JIGE 6 (4) (2025) 2871-2888
JURNAL ILMIAH GLOBAL EDUCATION
ejournal.nusantaraglobal.ac.id/index.php/jige
DOI: https://doi.org/10.55681/jige.v6i4.4704

Analisis Performa Metode Machine Learning dalam Mengidentifikasi Penyebab Ulasan Rating Satu Aplikasi MyBluebird

Almira Farradinda Azziizah[1], Hery Mustofa[1], Khothibul Umam[1*], Maya Rini Handayani[1]
[1] Teknologi Informasi, Universitas Islam Negeri Walisongo Semarang

*Corresponding author email: khothibul_umam@walisongo.ac.id

Article Info

Article history:

Received October 01, 2025
Approved November 20, 2025

Keywords:

Application, Classification, Machine Learning, Review, Transportation

ABSTRACT

This study addresses the increasing prevalence of negative user reviews for the MyBluebird ride-hailing application, focusing on the identification and classification of the main causes of one-star ratings. The research aims to compare the effectiveness of Support Vector Machine, Random Forest, and Naïve Bayes algorithms in classifying user complaints. Employing a quantitative experimental approach, the study utilizes a dataset of 1,399 one-star reviews collected purposively from Google Play Store. Data preprocessing includes cleaning, tokenization, and feature extraction using TF-IDF. The classification models are evaluated using accuracy, precision, recall, and F1-score metrics. Results indicate that Random Forest achieves the highest accuracy (90%), outperforming the other algorithms, with bugs/errors as the most frequent complaint, followed by driver performance, other issues, and price. The study concludes that machine learning-based classification can effectively map user dissatisfaction, though data imbalance remains a limitation. Future research should apply data balancing techniques and expand the dataset for broader generalization. Practical implications suggest that developers can utilize automated classification to improve service quality and address user needs more efficient.

ABSTRAK

Penelitian ini membahas meningkatnya prevalensi ulasan negatif pengguna pada aplikasi transportasi daring MyBluebird, dengan fokus pada identifikasi dan klasifikasi penyebab utama rating bintang satu. Tujuan penelitian adalah membandingkan efektivitas algoritma Support Vector Machine, Random Forest, dan Naïve Bayes dalam mengklasifikasikan keluhan pengguna. Penelitian menggunakan pendekatan kuantitatif eksperimental dengan dataset 1.399 ulasan bintang satu yang diambil secara purposive dari Google Play Store. Proses data meliputi pembersihan, tokenisasi, dan ekstraksi fitur menggunakan TF-IDF. Model klasifikasi dievaluasi dengan metrik akurasi, presisi, recall, dan F1-score. Hasil penelitian menunjukkan Random Forest memiliki akurasi tertinggi (90%) dibandingkan algoritma lain, dengan bug/error sebagai keluhan terbanyak, diikuti kinerja driver, kategori lain, dan harga. Kesimpulan penelitian

menyatakan klasifikasi berbasis machine learning efektif memetakan ketidakpuasan pengguna, meski ketidakseimbangan data masih menjadi keterbatasan. Penelitian selanjutnya disarankan menerapkan teknik penyeimbangan data dan memperluas dataset untuk generalisasi yang lebih luas. Implikasi praktisnya, pengembang dapat memanfaatkan klasifikasi otomatis untuk meningkatkan kualitas layanan dan merespons kebutuhan pengguna secara lebih efisien.

Copyright © 2025, The Author(s).
This is an open-access article under the CC–BY-SA license.

How to cite: Azziizah, A. F., Mustofa, H., Umam, K., & Handayani, M. R. (2025). Analisis Performa Metode Machine Learning dalam Mengidentifikasi Penyebab Ulasan Rating Satu Aplikasi MyBluebird. Jurnal Ilmiah Global Education, 6(4), 2871–2888. https://doi.org/10.55681/jige.v6i4.4704

INTRODUCTION

Dalam beberapa tahun terakhir, layanan transportasi daring telah mengalami pertumbuhan pesat, memenuhi kebutuhan mobilitas masyarakat urban yang mengutamakan efisiensi dan kenyamanan (Alfarobby & Irawan, 2024; Chamidy & Informatika, 2025). Salah satu aplikasi yang menonjol di Indonesia adalah MyBluebird, yang memfasilitasi pemesanan taksi secara digital. Namun, data dari Google Play Store menunjukkan tingginya jumlah ulasan negatif berupa rating bintang satu, yang menandakan adanya ketidakpuasan pengguna terhadap layanan tersebut (Ramadani et al., 2024; Subagja et al., 2021). Ulasan-ulasan ini mengindikasikan adanya permasalahan mendasar yang perlu diidentifikasi secara sistematis agar pengembang dapat meningkatkan kualitas layanan.

Fenomena ulasan negatif pada aplikasi transportasi daring tidak hanya terjadi pada MyBluebird, tetapi juga pada aplikasi serupa seperti Gojek dan Grab, di mana keluhan pengguna sering kali berkaitan dengan bug aplikasi, kinerja pengemudi, dan persepsi harga (Alfarobby & Irawan, 2024; Radiena & Nugroho, 2023). Kompleksitas data ulasan yang tidak terstruktur, dengan variasi gaya bahasa dan ekspresi, menjadi tantangan tersendiri dalam proses analisis dan klasifikasi penyebab utama ketidakpuasan pengguna (Gitacahyani et al., 2024; Meli et al, 2024).

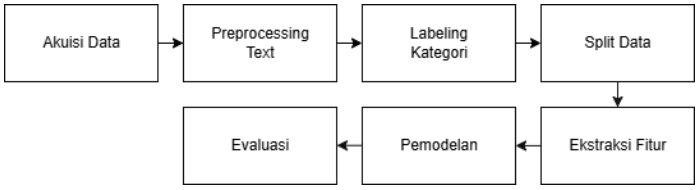
Permasalahan utama yang dihadapi dalam penelitian ini adalah bagaimana mengidentifikasi dan mengklasifikasikan penyebab utama ulasan bintang satu pada aplikasi MyBluebird secara otomatis dan akurat. Data ulasan yang bersifat teks bebas dan tidak terstruktur menyulitkan proses pemahaman keluhan pengguna secara manual (Khairunnisa et al., 2021; Septiani & Isabela, 2023). Selain itu, distribusi data yang tidak seimbang, khususnya pada kategori minoritas seperti harga, menambah kompleksitas dalam penerapan metode klasifikasi berbasis *machine learning* (Iqrom et al., n.d.; Vitalaya, 2024).

Penelitian sebelumnya telah membuktikan efektivitas metode *Support Vector Machine* (SVM) dan *Random Forest* (RF) dalam klasifikasi teks ulasan aplikasi digital, namun masih terdapat keterbatasan dalam menangani data tidak seimbang dan kategori minoritas (Larasati et al., 2022; Ramadani et al., 2024). Selain itu, model *baseline* seperti *Naïve Bayes* (NB) sering kali menunjukkan performa rendah pada kategori dengan jumlah data terbatas (Meli et al, 2024; Subagja et al., 2021). Oleh karena itu, diperlukan pendekatan komparatif untuk mengevaluasi performa ketiga algoritma tersebut dalam konteks klasifikasi penyebab ulasan negatif pada aplikasi MyBluebird.

Permasalahan lain yang diidentifikasi adalah kurangnya interpretabilitas hasil klasifikasi, terutama dalam mengidentifikasi fitur-fitur penting yang berkontribusi terhadap prediksi kategori keluhan (Larasati et al., 2022; Nababan & Hutagalung, 2023). Hal ini penting agar pengembang dapat memahami secara spesifik faktor-faktor yang memicu ketidakpuasan pengguna dan merumuskan strategi perbaikan yang tepat sasaran.

Penelitian ini bertujuan untuk mengidentifikasi dan mengklasifikasikan penyebab utama ulasan bintang satu pada aplikasi MyBluebird dengan membandingkan performa tiga algoritma machine learning, yaitu SVM, RF, dan NB. Urgensi penelitian terletak pada kebutuhan pengembang aplikasi untuk memperoleh pemahaman yang lebih terstruktur dan otomatis terhadap keluhan pengguna, sehingga dapat meningkatkan kualitas layanan secara efektif (Amalia & Yustanti, 2021; Prabowo & Kurniadi, 2023). Kebaruan penelitian ini terletak pada pendekatan komparatif yang tidak hanya fokus pada analisis sentimen, tetapi juga pada klasifikasi penyebab spesifik ulasan negatif berdasarkan kategori yang relevan secara bisnis, serta evaluasi interpretabilitas fitur penting dalam proses klasifikasi (Aditiya et al., 2025; Radhi et al., 2021).

METHODS



Gambar 1. Diagram Alur Penelitian

Akuisi Data

Dataset penelitian diperoleh melalui ekstraksi ulasan pengguna aplikasi MyBluebird dari Google Play Store menggunakan pustaka *google-play-scraper* berbasis Python (Subagja et al., 2021). Pengambilan difokuskan pada ulasan dengan rating bintang satu, yang dianggap mewakili pengalaman paling negatif dari pengguna terhadap aplikasi. Sebanyak 1.500 ulasan terbaru dikumpulkan, kemudian melalui proses pembersihan dan penghapusan duplikat, diperoleh 1.399 ulasan yang digunakan sebagai dataset. Ulasan dalam bentuk teks tidak terstruktur ini menjadi dasar untuk proses analisis lebih lanjut, dimulai dari preprocessing hingga klasifikasi.

Preprocessing Text

Data ulasan mentah yang diperoleh dari hasil scraping masih mengandung banyak unsur yang tidak relevan untuk analisis komputasional (Chamidy & Informatika, 2025). Oleh karena itu, dilakukan tahap *preprocessing text* guna meningkatkan kualitas data dan efektivitas ekstraksi fitur. Langkah-langkah preprocessing yang diterapkan meliputi:

Noise Removal

Menghapus elemen non-alfabet seperti angka, URL, tanda baca berlebihan, serta emoji. Tujuannya adalah untuk menghilangkan noise yang tidak memberikan kontribusi semantik pada konteks keluhan dan meningkatkan kualitas data sebelum dilakukan labeling kategori (Khairunnisa et al., 2021).

Tokenize dan Lowercase

Memecah kalimat menjadi satuan kata atau token menggunakan pustaka *Natural Language Toolkit* (nltk) dan teks diubah menjadi huruf kecil (*lowercase*) untuk menjaga konsistensi representasi kata (Radiena & Nugroho, 2023).

Stopword Removal

Mengeliminasi kata-kata fungsional yang tidak informatif dalam Bahasa Indonesia (seperti "yang", "dan", "itu") yang tidak memiliki informasi bermakna yang mendukung proses pengklasifikasian. Hal ini dilakukan menggunakan pustaka daftar *stopword* Bahasa Indonesia (Meli et al., 2024).

Stemming

Mengembalikan kata ke bentuk dasarnya (Gitacahyani et al., 2024) melalui pustaka Sastrawi, sehingga kata seperti "berjalan", "berjalanlah", dan "berjalanannya" akan dipetakan menjadi "jalan" untuk menyatukan varian kata dengan makna yang sama.

Labeling Kategori

Setiap data ulasan diklasifikasikan ke dalam empat kategori utama berdasarkan isi keluhan pengguna. Proses klasifikasi dilakukan secara semi-otomatis menggunakan pendekatan *rule-based*, yaitu melalui pencocokan kata kunci spesifik yang muncul dalam teks ulasan. Pendekatan ini bertujuan untuk mengarahkan proses pelabelan awal secara sistematis dan efisien agar sesuai dengan konteks keluhan pengguna (Shalihat, 2023).

Kategori klasifikasi yang digunakan terdiri dari Bug/Error Aplikasi, yang mencakup keluhan teknis seperti kesalahan login, force close, atau fitur tidak berfungsi dengan kata kunci “bug”, “error”, “verifikasi”, dan “gagal”. Kinerja Driver, meliputi ulasan terkait perilaku pengemudi seperti keterlambatan, sikap tidak profesional, atau ketidaksesuaian layanan dengan kata kunci “driver”, “jemput”, “pelayanan”, dan “ugal-ugalan”. Harga, mencakup keluhan mengenai tarif atau biaya perjalanan dengan kata kunci “harga”, “argo”, “mahal”, dan “discount”. Terakhir, Lainnya, berisi komentar umum atau keluhan yang tidak secara eksplisit termasuk dalam tiga kategori sebelumnya.

Split Data

Dataset yang telah dilabeli kemudian dibagi menjadi dua subset, yaitu data pelatihan (*train*) dan data pengujian (*test*), dengan rasio 80:20 menggunakan teknik *stratified sampling* untuk menjaga keseimbangan label (Iqrom et al., n.d.; Vitalaya, 2024). Total sebanyak 1.119 data digunakan untuk pelatihan dan 280 data untuk pengujian.

Ekstraksi Fitur

Tahap selanjutnya adalah konversi teks dinyatakan dalam bentuk fitur numerik agar sehingga siap digunakan sebagai input dalam model klasifikasi. Penelitian ini menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) sebagai teknik ekstraksi fitur (Septiani & Isabela, 2023). Metode TF-IDF menghitung bobot suatu kata berdasarkan dua komponen utama, yakni pengukuran seberapa sering sebuah kata muncul dalam suatu dokumen (*Term Frequency* (TF)) serta seberapa jarang kata tersebut muncul di seluruh kumpulan dokumen (*Inverse Document Frequency* (IDF)). Kata dengan frekuensi tinggi dalam satu dokumen dan rendah di dokumen lainnya akan diberikan bobot signifikan dalam perhitungan.

Pemodelan

Tahap pemodelan dilakukan untuk mengklasifikasikan penyebab ulasan bintang satu berdasarkan teks yang telah diproses. Pemodelan dilakukan menggunakan *pipeline* yang terdiri dari tiga algoritma, yaitu:

Support Vector Machine (SVM)

SVM digunakan dengan kernel linear karena cocok untuk data teks berdimensi tinggi dan bersifat sparsity (kepadatan rendah). Parameter C=1 digunakan untuk mengontrol margin soft terhadap pelatihan (Amalia & Yustanti, 2021).

Random Forest (RF)

Random forest digunakan sebagai model berbasis ensemble yang membangun banyak pohon keputusan (*decision trees*) (Larasati et al., 2022). Dalam penelitian ini, digunakan 100 *estimator* dengan pengaturan *random_state*=42 untuk menjaga reprodusibilitas hasil.

Naïve Bayes (NB)

Naive bayes digunakan sebagai baseline model untuk klasifikasi teks dengan mengasumsikan bahwa fitur-fitur yang digunakan bersifat independen satu sama lain. Model ini menggunakan pendekatan Multinomial Naive Bayes, yang umum digunakan untuk data teks dan berbasis distribusi kata. Tuning dilakukan terhadap parameter alpha untuk regularisasi (*Laplace smoothing*), dengan pemilihan nilai optimal melalui pencarian grid (Putri & Cahyono, 2024).

Evaluasi

Tahap ini memusatkan perhatian pada penilaian kinerja komparatif dari tiga algoritma machine learning SVM, Random Forest, dan Naïve Bayes, dalam mengklasifikasikan ulasan ke dalam kategori keluhan. Penilaian ini secara fundamental menggunakan *Confusion Matrix* yang terdiri dari empat komponen utama yaitu, *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) yang menjadi dasar perhitungan pada persamaan (1) sampai (3) sejumlah metrik utama diantaranya; *Accuracy* (proporsi prediksi benar secara keseluruhan), *Precision* (tingkat ketepatan prediksi positif yang benar), *Recall* (kemampuan model mengenali seluruh data positif yang sebenarnya), dan *F1-Score* (nilai rata-rata harmonis antara Precision dan Recall) (Prabowo & Kurniadi, 2023). Analisis ini juga mencakup diagnostik model melalui *tuning hyperparameter* dan analisis *learning curve* untuk mendeteksi *overfitting* atau kebutuhan data (Adi et al., 2025; Radhi et al., 2021). Secara krusial, evaluasi ini bertujuan mengidentifikasi dampak ketidakseimbangan data pada akurasi per kategori, yang pada akhirnya akan menentukan algoritma terbaik dan memberikan rekomendasi strategis bagi pengembang aplikasi berdasarkan temuan *feature importance*.

$$Accuracy = \frac{TP+TN}{(TP+FP+FN+TN)} \times 100\% \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

$$F1 - Score = \frac{2 \times precision \times recall}{precision+recall} \times 100\% \quad (4)$$

RESULTS AND DISCUSSION

Akuisi Data

Saat mengakuisi data menggunakan pustaka google-play-scraper sebanyak 1.399 ulasan dari 1.500 ulasan terbaru dikumpulkan. Berikut dataset rating satu pada gambar 2.

	content
0	mohon turun kan ongkos Nya
1	Sering trouble, tidak bisa d pilih menu pembay...
2	di bawa keliling padahall rute udah jelas
3	tdk bisa log in
4	CSnya udah mulai resek sekarang, hampir satu t...

Gambar 2. Akuisi Data

Preprocessing Text

Preprocessing text dilakukan untuk merubah kata ke dalam betuk dasar guna meningkatkan kualitas dataset. Terdapat beberapa tahapan preprocessing text yang diterapkan meliputi:

Noise Removal

Pada tabel 1 dilakukan *noise removal* untuk menghilangkan noise dan elemen non-alfabetik agar data keluhan menjadi murni dan siap untuk pelabelan kategori.

Tabel 1. Noise Removal

Sebelum Noise Removal	Noise Removal
1. barusan pesan, posisi maps salah ga saya cek karena sudah otomatis. 2. chat box nya kalau yg bukan gestur ketutup tombol, jadi ga bisa dipencet. kalau sudah diperbaiki app nya, saya ubah lagi rating nya :)	barusan pesan posisi maps salah ga saya cek karena sudah otomatis chat box nya kalau yg bukan gestur ketutup tombol jadi ga bisa dipencet kalau sudah diperbaiki app nya saya ubah lagi rating nya
Selalu Ngeblank Saat "Save" Data Email..	Selalu Ngeblank Saat Save Data Email
BANYAK SOPIR BLUE BIRD YG SENGAJA PUTARÂ² JALAN SUPAYA BAYAR MAHAL. BEDA DG GRAB CAR ATAU GOCAR	BANYAK SOPIR BLUE BIRD YG SENGAJA PUTAR2 JALAN SUPAYA BAYAR MAHAL BEDA DG GRAB CAR ATAU GOCAR

Tokenize dan Lowercase

Pada tabel 2 dilakukan *tokenize* dan *lowercase* dengan tujuan teks dipecah menjadi satuan kata dan dikonversi ke huruf kecil agar setiap kata memiliki representasi yang konsisten.

Tabel 2. Tokenize dan Lowercase

Sebelum Tokenize dan Lowercase	Tokenize dan Lowercase
barusan pesan posisi maps salah ga saya cek karena sudah otomatis chat box nya kalau yg bukan gestur ketutup tombol jadi ga bisa dipencet kalau sudah diperbaiki app nya saya ubah lagi rating nya	['barusan', 'pesan', 'posisi', 'maps', 'salah', 'ga', 'saya', 'cek', 'karena', 'sudah', 'otomatis', 'chat', 'box', 'nya', 'kalau', 'yg', 'bukan', 'gestur', 'ketutup', 'tombol', 'jadi', 'ga', 'bisa', 'dipencet', 'kalau', 'sudah', 'diperbaiki', 'app', 'nya', 'saya', 'ubah', 'lagi', 'rating', 'nya']
Selalu Ngeblank Saat Save Data Email	['selalu', 'ngeblank', 'saat', 'save', 'data', 'email']
BANYAK SOPIR BLUE BIRD YG SENGAJA PUTAR2 JALAN SUPAYA BAYAR MAHAL BEDA DG GRAB CAR ATAU GOCAR	[‘banyak’, ‘sopir’, ‘blue’, ‘bird’, ‘yg’, ‘sengaja’, ‘putar’, ‘jalan’, ‘supaya’, ‘bayar’, ‘mahal’, ‘beda’, ‘dg, ‘grab’, ‘car’, ‘gocar’]

Stopword Removal

Pada tabel 3 dilakukan tahap *stopword removal* dilakukan agar kata-kata konektor Bahasa Indonesia yang tidak signifikan, seperti 'dan' atau 'yang', disaring keluar dari data.

Tabel 3. Stopword Removal

Sebelum Stopword Removal	Stopword Removal
barusan pesan posisi maps salah ga saya cek karena sudah otomatis chat box nya kalau yg bukan gestur ketutup tombol jadi ga bisa dipencet kalau sudah diperbaiki app nya saya ubah lagi rating nya	[‘barusan', 'ga', ‘saya’, 'nya', ‘kalau’, 'yg', ‘bukan’, ‘jadi’, ‘bisa’, ‘lagi’]
Selalu ngeblank saat save data email	['selalu', ‘saat’]
Banyak sopir blue bird yg sengaja putar2 jalan supaya bayar mahal beda dg grab car atau gocar	[‘yg’, 'supaya', ‘atau’]

Stemming

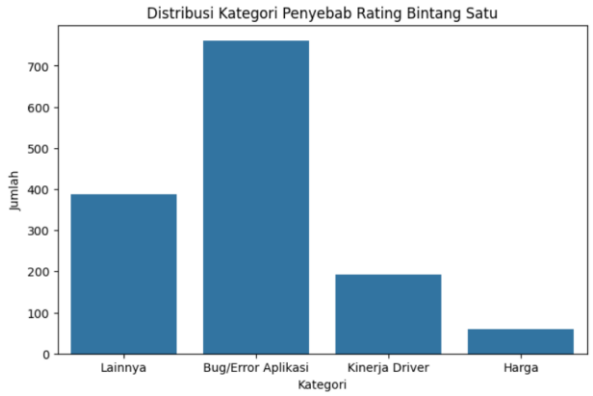
Pada tabel 4 dilakukan *stemming* untuk mereduksi kata berimbuhan menjadi kata dasar tunggal untuk meningkatkan efisiensi pengklasifikasian dan analisis.

Tabel 4. Stemming

Sebelum Stemming	Stemming
pesan posisi maps salah cek otomatis chat box gestur ketutup tombol dipencet diperbaiki app ubah rating	pesan posisi map salah cek otomatis chat box gestur tutup tombol pencet baik app ubah rating
ngeblank save data email	blank save data email
banyak sopir blue bird sengaja putar2 jalan bayar mahal beda dg grab car gocar	banyak sopir blue bird sengaja putar jalan bayar mahal beda dg grab car gocar

Labeling Kategori

Mengelompokkan setiap data ulasan ke dalam empat kategori utama melalui proses semi-otomatis berbasis aturan, di mana pencocokan kata kunci spesifik digunakan untuk penentuan label yang sistematis dan relevan dengan konteks keluhan pengguna. Kategori klasifikasi yang digunakan terdiri dari Bug/Error Aplikasi, Kinerja Driver, Harga, dan Lainnya yang ditampilkan pada gambar 3.



Gambar 3. Labeling Kategori

Split Data

Guna memastikan keseimbangan label, dataset yang telah dilabeli didistribusikan menjadi 80% data pelatihan (1.119) dan 20% data pengujian (280) melalui metode stratified sampling.

Ekstraksi Fitur

Metode *Term Frequency-Inverse Document Frequency* (TF-IDF) dipilih sebagai teknik ekstraksi fitur untuk mengonversi data teks berlabel menjadi representasi numerik, dengan membobot kata berdasarkan relevansi uniknya di setiap dokumen ulasan. Hasil disajikan pada gambar 4.

```
10 fitur dengan bobot TF-IDF tertinggi:
         fitur  rata_rata_bobot
126   aplikasi         0.049928
830       gak          0.029557
817        ga          0.029457
1843      nya          0.027901
708    driver          0.025736
2844       yg          0.023238
1898    order          0.019548
2508    susah          0.019440
563    daftar          0.017317
34        aja          0.017079
```

Gambar 4. Ekstraksi Fitur

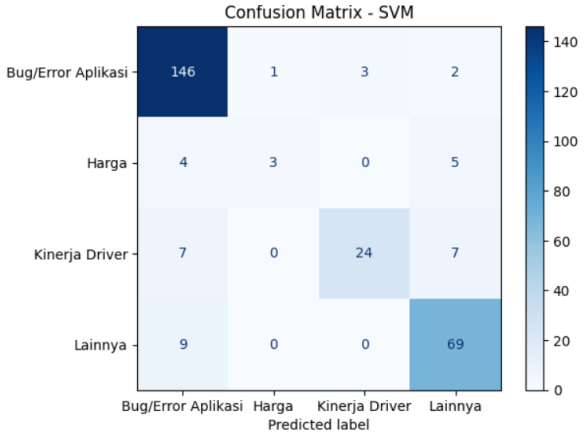
Evaluasi Support Vector Machine (SVM)

Klasifikasi SVM

Evaluasi awal terhadap performa algoritma SVM dalam mengklasifikasikan penyebab ulasan bintang satu pada aplikasi MyBluebird menunjukkan hasil yang cukup menjanjikan. Berdasarkan Tabel 5, model memperoleh akurasi keseluruhan sebesar 86% dari total 280 data uji. Secara umum, nilai *macro average* F1-score mencapai 0,72 yang menandakan bahwa meskipun distribusi data antar kelas tidak seimbang, model masih mampu mengklasifikasikan sebagian besar kategori dengan baik. Nilai *weighted average* F1-score yang mencapai 0,85 juga memperlihatkan bahwa performa model cenderung stabil, meskipun hasil ini lebih dipengaruhi oleh dominasi kelas mayoritas.

Tabel 5. Klasifikasi SVM Sebelum Tuning

	Precision	Recall	F1-Score	Support
Bug/error aplikasi	0,88	0,96	0,92	152
Harga	0,75	0,25	0,38	12
Kinerja driver	0,89	0,63	0,74	38
Lainnya	0,83	0,88	0,86	78
Accuracy			0,86	280
Macro avg	0,84	0,68	0,72	280
Weighted avg	0,86	0,86	0,85	280



Gambar 5. Confusion Matrix SVM Sebelum Tuning

Jika ditinjau lebih rinci, berdasarkan gambar 5 kategori dengan performa terbaik adalah Bug/Error Aplikasi, yang memiliki nilai F1-score sebesar 0,92. Tingginya nilai ini dipengaruhi oleh jumlah data yang sangat besar, yakni 152 sampel atau lebih dari separuh total dataset. Keberlimpahan data membuat model lebih mudah menangkap pola kosakata yang berhubungan dengan keluhan bug, seperti kata "crash", "error" atau "gagal".Hasil ini tercermin pula pada confusion matrix, di mana 146 dari 152 data pada kategori Bug/Error berhasil diprediksi dengan benar. Sebaliknya, performa terendah ditunjukkan pada kategori Harga, dengan F1-score hanya 0,38. Rendahnya performa ini terutama disebabkan oleh jumlah data yang sangat sedikit, yaitu hanya 12 ulasan. Minimnya data membuat model kesulitan mengenali pola kosakata khusus yang mengindikasikan keluhan harga. Selain itu, konteks ulasan harga sering kali bercampur dengan kategori lain, misalnya keluhan tarif yang tidak sesuai akibat bug atau promosi yang gagal masuk, sehingga model sering salah mengklasifikasikan keluhan harga sebagai bug.

Confusion matrix memperlihatkan bahwa empat ulasan terkait harga justru masuk ke kategori Bug/Error. Hal ini menunjukkan bahwa pada kondisi data minoritas, model kesulitan menjaga keseimbangan antara presisi dan recall, yang menyebabkan sebagian besar ulasan harga tidak teridentifikasi dengan benar. Untuk katefori Kinerja Driver, model mencatat F1-score sebesar 0,74, dengan presisi tinggi (0,89) namun recall yang lebih rendah (0,63). Artinya, ketika model memutuskan suatu ulasan berkaitan dengan kinerja driver, prediksi tersebut relatif akurat. Akan tetapi, banyak ulasan mengenai driver yang tidak berhasil dikenali dan justru diprediksi sebagai kategori lain, khususnya Bug/Error. Sebanyak sembilan ulasan tentang kinerja driver salah diklasifikasikan sebagai bug. Hal ini dimungkinkan karena adanya tumpang tindih kosakata, misalnya keluhan terkait driver sering disertai dengan kata-kata yang juga berhubungan dengan aplikasi, seperti "order", "titik jemput", atau "aplikasi error".

Kategori Lainnya menempati posisi menengah dengan F1-score 0,86, menunjukkan bahwa model cukup baik dalam mengenali ulasan yang tidak termasuk ke dalam kategori utama. Akan tetapi, tingginya akurasi pada kelas ini juga perlu dicermati secara hati-hati karena kategori "Lainnya" bersifat sangat heterogen dan luas, sehingga ada kemungkinan model menggunakan kelas ini sebagai kategori fallback ketika sinyal kelas lain kurang kuat. Secara keseluruhan, hasil evaluasi ini menegaskan bahwa distribusi data yang tidak seimbang memiliki pengaruh signifikan terhadap kinerja model. Kelas mayoritas seperti Bug/Error mendapatkan performa yang sangat baik karena dukungan jumlah data yang besar, sedangkan kelas minor seperti Harga mengalami kesulitan serius karena keterbatasan data dan kemiripan kosakata dengan kelas lain. Confusion matrix memberikan gambaran jelas bahwa kesalahan klasifikasi paling sering terjadi dengan mengalirnya data minoritas ke kelas mayoritas, terutama ke Bug/Error.

Dengan demikian, meskipun akurasi total terlihat tinggi, evaluasi mendalam melalui F1-score dan confusion matrix menunjukkan masih adanya kelemahan mendasar, khususnya pada kelas dengan jumlah data terbatas. Oleh karena itu, perbaikan model perlu diarahkan pada peningkatan performa kelas minor, baik melalui penambahan jumlah data, pengaturan bobot kelas dalam SVM, maupun rekayasa fitur yang lebih peka terhadap konteks bahasa pada ulasan. Perbaikan ini diharapkan dapat meningkatkan keseimbangan kinerja model lintas kelas, sehingga macro-F1 dapat meningkat tanpa mengorbankan performa pada kelas mayoritas.

Tuning Hyperparameter SVM

Untuk meningkatkan performa, dilakukan tuning hyperparameter dengan pendekatan *grid search* guna mencari parameter optimal (Adi et al., 2025). Setelah penyesuaian, performa SVM meningkat dengan akurasi mencapai 88%. Selain itu, macro average F1-score naik menjadi 0,76, menunjukkan peningkatan kestabilan model. Kategori bug/error aplikasi tetap menjadi yang paling akurat dikenali (F1-score: 0,94), sementara kategori harga masih menunjukkan performa rendah akibat keterbatasan jumlah data. Hasil ini mengindikasikan bahwa tuning berdampak positif dalam memperbaiki kemampuan generalisasi model.

Tabel 6. Tuning Hyperparameter SVM

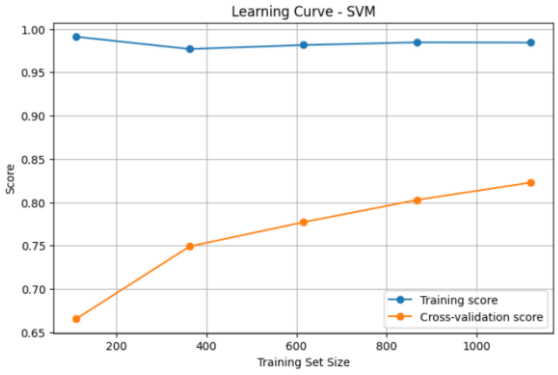
	Precision	Recall	F1-Score	Support
Bug/error aplikasi	0,92	0,95	0,94	152
Harga	0,80	0,33	0,47	12
Kinerja driver	0,87	0,68	0,76	38
Lainnya	0,84	0,94	0,88	78
Accuracy			0,89	280
Macro avg	0,86	0,73	0,76	280
Weighted avg	0,88	0,89	0,88	280

Hasil tuning hyperparameter menggunakan pendekatan grid search memberikan dampak positif terhadap performa algoritma SVM dalam mengklasifikasikan ulasan bintang satu pada aplikasi MyBluebird. Seperti ditunjukkan pada Tabel 6, Peningkatan akurasi menjadi 89% dan macro average F1-score 0,76 menunjukkan perbaikan kemampuan model dalam mengenali pola bahasa yang kompleks. Secara per kelas, kategori Bug/Error Aplikasi tetap dominan (F1-score: 0,94) karena kelimpahan data, dengan peningkatan presisi dari 0,88 menjadi 0,92 yang membuktikan model lebih jarang salah klasifikasi. Kategori Kinerja Driver juga mengalami sedikit kenaikan F1-score dari 0,74 menjadi 0,76, meskipun adanya tumpang tindih kosakata dengan keluhan bug masih membuat sebagian ulasan salah klasifikasi. Kategori Lainnya juga mencatat performa yang sedikit lebih baik (F1-score: 0,88), mengindikasikan model semakin mampu menangani ulasan heterogen.

Sementara itu, kenaikan F1-score kategori Harga dari 0,38 menjadi 0,47 juga menunjukkan bahwa tuning membantu model mengenali pola kosakata ulasan terkait harga. Meskipun demikian, nilai recall yang hanya mencapai 0,33 secara jujur memperlihatkan bahwa model telah mencapai batas kemampuannya karena keterbatasan jumlah data yang ekstrem (hanya 12 ulasan). Oleh karena itu, hasil ini membuktikan bahwa penyesuaian parameter telah mencapai batas optimalnya, dan untuk perbaikan lebih lanjut, strategi tambahan seperti penyeimbangan data (*oversampling* atau *undersampling*), pembobotan kelas, maupun augmentasi teks sangat direkomendasikan untuk mengatasi ketimpangan distribusi data yang menjadi akar masalah utama.

Learning Curve

Analisis learning curve dilakukan untuk mengevaluasi kecenderungan model terhadap *overfitting* atau *underfitting* (Radhi et al., 2021). Grafik pada Gambar 6 memperlihatkan bahwa nilai training score pada SVM cukup tinggi dan stabil, sedangkan *validation score* menunjukkan peningkatan seiring bertambahnya jumlah data latih, sebelum akhirnya mendatar pada titik tertentu.



Gambar 6. Grafik Learning Curve SVM

Berangkat dari Gambar 6, learning curve SVM memperlihatkan dua garis utama yaitu, skor pelatihan yang sejak awal tinggi dan relatif stabil, serta skor validasi yang mula-mula lebih rendah lalu meningkat seiring bertambahnya ukuran data latih sebelum akhirnya mendatar pada suatu

plateau (mendatar). Pola ini adalah ciri klasik model dengan kapasitas yang memadai untuk "menghafal" pola pada data latih (sehingga training score tinggi), namun tetap harus "belajar" generalisasi dari keragaman contoh terlihat dari validation score yang terus naik sampai titik tertentu. Selisih yang masih tersisa antara kedua kurva pada saat mendekati akhir (*generalization gap*) menunjukkan gejala overfitting ringan: model sedikit terlalu pas terhadap idiosinkrasi data pelatihan, tetapi belum pada taraf yang mengkhawatirkan karena kurva validasi stabil dan tidak menurun.

Secara *bias variance*, kurva ini menandakan bias yang tidak besar (karena training score tinggi dan tidak menunjukkan kesulitan mempelajari pola dasar), sementara varians masih hadir dalam derajat moderat (tercermin dari gap pelatihan validasi yang tidak sepenuhnya tertutup). Dengan kata lain, SVM sudah mampu menangkap struktur bahasa yang relevan untuk sebagian besar kelas, namun masih kehilangan sebagian sinyal ketika dihadapkan pada contoh baru yang lebih beragam. Plateau pada kurva validasi mengisyaratkan batas manfaat marjinal dari penambahan data acak: tambahan contoh yang sifatnya mirip dengan yang sudah ada tidak lagi mendorong kinerja naik secara berarti. Dalam konteks korpus ulasan MyBluebird yang tidak seimbang, plateau ini sangat mungkin dipengaruhi oleh minimnya representasi kelas minor khususnya "Harga" serta kemiripan kosakata antarkategori (misalnya keluhan harga yang sering beririsan dengan istilah teknis aplikasi), yang membuat ruang keputusan SVM condong ke kelas mayoritas.

Implikasi praktis untuk perbaikan model selanjutnya dapat dirumuskan menjadi dua fokus utama yang saling berkaitan. Oleh karena itu, kurva ini memandu strategi perbaikan: untuk mengangkat kinerja melewati batas plateau ini, penambahan data harus bersifat terarah dengan memperkaya contoh yang langka. Selain itu, kurva ini menggarisbawahi pentingnya penyesuaian regulerisasi. Pengaturan hyperparameter SVM, seperti nilai C, adalah kunci untuk menutup generalization gap dan menekan varians. Sementara algoritma lain cenderung menghasilkan kurva yang kurang informatif pada dataset berdimensi tinggi, bentuk kurva SVM-lah yang secara tuntas memvalidasi diagnosis penelitian ini, mengkonfirmasi kebutuhan data minoritas yang beragam dan memandu langkah-langkah *feature engineering* (seperti n-gram dan pembobotan domain-spesifik) untuk penelitian lanjutan.

Evaluasi Random Forest (RF)

Klasifikasi Random Forest

Evaluasi awal terhadap algoritma Random Forest (RF) menunjukkan hasil klasifikasi yang cukup baik. Berdasarkan Tabel 3, model mencapai akurasi sebesar 90% sebelum dilakukan tuning. Kategori Bug/error aplikasi dan kinerja driver memiliki nilai F1-score tertinggi, masing-masing sebesar 0,95 dan 0,87, menunjukkan bahwa model mampu mengenali dua kategori utama tersebut dengan akurasi yang tinggi. Namun, performa pada kategori harga masih rendah, ditunjukkan oleh nilai recall sebesar 0,42, yang mengindikasikan bahwa model belum optimal dalam mengenali ulasan terkait tarif. Hal ini kemungkinan disebabkan oleh ketidakseimbangan distribusi data antar kategori. Secara umum, model menunjukkan performa klasifikasi yang stabil di berbagai kategori, dengan macro average F1-score sebesar 0,81.

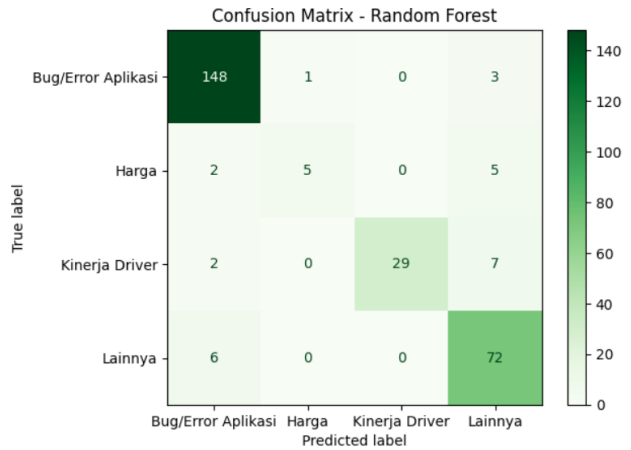
Tabel 7. Klasifikasi Random Forest Sebelum Tuning

	Precision	Recall	F1-Score	Support
Bug/error aplikasi	0,94	0,97	0,95	152
Harga	0,83	0,42	0,56	12
Kinerja driver	1,00	0,76	0,87	38
Lainnya	0,83	0,92	0,87	78
Accuracy			0,91	280
Macro avg	0,90	0,77	0,81	280
Weighted avg	0,91	0,91	0,90	280

Berdasarkan tabel 7 dapat perhatikan secara mendalam, hasil evaluasi awal algoritma Random Forest (RF) sebelum dilakukan tuning memberikan gambaran yang cukup kuat terkait keunggulan sekaligus keterbatasan model ini dalam klasifikasi teks ulasan pengguna MyBluebird. Secara umum, nilai akurasi sebesar 91% menunjukkan bahwa RF mampu mengenali sebagian besar data dengan benar. Lebih jauh, performa ini juga didukung oleh nilai weighted average F1-score sebesar 0,90, yang berarti model secara keseluruhan mampu menjaga keseimbangan antara presisi dan recall pada mayoritas kelas, terutama pada kategori dengan jumlah data besar. Pada kategori Bug/Error Aplikasi, model menunjukkan performa yang sangat baik dengan F1-score 0,95, presisi 0,94, dan recall 0,97. Angka recall yang sangat tinggi menandakan bahwa hampir seluruh data bug/error berhasil teridentifikasi dengan benar, sementara presisi yang relatif tinggi mengindikasikan bahwa sebagian besar prediksi yang diklasifikasikan sebagai bug/error memang benar adanya. Hal ini wajar karena kategori ini merupakan kelas mayoritas dengan jumlah 152 data (lebih dari separuh total dataset), sehingga model memiliki cukup banyak contoh untuk belajar pola-pola bahasa khas terkait keluhan bug aplikasi.

Pada kategori Kinerja Driver, model SVM tampil solid dengan F1-score 0,87 dan presisi sempurna (1,00). Tingginya presisi ini menandakan bahwa prediksi model sebagai "kinerja driver" sangat jarang salah, namun recall yang sedikit lebih rendah (0,76) mengindikasikan adanya tumpang tindih kosakata antara ulasan tentang driver dan bug/error, sehingga menyebabkan model terkadang kesulitan membedakan keduanya. Sementara itu, kategori Lainnya menunjukkan keseimbangan performa yang baik dengan F1-score 0,87. Tingginya recall (0,92) membuktikan kemampuan model dalam menangkap keragaman ulasan umum. Meskipun presisi sedikit lebih rendah (0,83), hal ini wajar karena kelas "lainnya" bersifat heterogen, yang memuat berbagai ulasan yang tidak mudah dipetakan secara jelas dan mungkin menyerap beberapa prediksi dari kategori spesifik.

Kelemahan paling nyata pada model Random Forest terlihat jelas pada kategori Harga, di mana meskipun presisi cukup tinggi (0,83), recall sangat rendah (0,42) dan F1-score hanya mencapai 0,56. Rendahnya recall ini secara langsung disebabkan oleh ketidakseimbangan distribusi data yang ekstrem pada kategori harga hanya memiliki 12 data (sekitar 4% dari total dataset), menyebabkan RF kesulitan membentuk pohon keputusan yang kaya pola dan cenderung bias ke kelas mayoritas. Hal ini ditegaskan oleh macro average F1-score sebesar 0,81 yang ditarik turun oleh performa buruk pada kelas minoritas, meskipun model unggul dalam stabilitas dan akurasi pada kelas mayoritas. Secara analitis, hasil ini menegaskan bahwa meskipun RF kuat pada data yang seimbang, model ini menghadapi kendala serius dalam generalisasi pada kelas minoritas. Oleh karena itu, perbaikan lebih lanjut mutlak diperlukan, terutama melalui upaya penyeimbangan data seperti *Synthetic Minority Over-sampling Technique* (SMOTE) yang secara spesifik menciptakan sampel sintetis baru untuk mengurangi bias terhadap kelas mayoritas, serta tuning parameter pohon (misalnya n_estimators dan max_depth) guna meningkatkan recall pada kategori harga.



Gambar 7. Confusion Matrix Random Forest Sebelum Tuning

Berdasarkan hasil analisis confusion matrix pada Gambar 7 terlihat secara lebih mendalam bahwa algoritma Random Forest memperlihatkan kecenderungan untuk bekerja optimal pada kelas dengan jumlah data yang besar, namun masih kesulitan pada kelas minoritas. Pada kategori Bug/Error Aplikasi, model menunjukkan performa yang hampir sempurna dengan 148 dari 152 data terklasifikasi benar. Tingginya tingkat keberhasilan ini menegaskan bahwa Random Forest mampu menangkap pola bahasa khas dari keluhan bug aplikasi, seperti istilah "error", "crash", atau "aplikasi tidak bisa dibuka", yang relatif konsisten muncul dalam dataset. Keberhasilan ini juga sejalan dengan tingginya proporsi data kategori bug/error yang membuat model lebih sering belajar pola dari kelas ini.

Kategori Lainnya menunjukkan tingkat klasifikasi yang cukup baik (72 dari 78 data benar), menegaskan kemampuan model membedakan ulasan umum meskipun bersifat heterogen. Namun, analisis Confusion Matrix memperlihatkan adanya tumpang tindih kosakata yang signifikan pada kelas Kinerja Driver; meskipun 29 dari 38 data dikenali, sejumlah kesalahan klasifikasi driver justru masuk ke kategori bug/error atau lainnya merupakan indikasi bahwa istilah seperti "lambat" atau "tidak responsif" merujuk pada performa aplikasi sekaligus pengemudi. Kesulitan paling signifikan terjadi pada kategori Harga, yang hanya berhasil dikenali 5 dari 12 data. Pola ini secara keseluruhan menegaskan adanya bias model terhadap kelas dengan distribusi data besar, terutama Bug/Error Aplikasi dan Lainnya. Kondisi ini umum pada algoritma berbasis pohon keputusan seperti RF; jumlah data harga yang sangat kecil membuat model tidak memiliki cukup representasi untuk membentuk cabang keputusan yang kuat, sehingga recall-nya sangat rendah. Oleh karena itu, hasil ini memberikan gambaran bahwa meskipun Random Forest memiliki performa umum yang baik, ketidakseimbangan data tetap menjadi faktor penentu kualitas klasifikasi, yang menuntut pertimbangan strategi resampling data (oversampling kelas minoritas) atau penggunaan algoritma yang lebih adaptif.

Tuning Hyperparameter RF

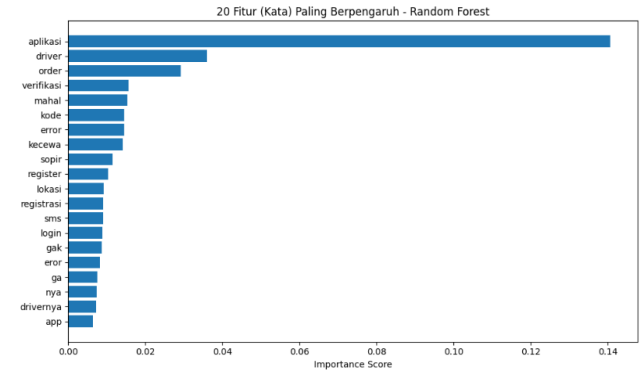
Penerapan grid search sebagai pendekatan dalam tuning hyperparameter telah banyak digunakan, termasuk pada studi prediksi sentimen ulasan dengan algoritma Random Forest yang menunjukkan peningkatan performa model (Nababan & Hutagalung, 2023) seperti terlihat pada Tabel 8. Akurasi naik menjadi 90%. Meskipun kategori "Harga" tetap menunjukkan recall yang rendah, secara keseluruhan tuning memberikan dampak positif terhadap stabilitas dan akurasi model.

Tabel 8. Tuning Hyperparameter RF

	Precision	Recall	F1-Score	Support
Bug/error aplikasi	0,94	0,95	0,95	152
Harga	0,83	0,42	0,56	12
Kinerja driver	0,97	0,79	0,87	38
Lainnya	0,82	0,94	0,87	78
Accuracy			0,90	280
Macro avg	0,94	0,95	0,95	152
Weighted avg	0,83	0,42	0,56	12

Feature Importance

Feature importance bertujuan mengidentifikasi kata-kata yang paling berpengaruh dalam klasifikasi. Visualisasi feature importance pada Gambar 8 menunjukkan kata-kata yang paling berpengaruh yaitu fitur seperti "aplikasi", "driver", "order", dan "mahal" memiliki skor penting yang tinggi, menunjukkan bahwa kemunculan kata-kata ini sangat membantu model dalam membedakan kategori keluhan. Hal ini memberikan wawasan mengenai fokus utama keluhan pengguna terhadap layanan.



Gambar 8. Feature Importance Random Forest

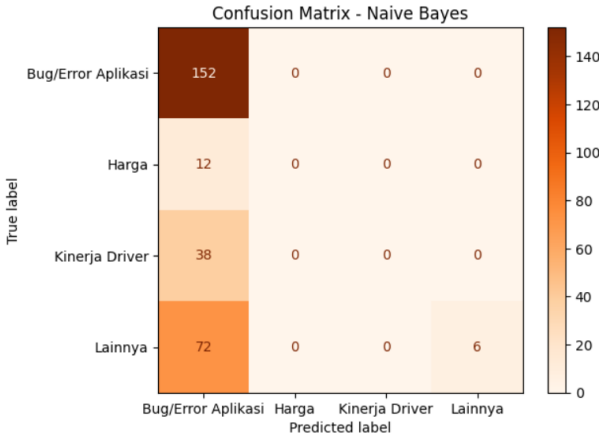
Evaluasi Naïve Bayes (NB)

Klasifikasi Naïve Bayes

Naive Bayes digunakan sebagai baseline model klasifikasi teks ulasan. Pada evaluasi awal, model ini memperoleh akurasi sebesar 56%, dengan performa terbaik pada kelas bug/error aplikasi (F1-score: 0,71), sedangkan kelas lain seperti kinerja driver dan harga tidak berhasil dikenali sama sekali (F1-score = 0). Sebagaimana terlihat pada tabel 9.

Tabel 9. Klasifikasi Naïve Bayes sebelum tunning

	Precision	Recall	F1-Score	Support
Bug/error aplikasi	0,55	1,00	0,71	152
Harga	0,00	0,00	0,00	12
Kinerja driver	0,00	0,00	0,00	38
Lainnya	1,00	0,08	0,14	74
Accuracy			0,56	280
Macro avg	0,39	0,27	0,21	280
Weighted avg	0,58	0,56	0,43	280



Gambar 9. Confusion Matrix Naïve Bayes Sebelum Tuning

Confusion matrix pada Gambar 9 memperlihatkan bahwa model memiliki kecenderungan kuat untuk mengklasifikasikan mayoritas data diklasifikasikan ke dalam satu kategori tertentu, yaitu bug/error aplikasi, termasuk semua data dari kategori harga dan kinerja driver yang salah diklasifikasikan ke dalam kelas tersebut.

Tuning Hyperparameter NB

Setelah tuning terhadap parameter alpha = 0.1, performa meningkat dengan akurasi mencapai 64%, dan macro average F1-score naik menjadi 0,39 seperti terlihat pada tabel 10. Model mulai

mengenali sebagian data dari kelas minoritas, walaupun recall masih rendah. Sebagai baseline, NB memberikan batas bawah performa klasifikasi teks dan mendemonstrasikan keterbatasan dalam menangani data tidak seimbang.

Tabel 10. Tuning Hyperparameter NB

	Precision	Recall	F1-Score	Support
Bug/error aplikasi	0,62	0,98	0,76	152
Harga	0,00	0,00	0,00	12
Kinerja driver	0,62	0,26	0,37	38
Lainnya	0,84	0,27	0,41	78
Accuracy			0,64	280
Macro avg	0,52	0,38	0,39	280
Weighted avg	0,66	0,64	0,58	280

Hasil evaluasi setelah dilakukan tuning terhadap parameter smoothing $\alpha = 0,1$ pada algoritma Naïve Bayes menunjukkan adanya peningkatan performa dibandingkan sebelum tuning, meskipun secara umum model masih relatif lemah dibandingkan SVM maupun Random Forest. Akurasi keseluruhan naik menjadi 64%, dan nilai macro average F1-score meningkat menjadi 0,39. Peningkatan ini mengindikasikan bahwa penyesuaian parameter memberikan efek positif terhadap kemampuan model dalam mengenali sebagian kelas minoritas, walaupun recall tetap rendah pada beberapa kategori. Secara lebih rinci, performa terbaik dicapai pada kelas Bug/Error Aplikasi dengan F1-score 0,76.

Hal ini menunjukkan bahwa model cukup mampu mengenali ulasan yang berkaitan dengan gangguan teknis, terutama karena kata-kata kunci yang muncul dalam keluhan bug atau error relatif konsisten dan mudah dipelajari oleh Naïve Bayes. Tingginya nilai recall (0,98) pada kategori ini memperlihatkan bahwa hampir semua data bug/error berhasil dikenali, meskipun tingkat presisinya tidak setinggi kelas mayoritas pada algoritma lain. Pada kelas Harga, meskipun jumlah data sangat kecil, nilai presisi tetap 0,00 yang berarti model tidak mampu memberikan prediksi yang benar pada kategori ini. Begitu juga dengan recall yang 0,00, menunjukkan bahwa seluruh data harga gagal dikenali dan justru terklasifikasi ke dalam kelas lain. Hal ini menunjukkan bahwa Naïve Bayes kurang mampu mendeteksi kelas dengan distribusi data sangat terbatas.

Sebaliknya, pada kategori Kinerja Driver dan Lainnya, performa model cukup rendah dengan F1-score 0,37 dan 0,41. Rendahnya nilai recall (0,26 pada Kinerja Driver dan 0,27 pada Lainnya) menegaskan bahwa sebagian besar data pada kedua kategori tersebut gagal dikenali dengan baik dan justru terklasifikasi ke kelas mayoritas. Hal ini dapat dijelaskan oleh asumsi independensi antar fitur pada Naïve Bayes yang membuat model kesulitan menangkap hubungan antar kata dalam kalimat ulasan, sehingga konteks keluhan driver atau kategori lainnya sering tertukar dengan bug/error aplikasi. Secara keseluruhan, hasil ini menegaskan bahwa meskipun tuning mampu meningkatkan performa Naïve Bayes, algoritma ini tetap kurang optimal dalam menangani ketidakseimbangan data dan klasifikasi teks dengan konteks yang kompleks. Oleh karena itu, peran Naïve Bayes lebih tepat dijadikan sebagai baseline untuk membandingkan performa algoritma lain, bukan sebagai metode utama dalam penelitian ini.

Analisis Komparatif dan Pilihan Metode Terbaik

Berdasarkan analisis komparatif, Random Forest (RF) secara tegas merupakan algoritma paling optimal untuk digunakan dalam penelitian ini. RF unggul dengan memberikan akurasi tertinggi (90%) dan menunjukkan stabilitas performa global yang superior, serta mampu mengurangi risiko overfitting berkat sifat ensemble yang kuat. Keunggulan RF ini dilengkapi dengan interpretabilitasnya, menjadikannya dasar yang kuat untuk mengidentifikasi faktor dominan penyebab ulasan negatif. Di sisi lain, SVM meskipun unggul dalam memisahkan kelas dengan margin yang jelas terbukti sensitif terhadap ketidakseimbangan data (seperti yang terlihat dari F1-score rendah pada kategori Harga), sementara Naïve Bayes menunjukkan hasil terendah

karena asumsi independensi antar fitur yang tidak terpenuhi. Dengan demikian, meskipun terdapat keterbatasan pada kelas minoritas, kestabilan dan kekuatan diagnostik RF melebihi algoritma lain, menjadikannya pilihan terbaik, sementara SVM dan NB tetap relevan sebagai metode baseline komparatif.

Penyebab Rating Bintang Satu pada Aplikasi MyBluebird

Berdasarkan hasil klasifikasi, dapat dipahami bahwa bug/error aplikasi menjadi penyebab dominan dalam terbentuknya ulasan bintang satu, menegaskan bahwa faktor teknis seperti aplikasi gagal login atau crash masih menjadi titik lemah utama yang mengurangi kepercayaan dan reliabilitas layanan. Selanjutnya, kategori kinerja driver muncul sebagai faktor kedua yang cukup berpengaruh, mencerminkan bahwa kualitas layanan manusia (keterlambatan, pembatalan sepihak, sikap sopir) tetap menjadi komponen penting yang menentukan persepsi keseluruhan pengguna di lapangan. Kategori Lainnya menunjukkan adanya dimensi permasalahan baru yang bersifat heterogen (seperti metode pembayaran atau promosi umum); meskipun jumlahnya tidak sebanyak bug/error, keberadaannya penting karena menandakan akumulasi isu-isu kecil. Sementara itu, meskipun keluhan terkait harga menempati posisi paling kecil dalam distribusi ulasan, faktor ini tidak bisa diabaikan karena menyangkut sensitivitas pengguna terhadap biaya (misalnya tarif yang dianggap mahal) dan dapat memperkuat persepsi negatif terhadap layanan secara keseluruhan ketika disandingkan dengan masalah teknis atau pelayanan.

Implikasi Hasil Terhadap Pengembang Aplikasi

Temuan penelitian ini memberikan implikasi langsung bagi pengembang aplikasi MyBluebird untuk meningkatkan kualitas layanan secara sistematis. Implikasi utama adalah urgensi perbaikan teknis pada aplikasi untuk mengatasi dominasi keluhan bug/error melalui pengujian dan pembaruan yang responsif. Selanjutnya, diperlukan perhatian strategis pada kinerja driver melalui pelatihan rutin dan sistem insentif/sanksi yang transparan guna menjaga loyalitas pelanggan. Selain itu, meskipun proporsinya kecil, keluhan harga/promo harus ditindaklanjuti dengan komunikasi tarif yang lebih jelas dan penerapan algoritma harga yang lebih transparan. Secara keseluruhan, pemanfaatan klasifikasi berbasis machine learning yang dihasilkan penelitian ini sangat direkomendasikan untuk mendukung *data-driven decision making* (pengambilan keputusan berdasarkan data), memungkinkan pengembang memetakan area masalah dengan cepat dan merumuskan strategi perbaikan yang lebih tepat sasaran dan adaptif.

KESIMPULAN

Penelitian ini menemukan bahwa bug atau error aplikasi merupakan penyebab utama ulasan rating bintang satu pada aplikasi MyBluebird, diikuti oleh keluhan terkait kinerja driver, kategori lainnya, dan harga. Dari sisi performa algoritma, Random Forest terbukti sebagai metode paling optimal dengan akurasi mencapai 90% dan kestabilan klasifikasi antarkategori, sementara Support Vector Machine (SVM) menunjukkan akurasi 89% namun masih sensitif terhadap distribusi data yang tidak seimbang. Naïve Bayes, sebagai baseline, memiliki akurasi terendah dan kesulitan dalam mengenali kelas minoritas. Temuan ini menegaskan pentingnya perbaikan teknis aplikasi dan peningkatan kualitas layanan driver sebagai prioritas utama pengembang. Selain itu, hasil penelitian ini memperlihatkan bahwa distribusi data yang tidak merata, khususnya pada kategori harga, menjadi tantangan signifikan dalam pengembangan model klasifikasi berbasis machine learning.

Keterbatasan utama penelitian ini terletak pada ketidakseimbangan distribusi data antar kategori, sehingga performa model pada kelas minoritas seperti harga masih rendah. Selain itu, penelitian hanya menggunakan data ulasan bintang satu dari satu aplikasi dan belum menguji generalisasi model pada aplikasi transportasi daring lain. Untuk penelitian selanjutnya, disarankan menerapkan teknik penyeimbangan data seperti SMOTE atau data augmentation, serta memperluas cakupan dataset agar model lebih robust dan generalis. Implikasi praktis dari penelitian ini adalah pengembang aplikasi dapat memanfaatkan sistem klasifikasi otomatis berbasis

machine learning untuk memetakan area masalah secara cepat dan akurat, sehingga strategi perbaikan layanan dapat dilakukan secara lebih terarah dan adaptif terhadap kebutuhan pengguna.

DAFTAR PUSTAKA

Adi, I. N., Putra, M., & Pramartha, C. (2025). *Optimasi Hyperparameter Algoritma Support Vector Machine dalam Klasifikasi Penyakit β -Thalassemia*. *3*, 283–294.

Aditiya, N., Setiaji, P., & Supriyono. (2025). Analisis Sentimen Kepuasan Masyarakat terhadap Aplikasi "INFO BMKG"menggunakan Naive Bayes, SVM, dan KNN. *Sistemasi: Jurnal Sistem Informasi*, *14*(3), 2540–9719. http://sistemasi.ftik.unisi.ac.id

Alfarobby, A. N., & Irawan, H. (2024). *Analisis Sentimen Kepuasan Konsumen Pengguna Transportasi Online Pada Ulasan Google Playstore Menggunakan Indobert Dan Topic Modeling (Studi kasus: Gojek dan Grab)* (Vol. 11, Issue 1).

Amalia, D. H., & Yustanti, W. (2021). Klasifikasi Buku Menggunakan Metode Support Vector Machine pada Digital Library. *Journal of Informatics and Computer Science (JINACS)*, *3*(01), 55–61. https://doi.org/10.26740/jinacs.v3n01.p55-61

Chamidy, T., & Informatika, M. (2025). *Application of SMOTE in Sentiment Analysis of MyXL User Reviews on Google Play Store*. *10*(1), 74–86.

Gitacahyani, A., Irma Purnamasari, A., & Ali, I. (2024). Klasifikasi Ulasan Aplikasi Linkedin Menggunakan Metode Naïve Bayes Classifier. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(1), 176–181. https://doi.org/10.36040/jati.v8i1.8310

Iqrom, M., Afdal, M., Novita, R., Rahmawita, M., & Khairil Ahsyar, T. (n.d.). *Sentiment analysis of Gojek, Grab, and Maxim applications using support vector machine algorithm analisis sentimen aplikasi Gojek, Grab, dan Maxim menggunakan algoritma support vector machine*. *10*(1), 2025.

Khairunnisa, S., Adiwijaya, A., & Faraby, S. Al. (2021). Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19). *Jurnal Media Informatika Budidarma*, *5*(2), 406. https://doi.org/10.30865/mib.v5i2.2835

Larasati, F. A., Ratnawati, D. E., & Hanggara, B. T. (2022). Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *6*(9), 4305–4313.

Meli et al. (2024). Implementasi analisis sentimen pada ulasan aplikasi Duolingo di Google Playstore menggunakan algoritma Naïve Bayes. *AITI: Jurnal Teknologi Informasi*, *21*(2), 298–311. https://doi.org/10.36040/jati.v8i1.8708

Nababan, A., & Hutagalung, M. Y. (2023). Hyperparameter Tuning Pada Model Stance Detection Menggunakan GridSearchCV. *Jurnal Sains Dan Teknologi*, *5*(1), 205–209. https://doi.org/10.55338/saintek.v5i1.1505

Prabowo, A. S., & Kurniadi, F. I. (2023). Analisis Perbandingan Kinerja Algoritma Klasifikasi dalam Mendeteksi Penyakit Jantung. *Jurnal SISKOM-KB (Sistem Komputer Dan Kecerdasan Buatan)*, *7*(1), 56–61. https://doi.org/10.47970/siskom-kb.v7i1.468

Putri, R. R., & Cahyono, N. (2024). *Publik Pemerintah Dki Jakarta Dengan Algoritma*. *8*(2), 2363–2371.

Radhi, T., Fitrah, M., & Nurdin, Y. (2021). 21428-72955-1-Pb. *KITEKTRO: Jurnal Komputer, Informasi Teknologi Dan Elektro*, *6*(2), 7–14.

Radiena, G., & Nugroho, A. (2023). Analisis Sentimen Berbasis Aspek Pada Ulasan Aplikasi Kai Access Menggunakan Metode Support Vector Machine. *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, *6*(1), 1–10. https://doi.org/10.37792/jukanti.v6i1.836

Ramadani, N. C., Tahyudin, I., & Shouni Barkah, A. (2024). Perbandingan Algoritma Support Vector Machine, Decision Tree, dan Logistic Regresion Pada Analisis Sentimen Ulasan Aplikasi Netflix. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *10*(2), 110–117. https://teknosi.fti.unand.ac.id/index.php/teknosi/article/view/2746

Septiani, D., & Isabela, I. (2023). Analisis Term Frequency Inverse Document Frequency (TF-IDF) Dalam Temu Kembali Informasi Pada Dokumen Teks. *SINTESIA: Jurnal Sistem Dan*

Teknologi Informasi Indonesia, *1*(2), 81–88.

Shalihat, B. (2023). *Implementasi Metode Rule-Based pada Proses Silabifikasi dalam Bahasa Aceh*. 1–68. https://repository.ar-raniry.ac.id/id/eprint/36191

Subagja, R. A., Widiastiwi, Y., & Chamidah, N. (2021). Klasifikasi Ulasan Aplikasi Jenius pada Google Play Store Menggunakan Algoritma Naive Bayes. *Informatik : Jurnal Ilmu Komputer*, *17*(3), 197. https://doi.org/10.52958/iftk.v17i3.3652

Vitalaya, N. A. R. (2024). *Perbandingan tipe sampling pada klasifikasi minat TIK bagi skripsi*.