

Analisis Sentimen Ulasan Aplikasi Detik.Com di Google Play Store Menggunakan Pendekatan *Lexicon-Based* dan *Machine Learning*

Talcha Ilham Putri^{1*}, Riski Annisa², Muhammad Fahmi Julianto³

^{1,2,3}Fakultas Teknik dan Informatika, Program Studi Informatika Kampus Kota Pontianak, Universitas Bina Sarana Informatika, Pontianak, Indonesia

Email: ¹15220788@bsi.ac.id, ²riski.rnc@bsi.ac.id, ³Fahmi.fjl@bsi.ac.id

(*Email Corresponding Author: 15220788@bsi.ac.id)

Received: June 12, 2026 | Revision: June 20, 2026 | Accepted: June 23, 2026

Abstrak

Ulasan pengguna pada Google Play Store merupakan sumber informasi yang dapat digunakan untuk mengetahui tingkat kepuasan pengguna terhadap suatu aplikasi. Namun, jumlah ulasan yang terus bertambah menyebabkan proses analisis secara manual menjadi kurang efektif. Oleh karena itu, diperlukan metode analisis sentimen untuk mengidentifikasi kecenderungan opini pengguna secara otomatis. Penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna aplikasi Detik.com menggunakan pendekatan *Lexicon-Based* serta membandingkan kinerja algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM). Data penelitian diperoleh dari Google Play Store melalui proses *web scraping*. Tahapan penelitian meliputi *preprocessing* data, pelabelan sentimen menggunakan pendekatan *Lexicon-Based*, ekstraksi fitur menggunakan TF-IDF, pembagian data latih dan data uji, proses klasifikasi menggunakan algoritma *machine learning*, serta evaluasi model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Hasil pelabelan sentimen menunjukkan bahwa dari 1.000 ulasan yang dianalisis, sebanyak 591 ulasan (59,10%) termasuk sentimen positif dan 409 ulasan (40,90%) termasuk sentimen negatif. Berdasarkan hasil pengujian, algoritma *Decision Tree* memperoleh performa terbaik dengan nilai akurasi sebesar 79,6%, *precision* sebesar 80,1%, *recall* sebesar 79,6%, dan *F1-score* sebesar 79,7%. Sementara itu, *Random Forest* memperoleh akurasi sebesar 77,6%, SVM sebesar 74,5%, dan *Naive Bayes* sebesar 71,9%. Hasil penelitian menunjukkan bahwa pendekatan *Lexicon-Based* yang dikombinasikan dengan algoritma *machine learning* mampu digunakan untuk menganalisis sentimen ulasan pengguna aplikasi Detik.com secara efektif, dengan *Decision Tree* sebagai algoritma yang memberikan kinerja terbaik pada dataset penelitian.

Kata Kunci: Analisis Sentimen, Detik.com, *Lexicon-Based*, *Naive Bayes*, *Decision Tree*, *Random Forest*, *Support Vector Machine*.

Abstract

User reviews on Google Play Store provide valuable information for evaluating user satisfaction with a mobile application. However, the increasing number of reviews makes manual analysis inefficient and time-consuming. Therefore, sentiment analysis is required to automatically identify users' opinions and sentiments. This study aims to analyze the sentiment of Detik.com user reviews using a *Lexicon-Based* approach and compare the performance of *Naive Bayes*, *Decision Tree*, *Random Forest*, and *Support Vector Machine* (SVM) algorithms. The dataset was collected from Google Play Store through a *web scraping* process. The research stages included data *preprocessing*, sentiment labeling using a *Lexicon-Based* approach, feature extraction using TF-IDF, training and testing data splitting, sentiment classification using machine learning algorithms, and model evaluation using accuracy, precision, recall, and F1-score metrics. The sentiment labeling results showed that out of 1,000 reviews analyzed, 591 reviews (59.10%) were classified as positive and 409 reviews (40.90%) as negative. Based on the experimental results, the *Decision Tree* algorithm achieved the best performance with an accuracy of 79.6%, precision of 80.1%, recall of 79.6%, and F1-score of 79.7%. Meanwhile, *Random Forest* achieved an accuracy of 77.6%, SVM 74.5%, and *Naive Bayes* 71.9%. The findings indicate that the combination of the *Lexicon-Based* approach and machine learning algorithms can effectively analyze user sentiment toward the Detik.com application, with *Decision Tree* demonstrating the best classification performance on the dataset used in this study.

Keywords: Sentiment Analysis, Detik.com, *Lexicon-Based*, *Naive Bayes*, *Decision Tree*, *Random Forest*, *Support Vector Machine*.

1. PENDAHULUAN

Perkembangan teknologi informasi telah mengubah cara masyarakat dalam memperoleh informasi. Di era sekarang berbagai informasi dapat diakses dengan melalui perangkat *smartphone* yang terhubung dengan internet. Salah satu bentuk perkembangan tersebut adalah aplikasi berita digital, dan Detik.com merupakan salah satu platform berita yang populer di Indonesia.

Sebagai aplikasi yang digunakan oleh banyak pengguna, Detik.com menerima berbagai ulasan melalui Google Play Store. Ulasan tersebut berisi pengalaman pengguna selama menggunakan aplikasi, baik berupa kritik, saran, maupun apresiasi terhadap layanan yang diberikan. Informasi yang terdapat dalam ulasan pengguna memiliki nilai yang penting karena dapat digunakan untuk mengetahui tingkat kepuasan pengguna sekaligus menjadi bahan evaluasi bagi pengembang aplikasi. Selain sebagai media evaluasi, ulasan pengguna juga banyak dimanfaatkan sebagai sumber data dalam penelitian analisis sentimen untuk mengidentifikasi persepsi pengguna terhadap kualitas layanan aplikasi secara otomatis [1]. Namun, jumlah ulasan yang terus bertambah menyebabkan analisis manual menjadi kurang efektif karena memerlukan waktu yang lama dan berpotensi menghasilkan penilaian yang subjektif. Oleh karena itu, diperlukan suatu metode yang mampu mengolah dan menganalisis data ulasan secara otomatis sehingga informasi yang terkandung di dalamnya dapat dimanfaatkan secara optimal.

Analisis sentimen adalah metode yang banyak digunakan untuk menyelesaikan masalah ini. Analisis sentimen mengkategorikan pendapat pengguna ke dalam kategori tertentu, seperti positif dan negatif, berdasarkan data teks yang tersedia. Analisis sentimen, menurut Sidik et al. [2] dapat digunakan untuk menilai kualitas aplikasi berdasarkan pengalaman pengguna dan membantu pengembang memahami kebutuhan pengguna lebih baik. Oleh karena itu, temuan analisis sentimen dapat menjadi dasar untuk proses membuat keputusan dan mengembangkan aplikasi yang lebih baik.

Dalam penelitian analisis sentimen terdapat beberapa pendekatan yang dapat digunakan. Salah satu pendekatan yang banyak diterapkan adalah *Lexicon-Based*. Pendekatan ini bekerja dengan mencocokkan kata-kata yang terdapat pada teks dengan kamus sentimen yang telah memiliki nilai positif maupun negatif. Ratnaswari et al. [3], menjelaskan bahwa metode *Lexicon-Based* mampu mengidentifikasi sentimen dalam suatu teks melalui proses pencocokan kata dengan kamus sentimen yang telah ditentukan sebelumnya. Keunggulan metode ini adalah mampu melakukan pelabelan sentimen secara otomatis sehingga dapat mengurangi proses pelabelan manual yang membutuhkan waktu cukup lama.

Penelitian tentang analisis sentimen sering menggunakan metode pembelajaran mesin selain pendekatan *lexicon-based*. Memiliki kemampuan untuk mempelajari pola dari data yang tersedia, algoritma pembelajaran mesin memungkinkan sistem untuk melakukan klasifikasi sentimen secara otomatis. *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM) adalah beberapa algoritma yang paling umum digunakan. Berbagai penelitian menunjukkan bahwa kombinasi metode TF-IDF dan algoritma klasifikasi mampu menghasilkan performa yang baik dalam analisis sentimen ulasan aplikasi digital [4]. Semua algoritma memiliki fitur dan tingkat performa yang berbeda, jadi pengujian diperlukan untuk menentukan algoritma mana yang paling cocok untuk dataset yang digunakan.

Sanjaya et al. [5] melakukan analisis sentimen ulasan aplikasi Shopee menggunakan algoritma *Naive Bayes* dan *Support Vector Machine*. Hasil penelitian menunjukkan bahwa *Naive Bayes* memperoleh performa yang lebih baik dibandingkan *Support Vector Machine* pada dataset yang digunakan. Zelina dan Afiyati [6] melakukan analisis sentimen ulasan pengguna aplikasi *Motion Banking* menggunakan pendekatan *Lexicon-Based*, TF-IDF, *Support Vector Machine*, dan *Decision Tree*. Hasil penelitian menunjukkan bahwa algoritma *Support Vector Machine* memperoleh nilai akurasi sebesar 93,7%, lebih tinggi dibandingkan *Decision Tree* yang memperoleh akurasi sebesar 83%. Penelitian lain mengenai analisis sentimen aplikasi *E-Commerce* menggunakan pendekatan *Lexicon* dan *Random Forest* menunjukkan bahwa kombinasi kedua metode tersebut mampu menghasilkan performa klasifikasi yang baik [7].

Selain itu, Astuti et al. [8] menunjukkan bahwa *text mining* dapat digunakan untuk menilai sentimen masyarakat terhadap aplikasi Digital Korlantas Polri yang dibeli dari Google Play Store dan menunjukkan bahwa analisis sentimen dapat digunakan untuk menilai kualitas layanan aplikasi berdasarkan pendapat pengguna. Nurwanda et al. [9] juga mengatakan bahwa pengolahan bahasa alami pada data ulasan aplikasi dapat membuat proses pengolahan pendapat pengguna lebih efisien.

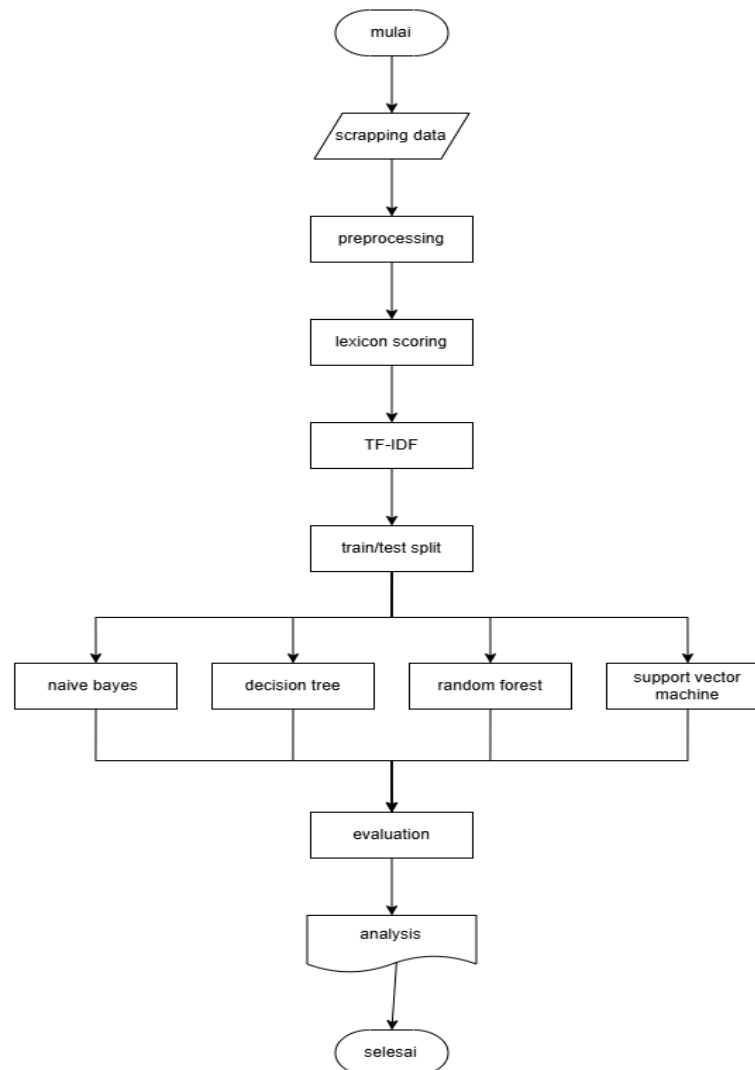
Berdasarkan penelitian sebelumnya, dapat dilihat bahwa analisis sentimen telah banyak diterapkan pada berbagai objek penelitian dengan menggunakan beragam algoritma klasifikasi. Namun, sebagian besar penelitian berfokus pada aplikasi *e-commerce*, layanan perbankan digital, maupun layanan publik. Karakteristik ulasan aplikasi berita digital berbeda dengan aplikasi lainnya karena pengguna tidak hanya menilai performa aplikasi, tetapi juga kualitas dan relevansi konten berita yang disajikan. Kondisi tersebut menyebabkan ulasan pengguna aplikasi berita cenderung lebih beragam dan kompleks sehingga membutuhkan metode analisis yang mampu mengidentifikasi sentimen secara lebih efektif.

Selain itu, penelitian yang mengombinasikan pendekatan *Lexicon-Based* sebagai proses pelabelan sentimen dengan perbandingan beberapa algoritma klasifikasi, yaitu *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM), pada ulasan aplikasi berita digital masih relatif terbatas. Oleh karena itu, penelitian ini dilakukan untuk menganalisis sentimen ulasan pengguna aplikasi Detik.com serta membandingkan kinerja keempat algoritma tersebut guna memperoleh model klasifikasi yang paling sesuai untuk karakteristik data ulasan aplikasi berita digital.

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk menganalisis sentimen ulasan yang diberikan pengguna pada aplikasi Detik.com di Google Play Store dengan menggunakan pendekatan berbasis *Lexicon* dan algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine*. Selain itu, penelitian ini juga bertujuan untuk membandingkan kinerja masing-masing algoritma berdasarkan nilai *accuracy*, *precision*, *recall*, dan *F1-score*. Tujuan akhir dari penelitian ini adalah untuk menentukan algoritma mana yang memiliki kinerja Hasil penelitian diharapkan dapat menunjukkan opini pengguna tentang aplikasi Detik.com dan membantu pengembang meningkatkan kualitas aplikasi pada masa yang akan datang.

2. METODOLOGI PENELITIAN

Dengan menggunakan pendekatan yang didasarkan pada *Lexicon* dan algoritma *machine learning*, penelitian ini bertujuan untuk menganalisis persepsi pengguna terhadap aplikasi Detik.com, yang diperoleh dari Google Play Store. Analisis sentimen digunakan untuk mengetahui kecenderungan pengguna terhadap aplikasi, baik positif maupun negatif. Penelitian dilakukan secara sistematis melalui beberapa tahapan, seperti *scraping data*, *preprocessing*, pelabelan sentimen menggunakan pendekatan yang didasarkan pada *Lexicon*, ekstraksi fitur menggunakan TF-IDF, klasifikasi sentimen menggunakan algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM), serta evaluasi dan analisis hasil klasifikasi. Tahapan penelitian yang dilakukan dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

2.1 Scraping data

Scraping web adalah metode pengambilan data secara otomatis dari situs *web* atau layanan digital. Ini dapat dilakukan dengan lebih cepat dan efisien daripada mengumpulkan data secara manual. Sebagai sumber data penelitian dalam penelitian analisis sentimen, *scraping* banyak digunakan untuk mendapatkan data ulasan pengguna dari berbagai platform digital [10].

Dalam penelitian ini, library *google-play-scraper* digunakan untuk mengumpulkan ulasan pengguna aplikasi Detik.com dari Google Play Store. Data hasil *scraping* digunakan sebagai dataset penelitian dan diproses melalui tahap *preprocessing*, pelabelan sentimen menggunakan pendekatan berbasis *lexicon*, ekstraksi fitur TF-IDF, dan klasifikasi menggunakan algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM).

2.2 Preprocessing

Preprocessing adalah serangkaian tindakan yang dilakukan pada data sebelum digunakan dalam tahap analisis atau pemodelan. Tujuan dari tahap ini adalah untuk membersihkan, menormalkan, dan mempersiapkan data sehingga dapat diolah lebih baik oleh algoritma analisis sentimen dan machine learning. Data mentah dapat diperbaiki dan diubah ke dalam bentuk yang lebih terstruktur melalui proses *preprocessing* [11].

Pada penelitian ini, data ulasan yang diperoleh dari Google Play Store terlebih dahulu melalui tahap *preprocessing* sebelum dilakukan analisis sentimen. Tahapan yang dilakukan meliputi *case folding* untuk mengubah seluruh huruf menjadi huruf kecil (*lowercase*), *cleansing* untuk menghapus karakter yang tidak diperlukan seperti angka, simbol, tanda baca, URL, dan karakter khusus lainnya, serta *tokenizing* untuk memecah kalimat menjadi kata-kata tunggal. Selanjutnya dilakukan *normalization* untuk mengubah kata tidak baku atau singkatan menjadi kata baku, *stopword removal* untuk menghapus kata-kata umum yang tidak memiliki pengaruh signifikan terhadap sentimen, dan *stemming* untuk mengubah kata berimbuhan menjadi kata dasar. Hasil dari proses *preprocessing* berupa data teks yang

lebih bersih dan terstruktur sehingga siap digunakan pada tahap pelabelan sentimen menggunakan pendekatan *Lexicon-Based*, ekstraksi fitur TF-IDF, dan klasifikasi menggunakan algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, serta *Support Vector Machine* (SVM).

2.3 Pelabelan Data

Pelabelan data dilakukan menggunakan pendekatan *Lexicon-Based*, yaitu metode yang menentukan sentimen suatu teks berdasarkan kamus sentimen yang telah memiliki nilai polaritas positif maupun negatif. Setiap ulasan dianalisis dengan mencocokkan kata-kata yang terdapat di dalamnya dengan kata-kata yang tersedia pada kamus sentimen. Selanjutnya, nilai polaritas dari setiap kata dijumlahkan untuk menghasilkan skor sentimen total yang digunakan sebagai dasar penentuan kategori sentimen. Pendekatan *Lexicon-Based* banyak digunakan dalam penelitian analisis sentimen karena mampu melakukan pelabelan data secara otomatis tanpa memerlukan proses anotasi manual yang memerlukan waktu dan biaya yang relatif besar [12].

Setelah tahap *preprocessing* selesai, pelabelan sentimen dilakukan menggunakan InSet Lexicon (Indonesian Sentiment Lexicon) yang dikembangkan oleh Koto dan Rahmaningtyas [13], InSet Lexicon merupakan kamus sentimen Bahasa Indonesia yang dirancang untuk analisis sentimen pada teks berbahasa Indonesia. Kamus ini terdiri dari 3.609 kata positif dan 6.609 kata negatif yang memiliki bobot sentimen pada rentang -5 hingga +5. Pada penelitian ini, setiap kata dalam ulasan dicocokkan dengan kata yang terdapat pada InSet Lexicon untuk memperoleh nilai sentimennya. Selanjutnya, seluruh skor sentimen dalam satu ulasan dijumlahkan untuk menghasilkan skor sentimen total. Ulasan dengan skor sentimen lebih besar dari nol diberi label positif, sedangkan ulasan dengan skor sentimen kurang dari nol diberi label negatif. Hasil pelabelan tersebut kemudian digunakan sebagai data target pada tahap ekstraksi fitur TF-IDF dan proses klasifikasi menggunakan algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM).

2.4 Ekstraksi Fitur TF-IDF

Setelah pelabelan sentimen selesai, langkah berikutnya adalah ekstraksi fitur menggunakan metode Frekuensi *Term-Inverse* Dokumen (TF-IDF). TF-IDF adalah metode pembobotan kata yang digunakan untuk mengubah data teks menjadi representasi numerik sehingga algoritma pembelajaran mesin dapat memprosesnya. Metode ini bekerja dengan menghitung frekuensi kata dalam dokumen (Frekuensi Kata) dan tingkat kepentingan kata tersebut dibandingkan dengan frekuensi dokumen secara keseluruhan.

Data ulasan yang telah melalui tahap *preprocessing* dalam penelitian ini diubah menjadi vektor numerik dengan TF-IDF. Setiap kata dalam ulasan diberi bobot berdasarkan tingkat kemunculannya dalam dataset dan kepentingannya. Selanjutnya, hasil pembobotan TF-IDF digunakan sebagai fitur masukan (*input features*) dalam proses klasifikasi sentimen menggunakan algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM).

$$\frac{TF-IDF(t, d) = TF(t, d) \times IDF(t)}{IDF(t) = \log \frac{N}{df(t)}}$$

Keterangan:

TF(t,d) = frekuensi kemunculan kata t pada dokumen d

IDF(t) = nilai kebalikan frekuensi dokumen

N = jumlah seluruh dokumen

df(t) = jumlah dokumen yang mengandung kata t

2.5 Klasifikasi Sentimen

Penelitian ini menggunakan empat algoritma *machine learning* yaitu *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM). Proses klasifikasi sentimen dilakukan untuk mengelompokkan ulasan pengguna ke dalam kategori sentimen positif dan negatif berdasarkan fitur yang diperoleh dari proses TF-IDF. Algoritma ini dipilih karena banyak digunakan dalam penelitian analisis sentimen dan memiliki karakteristik yang unik.

Teorema Bayes adalah dasar algoritma klasifikasi *Naive Bayes*. *Decision Tree* melakukan klasifikasi dengan membuat struktur pohon keputusan berdasarkan atribut yang digunakan. *Random Forest* adalah pengembangan dari *Decision Tree* yang menggabungkan beberapa pohon keputusan untuk meningkatkan akurasi dan mengurangi risiko *overfitting*. Dalam prosesnya, *Support Vector Machine* (SVM) menemukan *hyperplane* optimal, yang memiliki kemampuan untuk memisahkan data ke dalam berbagai kelas. Selanjutnya, hasil klasifikasi masing-masing algoritma dievaluasi dan dibandingkan untuk mengetahui algoritma mana yang dapat menganalisis sentimen ulasan pengguna aplikasi Detik.com dengan paling baik.

2.6 Evaluasi Model

Setelah proses klasifikasi sentimen selesai, langkah berikutnya adalah evaluasi model untuk mengetahui seberapa baik masing-masing algoritma yang digunakan. Untuk melakukan evaluasi, hasil prediksi model dibandingkan dengan label sentimen yang dihasilkan selama proses pelabelan data. Sebelum proses klasifikasi dilakukan, dataset dalam penelitian ini dibagi menjadi data latihan (*training data*) dan data uji (*testing data*).

Penggunaan *confusion matrix* dengan metrik evaluasi *accuracy*, *precision*, *recall*, dan *F1-score* digunakan untuk mengevaluasi performa model. *Accuracy* mengukur tingkat ketepatan model dalam mengklasifikasikan data, *precision* menunjukkan tingkat ketepatan yang diprediksi untuk suatu kelas, dan *recall* mengukur kemampuan model untuk menemukan seluruh data yang termasuk dalam kelas. *F1-score* adalah nilai harmonis antara *accuracy* dan *recall*. Selanjutnya, hasil evaluasi dari algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM) dibandingkan untuk menentukan algoritma yang melakukan analisis sentimen ulasan pengguna di aplikasi Detik.com dengan paling baik.

Accuracy:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Precision:

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Recall:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

F1-Score:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

Keterangan:

TP (*True Positive*) = data positif yang diprediksi positif

TN (*True Negative*) = data negatif yang diprediksi negatif

FP (*False Positive*) = data negatif yang diprediksi positif

FN (*False Negative*) = data positif yang diprediksi negatif

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan Data

	Review ID	Username	Rating	Review Text	Date
0	a094de08-9f21-4b94-a27b-34f1a1847ceb	MIRSA HAPIT NUGRAHA	5	biasanya ga perduli,cuma ini sbkm 2019 jadi ap...	2026-06-06 18:48:49
1	c8ae8eca-24d0-4869-a2f2-e5f8011a456c	Fahrudin	1	asu ni aplikasi, dibuka beritanya ketinggalan ...	2026-06-06 10:13:08
2	2b9ae13b-21a0-4a5a-9db0-e0badc5712b1	vita sarita	5	mantap om	2026-06-05 21:21:02
3	9110caa3-d7e5-4202-a543-4469c44a50ee	wanita binal	1	error mulu walau koneksi bagus, parah	2026-06-05 09:47:22

Gambar 2. Hasil Pengumpulan Data

Data menunjukkan bahwa pengguna memiliki berbagai tanggapan terhadap pengalaman mereka dengan aplikasi. Komentar menunjukkan perasaan positif, seperti komentar "mantap om" yang menunjukkan kepuasan terhadap aplikasi, tetapi juga komentar negatif, seperti keluhan tentang berita yang tertunda dan masalah dengan aplikasi. Keanekaragaman pendapat pengguna ini menunjukkan bahwa data ulasan dapat digunakan untuk studi analisis sentimen untuk mengetahui kecenderungan pendapat pengguna terhadap aplikasi Detik.com.

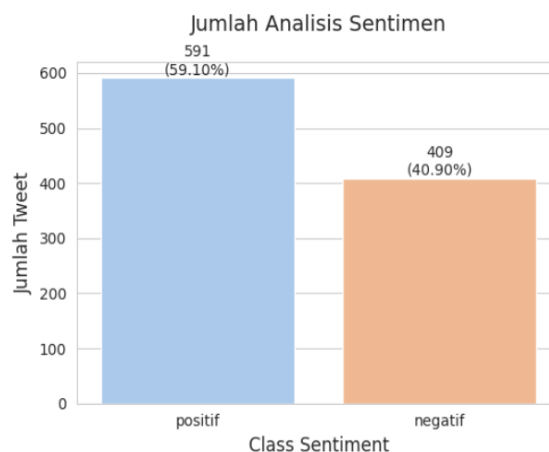
3.2 Hasil *Preprocessing* Teks

	Review Text	cleaning	case_folding	normalisasi	tokenize	stopword removal	stemming_data
0	biasanya ga perduli,cuma ini sbim 2019 jadi ap...	biasanya ga perdulicuma ini sbim jadi aplikas...	biasanya ga perdulicuma ini sbim jadi aplikas...	biasanya tidak perdulicuma ini sebelum jadi ap...	[biasanya, ga, perdulicuma, ini, sbim, jadi, a...	[ga, perdulicuma, sbim, aplikasi, berita, flavo...	ga perdulicuma sbim aplikasi berita favoritskr...
1	asu ni aplikasi, dibuka bertanya ketinggalan ...	asu ni aplikasi dibuka bertanya ketinggalan b...	asu ni aplikasi dibuka bertanya ketinggalan b...	asu nih aplikasi dibuka bertanya ketinggalan ...	[asu, ni, aplikasi, dibuka, bertanya, ketingg...	[asu, ni, aplikasi, dibuka, bertanya, ketingg...	asu ni aplikasi buka berita tinggal bas jam qu...
2	mantap om	mantap om	mantap om	mantap om	[mantap, om]	[mantap, om]	mantap om
3	error mulu walau koneksi bagus, parah	error mulu walau koneksi bagus parah	error mulu walau koneksi bagus parah	error mulu walau koneksi bagus parah	[error, mulu, walau, koneksi, bagus, parah]	[error, mulu, koneksi, bagus, parah]	error mulu koneksi bagus parah
4	Baik n bagus	Baik n bagus	baik n bagus	baik n bagus	[baik, n, bagus]	[n, bagus]	n bagus

Gambar 3. Hasil *Preprocessing* Teks

Setiap ulasan mengalami perubahan pada setiap tahapan *preprocessing*, seperti yang ditunjukkan pada gambar 3. Karakter yang tidak diperlukan dibersihkan dari teks ulasan pada tahap perbaikan. Selanjutnya, *case folding* mengubah huruf secara keseluruhan menjadi huruf kecil, atau *lowercase*. Proses normalisasi mengubah kata-kata yang tidak baku menjadi kata baku, misalnya, "ga" menjadi "tidak" dan "sbim" menjadi "sebelum". *Stopword removal* kemudian membagi kalimat menjadi kumpulan kata tunggal. *Stemming* mengubah kata berimbuhan menjadi kata dasar, sedangkan *stopword removal* menghilangkan kata-kata umum yang tidak memberikan sentimen yang signifikan. Proses ini menghasilkan data teks yang lebih ringkas, konsisten, dan siap digunakan pada tahap pelabelan sentimen menggunakan pendekatan berbasis *lexicon* serta proses klasifikasi yang menggunakan algoritma *Support Vector Machine (SVM)*, *Decision Tree*, *Random Forest*, dan *Naive Bayes*.

3.3 Hasil Pelabelan Menggunakan *Lexicon-Based*

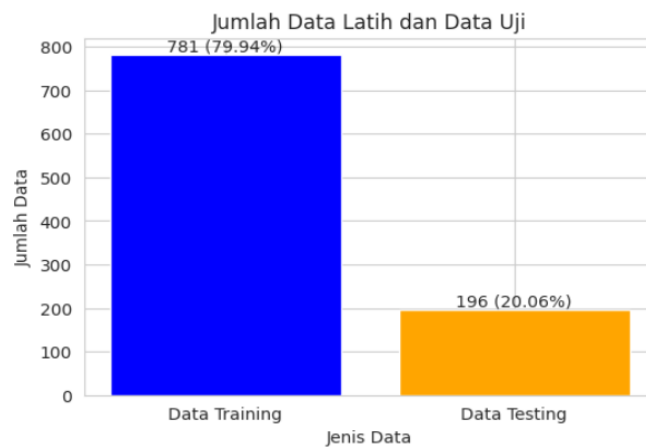


Gambar 4. Hasil Pelabelan Menggunakan *Lexicon*

Berdasarkan hasil pelabelan sentimen menggunakan pendekatan *Lexicon-Based*, diperoleh 591 ulasan dengan sentimen positif dan 409 ulasan dengan sentimen negatif. Hasil menunjukkan bahwa sentimen positif mendominasi dataset dengan persentase 59,10% dan sentimen negatif dengan persentase 40,90%. Dominasi sentimen positif menunjukkan bahwa sebagian besar pengguna memberikan respons yang baik terhadap aplikasi Detik.com. Setelah pelabelan ini selesai, label target digunakan dalam proses klasifikasi menggunakan algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine (SVM)*.

3.4 Pembagian Data

Sebanyak 781 data, atau 79,94 %, diperoleh sebagai data latih, dan 196 data, atau 20,06%, diperoleh sebagai data uji, seperti yang ditunjukkan pada Gambar 5. Untuk membangun model klasifikasi, algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine (SVM)* digunakan dengan data latih. Di sisi lain, data uji digunakan untuk mengevaluasi kemampuan model dalam klasifikasi sentimen. Untuk menghasilkan keseimbangan antara proses pelatihan dan pengujian model, pembagian data dengan rasio 80 : 20 dipilih.

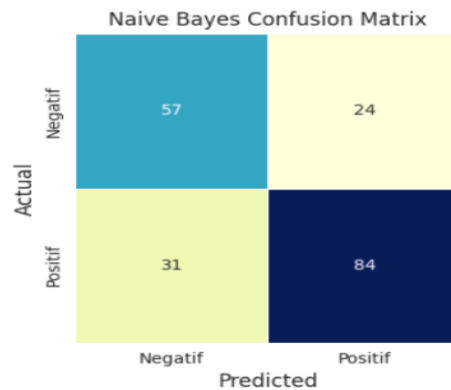


Gambar 5. Data *Training* dan Data *Testing*

3.5 Implementasi Algoritma

3.5.1 *Naive Bayes* (NB)

Confusion Matrix for Naive Bayes:

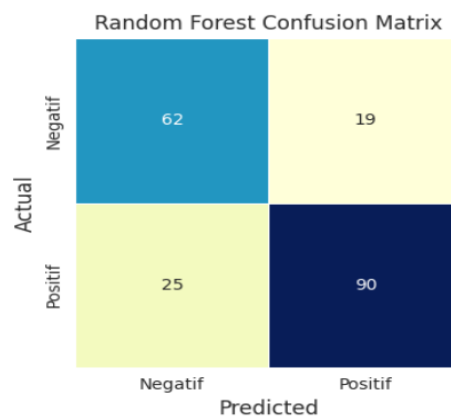


Gambar 6. *Confusion Matrix Naive Bayes*

Gambar 6 menunjukkan *confusion matrix* algoritma *Naive Bayes*. Model berhasil mengklasifikasikan 57 data negatif dan 84 data positif dengan benar. Sementara itu, masih terdapat beberapa kesalahan klasifikasi yang menyebabkan sebagian data negatif diprediksi sebagai positif dan sebaliknya. Hasil ini menunjukkan bahwa *Naive Bayes* mampu melakukan klasifikasi sentimen dengan cukup baik pada dataset penelitian.

3.5.2 *Random Forest*

Confusion Matrix for Random Forest:

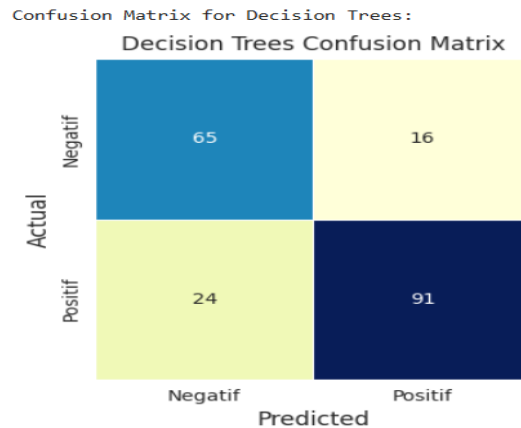


Gambar 7. *Confusion Matrix Random Forest*

Gambar 7 menunjukkan *confusion matrix* algoritma *Random Forest*. Model berhasil mengklasifikasikan sebagian besar data sentimen positif dan negatif dengan benar. Jumlah kesalahan klasifikasi yang dihasilkan lebih sedikit dibandingkan *Naive Bayes*, sehingga menunjukkan kemampuan *Random Forest* yang lebih baik dalam mengenali pola sentimen pada data ulasan pengguna.

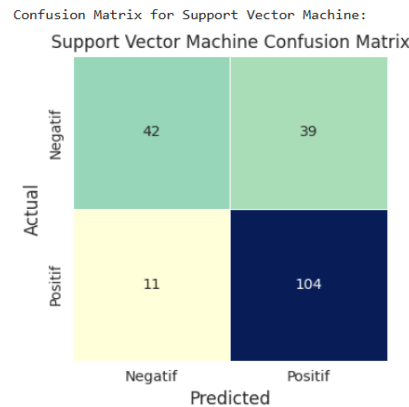
3.5.3 Decision Tree

Gambar 8 menunjukkan *confusion matrix* algoritma *Decision Tree*. Model mampu mengklasifikasikan sebagian besar data sentimen dengan benar serta menghasilkan jumlah kesalahan klasifikasi yang relatif rendah. Hasil ini menunjukkan bahwa *Decision Tree* mampu membentuk aturan keputusan yang sesuai dengan karakteristik data ulasan pengguna aplikasi Detik.com.



Gambar 8. Confusion Matrix Decision Tree

3.5.4 Support Vector Machine (SVM)

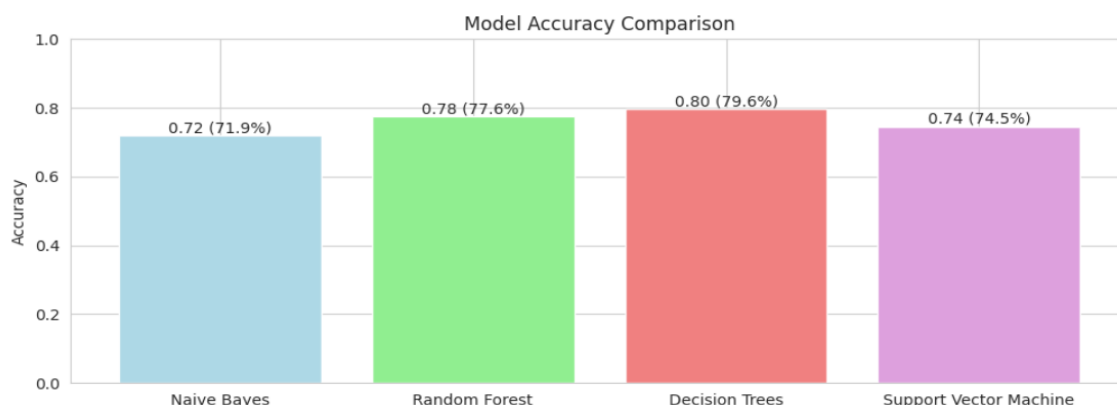


Gambar 9. Confusion Matrix SVM

Gambar 9 menunjukkan *confusion matrix* algoritma *Support Vector Machine* (SVM). Model mampu mengidentifikasi sebagian besar data sentimen positif dan negatif dengan benar, meskipun masih terdapat beberapa kesalahan klasifikasi. Hasil ini menunjukkan bahwa SVM memiliki kemampuan yang cukup baik dalam memisahkan data berdasarkan kelas sentimen yang berbeda.

3.6 Evaluasi dan Perbandingan Kinerja Model

Setelah proses klasifikasi selesai menggunakan algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM), tahap selanjutnya adalah membandingkan kinerja masing-masing model berdasarkan nilai akurasi. Tujuan dari perbandingan ini adalah untuk menentukan algoritma mana yang dapat mengklasifikasikan sentimen ulasan pengguna di aplikasi Detik.com dengan paling efisien.



Gambar 10. Perbandingan Accuracy

Tabel 1. Hasil Evaluasi Algoritmas

Algoritma	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	71,9	72,4	71,9	72,1
Random Forest	77,6	77,9	77,6	77,7
Decision Tree	79,6	berdasarkan 80,1	79,6	79,7
SVM	74,5	75,4	74,5	73,2

Berdasarkan Gambar 10 dan Tabel 1, algoritma *Decision Tree* memperoleh performa terbaik dengan nilai *accuracy* sebesar 79,6%, *precision* sebesar 80,1%, *recall* sebesar 79,6%, dan *F1-score* sebesar 79,7%. Hasil tersebut menunjukkan bahwa *Decision Tree* mampu menghasilkan klasifikasi sentimen yang lebih baik dibandingkan algoritma lainnya pada dataset yang digunakan.

3.7 Pembahasan Hasil

Hasil pelabelan sentimen menggunakan pendekatan *Lexicon-Based* menunjukkan bahwa dari 1.000 ulasan yang dianalisis, sebanyak 591 ulasan (59,10%) termasuk sentimen positif dan 409 ulasan (40,90%) termasuk sentimen negatif. Dominasi sentimen positif menunjukkan bahwa sebagian besar pengguna memberikan tanggapan yang baik terhadap aplikasi Detik.com. Namun demikian, masih terdapat sejumlah ulasan negatif yang menunjukkan adanya keluhan terkait performa aplikasi, tampilan, maupun pengalaman pengguna sehingga dapat menjadi bahan evaluasi bagi pengembang.

Berdasarkan hasil evaluasi model, algoritma *Decision Tree* memperoleh performa terbaik dengan nilai *accuracy* sebesar 79,6%, *precision* sebesar 80,1%, *recall* sebesar 79,6%, dan *F1-score* sebesar 79,7%. Hasil tersebut menunjukkan bahwa *Decision Tree* mampu mengklasifikasikan sentimen ulasan pengguna dengan lebih baik dibandingkan algoritma lainnya pada dataset yang digunakan.

Secara teoritis, *Random Forest* dan *Support Vector Machine* (SVM) sering menghasilkan performa yang lebih tinggi dibandingkan *Decision Tree* karena memiliki kemampuan yang lebih baik dalam menangani data yang kompleks. Namun, pada penelitian ini *Decision Tree* justru memperoleh hasil terbaik, temuan tersebut menunjukkan bahwa algoritma dengan kompleksitas yang lebih tinggi tidak selalu menghasilkan performa terbaik. Hasil yang diperoleh sangat dipengaruhi oleh karakteristik data dan representasi fitur yang digunakan dalam proses klasifikasi [14]. Hal ini diduga dipengaruhi oleh ukuran dataset yang relatif terbatas, yaitu 1.000 ulasan. Pada dataset yang tidak terlalu besar, model yang lebih sederhana seperti *Decision Tree* dapat mempelajari pola data dengan lebih efektif tanpa memerlukan kompleksitas model yang tinggi.

Selain itu, proses pelabelan menggunakan pendekatan *Lexicon-Based* menghasilkan pemisahan kelas sentimen yang relatif jelas. Setelah dilakukan ekstraksi fitur menggunakan TF-IDF, fitur-fitur yang terbentuk cenderung dapat dipisahkan menggunakan aturan keputusan yang sederhana sehingga *Decision Tree* mampu membangun model klasifikasi yang sesuai dengan karakteristik data. Sementara itu, *Random Forest* memperoleh akurasi sebesar 77,6%, SVM sebesar 74,5%, dan *Naive Bayes* sebesar 71,9%. Hasil tersebut menunjukkan bahwa performa algoritma klasifikasi dapat berbeda pada setiap dataset. Penelitian Suharman dan Sulaeman (2025) juga menunjukkan bahwa pemilihan metode ekstraksi fitur berpengaruh terhadap kinerja model klasifikasi sentimen [15]. Performa *Naive Bayes* yang lebih rendah diduga disebabkan oleh asumsi independensi antar fitur yang kurang sesuai dengan karakteristik data teks yang memiliki keterkaitan antar kata.

Secara keseluruhan, hasil penelitian menunjukkan bahwa pendekatan *Lexicon-Based* mampu melakukan pelabelan sentimen secara otomatis, sedangkan algoritma *machine learning* mampu mengklasifikasikan sentimen ulasan pengguna dengan cukup baik. Berdasarkan hasil evaluasi yang diperoleh, *Decision Tree* merupakan algoritma yang paling sesuai untuk klasifikasi sentimen ulasan pengguna aplikasi Detik.com pada penelitian ini.

4. KESIMPULAN

Studi ini berhasil menggunakan pendekatan berbasis lexicon untuk pelabelan sentimen serta algoritma *Naive Bayes*, *Decision Tree*, *Random Forest*, dan *Support Vector Machine* (SVM) untuk mengklasifikasikan ulasan pengguna aplikasi Detik.com yang diperoleh dari Google Play Store. Hasil pelabelan sentimen menunjukkan bahwa dari 1.000 ulasan yang dianalisis, sebanyak 591 ulasan (59,10%) termasuk sentimen positif dan 409 ulasan (40,90%) termasuk sentimen negatif. Hasil pengujian model menunjukkan bahwa masing-masing algoritma memiliki tingkat performa yang berbeda. Akurasi tertinggi diperoleh oleh algoritma *Decision Tree* sebesar 79,6%, diikuti oleh *Random Forest* sebesar 77,6%, *Support Vector Machine* (SVM) sebesar 74,5%, dan *Naive Bayes* sebesar 71,9%. Hasil tersebut menunjukkan bahwa algoritma *Decision Tree* merupakan algoritma terbaik dalam mengklasifikasikan sentimen ulasan pengguna aplikasi Detik.com pada dataset yang digunakan dalam penelitian ini. Penelitian ini menunjukkan bahwa pendekatan *Lexicon-Based* dan algoritma *machine learning* dapat digunakan secara efektif untuk menganalisis sentimen ulasan pengguna aplikasi Detik.com. Hasil penelitian diharapkan dapat membantu pengembang dalam memahami opini pengguna serta menjadi bahan evaluasi untuk meningkatkan kualitas layanan dan pengalaman pengguna aplikasi. Penelitian ini masih memiliki keterbatasan, yaitu jumlah data yang digunakan relatif terbatas sebanyak 1.000 ulasan dan proses pelabelan sentimen masih bergantung pada kamus lexicon sehingga belum mampu menangkap konteks bahasa yang lebih kompleks, seperti sarkasme, ironi, atau makna yang bergantung pada konteks kalimat. Oleh karena itu, penelitian selanjutnya disarankan menggunakan jumlah data yang lebih besar serta mencoba pendekatan *Deep Learning*, seperti LSTM atau BERT, untuk meningkatkan kemampuan model dalam memahami konteks bahasa dan menghasilkan performa klasifikasi yang lebih baik.

REFERENCES

- [1] N. K. A. Juliana and M. A. Raharja, "Analisis Sentimen pada Ulasan Aplikasi myIM3 Menggunakan Multinomial Naive Bayes dengan TF-IDF," *J. Nas. Teknol. Inf. dan Apl.*, vol. 2, no. 3, pp. 649–656, 2024, [Online]. Available: <https://ojs.unud.ac.id/index.php/jnatia/article/view/115818>
- [2] K. Sidik, D. Herdiana, M. A. Helmiawan, and A. Saeppani, "ANALISIS SENTIMEN ULASAN GOOGLE PLAY PADA APLIKASI TAHU SUMEDANG MENGGUNAKAN LEXICON-BASED," vol. 5, no. 2, pp. 356–361, 2025.
- [3] S. Ratnaswari, N. C. Wibowo, D. Satria, and Y. Kartika, "ANALISIS SENTIMEN MENGGUNAKAN METODE LEXICON-BASED DAN SUPPORT VECTOR MACHINE PADA PRESIDEN DAN WAKIL PRESIDEN INDONESIA," vol. 13, no. 1, pp. 362–368, 2025.
- [4] J. Sutrisno and G. Eka, "Analisis Sentimen Pada Aplikasi Livin ' By Mandiri Menggunakan Metode Tf-Idf Dan Naive Bayes Sentiment Analysis On Livin ' By Mandiri App Using Tf -Idf And Naive Bayes Method," vol. 17, no. 1, 2024.
- [5] T. P. R. Sanjaya, A. Fauzi, and A. F. N. Masruriyah, "Analisis sentimen ulasan pada e-commerce shopee menggunakan algoritma naive bayes dan support vector machine," *INFOTECH J. Inform. Teknol.*, vol. 4, no. 1, pp. 16–26, 2023, doi: 10.37373/infotech.v4i1.422.
- [6] N. Zelina and A. Afyati, "Analisis Sentimen Ulasan Pengguna Aplikasi M- Banking Menggunakan Algoritma Support Vector Machine dan Decision Tree," vol. 7, no. 1, pp. 31–37, 2024.
- [7] A. Syah, F. Nurdiansyah, and A. Y. Rahman, "Analisis sentimen aplikasi shopee, tokopedia, lazada dan blibli menggunakan leksikon dan random forest," vol. 12, no. 3, 2024.
- [8] K. C. Astuti, A. Firmansyah, and A. Riyadi, "Implementasi Text Mining untuk Analisis Sentimen Masyarakat terhadap Ulasan Aplikasi Digital Korlantas Polri pada Google Play Store," *Ris. dan E-Jurnal Manaj. Inform. Komput.*, vol. 8, pp. 383–394, 2024.
- [9] N. Nurwanda, N. Suarna, and W. Prihartono, "Penerapan Nlp (Natural Language Processing) Dalam Analisis Sentimen Pengguna Telegram Di Playstore," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 2, pp. 1841–1846, 2024, doi: 10.36040/jati.v8i2.8469.
- [10] K. Dwicahyo and C. Indah Ratnasari, "Perbandingan Metode Web Scraping Dalam Pengambilan Data: Kajian Literatur," *Automata*, vol. 4, no. 2, pp. 200–205, 2023.
- [11] F. Putra, H. F. Tahiyat, R. M. Ihsan, R. Rahmaddeni, and L. Efrizoni, "Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia," *MALCOM*

Indones. J. Mach. Learn. Comput. Sci., vol. 4, no. 1, pp. 273–281, 2024, doi: 10.57152/malcom.v4i1.1085.

- [12] T. D. Septyorini, K. Umam, and M. R. Handayani, “Jurnal Informatika : Jurnal pengembangan IT Dinamika Opini Publik Terkait Quarter Life Crisis Pada Media Sosial X Menggunakan Support Vector Machine,” vol. 10, no. 3, pp. 617–627, 2025, doi: 10.30591/jpit.v10i3.8648.
- [13] F. Koto and G. Y. Rahmaningtyas, “Inset lexicon: Evaluation of a word list for Indonesian sentiment analysis in microblogs,” in *2017 International Conference on Asian Language Processing (IALP)*, 2017, pp. 391–394. doi: 10.1109/IALP.2017.8300625.
- [14] M. T. Akhir, M. Syarat, G. Memperoleh, G. Sarjana, S. Satu, and T. Informasi, “Analisis sentimen pengguna aplikasi livin’ by mandiri berdasarkan ulasan pada google playstore menggunakan metode random forest skripsi,” 2024.
- [15] A. Suharman and M. Kamayani Sulaeman, “Analisis Sentimen Pengguna Aplikasi Livin’ by Mandiri Menggunakan Metode Support Vector Machine (SVM) dengan Ekstraksi Fitur TF-IDF dan Word2Vec,” *J. Pendidik. dan Teknol. Indones.*, vol. 5, no. 8, pp. 2201–2212, 2025, doi: 10.52436/1.jpti.941.