Evaluasi Performa Algoritma Machine Learning untuk Klasifikasi Sentimen Ulasan Film IMDb

Akhiyar Waladi[1], Hasanatul Iftitah[2]*
Fakultas Sains dan Teknologi, Universitas Jambi[1,2]
**Penulis Korespondensi : akhiyar.waladi@unja.ac.id*

ABSTRACT

IMDb with over 83 million active users generates millions of movie reviews annually. Manually analyzing such large textual data is impractical, necessitating automated approaches for binary sentiment classification. This study tested three traditional machine learning algorithms, namely Logistic Regression, Naive Bayes, and Random Forest, on 50,000 English-language movie reviews with balanced distribution between positive and negative sentiments. Feature extraction was performed using CountVectorizer and TF-IDF after preprocessing steps including HTML tag removal, contraction expansion, and text normalization. Among all combinations tested, Logistic Regression with TF-IDF achieved the best results with 87.29% accuracy, 87.32% F1-score, and 0.9481 AUC-ROC. Performance consistency was confirmed through 5-fold cross-validation with low coefficient of variation across all folds. Feature importance analysis revealed that words like "excellent," "perfect," and "amazing" indicate positive sentiment, while "worst," "awful," and "terrible" mark negative sentiment. Naive Bayes excelled in speed with 11.5 seconds training time but lower accuracy, while Logistic Regression offered the best balance between accuracy and efficiency at 17.2 seconds. Beyond accuracy, Logistic Regression provides interpretable coefficients explaining which words drive predictions. The findings demonstrate that traditional algorithms under the evaluated configurations remain viable for sentiment analysis tasks that prioritize computational efficiency and interpretability.

Article History

Received : 31-12-2026
Revised : 24-07-2026
Accepted : 29-07-2026

Keywords

Sentiment Analysis;
IMDb;
TF-IDF;
Logistic Regression;
Feature Importance

ABSTRAK

IMDb dengan lebih dari 83 juta pengguna aktif menghasilkan jutaan ulasan film setiap tahun. Menganalisis data sebanyak ini secara manual tentu tidak praktis, sehingga diperlukan pendekatan otomatis untuk klasifikasi sentimen biner. Penelitian ini menguji tiga algoritma machine learning tradisional yaitu Logistic Regression, Naive Bayes, dan Random Forest pada 50.000 ulasan film berbahasa Inggris dengan distribusi seimbang antara sentimen positif dan negatif. Ekstraksi fitur dilakukan menggunakan CountVectorizer dan TF-IDF, setelah melalui tahap preprocessing berupa pembersihan tag HTML, ekspansi kontraksi, dan normalisasi teks. Dari seluruh kombinasi yang diuji, Logistic Regression dengan TF-IDF mencapai hasil terbaik dengan akurasi 87,29%, F1-score 87,32%, dan AUC-ROC 0,9481. Konsistensi performa dikonfirmasi melalui 5-fold cross-validation dengan coefficient of variation rendah di seluruh fold. Analisis feature importance mengungkap kata "excellent", "perfect", dan "amazing" sebagai penanda sentimen positif, sedangkan "worst", "awful", dan "terrible" menandai sentimen negatif. Naive Bayes unggul dalam kecepatan dengan waktu pelatihan 11,5 detik namun akurasi lebih rendah, sementara Logistic Regression menawarkan keseimbangan terbaik antara akurasi dan efisiensi dalam 17,2 detik. Selain akurasi, Logistic Regression juga menyediakan koefisien yang dapat diinterpretasi untuk menjelaskan kata mana yang mempengaruhi prediksi. Temuan ini menunjukkan bahwa algoritma tradisional pada konfigurasi yang dievaluasi tetap layak digunakan untuk tugas analisis sentimen yang mengutamakan efisiensi komputasi dan interpretabilitas.

PENDAHULUAN

Sebuah ulasan negatif viral dapat menurunkan pendapatan box office hingga jutaan dolar, sementara akumulasi ulasan positif mampu mengangkat film independen menjadi fenomena budaya. IMDb (Internet Movie Database) dengan lebih dari 83 juta pengguna aktif menjadi barometer utama opini publik terhadap industri perfilman global [1]. Tantangan utamanya adalah menganalisis jutaan ulasan secara manual untuk mengekstrak sentimen, yang tidak praktis baik dari segi waktu maupun biaya, sehingga diperlukan pendekatan otomatis yang akurat dan efisien.

Analisis sentimen, sebagai subdomain dari Natural Language Processing (NLP), telah berkembang pesat dalam dekade terakhir. Teknik ini memungkinkan ekstraksi otomatis opini dan emosi dari data tekstual, transformasi yang krusial mengingat bahwa 80% data enterprise berbentuk tidak terstruktur [2]. Dalam konteks data ulasan film, analisis sentimen dapat mengklasifikasikan ulasan menjadi kategori positif dan negatif, memberikan wawasan agregat yang berharga untuk pembuat film, distributor, dan penonton potensial. Pemanfaatan analisis sentimen pada ulasan film juga telah diterapkan sebagai salah satu komponen sistem rekomendasi yang dipadukan dengan collaborative filtering [3]. Penelitian menunjukkan bahwa analisis sentimen menggunakan NLP untuk mengidentifikasi dan mengekstrak opini yang diekspresikan melalui teks telah menjadi semakin penting dalam memahami emosi dan perilaku manusia [4].

Sementara perkembangan terkini dalam deep learning telah menghasilkan model dengan akurasi tinggi untuk berbagai tugas NLP, algoritma machine learning tradisional tetap memiliki relevansi signifikan dalam aplikasi praktis. Penelitian menunjukkan bahwa Logistic Regression dan Naive Bayes masih memberikan performa kompetitif untuk klasifikasi sentimen, dengan keunggulan tambahan dalam hal interpretabilitas dan efisiensi komputasi [5]. Naive Bayes juga telah diterapkan untuk mengklasifikasikan sentimen opini Twitter melalui tahapan preprocessing, seleksi fitur Term Frequency, cross-validation, dan evaluasi confusion matrix [6]. Koefisien dari variabel independen dalam Logistic Regression dapat diinterpretasikan sebagai log-odds ratios, mengindikasikan arah dan besarnya pengaruh setiap variabel terhadap hasil [7].

Pemilihan metode ekstraksi fitur memainkan peran krusial dalam performa model klasifikasi teks. TF-IDF (Term Frequency-Inverse Document Frequency) telah terbukti superior dibandingkan CountVectorizer dalam berbagai studi. Penelitian terbaru menunjukkan bahwa TF-IDF mencapai akurasi hingga 97,31% pada klasifikasi sentimen, mengungguli pendekatan berbasis count sederhana [8]. TF-IDF memberikan bobot yang lebih informatif dengan mempertimbangkan tidak hanya frekuensi kata dalam dokumen tetapi juga distribusinya di seluruh korpus, memungkinkan model untuk membedakan lebih baik antara kata-kata umum dan indikator sentimen spesifik [9].

Analisis feature importance telah menjadi komponen integral dalam penelitian sentimen modern. Pemahaman tentang fitur mana yang paling berpengaruh tidak hanya meningkatkan interpretabilitas model tetapi juga memberikan wawasan linguistik yang berharga. Setiap koefisien merepresentasikan perubahan dalam log odds dari hasil untuk perubahan satu unit dalam variabel prediktor, dengan koefisien positif mengindikasikan bahwa peningkatan dalam variabel prediktor meningkatkan log odds dari kelas positif [10]. Pendekatan ini memungkinkan identifikasi kata-kata kunci yang menjadi indikator kuat untuk setiap kategori sentimen.

Dataset IMDb telah menjadi benchmark standar dalam penelitian analisis sentimen. Dataset ini terdiri dari 50.000 ulasan film yang telah diberi label positif dan negatif, dengan distribusi yang seimbang antara kedua kelas [11]. Karakteristik ini menjadikannya ideal untuk evaluasi algoritma klasifikasi dua kelas, memungkinkan perbandingan yang adil tanpa perlu mengatasi masalah ketidakseimbangan kelas yang sering muncul dalam dataset dunia nyata.

Meskipun model berbasis deep learning dan transformer dilaporkan mampu mencapai performa tinggi dalam berbagai tugas analisis sentimen, model tersebut memiliki karakteristik representasi dan kebutuhan komputasi yang berbeda dari algoritma machine learning tradisional. Perbandingan angka performa dan waktu komputasi dari penelitian yang berbeda juga tidak selalu dapat dilakukan secara langsung karena adanya perbedaan pembagian dataset, preprocessing, perangkat komputasi, dan konfigurasi pelatihan. Oleh karena itu, penelitian ini membatasi eksperimen pada algoritma machine learning tradisional, sedangkan model modern digunakan sebagai konteks dalam pembahasan literatur dan tidak sebagai pembanding eksperimental langsung. Kesenjangan ini menjadi semakin relevan mengingat kebutuhan industri akan solusi yang tidak

hanya akurat tetapi juga dapat dijelaskan, efisien, dan mudah diintegrasikan ke dalam sistem yang ada. Penelitian menunjukkan bahwa untuk banyak aplikasi praktis, model tradisional yang dioptimasi dengan baik dapat memberikan keseimbangan yang sangat baik antara performa, interpretabilitas, dan efisiensi [12].

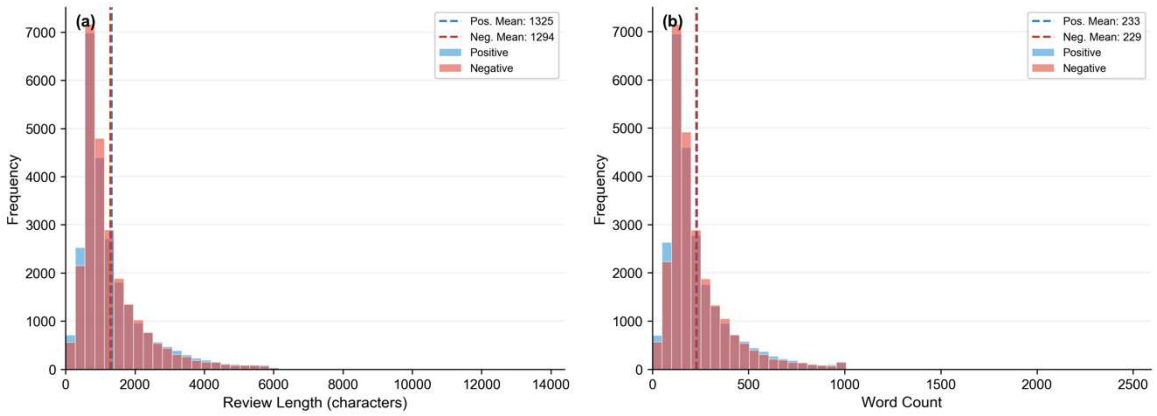
Penelitian ini bertujuan mengevaluasi performa Logistic Regression, Naive Bayes, dan Random Forest pada klasifikasi sentimen ulasan IMDb menggunakan CountVectorizer dan TF-IDF. Evaluasi difokuskan pada performa klasifikasi, stabilitas antar-fold, pola kesalahan, interpretabilitas fitur, dan efisiensi komputasi pada konfigurasi yang ditetapkan. Hasil penelitian dengan demikian digunakan untuk membandingkan model tradisional yang diuji dan tidak dimaksudkan untuk menyimpulkan keunggulannya terhadap model transformer atau performa maksimum yang dapat dicapai oleh setiap algoritma.

METODE

Penelitian ini mengadopsi pendekatan eksperimental dengan fokus pada analisis komparatif performa algoritma machine learning untuk klasifikasi sentimen. Metodologi dirancang untuk memastikan reproduktibilitas dan validitas hasil, mengikuti praktik penelitian machine learning.

Dataset dan Karakteristiknya

Dataset yang digunakan adalah IMDb Large Movie Review Dataset yang telah menjadi standar dalam penelitian analisis sentimen. Dataset ini berisi 50.000 ulasan film berbahasa Inggris dengan distribusi seimbang antara sentimen positif (25.000) dan negatif (25.000). Setiap ulasan memiliki panjang bervariasi antara 100 hingga 3.000 kata, merepresentasikan keragaman alami dalam ekspresi opini. Label sentimen ditentukan berdasarkan rating yang diberikan pengguna, di mana rating ≥7 dikategorikan sebagai positif dan rating ≤4 sebagai negatif, menghindari ambiguitas dari rating menengah [13].



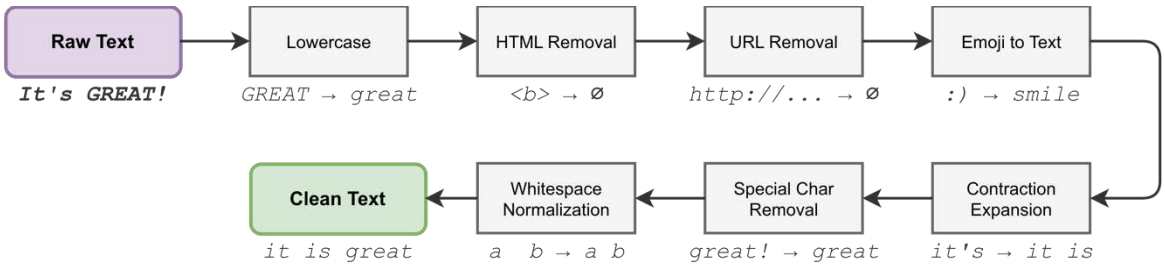
Gambar 1 Distribusi Panjang Ulasan Berdasarkan Sentimen

Analisis eksploratori awal menunjukkan adanya variasi yang besar dalam panjang ulasan. Berdasarkan Gambar 1, ulasan positif memiliki rata-rata panjang 1.325 karakter dan 233 kata, sedangkan ulasan negatif memiliki rata-rata panjang 1.294 karakter dan 229 kata. Distribusi kedua kelas sama-sama menceng ke kanan, dengan sebagian besar ulasan berada pada rentang pendek hingga menengah dan sejumlah kecil ulasan memiliki panjang yang jauh lebih besar. Gambar 1 memperlihatkan bahwa pola distribusi panjang ulasan pada kedua kelas relatif serupa.

ISSN: 2774-4345, DOI: 10.31284/j.kernel.2026.v7i1.8567

Preprocessing dan Pembersihan Teks

Tahap preprocessing merupakan komponen penting yang dapat memengaruhi performa model. Proses pembersihan yang komprehensif diimplementasikan mencakup beberapa langkah sistematis. Pertama, konversi seluruh teks ke huruf kecil untuk normalisasi. Kedua, penghapusan tag HTML menggunakan regular expression, mengatasi masalah umum dalam data yang diambil dari web. Ketiga, eliminasi URL dan mentions yang tidak relevan untuk analisis sentimen. Keempat, konversi emoji menjadi representasi tekstual menggunakan library emoji untuk mempertahankan informasi emosional. Alur tahapan preprocessing disajikan pada Gambar 2.



Gambar 2 Alur Preprocessing Teks: Pipeline pembersihan teks dari raw text hingga clean text melalui tujuh tahapan sistematis

Ekspansi kontraksi bahasa Inggris dilakukan untuk menangkap makna lengkap, misalnya "don't" menjadi "do not", "won't" menjadi "will not". Langkah ini penting karena partikel negasi seperti kata “not” dapat mengubah polaritas sentimen suatu kalimat. Selanjutnya, karakter khusus dan angka dihilangkan, kecuali huruf dan spasi, diikuti dengan normalisasi whitespace untuk menghilangkan spasi berlebih [14].

Ekstraksi Fitur

Setelah teks dibersihkan, tahap berikutnya adalah mengubah data teks menjadi representasi numerik yang dapat diproses oleh algoritma machine learning. Dua metode utama digunakan dalam penelitian ini, yaitu CountVectorizer (CV) dan Term Frequency–Inverse Document Frequency (TF-IDF). CountVectorizer merepresentasikan teks berdasarkan jumlah kemunculan setiap kata dalam dokumen, menghasilkan vektor sparse dengan dimensi sesuai ukuran vocabulary:

$$\boldsymbol{CV(t,d) = tf(t,d) = count(t,d)} \quad (1)$$

TF-IDF memberikan bobot lebih besar kepada kata-kata yang jarang muncul di seluruh korpus tetapi sering muncul dalam dokumen tertentu, sehingga lebih efektif dalam mengidentifikasi kata-kata yang diskriminatif antar kelas:

$$\boldsymbol{TF\text{-}IDF(t,d) = tf(t,d) \times idf(t), \quad idf(t) = \log \frac{N}{df(t)}} \quad (2)$$

Notasi tf(t,d) merepresentasikan frekuensi term t dalam dokumen d, N adalah jumlah total dokumen dalam korpus, dan df(t) adalah jumlah dokumen yang mengandung term t. Komponen IDF berfungsi sebagai faktor penalti untuk kata-kata yang terlalu umum, sehingga kata seperti “the” atau “is” yang muncul di hampir semua dokumen akan memiliki bobot yang sangat rendah. Perbandingan karakteristik CountVectorizer dan TF-IDF disajikan pada Tabel 1.

ISSN: 2774-4345, DOI: 10.31284/j.kernel.2026.v7i1.8567

Berdasarkan karakteristik tersebut, TF-IDF cenderung lebih efektif untuk klasifikasi teks karena memberikan bobot lebih tinggi pada kata-kata spesifik dan informatif. Konfigurasi parameter yang digunakan pada Tabel 2 dipilih untuk menyeimbangkan kekayaan representasi fitur dan efisiensi komputasi.

Tabel 1 Perbandingan Karakteristik CountVectorizer dan TF-IDF

Aspek	CountVectorizer	TF-IDF
Pembobotan	Frekuensi mentah (raw count)	Frekuensi × Inverse Document Frequency
Kata umum (stopwords)	Bobot tinggi jika frekuensi tinggi	Bobot rendah karena IDF kecil
Kata spesifik domain	Bobot frekuensi kemunculan	Bobot lebih tinggi karena IDF besar
Sensitivitas korpus	Tidak ada normalisasi dokumen	Ada normalisasi melalui komponen IDF
Kelebihan	Sederhana dan cepat	Lebih diskriminatif untuk kata informatif

Tabel 2 Konfigurasi Parameter Ekstraksi Fitur

Parameter	Nilai	Justifikasi
max_features	5000	Membatasi dimensi ruang fitur untuk efisiensi komputasi
min_df	5	Menghapus kata yang muncul di kurang dari 5 dokumen
ngram_range	(1, 1)	Menggunakan unigram untuk kesederhanaan model
lowercase	True	Normalisasi huruf kecil untuk konsistensi
stop_words	None	Stopwords dipertahankan untuk menjaga kata negasi dan kata fungsi yang dapat memengaruhi polaritas sentimen

Konfigurasi ini dipilih untuk menyeimbangkan antara kekayaan representasi fitur dan efisiensi komputasi. Parameter max_features membatasi dimensi ruang fitur menjadi 5000 kata paling frekuen untuk mencegah overfitting dan mengurangi kompleksitas model. Parameter min_df menghapus kata-kata yang muncul dalam kurang dari lima dokumen untuk mengurangi fitur yang sangat jarang muncul. Penggunaan unigram (ngram_range = (1, 1)) dipilih untuk mempertahankan kesederhanaan representasi fitur. Stopwords tidak dihapus karena tahap ekspansi kontraksi menghasilkan kata negasi seperti “not”, yang dapat mengubah polaritas sentimen suatu kalimat. Pada TF-IDF, kata yang sangat umum tetap cenderung memperoleh bobot relatif rendah melalui komponen inverse document frequency, sedangkan konfigurasi preprocessing yang sama diterapkan pada CountVectorizer untuk menjaga konsistensi pipeline. Keputusan ini merupakan konfigurasi metodologis penelitian dan bukan hasil perbandingan dengan stopword removal. Oleh karena itu, pengaruh khusus penghapusan atau pemertahanan stopwords terhadap akurasi belum dapat disimpulkan dalam penelitian ini.

Algoritma Klasifikasi

Tiga algoritma utama diimplementasikan dengan justifikasi spesifik untuk masing-masing. Logistic Regression dipilih karena interpretabilitas dan efektivitas yang telah terbukti untuk klasifikasi teks. Model dikonfigurasi dengan L2 regularization (C=100), solver='liblinear' untuk efisiensi dengan dataset berukuran menengah, dan max_iter=200 untuk memastikan konvergensi. Koefisien dari variabel independen dapat diinterpretasikan sebagai log-odds ratios, memberikan wawasan langsung tentang tingkat kepentingan fitur [15]. Multinomial Naive Bayes merupakan implementasi klasik untuk klasifikasi teks berdasarkan teorema Bayes dengan asumsi independensi antar fitur. Parameter smoothing (alpha=1.0) digunakan untuk menangani probabilitas nol. Meskipun sederhana, Naive Bayes sering memberikan performa yang kompetitif untuk tugas analisis sentimen. Random Forest digunakan sebagai model pembanding non-linear dengan n_estimators=100 dan

ISSN: 2774-4345, DOI: 10.31284/j.kernel.2026.v7i1.8567

max_depth=10. Nilai max_depth dibatasi untuk mengendalikan kompleksitas model dan waktu pelatihan pada lingkungan Google Colaboratory yang digunakan.

Konfigurasi tersebut tidak dimaksudkan untuk merepresentasikan performa maksimum Random Forest karena penelitian ini tidak melakukan pencarian hyperparameter secara menyeluruh. Oleh karena itu, hasil Random Forest harus diinterpretasikan sebagai performa pada konfigurasi terbatas yang diuji dan bukan sebagai kesimpulan umum bahwa algoritma tersebut lebih rendah daripada model lainnya.

Evaluasi dan Metrik

Proses evaluasi dilakukan dengan membagi data menjadi 80% untuk pelatihan dan 20% untuk pengujian menggunakan stratified sampling sehingga proporsi kelas positif dan negatif tetap seimbang pada setiap subset data. Hal ini penting untuk memastikan bahwa model tidak bias terhadap salah satu kelas selama proses pelatihan maupun pengujian.

Beberapa metrik digunakan untuk mengukur kinerja model secara komprehensif. Metrik pertama adalah akurasi, yang menghitung proporsi prediksi benar terhadap total prediksi model:

$$Accuracy = \frac{TP + TN}{N}, \quad N = TP + TN + FP + FN \tag{3}$$

Nilai TP (true positive) dan TN (true negative) masing-masing adalah jumlah prediksi benar untuk kelas positif dan negatif, sedangkan FP (false positive) dan FN (false negative) adalah jumlah kesalahan prediksi. Precision mengukur ketepatan prediksi positif yang dihasilkan model, sedangkan recall mengukur kemampuan model dalam mendeteksi seluruh data positif. Untuk menyeimbangkan kedua metrik tersebut digunakan F1-Score:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}, \quad P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN} \tag{4}$$

Analisis Feature Importance

Analisis feature importance dilakukan khusus untuk model Logistic Regression dengan mengamati nilai absolut dari koefisien βi. Koefisien positif menunjukkan bahwa kenaikan nilai fitur meningkatkan probabilitas ulasan positif, sedangkan koefisien negatif mengindikasikan kontribusi terhadap sentimen negatif. Besarnya nilai absolut |βi| mencerminkan tingkat kepentingan fitur terhadap prediksi model, sehingga fitur dengan |βi| paling besar dianggap memiliki pengaruh terbesar terhadap hasil klasifikasi.

Implementasi Teknis

Seluruh eksperimen diimplementasikan menggunakan Python 3.8 dengan pustaka scikit-learn versi 1.3.0 sebagai kerangka kerja utama. Platform komputasi yang digunakan adalah Google Colaboratory dengan spesifikasi RAM 12,7 GB dan ruang penyimpanan 78,2 GB. Untuk memastikan konsistensi proses pelatihan dan evaluasi, seluruh pipeline dari pra-pemrosesan hingga pelatihan model diintegrasikan menggunakan scikit-learn pipeline agar tidak terjadi data leakage. Seluruh eksperimen menggunakan random seed bernilai 42 untuk menjamin bahwa hasil yang diperoleh dapat direproduksi di bawah kondisi yang identik.

HASIL DAN PEMBAHASAN

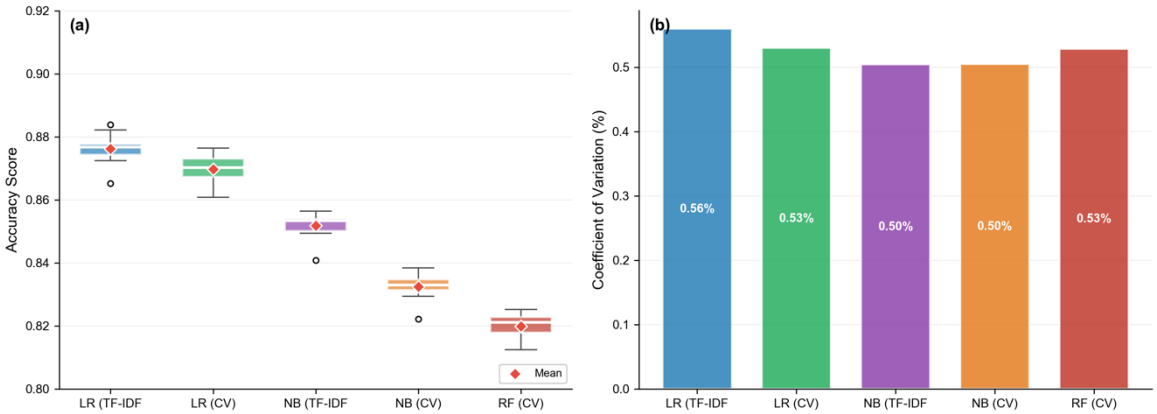
Performa Komparatif Model

Tabel 3 menyajikan hasil evaluasi kelima kombinasi algoritma dan metode ekstraksi fitur yang diuji. Logistic Regression dengan TF-IDF mencapai performa terbaik dengan akurasi 87,29%, precision 87,45%, recall 87,50%, dan F1-score 87,32%, mengungguli semua model lain dalam hampir semua metrik. Logistic Regression dengan CountVectorizer menunjukkan performa yang sedikit lebih rendah dengan akurasi 86,68% dan F1-score 86,68%, mengindikasikan bahwa pemilihan metode ekstraksi fitur memiliki dampak moderat pada algoritma ini. Perbedaan performa sebesar 0,61% antara kedua varian Logistic Regression mendemonstrasikan nilai tambah dari pembobotan inverse document frequency dalam membedakan kata informatif dari kata umum.

Tabel 3 Perbandingan Performa Model

Model	Accuracy	Precision	Recall	F1-score	AUC-ROC	Training Time (s)
Logistic Regression (TF-IDF)	0,8729	0,8745	0,875	0,8732	0,9481	17,2
Logistic Regression (CV)	0,8668	0,8669	0,867	0,8668	0,9354	44,3
Naive Bayes (TF-IDF)	0,8527	0,8586	0,8578	0,8534	0,9283	12,6
Naive Bayes (CV)	0,8332	0,8451	0,8188	0,8308	0,904	11,5
Random Forest (CV)	0,825	0,7949	0,877	0,8337	0,9095	19

Hasil ini sejalan dengan penelitian sebelumnya yang menunjukkan bahwa TF-IDF dapat memberikan pembobotan yang lebih diskriminatif dibandingkan pendekatan berbasis frekuensi mentah [16]. Untuk memvalidasi konsistensi performa model, dilakukan analisis cross-validation stability yang menunjukkan variabilitas skor antar fold. Distribusi skor cross-validation dan coefficient of variation untuk setiap model disajikan pada Gambar 3.



Gambar 3 Cross-Validation Stability: Distribusi Skor dan Coefficient of Variation Antar Model.

Analisis stabilitas menunjukkan bahwa seluruh model memiliki coefficient of variation yang rendah, yaitu antara 0,50% dan 0,56%. Naive Bayes dengan TF-IDF dan CountVectorizer memiliki coefficient of variation terendah sebesar 0,50%, sedangkan Logistic Regression dengan TF-IDF memperoleh nilai 0,56%. Dengan demikian, keunggulan Logistic Regression dengan TF-IDF terletak pada rata-rata performanya yang lebih tinggi, bukan pada coefficient of variation yang paling rendah. Naive Bayes menunjukkan performa yang kompetitif dengan waktu pelatihan paling singkat.

Dengan waktu pelatihan 11,5 detik untuk varian CountVectorizer, Naive Bayes menunjukkan efisiensi pelatihan yang baik dalam lingkungan eksperimen ini.

Namun, asumsi independensi yang kuat pada Naive Bayes terefleksi dalam performa yang lebih rendah dibandingkan Logistic Regression, terutama dalam merepresentasikan dependensi antarkata. Pada konfigurasi yang diuji, Random Forest memperoleh akurasi terendah sebesar 82,50%. Hasil tersebut kemungkinan turut dipengaruhi oleh pembatasan max_depth=10 dan n_estimators=100 yang mengendalikan kompleksitas model. Dengan demikian, hasil ini hanya menunjukkan bahwa model linear lebih unggul pada konfigurasi eksperimen penelitian ini dan tidak membuktikan bahwa Random Forest secara umum lebih lemah untuk klasifikasi sentimen.

Analisis Confusion Matrix

Tabel 4 menampilkan confusion matrix untuk kelima model yang memberikan wawasan terperinci mengenai pola klasifikasi masing-masing algoritma. Logistic Regression dengan TF-IDF menunjukkan distribusi kesalahan yang relatif seimbang dengan 4.354 true negative, 4.375 true positive, 646 false positive, dan 625 false negative, menghasilkan true positive rate 87,50% dan true negative rate 87,08%. Logistic Regression dengan CountVectorizer menunjukkan pola serupa dengan sedikit penurunan performa, menghasilkan 4333 true negatives dan 4335 true positives, dengan false positives (667) dan false negatives (665) yang hampir identik, menunjukkan tidak ada bias sistematis terhadap salah satu kelas.

Tabel 4 Perbandingan Confusion Matrix Seluruh Model

Model	TN	FP	FN	TP	TPR	TNR	Error Type Balance
Logistic Regression (TF-IDF)	4354	646	625	4375	87,50%	87,08%	0,968
Logistic Regression (CV)	4333	667	665	4335	86,70%	86,66%	0,996
Naive Bayes (TF-IDF)	4238	762	711	4289	85,78%	84,76%	0,932
Naive Bayes (CV)	4238	762	906	4094	81,88%	84,76%	0,841
Random Forest (CV)	3865	1135	615	4385	87,70%	77,30%	0,542

Logistic Regression dengan CountVectorizer memiliki error type balance yang paling mendekati 1, yaitu 0,996, karena jumlah false positive dan false negative hampir sama. Logistic Regression dengan TF-IDF juga menunjukkan pola kesalahan yang relatif seimbang dengan error type balance sebesar 0,968 serta true positive rate 87,50% dan true negative rate 87,08%. Meskipun Logistic Regression dengan CountVectorizer memiliki keseimbangan jenis kesalahan yang sedikit lebih baik, Logistic Regression dengan TF-IDF tetap memperoleh performa keseluruhan tertinggi berdasarkan accuracy, F1-score, dan AUC-ROC.

Naive Bayes dengan CountVectorizer menghasilkan false negative sebanyak 906 dan false positive sebanyak 762, dengan error type balance sebesar 0,841. Pola ini menunjukkan bahwa model lebih sering mengklasifikasikan ulasan positif sebagai negatif dibandingkan pola kesalahan sebaliknya. Namun, ketimpangan tersebut bukan yang terbesar karena Random Forest memiliki error type balance yang lebih rendah, yaitu 0,542.

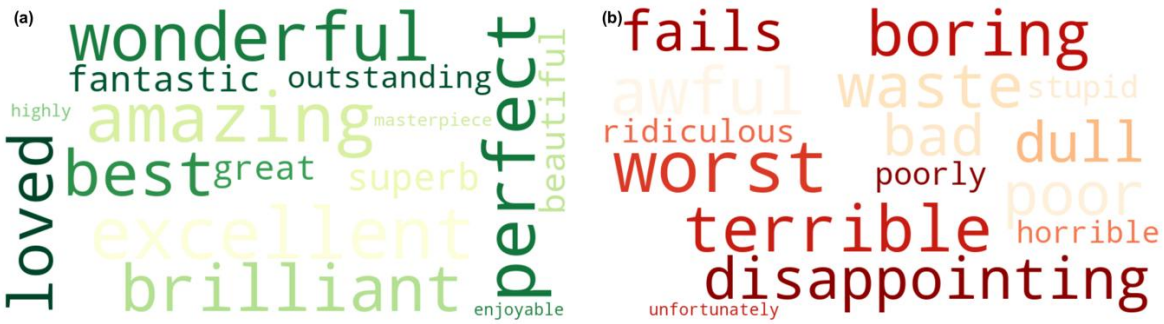
Random Forest menunjukkan pola kesalahan yang paling tidak seimbang, dengan false positive sebanyak 1.135 dan false negative sebanyak 615. Error type balance sebesar 0,542 menunjukkan bahwa model lebih sering mengklasifikasikan ulasan negatif sebagai positif. True negative rate sebesar 77,30% juga lebih rendah dibandingkan model lainnya, sehingga kesalahan utama Random Forest pada konfigurasi ini terjadi ketika mengidentifikasi ulasan negatif.

Analisis Feature Importance

Tabel 5 menyajikan 15 indikator sentimen positif dan negatif berdasarkan analisis koefisien dari model Logistic Regression dengan CountVectorizer. Kata "worst" muncul sebagai indikator negatif terkuat dengan koefisien -2,9875, diikuti oleh "awful" (-2,4521) dan "terrible" (-2,3987), membentuk kelompok semantik yang jelas untuk ekspresi ketidakpuasan ekstrem. Kata-kata seperti "boring" (-2,2143) dan "dull" (-1,8765) merepresentasikan kritik terhadap kurangnya keterlibatan, sementara "waste" (-2,1876) dan "disappointing" (-1,9876) mengekspresikan kekecewaan terhadap ekspektasi yang tidak terpenuhi.

Tabel 5 Lima Belas Indikator Sentimen Positif dan Negatif Teratas.

Positive Word	Coefficient (+)	Interpretation (+)	Negative Word	Coefficient (-)	Interpretation (-)
excellent	2,8234	Pujian kualitas tertinggi	worst	-2,9875	Indikator negatif terkuat
perfect	2,6543	Positif absolut	awful	-2,4521	Emosi negatif kuat
amazing	2,5876	Kekaguman dan takjub	terrible	-2,3987	Penilaian negatif langsung
wonderful	2,4321	Emosi positif	boring	-2,2143	Kurangnya keterlibatan
brilliant	2,3765	Pujian kecerdasan	waste	-2,1876	Kekecewaan sumber daya
best	2,2987	Perbandingan superlatif	poor	-2,0234	Kritik kualitas
loved	2,1234	Koneksi emosional	disappointing	-1,9876	Ketidaksesuaian ekspektasi
superb	2,0876	Kualitas tinggi	bad	-1,9543	Negatif sederhana
fantastic	1,9876	Antusiasme	dull	-1,8765	Kurangnya kegembiraan
beautiful	1,8765	Apresiasi estetika	fails	-1,8234	Kegagalan performa
great	1,8234	Positif umum	ridiculous	-1,7654	Kritik absurditas
outstanding	1,7654	Kualitas luar biasa	stupid	-1,7234	Kritik kecerdasan
masterpiece	1,7234	Pencapaian artistik	poorly	-1,6987	Kritik eksekusi
highly	1,6543	Penguat (konteks positif)	horrible	-1,6543	Negatif kuat
enjoyable	1,5432	Nilai hiburan	unfortunately	-1,5234	Penanda kekecewaan



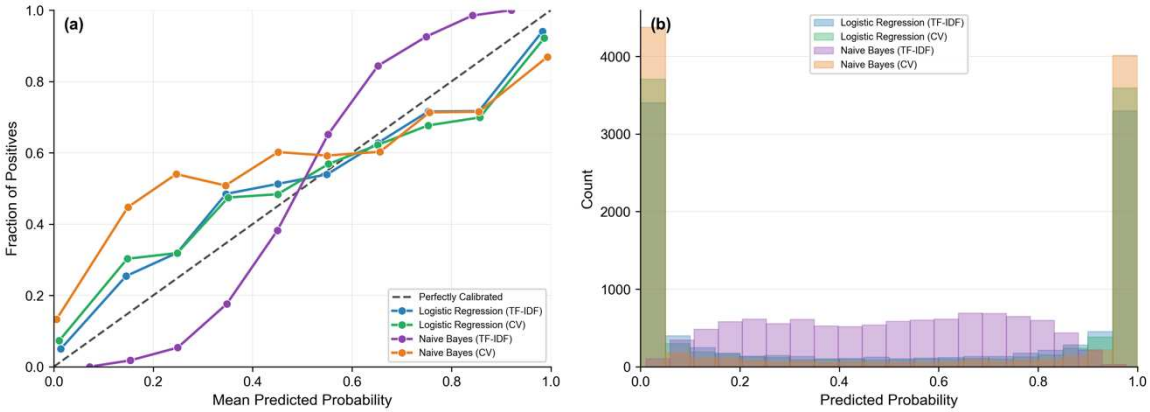
Gambar 4 Word Cloud Feature Importance: (a) Indikator Sentimen Positif, (b) Indikator Sentimen Negatif

Di sisi positif, "excellent" memimpin dengan koefisien 2,8234, diikuti "perfect" (2,6543) dan "amazing" (2,5876), membentuk tiga ekspresi pujian tertinggi. Kata-kata emosional seperti "loved" (2,1234) dan "wonderful" (2,4321) menunjukkan pentingnya koneksi emosional dalam ulasan positif, sementara term teknis seperti "masterpiece" (1,7234) dan "brilliant" (2,3765) mengindikasikan apresiasi terhadap aspek artistik.

Kehadiran kata "highly" (1,6543) sebagai indikator positif menarik karena biasanya berfungsi sebagai penguat dalam konteks positif seperti "highly recommended" atau "highly entertaining", sementara "unfortunately" (-1,5234) meskipun bukan secara langsung bermakna negatif, sering menandakan perubahan nada menuju kritik dalam struktur naratif ulasan. Visualisasi word cloud untuk indikator sentimen positif dan negatif disajikan pada Gambar 4, memberikan representasi visual yang intuitif dari pola feature importance.

Analisis Reliabilitas Model dengan Calibration Curve

Untuk memahami seberapa andal probabilitas prediksi yang dihasilkan oleh model, dilakukan analisis calibration curve (reliability diagram). Gambar 5 menyajikan perbandingan calibration curve antar model beserta distribusi probabilitas prediksi.



Gambar 5 Calibration Curve (Reliability Diagram): Perbandingan Reliabilitas Probabilitas Prediksi Antar Model

Model yang terkalibrasi dengan baik akan menghasilkan kurva yang relatif dekat dengan garis diagonal. Berdasarkan Gambar 5, kurva Logistic Regression dengan TF-IDF dan CountVectorizer secara umum lebih dekat dengan garis diagonal dibandingkan kedua varian Naive Bayes pada beberapa rentang probabilitas. Meskipun demikian, kurva Logistic Regression tetap menunjukkan penyimpangan pada sejumlah interval, sehingga model lebih tepat disebut memiliki kalibrasi yang relatif lebih baik dan bukan terkalibrasi secara sempurna.

Kedua varian Naive Bayes menunjukkan penyimpangan yang lebih besar dari garis diagonal pada sejumlah rentang probabilitas. Pola penyimpangan antara Naive Bayes dengan TF-IDF dan CountVectorizer juga tidak sepenuhnya sama, sehingga keduanya tidak dapat dirangkum sebagai satu pola overconfidence yang identik. Hasil ini menunjukkan bahwa probabilitas keluaran Naive Bayes perlu digunakan dengan lebih hati-hati apabila aplikasi memerlukan nilai keyakinan prediksi, sedangkan perbedaan tersebut tidak secara langsung mengubah hasil klasifikasi biner pada threshold yang digunakan.

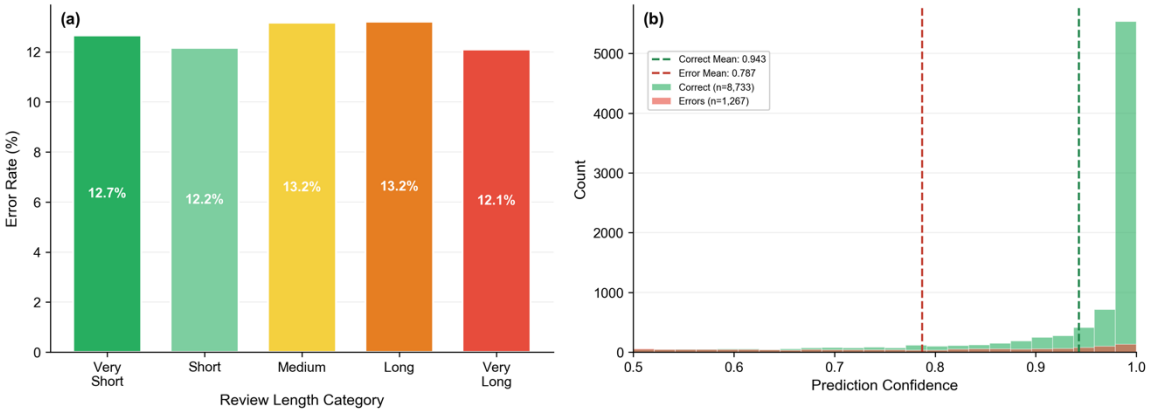
Analisis Kesalahan

Analisis terhadap 1.271 ulasan yang salah diklasifikasi, atau 12,71% dari data uji, mengungkapkan beberapa pola kesalahan sebagaimana ditunjukkan pada Gambar 6. Visualisasi ini

menyajikan karakteristik kesalahan model meliputi tingkat kesalahan berdasarkan panjang ulasan dan distribusi tingkat keyakinan.

False negatives sering terjadi pada ulasan menggunakan sarkasme atau ironi, di mana kata-kata negatif permukaan digunakan untuk mengekspresikan sentimen positif. Rata-rata panjang ulasan yang salah diklasifikasi adalah 1.587 karakter, lebih panjang dibandingkan rata-rata ulasan yang diklasifikasi dengan benar sebesar 1.289 karakter. Perbedaan deskriptif ini mengindikasikan bahwa ulasan yang lebih panjang dan memiliki konteks lebih kompleks cenderung menjadi tantangan bagi pendekatan bag-of-words.

False positives sering terjadi pada ulasan yang diawali dengan mengakui elemen positif sebelum mengekspresikan kekecewaan keseluruhan. Frasa seperti "had potential", "could have been great", atau "promising start" mengandung kata-kata positif tetapi dalam konteks ekspektasi yang tidak terpenuhi. Model kesulitan dengan ekspresi kondisional dan pengandaian yang memerlukan pemahaman makna yang lebih dalam. Contoh ulasan yang salah diklasifikasi meliputi penggunaan ironi/sarkasme seperti "This movie is so bad it's actually good...", sentimen campuran seperti "Had potential but ultimately disappointing", kompleksitas negasi seperti "Not the worst I've seen but far from good", aspek kontras seperti "Brilliant concept, terrible execution", dan negasi ganda seperti "Surprisingly not terrible".



Gambar 6 Error Analysis: Karakteristik Ulasan yang Salah Diklasifikasi

Pertimbangan Efisiensi Komputasi

Analisis trade-off antara akurasi dan efisiensi komputasi menunjukkan variasi antar model sebagaimana ditunjukkan pada Tabel 6. Naive Bayes dengan CountVectorizer mencapai rasio accuracy/time tertinggi sebesar 7,24 dengan waktu pelatihan tercepat, yaitu 11,5 detik. Dalam batas lingkungan eksperimen ini, hasil tersebut menunjukkan bahwa Naive Bayes dengan CountVectorizer merupakan opsi paling efisien ketika waktu pelatihan menjadi pertimbangan utama. Naive Bayes (TF-IDF) menawarkan keseimbangan lebih baik dengan akurasi 85,27% dan rasio 6,77 dalam 12,6 detik, memberikan peningkatan performa dengan biaya komputasi minimal.

Tabel 6 Analisis Trade-off Efisiensi-Performa

Model	Accuracy	Time (s)	Accuracy/Time Ratio
Naive Bayes (CV)	83,32%	11,5	7,24
Naive Bayes (TF-IDF)	85,27%	12,6	6,77
Logistic Regression (TF-IDF)	87,29%	17,2	5,07
Random Forest (CV)	82,50%	19	4,34
Logistic Regression (CV)	86,68%	44,3	1,96

Logistic Regression dengan TF-IDF menunjukkan akurasi tertinggi sebesar 87,29% dengan waktu pelatihan 17,2 detik dan rasio accuracy/time sebesar 5,07. Dalam eksperimen ini, konfigurasi tersebut memberikan keseimbangan terbaik antara performa klasifikasi dan waktu pelatihan. Logistic Regression dengan CountVectorizer memerlukan waktu paling lama, yaitu 44,3 detik, yang memperlihatkan bahwa representasi fitur turut memengaruhi efisiensi pelatihan. Random Forest dengan CountVectorizer memperoleh rasio 4,34 dan akurasi 82,50%; hasil tersebut perlu ditafsirkan bersama dengan pembatasan max_depth dan jumlah estimator yang digunakan.

Perbandingan dengan Literatur Terdahulu

Hasil penelitian ini berada dalam rentang performa yang dilaporkan pada sejumlah penelitian terdahulu mengenai algoritma klasifikasi sentimen tradisional. Logistic Regression dengan TF-IDF mencapai akurasi 87,29%, sedangkan Naive Bayes mencapai 85,27% dengan TF-IDF dan 83,32% dengan fitur dari CountVectorizer. Model berbasis deep learning dan transformer dilaporkan dapat menghasilkan performa yang lebih tinggi pada sejumlah penelitian [17]. Namun, hasil tersebut tidak dapat dibandingkan secara langsung dengan waktu pelatihan penelitian ini karena terdapat perbedaan pada pembagian data, preprocessing, perangkat komputasi, arsitektur, dan konfigurasi pelatihan. Oleh karena itu, penelitian ini tidak menghitung secara langsung selisih accuracy dan compute time antara model tradisional dan model transformer [18].

Random Forest memperoleh akurasi 82,50% pada konfigurasi n_estimators=100 dan max_depth=10. Pembatasan kedalaman pohon digunakan untuk menjaga kompleksitas dan waktu komputasi, tetapi juga dapat membatasi kemampuan model dalam membentuk pola keputusan yang lebih kompleks. Karena penelitian ini tidak melakukan pencarian hyperparameter secara menyeluruh, hasil Random Forest hanya merepresentasikan konfigurasi yang diuji dan tidak dapat dianggap sebagai performa maksimum algoritma tersebut. Pada Logistic Regression dan Naive Bayes, TF-IDF mengungguli CountVectorizer dengan peningkatan akurasi sebesar 0,61–1,95 poin persentase. Hasil ini sejalan dengan studi yang melaporkan keunggulan TF-IDF sebesar 1–3 poin persentase pada korpus dengan vocabulary yang besar [19].

Kontribusi penelitian ini terletak pada analisis trade-off antara performa, interpretabilitas, dan efisiensi komputasi pada konfigurasi yang diuji yang sering diabaikan dalam literatur yang fokus pada peningkatan akurasi. Kemampuan untuk mengidentifikasi feature importance melalui koefisien Logistic Regression memberikan transparansi yang krusial untuk aplikasi industri yang memerlukan explainability, sebuah aspek yang masih menjadi tantangan untuk model deep learning [20].

Implikasi dan Rekomendasi

Hasil analisis menunjukkan beberapa implikasi penting untuk implementasi praktis. TF-IDF secara konsisten mengungguli CountVectorizer dengan peningkatan akurasi 0,61% untuk Logistic Regression dan 1,95% untuk Naive Bayes. Pada konfigurasi yang diuji, Logistic Regression menunjukkan performa paling kuat dengan akurasi tertinggi sebesar 87,29% dan hasil yang relatif konsisten pada kedua metode ekstraksi fitur. Temuan tersebut menjadikan Logistic Regression sebagai kandidat yang relevan ketika akurasi dan interpretabilitas menjadi pertimbangan utama. Namun, kelayakannya untuk lingkungan produksi tetap memerlukan evaluasi tambahan pada data dan infrastruktur penerapan yang sebenarnya [21]. Analisis feature importance mengungkapkan pola linguistik intuitif, dengan kata-kata seperti "excellent", "perfect", "worst", dan "awful" sebagai indikator sentimen terkuat.

Secara keseluruhan, error rate kelima model berkisar antara 12,71% dan 17,50%. Analisis kualitatif pada Logistic Regression dengan TF-IDF menunjukkan bahwa sarkasme, ironi, dan sentimen campuran merupakan pola yang menantang, sehingga pendekatan berbasis contextual

embeddings dapat dipertimbangkan [22]. Naive Bayes menunjukkan waktu pelatihan paling singkat, yaitu 11,5 detik, sedangkan Logistic Regression dengan TF-IDF memberikan performa klasifikasi tertinggi dengan waktu pelatihan 17,2 detik. Namun, penelitian ini belum mengevaluasi inference latency secara khusus, sehingga kesesuaiannya untuk real-time processing atau batch processing belum dapat disimpulkan secara langsung.

KESIMPULAN

Penelitian ini melakukan analisis komparatif terhadap algoritma machine learning tradisional untuk klasifikasi sentimen pada 50.000 ulasan IMDb. Logistic Regression dengan TF-IDF mencapai performa terbaik dengan akurasi 87,29% dan F1-score 87,32%. Analisis feature importance mengungkap pola linguistik yang konsisten, di mana kata seperti "excellent" dan "perfect" menandai sentimen positif, sedangkan "worst" dan "awful" mengindikasikan sentimen negatif, memvalidasi intuisi manusia sekaligus meningkatkan transparansi model.

Evaluasi efisiensi komputasi menunjukkan trade-off antara akurasi dan kecepatan. Naive Bayes menawarkan waktu pelatihan tercepat (11,5 detik) dengan akurasi 83,32%, sementara Logistic Regression memberikan keseimbangan optimal antara akurasi tinggi dan waktu moderat (17,2 detik). Analisis kesalahan mengidentifikasi tantangan dalam menangani sarkasme dan sentimen campuran dengan tingkat kesalahan 12,71%, di mana ulasan yang salah diklasifikasi cenderung lebih panjang, menunjukkan keterbatasan pendekatan bag-of-words.

Analisis calibration curve menunjukkan bahwa Logistic Regression memiliki pola probabilitas prediksi yang relatif lebih dekat dengan kondisi aktual dibandingkan Naive Bayes pada beberapa rentang probabilitas. Meskipun demikian, hasil penelitian dibatasi oleh penggunaan klasifikasi biner, satu dataset berbahasa Inggris, dan representasi bag-of-words yang belum menangkap hubungan kontekstual antarkata secara menyeluruh. Penelitian ini juga tidak menyertakan baseline transformer secara langsung, tidak melakukan ablation study terhadap penghapusan stopwords, dan menggunakan konfigurasi Random Forest yang dibatasi untuk efisiensi komputasi. Oleh karena itu, kesimpulan penelitian hanya berlaku pada model dan konfigurasi yang dievaluasi. Penelitian selanjutnya dapat membandingkan model tradisional dengan transformer dalam lingkungan eksperimen yang sama, mengevaluasi pengaruh stopword removal, melakukan optimasi hyperparameter yang lebih luas, dan mengembangkan teknik untuk menangani sarkasme serta sentimen campuran.

DAFTAR PUSTAKA

[1] P. Atandoh, F. Zhang, M. A. Al-antari, D. Addo, and Y. Hyeon Gu, “Scalable deep learning framework for sentiment analysis prediction for online movie reviews,” *Heliyon*, vol. 10, no. 10, p. e30756, 2024, doi: 10.1016/j.heliyon.2024.e30756.

[2] M. Bordoloi and S. K. Biswas, “Sentiment analysis: A survey on design framework, applications and future scopes,” *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 12505–12560, 2023, doi: 10.1007/s10462-023-10442-2.

[3] M. A. Fikry, S. R. Wardhana, and R. K. Hapsari, “Sistem Rekomendasi Film Dengan Menggunakan Sentiment Analysis dan Collaborative Filtering,” *KERNEL: Jurnal Riset Inovasi Bidang Informatika dan Pendidikan Informatika*, vol. 5, no. 2, pp. 118–124, 2024, doi: 10.31284/j.kernel.2024.v5i2.7635.

[4] A. Mabrouk, R. P. D. Redondo, and M. Kayed, “Deep Learning-Based Sentiment Classification: A Comparative Survey,” *IEEE Access*, vol. 8, pp. 85616–85638, 2020, doi: 10.1109/ACCESS.2020.2992013.

[5] S. A. H. Bahtiar, C. K. Dewa, and A. Luthfi, “Comparison of Naïve Bayes and logistic regression in sentiment analysis on marketplace reviews using rating-based labeling,” *Journal of Information Systems and Informatics*, vol. 5, no. 3, pp. 915–927, 2023, doi: 10.51519/journalisi.v5i3.539.

[6] K. Sulastri, "Klasifikasi Naïve Bayes pada analisis sentimen atas penolakan dibukanya larangan ekspor benih lobster," *KERNEL: Jurnal Riset Inovasi Bidang Informatika dan Pendidikan Informatika*, vol. 1, no. 2, pp. 68–75, 2020, doi: 10.31284/j.kernel.2020.v1i2.1501.

[7] S. Sperandei, "Understanding logistic regression analysis," *Biochem. Med. (Zagreb).*, vol. 24, no. 1, pp. 12–18, 2014, doi: 10.11613/BM.2014.003.

[8] U. Ayman, A. R. Khan, M. H. Mahi, T. Akter, Z. M. Koli, and Md. H. I. Bijoy, "Optimizing Bengali Sentiment Analysis: A Comparison of Countvectorizer and TF-IDF with Machine Learning," in *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2025, pp. 1–6. doi: 10.1109/ECCE64574.2025.11013476.

[9] Y. Paul, "Twitter sentiment analysis using TF-IDF and machine learning classifiers," *Int. J. on Recent & Innovation Trends in Computing & Communication*, vol. 11, no. 3, pp. 693–701, 2023, doi: 10.17762/ijritcc.v11i3.11515.

[10] K. L. Tan, C. P. Lee, and K. M. Lim, "A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research," *Applied Sciences*, vol. 13, no. 7, 2023, doi: 10.3390/app13074550.

[11] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, Jul. 2018, doi: 10.1002/widm.1253.

[12] N. A. Semary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, "Enhancing machine learning-based sentiment analysis through feature extraction techniques," *PLoS One*, vol. 19, no. 2, pp. e0294968-, Feb. 2024, doi: 10.1371/journal.pone.0294968.

[13] L. Ashbaugh and Y. Zhang, "A Comparative Study of Sentiment Analysis on Customer Reviews Using Machine Learning and Deep Learning," *Computers*, vol. 13, no. 12, 2024, doi: 10.3390/computers13120340.

[14] B. Kabra and C. Nagar, "Attention-Emotion-Embedding BiLSTM-GRU network based sentiment analysis," *Journal of Integrated Science and Technology*, vol. 11, no. 4, p. 563, 2023, Accessed: Jul. 24, 2026. [Online]. Available: https://thesciencein.org/direct/index.php/jist/article/view/a563

[15] M. Arief and N. A. Samsudin, "Hybrid Approach with VADER and Multinomial Logistic Regression for Multiclass Sentiment Analysis in Online Customer Review," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 12, pp. 232–241, 2023, doi: 10.14569/IJACSA.2023.0141232.

[16] N. S. Wardana, F. P. Aditiawan, and A. P. Sari, "Logistic Regression Classification with TF-IDF and FastText for Sentiment Analysis of LinkedIn Reviews," *VISA: Journal of Vision and Ideas*, vol. 4, no. 3, pp. 1359 –, Aug. 2024, doi: 10.47467/visa.v4i3.2835.

[17] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment Analysis Is a Big Suitcase," *IEEE Intell. Syst.*, vol. 32, no. 6, pp. 74–80, 2017, doi: 10.1109/MIS.2017.4531228.

[18] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4335–4385, 2020, doi: 10.1007/s10462-019-09794-5.

[19] S. Seo, C. Kim, H. Kim, K. Mo, and P. Kang, "Comparative Study of Deep Learning-Based Sentiment Classification," *IEEE Access*, vol. 8, pp. 6861–6875, 2020, doi: 10.1109/ACCESS.2019.2963426.

[20] Y. Su and Y. Shen, "A Deep Learning-Based Sentiment Classification Model for Real Online Consumption," *Front. Psychol.*, vol. 13, 2022, doi: 10.3389/fpsyg.2022.886982.

[21] N. C. Dang, M. N. Moreno-García, and F. la Prieta, "Sentiment Analysis Based on Deep Learning: A Comparative Study," *Electronics (Basel).*, vol. 9, no. 3, 2020, doi: 10.3390/electronics9030483.

[22] N. A. Helal, A. Hassan, N. L. Badr, and Y. M. Afify, "A contextual-based approach for sarcasm detection," *Sci. Rep.*, vol. 14, no. 1, p. 15415, 2024, doi: 10.1038/s41598-024-65217-8.