



Sentiment Analysis of Jamsostek Mobile Application Reviews on Google Play Store Using Support Vector Machine and Naive Bayes Algorithms

Analisis Sentimen Pengguna Aplikasi Jamsostek Mobile Berdasarkan Ulasan Google Play Store Menggunakan Algoritma Support Vector Machine dan Naive Bayes

Tria Setyani^{1*}, Kevinda Sari², Helma Nopijani Heidy³, Ryan Randy Suryono⁴

^{1,2,3,4}Magister Ilmu Komputer, Fakultas Teknik dan Ilmu Komputer,
Universitas Teknokrat Indonesia, Indonesia

E-mail : ¹tria_setyani@teknokrat.ac.id , ²kevinda_sari@teknokrat.ac.id ,
³helma_nopijani_heidy@teknokrat.ac.id, ⁴ryan@teknokrat.ac.id

Received Nov 16th 2025; Revised Dec 12th 2025; Accepted Jan 13th 2026; Available Online Jan 31th 2026
Corresponding Author: Ryan Randy Suryono
Copyright ©2026 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

Abstract

This study aims to analyze user sentiment toward the Jamsostek Mobile (JMO) application using Support Vector Machine (SVM) and Naive Bayes algorithms, based on reviews collected from the Google Play Store. A total of 6,000 user reviews were collected via web scraping. The research stages include text preprocessing (cleaning, case folding, normalization, tokenization, stopword removal, and stemming), feature weighting using Term Frequency–Inverse Document Frequency (TF-IDF), data labeling using a lexicon-based approach, and sentiment classification into three classes: positive, negative, and neutral. Model performance was evaluated using a confusion matrix and metrics such as accuracy, precision, recall, and F1-score. The results show that the SVM algorithm achieved superior performance with an accuracy of 88.9%, while Naive Bayes achieved 64.1%. SVM also demonstrated more consistent F1-score values across all sentiment classes. Therefore, SVM has been shown to be a more effective and reliable algorithm for sentiment classification of JMO application reviews.

Keywords: Google Play Store, Jamsostek Mobile, Naive Bayes, Sentiment Analysis, Support Vector Machine

Abstrak

Penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna aplikasi Jamsostek Mobile (JMO) yang tersedia pada Google Play Store menggunakan algoritma Support Vector Machine (SVM) dan Naive Bayes. Data yang digunakan sebanyak 6.000 ulasan pengguna yang dikumpulkan melalui teknik web scraping. Tahapan penelitian meliputi preprocessing teks (cleaning, case folding, normalisasi, tokenizing, stopword removal, dan stemming), pembobotan fitur menggunakan Term Frequency–Inverse Document Frequency (TF-IDF), pelabelan data dengan metode lexicon-based, serta klasifikasi sentimen ke dalam tiga kelas, yaitu positif, negatif, dan netral. Evaluasi performa model dilakukan menggunakan confusion matrix dan metrik akurasi, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa algoritma SVM menghasilkan performa yang lebih unggul dengan nilai akurasi sebesar 88,9%, sedangkan Naive Bayes memperoleh akurasi sebesar 64,1%. SVM juga menunjukkan nilai F1-score yang lebih konsisten pada seluruh kelas sentimen dibandingkan Naive Bayes. Dengan demikian, algoritma SVM terbukti lebih efektif dan andal dalam mengklasifikasikan sentimen ulasan pengguna aplikasi JMO.

Kata Kunci: Analisis Sentimen, Google Play Store, Jamsostek Mobile, Naive Bayes, Support Vector Machine

1. PENDAHULUAN

Perkembangan teknologi digital telah mendorong perubahan signifikan dalam penyelenggaraan layanan publik di Indonesia, khususnya melalui pemanfaatan aplikasi mobile yang bertujuan meningkatkan efisiensi, transparansi, serta kemudahan akses bagi masyarakat. Salah satu bentuk implementasi digitalisasi layanan publik tersebut adalah pengembangan aplikasi Jamsostek Mobile (JMO) oleh BPJS Ketenagakerjaan



sebagai bagian dari inovasi pelayanan berbasis e-government. Aplikasi JMO dirancang untuk memfasilitasi peserta dalam mengakses informasi kepesertaan, pengecekan saldo, simulasi manfaat Jaminan Hari Tua (JHT), hingga pengajuan klaim secara daring.

Berdasarkan data Google Play Store per Februari 2025, aplikasi JMO telah diunduh lebih dari 10 juta kali dan memperoleh rating sebesar 4,8, yang menunjukkan tingkat adopsi serta kepuasan pengguna yang relatif tinggi. Temuan ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa ulasan pada Google Play Store dapat menjadi indikator penting dalam mengukur kualitas layanan aplikasi melalui pendekatan analisis sentimen [1], [2]. Penelitian terkait analisis sentimen pada aplikasi JMO menggunakan metode Support Vector Machine (SVM) juga menunjukkan bahwa sebagian besar ulasan pengguna berada pada kategori positif dengan menunjukkan tingkat akurasi yang tinggi. Meskipun demikian, keberadaan rating tinggi dan dominasi sentimen positif tidak sepenuhnya mencerminkan ketiadaan permasalahan. Hasil observasi langsung terhadap ulasan pengguna masih menunjukkan adanya keluhan terkait kesulitan login, error aplikasi, keterlambatan sistem, kegagalan pembaruan data, serta kendala pada fitur klaim.

Fenomena tersebut menunjukkan adanya kemungkinan ketidaksesuaian antara rating numerik dan isi teks ulasan, sehingga diperlukan pendekatan analisis sentimen berbasis teks untuk memperoleh gambaran yang lebih komprehensif mengenai kualitas layanan aplikasi JMO. Kondisi ini menunjukkan bahwa tingginya tingkat adopsi aplikasi tidak selalu sejalan dengan kualitas layanan yang dirasakan pengguna. Fenomena lain yang menarik adalah adanya ketidaksesuaian antara rating aplikasi dan isi ulasan, di mana sebagian pengguna memberikan rating tinggi namun menyampaikan komentar bernada negatif, sehingga menimbulkan potensi bias apabila evaluasi kualitas aplikasi hanya didasarkan pada rating tanpa memperhatikan isi opini pengguna secara mendalam [3]. Untuk menangani hal tersebut, penelitian ini tidak hanya menggunakan rating sebagai dasar pelabelan, tetapi juga melakukan proses validasi teks melalui tahapan preprocessing dan analisis manual terbatas pada sebagian sampel data. Selain itu, pelabelan menggunakan pendekatan lexicon-based sebelum proses klasifikasi supervised learning untuk meminimalkan bias akibat ketidaksesuaian rating.

Ulasan pengguna pada platform Google Play Store menjadi sumber data yang sangat penting karena merepresentasikan pengalaman nyata pengguna dalam menggunakan aplikasi, sekaligus dapat dimanfaatkan sebagai bahan evaluasi dan pengambilan keputusan bagi pengembang layanan publik digital. Untuk menggali informasi dari data ulasan yang bersifat tidak terstruktur tersebut, berbagai penelitian telah menerapkan pendekatan analisis sentimen berbasis *text mining* dengan tahapan pengumpulan data melalui *web scraping* atau crawling, preprocessing teks (cleaning, case folding, tokenization, stopword removal, dan stemming), serta pembobotan kata menggunakan Term Frequency–Inverse Document Frequency (TF-IDF). Proses klasifikasi sentimen selanjutnya dilakukan menggunakan algoritma *machine learning* seperti Naive Bayes, Logistic Regression, SVM, dan Random Forest, dengan evaluasi performa model melalui *confusion matrix* serta metrik akurasi, *precision*, *recall*, dan *F1-score* [4].

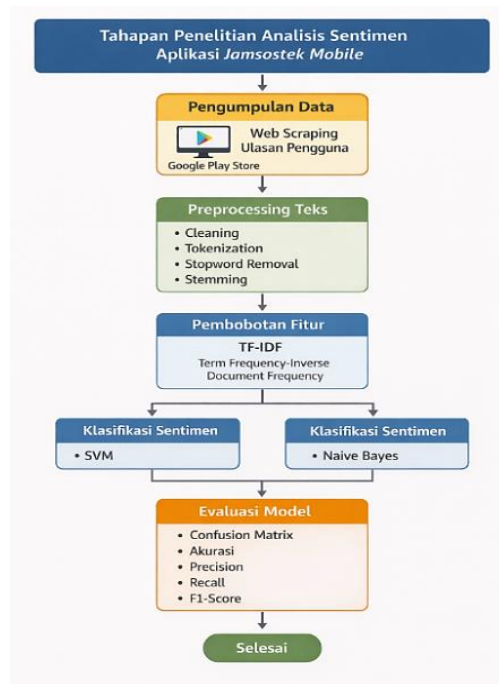
Sejumlah penelitian terdahulu telah mengkaji analisis sentimen terhadap ulasan aplikasi berbasis Google Play Store menggunakan berbagai algoritma klasifikasi *machine learning*. Penelitian sebelumnya menerapkan metode Naive Bayes untuk menganalisis sentimen ulasan aplikasi JMO dan menunjukkan bahwa metode tersebut mampu mengklasifikasikan sentimen pengguna dengan tingkat akurasi yang cukup baik [5], [6]. Penelitian lain telah membandingkan beberapa algoritma seperti SVM, Random Forest, dan Logistic Regression dan menemukan bahwa SVM memiliki performa yang lebih stabil dibandingkan metode lainnya. Penelitian lain juga menunjukkan bahwa kombinasi preprocessing teks dan pembobotan TF-IDF berpengaruh terhadap peningkatan performa model klasifikasi sentimen [7]. Namun, sebagian besar penelitian sebelumnya masih terbatas pada penggunaan satu algoritma atau belum mengevaluasi secara mendalam konsistensi performa antar kelas sentimen, khususnya pada kelas netral yang cenderung ambigu.

Berdasarkan keterbatasan tersebut, terdapat celah penelitian (*research gap*) berupa perlunya analisis sentimen ulasan aplikasi JMO yang dilakukan secara lebih sistematis dengan membandingkan kinerja algoritma klasifikasi yang berbeda serta mengevaluasi performa model menggunakan metrik yang lebih komprehensif. Oleh karena itu, penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna aplikasi JMO secara menyeluruh dengan membandingkan algoritma SVM dan Naive Bayes, sehingga diharapkan dapat memberikan kontribusi akademik dalam pengembangan studi analisis sentimen serta menjadi bahan evaluasi dan rekomendasi bagi BPJS Ketenagakerjaan dalam upaya peningkatan kualitas layanan aplikasi JMO.

2. BAHAN DAN METODE

Penelitian ini menggunakan pendekatan eksperimen komparatif dengan tujuan menganalisis dan mengevaluasi kinerja algoritma *machine learning* dalam mengklasifikasikan sentimen ulasan pengguna aplikasi Jamsostek Mobile (JMO). Pendekatan eksperimen digunakan karena penelitian ini menguji performa dua algoritma klasifikasi, yaitu Support Vector Machine (SVM) dan Naive Bayes, berdasarkan hasil evaluasi statistik untuk menentukan model yang memiliki kinerja terbaik dalam analisis sentimen. Data yang digunakan merupakan data sekunder berupa ulasan pengguna yang diperoleh dari platform Google Play

Store. Metode yang diterapkan dalam penelitian ini mencakup tahapan pengumpulan data, *preprocessing* teks, pembobotan fitur, klasifikasi sentimen, dan evaluasi model. Alur tahapan penelitian ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian Analisis Sentimen Aplikasi JMO

2.1. Pengumpulan Data

Proses pengambilan data dilakukan melalui teknik *web scraping* pada Google Play Store pada bulan Agustus tahun 2025, dengan mengambil ulasan yang tersedia pada rentang waktu tersebut. Berdasarkan proses tersebut, diperoleh sebanyak 6.000 data ulasan yang terdiri dari teks komentar pengguna, rating aplikasi, serta informasi waktu ulasan. Data hasil *scraping* ini kemudian disimpan dalam format terstruktur untuk memudahkan proses pengolahan pada tahap selanjutnya. Data tidak langsung diperoleh dalam kondisi siap analisis, melainkan melalui tahapan seleksi dan pembersihan data (*data cleaning*) untuk menghapus duplikasi dan data yang tidak relevan. Terkait distribusi data, dataset awal bersifat tidak seimbang apabila dilihat dari distribusi rating yang tersedia, di mana rating tinggi cenderung lebih dominan. Oleh karena itu, penelitian ini tidak menggunakan rating sebagai satu-satunya dasar pelabelan, tetapi melakukan analisis berbasis teks untuk mengklasifikasikan sentimen secara lebih representative [8].

2.2. Preprocessing Data

Tahap *preprocessing* data dilakukan untuk meningkatkan kualitas data teks sebelum dilakukan pembobotan fitur dan klasifikasi sentimen. Data ulasan pengguna yang diperoleh dari Google Play Store umumnya bersifat tidak terstruktur, mengandung kata tidak baku, simbol, serta variasi penulisan yang dapat menurunkan kinerja algoritma klasifikasi. Oleh karena itu, tahap *preprocessing* diperlukan untuk mengurangi *noise* dan menyeragamkan format data [9].

Proses *preprocessing* dalam penelitian ini meliputi *data cleaning*, *case folding*, *tokenization*, *stopword removal*, dan *stemming*. Tahap *data cleaning* bertujuan menghapus karakter khusus, tanda baca, angka, URL, serta data duplikat yang tidak relevan. Selanjutnya, *case folding* dilakukan dengan mengubah seluruh teks menjadi huruf kecil untuk menyeragamkan bentuk kata. Pendekatan ini bertujuan meningkatkan konsistensi data teks sebelum dilakukan proses klasifikasi [10].

Tahap *tokenization* dilakukan dengan memecah kalimat ulasan menjadi unit kata, kemudian dilanjutkan dengan *stopword removal* untuk menghilangkan kata-kata umum yang tidak memiliki makna signifikan terhadap sentimen. Tahap akhir *preprocessing* adalah *stemming*, yaitu proses mengembalikan kata berimbuhan ke bentuk kata dasar agar kata-kata yang memiliki makna serupa dapat diperlakukan sebagai satu representasi fitur. Kombinasi tahapan *preprocessing* tersebut terbukti mampu meningkatkan efektivitas analisis sentimen berbasis *machine learning* [11].

2.3. Pembobotan Fitur

Tahap selanjutnya adalah *transformation*, yaitu proses pembobotan kata menggunakan metode Term Frequency–Inverse Document Frequency (TF-IDF) untuk mengekstraksi fitur dan mengubah teks ulasan menjadi representasi numerik. Metode TF-IDF digunakan untuk mengukur tingkat kepentingan suatu kata dalam dokumen dengan mempertimbangkan frekuensi kemunculan kata serta tingkat kelangkaannya dalam keseluruhan dataset, sehingga menghasilkan representasi fitur yang lebih informatif dan efektif untuk digunakan pada proses klasifikasi sentimen [12].

2.4. Klasifikasi Sentimen

Klasifikasi sentimen merupakan proses pengelompokan data teks ke dalam kategori sentimen tertentu, seperti positif, negatif, dan netral, berdasarkan pola dan karakteristik kata yang terkandung dalam teks. Pada penelitian ini, proses klasifikasi sentimen dilakukan menggunakan dua algoritma *supervised learning*, yaitu SVM dan Naive Bayes, yang dipilih karena kemampuannya dalam menangani data teks berdimensi tinggi serta sering digunakan pada penelitian analisis sentimen berbasis *machine learning*.

2.4.1 Support Vector Machine (SVM)

SVM merupakan algoritma *supervised learning* yang banyak digunakan dalam analisis sentimen karena kemampuannya dalam menangani data berdimensi tinggi dan memisahkan kelas secara optimal menggunakan *hyperplane*. Dalam penelitian analisis sentimen berbasis teks, SVM dinilai efektif ketika dikombinasikan dengan pembobotan fitur TF-IDF karena mampu memaksimalkan jarak antar kelas sentimen positif dan negatif [13].

Secara umum, tujuan utama SVM adalah mencari *hyperplane* optimal dengan margin maksimum yang dapat memisahkan data ke dalam kelas-kelas tertentu. Permasalahan optimasi pada SVM dapat dirumuskan pada persamaan 1-2.

$$\min_{\mathbf{w}, b, \varepsilon} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \varepsilon_i \quad (1)$$

dengan kendala:

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0 \quad (2)$$

di mana \mathbf{w} dan b merepresentasikan parameter model, sedangkan ε_i digunakan untuk mengakomodasi kesalahan klasifikasi. Persamaan tersebut menunjukkan bahwa SVM berupaya memaksimalkan margin pemisah antar kelas dengan tetap meminimalkan kesalahan klasifikasi.

Penerapan SVM pada analisis sentimen ulasan aplikasi menunjukkan kinerja yang stabil dan konsisten, khususnya pada data teks yang bersifat tidak terstruktur. Hasil penelitian sebelumnya menunjukkan bahwa algoritma SVM mampu memberikan nilai akurasi, *precision*, *recall*, dan *F1-score* yang relatif lebih baik dibandingkan metode probabilistik pada beberapa kasus analisis sentimen berbasis Google Play Store [2], [8]. Oleh karena itu, dalam penelitian ini algoritma SVM digunakan sebagai salah satu metode klasifikasi untuk mengidentifikasi sentimen positif, negatif, dan netral pada ulasan pengguna aplikasi .

2.4.2 Naive Bayes

Naive Bayes merupakan algoritma klasifikasi berbasis probabilistik yang bekerja berdasarkan Teorema Bayes dengan asumsi independensi antar fitur. Algoritma ini banyak digunakan dalam analisis sentimen karena memiliki proses komputasi yang sederhana, cepat, dan efisien, terutama pada dataset teks berukuran besar. Dalam konteks analisis sentimen, Naive Bayes mampu menghitung probabilitas kemunculan kata pada masing-masing kelas sentimen untuk menentukan kategori sentimen suatu dokumen [14], [15]. Perhitungan probabilitas posterior suatu kelas terhadap dokumen dirumuskan pada persamaan 3-4.

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} \quad (3)$$

Karena $P(X)$ bersifat konstan, maka proses klasifikasi dapat disederhanakan menjadi:

$$C_k = \arg \max_{C_k} P(X|C_k) P(C_k) \quad (4)$$

Penerapan algoritma Naive Bayes pada data ulasan pengguna aplikasi menunjukkan performa yang cukup baik dengan waktu pemrosesan yang lebih cepat dibandingkan algoritma lain. Beberapa penelitian menunjukkan bahwa Naive Bayes mampu menghasilkan tingkat akurasi yang kompetitif, khususnya ketika dikombinasikan dengan preprocessing teks yang optimal dan pembobotan fitur TF-IDF [16]. Oleh karena itu,

algoritma Naive Bayes digunakan dalam penelitian ini sebagai metode pembandingan terhadap SVM untuk menentukan model klasifikasi sentimen yang paling optimal.

2.5. Evaluasi Model

Evaluasi performa model klasifikasi dilakukan menggunakan *confusion matrix* yang terdiri atas True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). *Confusion matrix* digunakan untuk menggambarkan perbandingan antara hasil prediksi model dengan label aktual, sehingga dapat diketahui tingkat kesalahan dan keberhasilan model dalam melakukan klasifikasi sentimen.

Berdasarkan nilai-nilai tersebut, kinerja model dievaluasi menggunakan beberapa metrik, yaitu akurasi, *precision*, *recall*, dan *F1-score*, yang dirumuskan pada persamaan 5-8 [17].

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

Akurasi digunakan untuk mengukur tingkat ketepatan model secara keseluruhan dalam mengklasifikasikan data sentimen. *Precision* menunjukkan kemampuan model dalam memprediksi data sentimen secara benar dari seluruh data yang diprediksi ke dalam suatu kelas tertentu. *Recall* menggambarkan kemampuan model dalam menemukan kembali seluruh data yang benar-benar termasuk ke dalam kelas tertentu. Sementara itu, *F1-score* merupakan rata-rata harmonis antara *precision* dan *recall* yang digunakan untuk menyeimbangkan kedua metrik tersebut, terutama pada kondisi data yang tidak seimbang.

Evaluasi ini bertujuan untuk mengukur tingkat ketepatan dan keandalan model dalam mengklasifikasikan sentimen ulasan pengguna. Penggunaan metrik evaluasi tersebut umum diterapkan dalam penelitian analisis sentimen karena mampu memberikan gambaran performa model secara komprehensif dan objektif.

3. HASIL DAN PEMBAHASAN

3.1 Tahap Preprocessing Data

Tahap Preprocessing merupakan tahapan penting dalam penelitian ini yang bertujuan untuk membersihkan dan menyiapkan data teks ulasan pengguna aplikasi Jamsostek Mobile (JMO) agar siap dianalisis pada tahap selanjutnya. Data hasil crawling masih mengandung berbagai bentuk noise, seperti tanda baca, perbedaan huruf kapital, kata tidak baku, serta kata-kata umum yang tidak memiliki pengaruh terhadap analisis sentimen. Oleh karena itu, dilakukan beberapa tahapan Preprocessing yang meliputi cleaning, case folding, normalisasi data, Tokenization, stopword removal, dan stemming.

Sebelum dilakukan tahap Preprocessing, data yang digunakan dalam penelitian ini merupakan data mentah (*raw data*) hasil crawling ulasan pengguna aplikasi JMO dari Google Play Store. Data mentah tersebut masih mengandung berbagai atribut non-teks dan variasi penulisan, seperti tautan ulasan, nama pengguna, tanggal ulasan, isi komentar, jumlah pengguna yang menilai ulasan bermanfaat, serta respons resmi dari pihak BPJS Ketenagakerjaan. Contoh data mentah hasil crawling sebelum dilakukan Preprocessing ditunjukkan pada Tabel 1.

Berdasarkan Tabel 1 dapat diketahui bahwa data mentah hasil crawling masih mengandung berbagai elemen yang tidak diperlukan dalam analisis sentimen, seperti URL, metadata pengguna, serta respons resmi instansi. Selain itu, teks ulasan juga masih mengandung tanda baca, perbedaan huruf kapital, dan kata tidak baku. Oleh karena itu, tahap awal Preprocessing yang dilakukan dalam penelitian ini adalah proses cleaning, yang bertujuan untuk membersihkan teks ulasan dari unsur-unsur tersebut.

Table 1. Contoh Data Mentah Hasil Crawling Ulasan Aplikasi JMO

No	URL Ulasan	Nama Pengguna	Tanggal Ulasan	Isi Ulasan	Jumlah Penilai Bermanfaat	Sumber	Tanggal Respons	Respons Resmi
1	https://play.google.com/...	Rasyid Firdaus	19 Agustus 2025	Ini gimana sih aplikasi, udah daftar tinggal b...	117 orang	BPJS Ketenagakerjaan Official	29 Agustus 2025	Yth. Bapak/Ibu pengguna aplikasi BPJS

No	URL Ulasan	Nama Pengguna	Tanggal Ulasan	Isi Ulasan	Jumlah Penilai Bermanfaat	Sumber	Tanggal Respons	Respons Resmi
2	https://play.google.com/...	Bagus Supriadi	26 Juni 2025	Ketika akun log out dan mau log in mesti berma...	438 orang	BPJS Ketenaga kerjaan Official	25 Juli 2025	Ketenaga... Yth. Bapak/Ibu pengguna aplikasi BPJS
3	https://play.google.com/...	Ruth Girsang	18 Juni 2025	Udah dari mei coba login ga bisa keterangannya...	344 orang	BPJS Ketenaga kerjaan Official	25 Juli 2025	Ketenaga... Yth. Bapak/Ibu pengguna aplikasi BPJS
4	https://play.google.com/...	LuluBisay Feliz	3 September 2025	Aplikasinya mudah digunakan, hanya saja mainte...	4 orang	-	-	-
5	https://play.google.com/...	Muhammad Alfi Erfani	3 September 2025	Aplikasi aneh, kita mau klaim JHT lewat portal...	15 orang	-	-	-
...	-	-	-
6000	https://play.google.com/...	Riyan Firman Syah	16 Juni 2025	klaim hak sendiri aja di persulit. pada ga becus bikin aplikasi yang mateng. buat apa jmo klo harus di alihkan ke lapak asik	-	-	-

3.1.1 Proses Cleaning

Proses *cleaning* dilakukan untuk menghilangkan karakter yang tidak diperlukan dalam teks ulasan, seperti tanda baca, angka, simbol, serta karakter khusus lainnya. Tahap ini bertujuan untuk mengurangi noise sehingga teks menjadi lebih bersih dan mudah diproses. Hasil *cleaning* menunjukkan bahwa teks ulasan menjadi lebih sederhana tanpa menghilangkan makna utama dari kalimat. Contoh hasil proses cleaning dapat dilihat pada Tabel 2.

Tabel 2. Contoh Hasil Proses Cleaning

No	Teks Asli (h3YV2d)	Hasil Cleaning
1	ini gimana sih aplikasi, udah daftar tinggal b...	ini gimana sih aplikasi udah daftar tinggal ba...
2	Ketika akun Log Out dan mau Log In mesti berma...	Ketika akun Log Out dan mau Log In mesti berma...
3	Udah dari mei coba login ga bisa. Keterangannya...	Udah dari mei coba login ga bisa Keterangannya...
4	Aplikasinya mudah digunakan. hanya saja mainte...	Aplikasinya mudah digunakan hanya saja mainte...
5	Aplikasi Aneh, kita mau Klaim JHT lewat Portal...	Aplikasi Aneh kita mau Klaim JHT lewat Portal...

3.1.2 Proses Case Folding dan Normalisasi

Tahap *case folding* dan normalisasi data merupakan bagian dari proses *preprocessing* teks yang bertujuan untuk meningkatkan kualitas data sebelum dilakukan analisis sentimen. Pada tahap *case folding*, seluruh teks ulasan pengguna diubah menjadi huruf kecil (*lowercase*). Proses ini dilakukan untuk menyeragamkan bentuk kata sehingga tidak terjadi perbedaan representasi akibat penggunaan huruf kapital, seperti pada kata “Login”, “LOGIN”, dan “login” yang secara makna sama namun berbeda secara penulisan.

Setelah proses *case folding* selesai, tahap selanjutnya adalah normalisasi data. Normalisasi dilakukan dengan mengubah kata tidak baku, singkatan, atau istilah informal yang sering digunakan dalam ulasan pengguna menjadi bentuk kata baku sesuai dengan kaidah bahasa Indonesia. Contoh normalisasi yang dilakukan antara lain perubahan kata “gimana” menjadi “bagaimana”, “udah” menjadi “sudah”, serta “ga”

menjadi “tidak”. Tahap ini penting untuk meningkatkan konsistensi data teks serta membantu algoritma klasifikasi dalam mengenali pola kata secara lebih akurat.

Penerapan *case folding* dan normalisasi data secara berurutan terbukti mampu mengurangi variasi kata yang tidak perlu dan memperbaiki kualitas representasi teks. Dengan data yang lebih bersih dan konsisten, proses selanjutnya seperti tokenisasi, pembobotan fitur menggunakan TF-IDF, serta klasifikasi sentimen dapat menghasilkan performa yang lebih optimal. Contoh hasil penerapan *case folding* dan normalisasi data ditunjukkan pada Tabel 3.

Tabel 3. Contoh Case Folding dan Normalisasi Data

No	Teks Asli (h3YV2d)	Case Folding	Normalisasi
1	ini gimana sih aplikasi, udah daftar tinggal b...	ini gimana sih aplikasi udah daftar tinggal ba...	ini bagaimana sih aplikasi sudah daftar tinggal ba...
2	Ketika akun Log Out dan mau Log In mesti berma...	ketika akun log out dan mau log in mesti berma...	ketika akun log out dan mau log in mesti berma...
3	Udah dari mei coba login ga bisa. Keteranganny...	udah dari mei coba login ga bisa keterangannya...	sudah dari mei coba login tidak bisa keterangannya...
4	Aplikasinya mudah digunakan. hanya saja mainte...	aplikasinya mudah digunakan hanya saja mainte...	aplikasinya mudah digunakan hanya saja mainte...
5	Aplikasi Aneh, kita mau Klaim JHT lewat Portal...	aplikasi aneh kita mau klaim jht lewat portal...	aplikasi aneh kita mau klaim jht lewat portal...

3.1.3 Tokenization, Stopword Removal, dan Stemming

Setelah proses *case folding* dan normalisasi data, tahap *preprocessing* dilanjutkan dengan tokenization, stopwords removal, dan stemming yang dilakukan secara berurutan. *Tokenization* bertujuan untuk memecah teks ulasan menjadi unit kata (token) agar memudahkan proses analisis lebih lanjut. Tahap ini menghasilkan daftar kata yang merepresentasikan isi ulasan pengguna secara terstruktur.

Selanjutnya, dilakukan *stopword removal* untuk menghapus kata-kata umum yang tidak memiliki pengaruh signifikan terhadap penentuan sentimen, seperti kata penghubung dan kata depan. Tahap terakhir adalah stemming, yaitu proses mengubah kata berimbuhan menjadi kata dasar untuk mengurangi variasi kata dan menyatukan kata-kata yang memiliki makna serupa. Melalui ketiga tahapan ini (lihat Tabel 4), data teks menjadi lebih ringkas, konsisten, dan siap digunakan pada tahap pembobotan fitur serta klasifikasi sentimen.

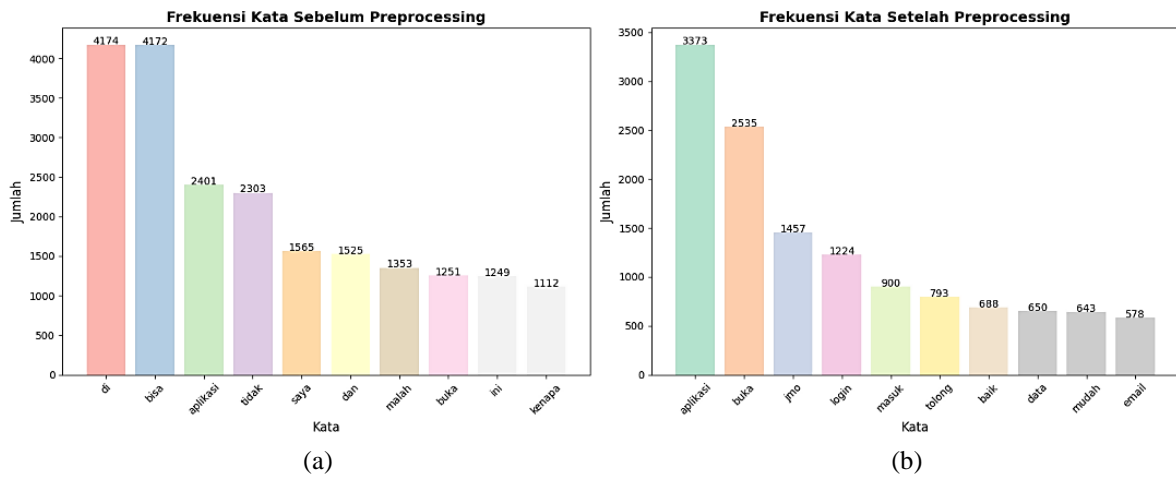
Tabel 4. Tokenization, Stopword Removal, dan Stemming

No	h3YV2d	Tokenize	Stopword Removal	Stemming Data
0	ini gimana sih aplikasi, udah daftar tinggal b...	[ini, bagaimana, sih, aplikasi,...]	[aplikasi, daftar, tinggal, bayar,...]	aplikasi daftar tinggal bayar
1	Ketika akun Log Out dan mau Log In mesti berma...	[ketika, akun, log, out,...]	[akun, log, out, mesti, bermasalah,...]	akun log out mesti masalah
2	Udah dari mei coba login ga bisa. Keteranganny...	[sudah, dari, mei, coba,...]	[mei, coba, login, keterangannya,...]	mei coba login terang
3	Aplikasinya mudah digunakan. hanya saja mainte...	[aplikasinya, mudah, digunakan,...]	[aplikasinya, mudah, maintenance,...]	aplikasi mudah maintenance
4	Aplikasi Aneh, kita mau Klaim JHT lewat Portal...	[aplikasi, aneh, kita, mau,...]	[aplikasi, aneh, klaim, jht,...]	aplikasi aneh klaim jht

3.2 Pembobotan Fitur Menggunakan TF-IDF

Pembobotan TF-IDF bertujuan untuk mengukur tingkat kepentingan suatu kata dalam sebuah dokumen dengan mempertimbangkan frekuensi kemunculan kata tersebut pada dokumen tertentu serta tingkat kelangkaannya pada keseluruhan dataset. Hasil pembobotan TF-IDF menghasilkan matriks fitur berdimensi tinggi, di mana setiap baris merepresentasikan satu ulasan pengguna dan setiap kolom merepresentasikan kata hasil preprocessing. Nilai numerik pada matriks menunjukkan bobot kepentingan suatu kata dalam ulasan tertentu. Sebagian besar nilai pada matriks bernilai nol, yang menunjukkan bahwa representasi TF-IDF bersifat *sparse*, yaitu hanya kata-kata tertentu yang memiliki kontribusi signifikan dalam merepresentasikan dokumen ulasan. Untuk mengidentifikasi fitur yang paling berpengaruh, dilakukan perhitungan rata-rata bobot TF-IDF pada seluruh data ulasan, dapat dilihat pada Table 5.

Hasil pada Tabel 5, yang menampilkan kata-kata dengan bobot TF-IDF tertinggi. Berdasarkan tabel tersebut, kata *aplikasi*, *update*, *buka*, *jmo*, dan *login* memiliki bobot yang relatif tinggi dibandingkan kata



Gambar 3. (a) Frekuensi Kata sebelum Preprocessing. (b) Frekuensi Kata setelah Preprocessing

3.5 Klasifikasi Sentimen

Tahap klasifikasi sentimen dilakukan untuk mengelompokkan ulasan pengguna aplikasi JMO ke dalam tiga kelas sentimen, yaitu positif, negatif, dan netral. Proses klasifikasi ini dilakukan menggunakan dua algoritma, yaitu SVM dan Naive Bayes, dengan tujuan untuk membandingkan kinerja kedua metode dalam mengklasifikasikan sentimen ulasan pengguna berdasarkan fitur teks yang telah diproses pada tahap sebelumnya.

3.5.1 Support Vector Machine (SVM)

Algoritma SVM digunakan dalam penelitian ini untuk mengklasifikasikan sentimen ulasan pengguna aplikasi JMO ke dalam tiga kelas, yaitu sentimen negatif, netral, dan positif. SVM dipilih karena kemampuannya dalam menangani data teks berdimensi tinggi serta membentuk batas pemisah (hyperplane) yang optimal antar kelas sentimen.

Tabel 7. Classification SVM

Kelas	Precision	Recall	F1-Score	Support
Negatif	0.948	0.918	0.933	561
Netral	0.789	0.846	0.816	331
Positif	0.900	0.883	0.891	307
Accuracy	0.889	0.889	0.889	0.889
Macro Avg	0.879	0.882	0.880	1199
Weighted Avg	0.892	0.889	0.890	1199

Hasil klasifikasi sentimen menggunakan algoritma SVM ditunjukkan pada Table 7, yang menyajikan *classification report* berdasarkan metrik *precision*, *recall*, dan *F1-score*. Berdasarkan Tabel 7, SVM menunjukkan performa yang sangat baik pada seluruh kelas sentimen. Pada kelas sentimen negatif, SVM memperoleh nilai *precision* sebesar 0,948, *recall* sebesar 0,918, dan *F1-score* sebesar 0,933, yang menunjukkan bahwa model sangat akurat dan konsisten dalam mengidentifikasi ulasan yang bersifat negatif. Untuk kelas sentimen netral, diperoleh nilai *precision* sebesar 0,789, *recall* 0,846, dan *F1-score* 0,816, yang mengindikasikan bahwa model mampu mengenali ulasan netral dengan cukup baik meskipun memiliki karakteristik yang cenderung ambigu. Sementara itu, pada kelas sentimen positif, SVM menghasilkan nilai *precision* 0,900, *recall* 0,883, dan *F1-score* 0,891, yang menunjukkan kinerja klasifikasi yang stabil dan seimbang. Secara keseluruhan, algoritma SVM menghasilkan nilai akurasi sebesar 0,889 (88,9%), sebagaimana ditunjukkan pada table 7. Nilai *F1-score* yang tinggi dan relatif konsisten pada setiap kelas sentimen menunjukkan bahwa SVM merupakan model yang andal dan efektif dalam mengklasifikasikan sentimen ulasan pengguna aplikasi JMO.

3.5.2 Naive Bayes

Algoritma Naive Bayes digunakan dalam penelitian ini sebagai metode pembanding terhadap SVM untuk mengklasifikasikan sentimen ulasan pengguna aplikasi JMO ke dalam tiga kelas, yaitu sentimen negatif, netral, dan positif. Naive Bayes merupakan algoritma klasifikasi berbasis probabilistik yang bekerja berdasarkan Teorema Bayes dengan asumsi independensi antar fitur, sehingga memiliki proses komputasi yang lebih sederhana dan cepat.

Hasil klasifikasi sentimen menggunakan algoritma Naive Bayes ditunjukkan pada Table 8, yang menyajikan *classification report* berdasarkan metrik *precision*, *recall*, dan *F1-score*. Berdasarkan Table 8, terlihat bahwa Naive Bayes memiliki kemampuan yang cukup baik dalam mendeteksi sentimen negatif, yang ditunjukkan oleh nilai *recall* sebesar 0,927. Hal ini mengindikasikan bahwa sebagian besar ulasan negatif berhasil teridentifikasi oleh model. Namun demikian, nilai *precision* pada kelas sentimen negatif sebesar 0,646 menunjukkan masih adanya kesalahan prediksi pada kelas tersebut. Pada kelas sentimen netral, performa Naive Bayes relatif rendah dengan nilai *precision* sebesar 0,461, *recall* sebesar 0,178, dan *F1-score* sebesar 0,257. Rendahnya nilai *recall* pada kelas ini menunjukkan bahwa model mengalami kesulitan dalam membedakan ulasan netral dengan ulasan yang memiliki kecenderungan sentimen negatif maupun positif. Sementara itu, pada kelas sentimen positif, Naive Bayes memperoleh nilai *precision* sebesar 0,711, *recall* 0,616, dan *F1-score* 0,660, yang menunjukkan performa klasifikasi yang cukup, namun belum optimal. Secara keseluruhan, algoritma Naive Bayes menghasilkan nilai akurasi sebesar 0,641 (64,1%), sebagaimana ditunjukkan pada table 8. Nilai akurasi dan *F1-score* yang lebih rendah dibandingkan SVM menunjukkan bahwa meskipun Naive Bayes unggul dalam mendeteksi sentimen negatif, model ini kurang efektif dalam mengklasifikasikan sentimen netral, sehingga berdampak pada performa klasifikasi secara keseluruhan.

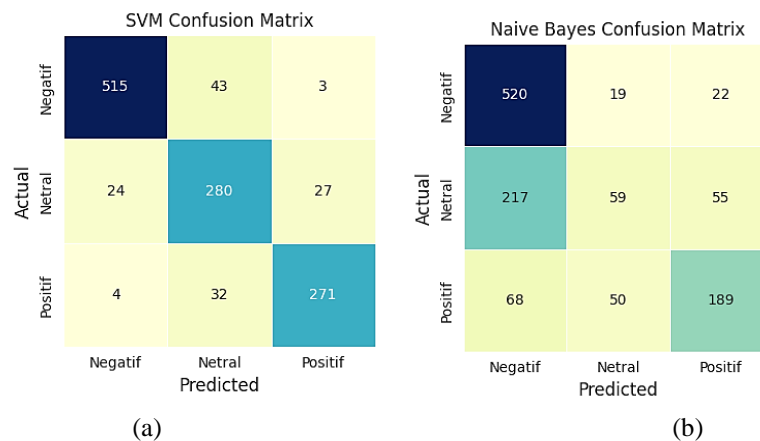
Tabel 8. *Classification* Naive Bayes

Kelas	Precision	Recall	F1-Score	Support
Negatif	0.646	0.927	0.761	561
Netral	0.461	0.178	0.257	331
Positif	0.711	0.616	0.660	307
Accuracy	0.641	0.641	0.641	0.641
Macro Avg	0.606	0.574	0.559	1199
Weighted Avg	0.611	0.641	0.596	1199

3.6 Evaluasi Model

3.6.1 Confusion matrix

Evaluasi performa model klasifikasi dilakukan menggunakan *confusion matrix*, yang berfungsi untuk menggambarkan perbandingan antara label sentimen aktual dengan hasil prediksi model. *Confusion matrix* memungkinkan analisis kesalahan klasifikasi secara lebih rinci, termasuk kesalahan prediksi antar kelas sentimen negatif, netral, dan positif.



Gambar 4. (a) *Confusion Matrix* SVM. (b) *Confusion Matrix* Naive Bayes

Hasil confusion matrix untuk algoritma SVM dan Naive Bayes ditunjukkan pada Gambar 6. Berdasarkan confusion matrix SVM pada Gambar 6(a), terlihat bahwa model SVM mampu mengklasifikasikan sebagian besar data dengan benar pada seluruh kelas sentimen. Pada kelas sentimen negatif, sebanyak 515 data berhasil diprediksi dengan tepat, dengan kesalahan klasifikasi yang relatif kecil, yaitu 43 data diprediksi sebagai netral dan 3 data sebagai positif. Pada kelas sentimen netral, sebanyak 280 data berhasil diklasifikasikan dengan benar, sedangkan kesalahan prediksi relatif seimbang ke kelas negatif (24 data) dan positif (27 data). Untuk kelas sentimen positif, SVM mampu mengklasifikasikan 271 data secara tepat, dengan jumlah kesalahan yang rendah.

Sebaliknya, confusion matrix Naive Bayes pada Gambar 6(b) menunjukkan pola kesalahan klasifikasi yang lebih besar, khususnya pada kelas sentimen netral. Meskipun Naive Bayes mampu mengklasifikasikan sentimen negatif dengan cukup baik, yaitu sebanyak 520 data terklasifikasi dengan benar, pada kelas sentimen netral hanya 59 data yang berhasil diprediksi dengan tepat. Sebagian besar data netral justru salah diklasifikasikan sebagai sentimen negatif (217 data) dan positif (55 data). Pada kelas sentimen positif, Naive

Bayes mengklasifikasikan 189 data dengan benar, namun masih terdapat kesalahan prediksi yang cukup signifikan ke kelas negatif dan netral.

Berdasarkan perbandingan confusion matrix tersebut, dapat disimpulkan bahwa model SVM memiliki tingkat kesalahan klasifikasi yang lebih rendah dan distribusi kesalahan yang lebih seimbang antar kelas sentimen dibandingkan dengan Naive Bayes. Sementara itu, Naive Bayes menunjukkan kecenderungan bias terhadap kelas sentimen negatif dan mengalami kesulitan dalam membedakan sentimen netral, yang berdampak pada menurunnya performa klasifikasi secara keseluruhan.

3.6.2 Evaluasi Menggunakan Akurasi, Precision, Recall, dan F1-score

Evaluasi performa model klasifikasi pada penelitian ini dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score* untuk memberikan gambaran kinerja model secara komprehensif. Nilai akurasi digunakan untuk mengukur tingkat ketepatan model secara keseluruhan dalam mengklasifikasikan data sentimen, sedangkan *precision* dan *recall* digunakan untuk mengevaluasi kemampuan model dalam memprediksi masing-masing kelas sentimen. *F1-score* digunakan sebagai ukuran keseimbangan antara *precision* dan *recall*, khususnya pada kondisi distribusi data yang tidak sepenuhnya seimbang.

Hasil evaluasi performa algoritma SVM disajikan pada Gambar 4. Berdasarkan gambar tersebut, SVM menghasilkan nilai akurasi sebesar 0,889, yang menunjukkan tingkat ketepatan klasifikasi yang tinggi. Pada kelas sentimen negatif, SVM memperoleh nilai *precision* 0,948, *recall* 0,918, dan *F1-score* 0,933, yang mengindikasikan kemampuan model yang sangat baik dalam mendeteksi ulasan negatif. Untuk kelas sentimen netral dan positif, nilai *F1-score* masing-masing sebesar 0,816 dan 0,891, yang menunjukkan bahwa performa SVM relatif konsisten pada seluruh kelas sentimen. Nilai *weighted average F1-score* sebesar 0,890 menegaskan semakin stabil kinerja model SVM.

Sementara itu, hasil evaluasi algoritma Naive Bayes ditunjukkan pada Gambar 5. Berdasarkan gambar tersebut, Naive Bayes menghasilkan nilai akurasi sebesar 0,641, yang lebih rendah dibandingkan dengan SVM. Meskipun Naive Bayes memiliki nilai *recall* yang tinggi pada kelas sentimen negatif (0,927), nilai *precision* pada kelas ini relatif lebih rendah (0,646). Performa terendah terlihat pada kelas sentimen netral, dengan nilai *precision* 0,461, *recall* 0,178, dan *F1-score* 0,257, yang menunjukkan bahwa model mengalami kesulitan dalam membedakan sentimen netral dari sentimen lainnya. Nilai *weighted average F1-score* sebesar 0,596 menunjukkan bahwa performa Naive Bayes secara keseluruhan masih kurang optimal.

Berdasarkan perbandingan hasil evaluasi yang ditunjukkan pada Gambar 4 dan Tabel Gambar 5, dapat disimpulkan bahwa algoritma SVM memiliki performa yang lebih unggul dibandingkan dengan Naive Bayes dalam mengklasifikasikan sentimen ulasan pengguna aplikasi JMO. Keunggulan SVM ditunjukkan oleh nilai akurasi yang lebih tinggi serta nilai *F1-score* yang lebih konsisten pada setiap kelas sentimen. Dengan demikian, SVM merupakan algoritma yang lebih efektif dan andal untuk digunakan dalam analisis sentimen pada penelitian ini.

4. KESIMPULAN

Berdasarkan hasil analisis sentimen terhadap ulasan pengguna aplikasi JMO pada Google Play Store, dapat disimpulkan bahwa algoritma SVM memiliki kinerja yang lebih unggul dibandingkan dengan algoritma Naive Bayes dalam mengklasifikasikan sentimen ulasan ke dalam kategori positif, negatif, dan netral. SVM menghasilkan nilai akurasi sebesar 88,9% dengan nilai *F1-score* yang tinggi dan relatif konsisten pada seluruh kelas sentimen, sehingga menunjukkan kemampuan yang andal dalam menangani data teks berdimensi tinggi. Sebaliknya, algoritma Naive Bayes memperoleh nilai akurasi sebesar 64,1% dan menunjukkan keterbatasan dalam mengklasifikasikan sentimen netral, yang berdampak pada rendahnya performa klasifikasi secara keseluruhan. Hasil analisis juga menunjukkan bahwa sentimen negatif pengguna aplikasi JMO didominasi oleh keluhan terkait permasalahan teknis, seperti kesulitan login, error sistem, serta kendala pada proses klaim dan pembaruan data.

Temuan dalam penelitian ini dapat dimanfaatkan sebagai bahan evaluasi bagi BPJS Ketenagakerjaan dalam upaya meningkatkan kualitas dan keandalan layanan aplikasi JMO. Selain itu, penelitian ini membuktikan bahwa algoritma SVM merupakan metode yang lebih efektif untuk digunakan dalam analisis sentimen ulasan aplikasi berbasis teks. Untuk penelitian selanjutnya, disarankan untuk mengembangkan pendekatan dengan menggunakan metode *deep learning* atau analisis sentimen berbasis aspek guna memperoleh hasil yang lebih mendalam dan komprehensif.

REFERENSI

- [1] K. Kusuma Dewi, I. Kaniawulan, C. Dewi Lestari, "Analisis Sentimen Pengguna Aplikasi Jmo Pada App Store Menggunakan Metode Naive Bayes," vol. 8, no. 2, 2023.
- [2] S. Alpin Rizaldi, S. Alam, and I. Kurniawan, "Analisis Sentimen Pengguna Aplikasi Jmo () Pada Google Play Store Menggunakan Metode Naive Bayes 1)," vol. 2, no. 3, pp. 109–117, 2023, doi: 10.55123.

- [3] S. Butsianto and A. M. Rifa'i, "Analisis Sentimen Ulasan Aplikasi Jamsostek dengan SVM, Random Forest, dan Logistic Regression," *Jurnal Informatika Ekonomi Bisnis*, pp. 700–706, Sep. 2025, doi: 10.37034/infv7i3.1266.
- [4] Efrata Madao, A. Irsyad, and M. Rivani Ibrahim, "Analisis Sentimen Pada Ulasan Pengguna Aplikasi Dengan Menggunakan Naive Bayes Dan Logistic Regression," *Jurnal Teknologi Informasi*, vol. 6, no. 2, 2025, doi: 10.46576/djtechno.
- [5] K. Kusuma Dewi, I. Kaniawulan, and C. Dewi Lestari, "Analisis Sentimen Pengguna Aplikasi Jmo Pada App Store Menggunakan Metode Naive Bayes," vol. 8, no. 2, 2023.
- [6] S. Alpin Rizaldi, S. Alam, and I. Kurniawan, "Analisis Sentimen Pengguna Aplikasi Jmo () Pada Google Play Store Menggunakan Metode Naive Bayes 1)," vol. 2, no. 3, pp. 109–117, 2023, doi: 10.55123.
- [7] S. Butsianto and A. M. Rifa'i, "Analisis Sentimen Ulasan Aplikasi Jamsostek dengan SVM, Random Forest, dan Logistic Regression," *Jurnal Informatika Ekonomi Bisnis*, pp. 700–706, Sep. 2025, doi: 10.37034/infv7i3.1266.
- [8] V. Fitriyana *et al.*, "Analisis Sentimen Ulasan Aplikasi Menggunakan Metode Support Vector Machine," 2023.
- [9] A. Ariyo Munandar and C. Edi Widodo, "Sentimen Analisis Aplikasi Belajar Online Menggunakan Klasifikasi SVM," 2026.
- [10] N. C. Ramadani, "Analisis Sentimen Untuk Mengukur Ulasan Pengguna Aplikasi Mobile Legend Menggunakan Algoritma Naive Bayes, SVM, Random Fores, Decision Tree, dan Logistic Regression," *JSI : Jurnal Sistem Informasi (E-Journal)*, vol. 16, no. 1, 2024.
- [11] T. Gori, A. Sunyoto, and H. Al Fatta, "Preprocessing Data dan Klasifikasi untuk Prediksi Kinerja Akademik Siswa," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 1, pp. 215–224, Feb. 2024, doi: 10.25126/jtiik.20241118074.
- [12] B. Hakim, "Analisa Sentimen Data Text Preprocessing Pada Data Mining Dengan Menggunakan Machine Learning," *JBASE - Journal of Business and Audit Information Systems*, vol. 4, no. 2, Aug. 2021, doi: 10.30813/jbase.v4i2.3000.
- [13] S. N. Rismanah, R. Astuti, and F. M. Basysyar, "Penerapan Algoritma Support Vector Machine Dalam Menganalisis Sentimen Ulasan Pelanggan Shopeefood Berdasarkan Twitter," 2024.
- [14] A. Selawati, Yan Rianto, Rachmawati Darma Astuti, Ainun Zumarniansyah, and Deny Novianti, "Analisis Sentimen Berita Online Terhadap Transportasi Online di Indonesia dengan Metode Naive Bayes Classifier, Support Vector Machine dan K-Nearest Neighbor," *Bulletin of Computer Science Research*, vol. 5, no. 2, pp. 105–111, Feb. 2025, doi: 10.47065/bulletincsr.v5i2.477.
- [15] P. R. Sari *et al.*, "Comparison of Naive Bayes and SVM Algorithms for Sentiment Analysis of PUBG Mobile on Google Play Store." [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [16] F. Yani, K. Imelda, and N. T. Kurniadi, "Analisis Sentiment Pengguna Aplikasi Mobile Legend Di Playstore Dengan Menggunakan Algoritma Naive Bayes Dan Support Vector Machine (Svm)," *Jurnal Teknologi Informasi*, vol. 6, no. 2, 2025, doi: 10.46576/djtechno.
- [17] L. Rhomeantias, A. Khairunisa, S. Shella, M. Wara, and K. M. Hindrayani, "Analisis Sentimen Ulasan Aplikasi Smile Indonesia Menggunakan Metode Naive Bayes Dan Support Vector Machine (SVM)," vol. 16, May 2025, doi: 10.52972/hoaq.voll6no1.
- [18] R. Maheri, F. N. Salisah, F. Muttakin, and M. Megawati, "Analisis Sentimen Ulasan Aplikasi M-Paspor Menggunakan Naive Bayes Dan Support Vector Machine," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 1, pp. 448–458, Jan. 2025, doi: 10.29100/jupi.v10i1.5826.
- [19] K. Kevin, M. Enjeli, and A. Wijaya, "Analisis Sentimen Penggunaan Aplikasi Kinemaster Menggunakan Metode Naive Bayes," *Jurnal Ilmiah Computer Science*, vol. 2, no. 2, pp. 89–98, Jan. 2024, doi: 10.58602/jics.v2i2.24.
- [20] S. Lestari and S. Febrianti, "Analisis Sentimen Ulasan Produk Shopee di Aplikasi Instagram Menggunakan Algoritma K-Nearest Neighbors," *JURNAL MALCOM*, vol. 5, pp. 1172–1180, 2025, doi: 10.57152/malcom.v5i4.1595.