur flowchart tersebut menggambarkan mekanisme kerja sistem analisis sentimen yang diawali dari tahap Mulai, di mana pengguna memasukkan teks yang akan dianalisis. Setelah input diberikan, sistem menyediakan opsi bagi pengguna untuk melanjutkan proses atau kembali ke tahap sebelumnya. Jika proses diteruskan, sistem menjalankan tahap preprocessing sebagai bagian dari pembersihan dan penyiapan data. Tahap ini diikuti oleh pengolahan yang menghasilkan output analisis sentimen. Pengguna kemudian dapat meninjau detail proses beserta penjabaran hasil yang diperoleh, sebelum diberi pilihan untuk membatalkan atau melanjutkan. Apabila proses dilanjutkan, sistem menampilkan informasi lengkap mengenai tahapan preprocessing dan skor sentimen. Selanjutnya, pengguna diberikan kesempatan untuk melakukan analisis baru atau mengakhiri proses. Jika pengguna memilih untuk selesai, sistem akan menuju tahap akhir yang menandai berakhirnya seluruh rangkaian prosedur.

#### 4. Kesimpulan

Hasil evaluasi menunjukkan bahwa kedua model mampu melakukan klasifikasi sentimen secara memadai, namun Support Vector Machine (SVM) memperlihatkan performa yang lebih unggul, dengan tingkat akurasi mencapai 97,57 serta nilai precision, recall, dan F1-score yang lebih tinggi dibandingkan Naïve Bayes.

Keunggulan SVM ini disebabkan oleh kemampuannya dalam membentuk decision boundary optimal pada data berdimensi tinggi yang dihasilkan dari representasi TF IDF, sekaligus efektivitas model dalam menangani pola non-linear melalui penyesuaian hyperparameter. Di sisi lain, Naïve Bayes hanya mencapai akurasi 85,90% dan menunjukkan performa yang kurang stabil pada beberapa kelas. Hal ini disebabkan oleh ketergantungan metode pada asumsi independensi antar fitur, yang tidak sepenuhnya sesuai dengan karakteristik data teks. Kesalahan prediksi pada kedua model umumnya terjadi pada kelas dengan pola kata yang serupa atau konteks yang saling tumpang tindih, misalnya ulasan negatif yang disampaikan secara halus atau ulasan positif yang memuat keluhan, sehingga beberapa sampel salah diklasifikasikan dan menurunkan nilai recall. Secara keseluruhan, SVM terbukti memberikan klasifikasi yang lebih akurat dan konsisten di seluruh kelas, menjadikannya metode yang paling efektif untuk analisis sentimen pada ulasan Aplikasi X.

#### Daftar Rujukan

- [1] J. Liu *et al.*, "Traffic signal control using reinforcement learning based on the teacher-student framework," *Expert Syst. Appl.*, vol. 228, p. 120458, Oct. 2023, doi: 10.1016/J.ESWA.2023.120458.
- [2] K. Z. K. Zhang, S. J. Zhao, C. M. K. Cheung, and M. K. O. Lee, "Examining the influence of online reviews on consumers' decision-making: A heuristic-systematic model," *Decis. Support Syst.*, vol. 67, pp. 78–89, Nov. 2014, doi: 10.1016/J.DSS.2014.08.005.
- [3] P. Kumar, R. Bhatnagar, K. Gaur, and A. Bhatnagar, "Classification of Imbalanced Data: Review of Methods and Applications," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1099, no. 1, p. 012077, 2021, doi: 10.1088/1757-899x/1099/1/012077.
- [4] Syahril Dwi Prasetyo, Shofa Shofiah Hilabi, and Fitri Nurapriani, "Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN," *J. KomtekInfo*, vol. 10, pp. 1–7, 2023, doi: 10.35134/komtekinfo.v10i1.330.
- [5] S. K. I. Irma, A. M. Yasin, and A. S. Irvan, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM)," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 32–35, 2023.
- [6] M. I. Alfarizi, L. Syafaah, and M. Lestandy, "Emotional Text Classification Using TF-IDF (Term Frequency-Inverse Document Frequency) And LSTM (Long Short-Term Memory)," *JUITA J. Inform.*, vol. 10, no. 2, p. 225, 2022, doi: 10.30595/juita.v10i2.13262.
- [7] Gishella Septania Al-Husna, Dian Asmarajati, Iman Ahmad Ihsannuddin, and Rina Mahmudati, "Perbandingan Metode Naïve Bayes Dan Support Vector Machine Untuk Analisis Sentimen Pada Ulasan Pengguna Aplikasi Linkedln," *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 3, no. 2, pp. 139–144, 2024, doi: 10.55123/storage.v3i2.3602.
- [8] Y. Wang, W. Meijer, J. Antonio Hernández López, U. Nilsson, and D. Varró, "Why Do Machine Learning Notebooks Crash? An Empirical Study on Public Python Jupyter Notebooks," *IEEE Trans. Softw. Eng.*, vol. 51, no. 7, pp. 2181–2196, 2025, doi: 10.1109/TSE.2025.3574500.

- [9] R. T. Aldisa and P. Maulana, "Analisis Sentimen Opini Masyarakat Terhadap Vaksinasi Booster COVID-19 Dengan Perbandingan Metode Naive Bayes, Decision Tree dan SVM," *Build. Informatics, Technol. Sci.*, vol. 4, no. 1, pp. 106–109, 2022, doi: 10.47065/bits.v4i1.1581.
- [10] P. F. Supriyadi and Y. Sibaroni, "Xiaomi smartphone sentiment analysis on Twitter social media using IndoBERT," *JURIKOM (Jurnal Ris. Komputer)*, vol. 10, no. 1, pp. 19–30, 2023.
- [11] Z. Fitriyah and M. D. Kartikasari, "Text Classification of Twitter Opinion Related To Permendikbud 30/2021 Using Bidirectional Lstm," *Barekeng*, vol. 17, no. 2, pp. 1113–1122, 2023, doi: 10.30598/barekengvol17iss2pp1113-1122.
- [12] D. Ananda and R. R. Suryono, "Analisis Sentimen Publik Terhadap Pengungsi Rohingya di Indonesia dengan Metode Support Vector Machine dan Naive Bayes," *vol*, vol. 8, pp. 748–757, 2024.
- [13] P. Arsi and R. Waluyo, "Sentiment Analysis of Discourse on Moving the Indonesian Capital City Using the Support Vector Machine (SVM) Algorithm," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 1, p. 147, 2021, doi: 10.25126/jtiik.202183944.
- [14] T. A. Dewi and E. Mailoa, "Perbandingan Implementasi Metode Smote Pada Algoritma Support Vector Machine (Svm) Dalam Analisis Sentimen Opini Masyarakat Tentang Mixue," *J. Indones. Manaj. Inform. dan Komun.*, vol. 4, no. 3, pp. 849–855, 2023, doi: 10.35870/jimik.v4i3.289.
- [15] R. K. Dinata, N. Hasdyna, and R. Mahendra, "Kombinasi Algoritma Brute Force Dan Stemming," vol. 5, no. 2, pp. 273–278, 2020.
- [16] A. Putri, C. S. Hardiana, E. Novfuja, F. T. P. Siregar, Y. Fatma, and R. Wahyuni, "Comparison of K-NN, Naive Bayes and SVM Algorithms for Final-Year Student Graduation Prediction Komparasi Algoritma K-NN, Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir," *Inst. Ris. dan Publ. Indones. MALCOM Indones. J. Mach. Learn. Comput. Sci. J. Homepage*, vol. 3, no. 1, pp. 20–26, 2023.
- [17] R. Wati, S. Ernawati, and H. Rachmi, "TF-IDF Weighting Using Naive Bayes on Public Sentiment on The Issue of Rising BIPIH," *J. Manaj. Inform.*, vol. 13, no. April, pp. 84–93, 2023.
- [18] C. H. Yutika, A. Adiwijaya, and S. Al Faraby, "Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naive Bayes," *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 422, 2021, doi: 10.30865/mib.v5i2.2845.
- [19] S. Wang, Y. Dai, J. Shen, and J. Xuan, "Research on expansion and classification of imbalanced data based on SMOTE algorithm," *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, 2021, doi: 10.1038/s41598-021-03430-5.
- [20] P. For, "T Heory and E Valuation M Etrics," *Iclr*, vol. 55, no. 1, pp. 1–14, 2021.
- [21] F. S. Gomiasti, W. Warto, E. Kartikadarma, J. Gondohanindijo, and D. R. I. M. Setiadi, "Enhancing Lung Cancer Classification Effectiveness Through Hyperparameter-Tuned Support Vector Machine," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 396–406, 2024, doi: 10.62411/jcta.10106.
- [22] Iis Dewi Ratih, S M Retnaningsih, and V M Dewi, "Klasifikasi Kualitas Tanah Menggunakan Metode Naive Bayes Classifier," *J. Apl. Mat. dan Stat.*, vol. 1, no. 1, pp. 11–20, 2022, doi: 10.53625/jams.v1i1.4427.
- [23] A. Putri *et al.*, "Komparasi Algoritma K-NN, Naive Bayes dan SVM untuk Prediksi Kelulusan Mahasiswa Tingkat Akhir," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 1, pp. 20–26, 2023, doi: 10.57152/malcom.v3i1.610.
- [24] A. A. Munandar, F. Farikhin, and C. E. Widodo, "Sentimen Analisis Aplikasi Belajar Online Menggunakan Klasifikasi SVM," *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 8, no. 2, p. 77, 2023, doi: 10.31328/jointecs.v8i2.4747.