

Analisis Sentimen Ulasan Aplikasi Loklok Menggunakan Metode *Support Vector Machine* (SVM)

I Gusti Ngurah Adhiwangsa Devananda^{a1}, Luh Arida Ayu Rahning Putri^{a2}, I Komang Arya Ganda Wiguna^{a2}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
¹adhiwangsa-devananda@email.com
²rahningputri@unud.ac.id
³arya.ganda@unud.ac.id

Abstract

Rapid advances in digital technology have led to an increase in the amount of text data available online, including user reviews of mobile applications. The Loklok application, as a popular entertainment platform, is one source of review data that is rich in user opinions. This research focuses on performing sentiment analysis on user reviews of the Loklok application by employing the Support Vector Machine (SVM) algorithm alongside the Term Frequency-Inverse Document Frequency (TF-IDF) method for feature extraction. The review dataset was sourced from the Kaggle platform and underwent several text preprocessing steps, including data cleaning, tokenization, stopwords elimination, and stemming. The evaluation results indicate that the SVM model, combined with TF-IDF, achieved an accuracy of 86%, a precision of 88%, a recall of 86%, and an F1-score of 87%. Classification performance tends to be better for positive sentiment classes compared to negative ones, due to data imbalance. This finding demonstrates that the combination of TF-IDF and SVM methods is effective in classifying user review sentiment and can serve as a foundation for decision-making in the development of digital applications.

Keywords: Sentiment Analysis, Application Reviews, TF-IDF, Support Vector Machine, Text Classification

1. Pendahuluan

Kemajuan teknologi informasi yang sangat cepat telah memberikan pengaruh besar terhadap berbagai bidang kehidupan, mulai dari cara berkomunikasi, menikmati hiburan, hingga pemanfaatan layanan digital. Salah satu wujud dari kemajuan ini adalah meningkatnya penggunaan aplikasi mobile yang memengaruhi cara masyarakat mengakses dan mengonsumsi konten di perangkat mobile berupa aplikasi, sejalan dengan temuan bahwa bagi orang dewasa di Inggris, waktu penggunaan ponsel (3 jam 21 menit per hari) kini melebihi waktu menonton TV (3 jam 16 menit)[1]. Salah satu aplikasi yang cukup populer adalah Loklok, platform berbagi konten hiburan yang memungkinkan pengguna untuk menonton dan memberikan ulasan terhadap film maupun serial yang mereka tonton.

Ulasan pengguna yang tersedia di Google Play Store menjadi sumber informasi penting bagi pengembang untuk memahami persepsi, pengalaman, dan tingkat kepuasan pengguna[2]. Melalui pendekatan analisis sentimen, opini-opini tersebut dapat diolah dan diklasifikasikan ke dalam kategori seperti positif, negatif, atau netral, guna memberikan wawasan yang lebih dalam dalam proses evaluasi dan pengembangan produk digital.

Namun, analisis sentimen terhadap aplikasi Loklok memiliki sejumlah tantangan, seperti ketidakseimbangan distribusi antara ulasan positif dan negatif yang dapat menyebabkan bias pada hasil klasifikasi [3]. Tantangan lainnya adalah kompleksitas bahasa yang digunakan dalam ulasan pengguna, seperti penggunaan slang, singkatan, dan campuran bahasa yang memerlukan pendekatan prapemrosesan teks yang tepat agar hasil analisis lebih akurat

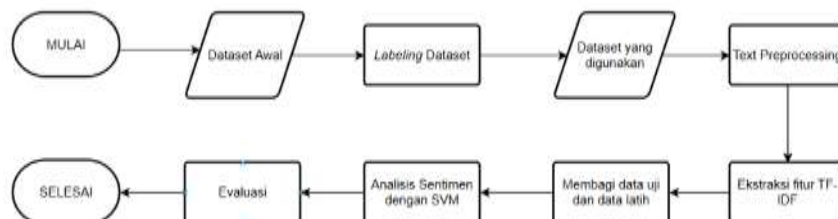
Sebagai upaya untuk mengatasi permasalahan tersebut, sejumlah penelitian telah menggunakan berbagai metode pra-pemrosesan, seperti pemecahan teks menjadi token (tokenisasi), penghilangan kata-kata yang tidak memiliki makna signifikan (*stopword*), serta pengembalian kata ke bentuk dasarnya (*stemming*), dengan tujuan untuk menyederhanakan dan membersihkan data teks yang akan dianalisis. Selain itu, pembobotan fitur menggunakan TF-IDF juga dinilai efektif dalam meningkatkan performa model klasifikasi karena membantu menonjolkan kata-kata penting dalam setiap dokumen[4]. Penelitian sebelumnya oleh Alfian dan Azhar (2023) membuktikan bahwa penerapan TF-IDF yang dikombinasikan dengan Algoritma *Support Vector Machine* (SVM) memiliki kemampuan dalam mengklasifikasikan sentimen ulasan pengguna dengan tingkat akurasi yang tinggi. aplikasi *Digitalent Mobile*, sehingga metode ini dinilai relevan untuk diterapkan dalam konteks aplikasi lainnya[5].

Meskipun metode Naïve Bayes sering digunakan dalam klasifikasi teks karena kesederhanaannya, namun algoritma ini memiliki keterbatasan dalam menghadapi data yang tidak seimbang atau ketika fitur antar kata tidak independen. Oleh karena itu, penelitian ini mengusulkan penggunaan Support Vector Machine (SVM), Algoritma yang dikenal mampu melakukan generalisasi dengan baik serta menunjukkan performa tinggi dalam menangani data dengan dimensi besar maupun distribusi kelas yang tidak seimbang.

Penelitian ini bertujuan untuk melakukan analisis sentimen terhadap ulasan pengguna aplikasi Loklok dengan menggunakan algoritma Support Vector Machine (SVM). Tahapan penelitian meliputi pengumpulan data dari platform Kaggle, dilanjutkan dengan proses pra-pemrosesan teks, ekstraksi fitur menggunakan metode TF-IDF, hingga pembuatan model klasifikasi. Diharapkan, hasil dari penelitian ini dapat memberikan pemahaman yang lebih mendalam mengenai opini pengguna serta menjadi masukan strategis bagi pengembangan aplikasi Loklok ke depannya.

2. Metode Penelitian

2.1 Alur Penelitian



Gambar 1. Alur Penelitian

Penelitian ini dimulai dengan mengunduh dataset ulasan pengguna aplikasi Loklok dalam bahasa Indonesia dari situs Kaggle. Setelah data dikumpulkan, setiap ulasan diberi label sentimen berdasarkan skornya: skor antara 1 sampai 3 dimasukkan ke dalam kategori sentimen negatif, sedangkan skor 4 dan 5 diklasifikasikan sebagai sentimen positif. Setelah proses labeling, data disaring dan dipilih untuk digunakan dalam proses analisis.

Langkah berikutnya adalah text preprocessing, Tahapan ini mencakup beberapa proses, seperti pembersihan teks, pemecahan kalimat menjadi token, penghapusan kata-kata umum (*stopword*), serta proses stemming untuk mengembalikan kata ke bentuk dasarnya. Setelah melalui tahapan tersebut, teks kemudian dikonversi menjadi representasi numerik menggunakan pendekatan TF-IDF, sehingga dapat dipahami oleh algoritma klasifikasi. Data selanjutnya dibagi menjadi data latih dan data uji.

Proses inti dari penelitian ini adalah klasifikasi sentimen menggunakan algoritma Support Vector

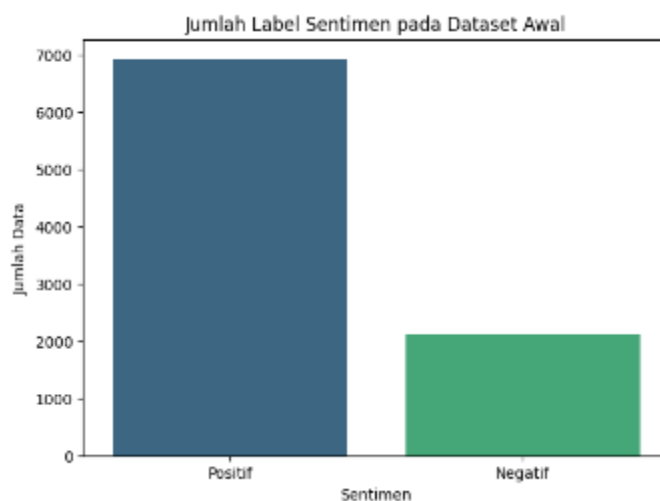
Machine (SVM), yang dilatih menggunakan data latih dan diuji pada data uji. Terakhir, model dievaluasi dengan menghitung metrik akurasi, precision, recall, dan F1-score untuk menilai performa klasifikasi.

2.2 Pengumpulan Data

Dataset yang dianalisis dalam studi ini diambil dari Kaggle dan terdiri atas ulasan berbahasa Indonesia yang ditulis oleh pengguna aplikasi Loklok. Dataset ini merupakan hasil *scraping* dari Google Play Store yang telah tersedia secara publik dalam format CSV. Data ulasan mencakup periode waktu dari tanggal 30 November 2023 hingga 30 Desember 2024, dengan total sebanyak 9.031 entri ulasan.

Informasi yang tersedia dalam dataset meliputi nama pengguna, gambar profil pengguna, teks ulasan (content), rating atau skor ulasan (score), jumlah likes pada ulasan, serta tanggal ulasan diberikan. Data ini menjadi dasar dalam proses analisis sentimen terhadap persepsi pengguna terhadap aplikasi Loklok.

Pada tahap selanjutnya, setiap data ulasan diberi label berdasarkan nilai skor yang dimilikinya. Ulasan yang memperoleh nilai antara 1 hingga 3 dianggap merepresentasikan sentimen bernuansa negatif, sedangkan ulasan dengan skor 4 maupun 5 diperlakukan sebagai representasi dari sentimen positif. Pendekatan klasifikasi biner ini dipilih untuk menyederhanakan analisis dan fokus pada perbedaan persepsi pengguna secara tegas antara positif dan negatif.



Gambar 2. Diagram Batang Jumlah Label Sentimen pada Dataset Awal

Setelah dilakukan proses pelabelan berdasarkan nilai skor ulasan, dataset terbagi ke dalam dua kategori sentimen, yaitu positif dan negatif. Ulasan dengan skor 4 dan 5 diklasifikasikan sebagai sentimen positif, sedangkan skor 1, 2, dan 3 diklasifikasikan sebagai sentimen negatif. Berdasarkan hasil pelabelan, terdapat sebanyak 6.916 ulasan positif dan 2.115 ulasan negatif. Distribusi ini menunjukkan bahwa mayoritas pengguna memberikan ulasan yang bersifat positif terhadap aplikasi Loklok, sementara ulasan negatif jumlahnya lebih sedikit. Ketidakseimbangan jumlah data antar kelas ini menjadi perhatian dalam proses pemodelan, karena berpotensi mempengaruhi performa model klasifikasi sentimen yang digunakan.

2.3 Text Preprocessing

Tahapan pra-pemrosesan data bertujuan untuk membersihkan serta mengubah teks mentah menjadi bentuk yang terstruktur dan dapat diproses oleh algoritma. Langkah pertama yang dilakukan adalah case folding, yaitu mengubah seluruh huruf dalam teks menjadi huruf kecil. Setelah itu, dilakukan pembersihan teks dengan menghapus karakter non-alfabet seperti angka

dan tanda baca. Proses selanjutnya adalah stemming menggunakan pustaka Sastrawi untuk mengembalikan setiap kata ke bentuk dasarnya, diikuti dengan stopword removal guna menghapus kata-kata umum yang tidak memiliki nilai sentimen[6]. Terakhir, dilakukan tokenisasi menggunakan library NLTK untuk memecah kalimat bersih menjadi kata-kata terpisah (token), yang nantinya akan digunakan pada proses ekstraksi fitur[7].

2.4 TF-IDF (Term Frequency-Inverse Document Frequency)

Setelah melalui proses pra-pemrosesan, tahap selanjutnya adalah melakukan ekstraksi fitur serta pemberian bobot pada setiap kata dengan menggunakan pendekatan TF-IDF (Term Frequency-Inverse Document Frequency). Metode ini berfungsi untuk menilai seberapa penting suatu kata dalam suatu dokumen jika dibandingkan dengan seluruh dokumen dalam korpus. Bobot TF-IDF diperoleh dengan mengalikan frekuensi kemunculan kata dalam satu dokumen (TF) dengan nilai logaritmik dari kebalikan proporsi dokumen yang memuat kata tersebut (IDF), yang menunjukkan seberapa jarang kata itu muncul [8]. Kata dengan nilai TF-IDF yang tinggi dianggap memiliki tingkat kepentingan yang lebih besar. Nilai TF-IDF untuk sebuah kata t dalam dokumen d dihitung menggunakan rumus sebagai berikut.:

$$TF(t, d) = \frac{\text{Jumlah frekuensi kata } t \text{ dalam dokumen } d}{\text{Jumlah seluruh kata dalam dokumen } d} \quad (1)$$

$$IDF(t) = \log \left(\frac{\text{Jumlah seluruh dokumen}}{\text{Jumlah dokumen yang mengandung kata } t} \right) \quad (2)$$

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

2.5 Support Vector Machine (SVM)

Proses pemodelan klasifikasi dimulai dengan membagi dataset menjadi dua bagian: dua puluh persen digunakan sebagai data pelatihan dan dua puluh persen digunakan sebagai data pengujian. Dalam penelitian ini, algoritma Support Vector Machine (SVM) digunakan; SVM adalah salah satu metode pengajaran di bawah pengawasan yang paling efektif untuk tugas klasifikasi.[9]. Prinsip dasar dari SVM adalah mencari hyperplane atau bidang pemisah terbaik yang mampu membedakan dua kelas sentimen, yaitu positif dan negatif, dengan margin maksimum[4]. Margin ini merujuk pada support vector, jarak antara *hyperplane* dengan titik data terdekat dari masing-masing kelas. *Hyperplane* secara matematis dinyatakan melalui persamaan:

$$w \cdot x - b = 0 \quad (4)$$

di mana x adalah vektor fitur, w adalah vektor normal, dan b adalah nilai bias. Tujuan SVM adalah untuk menemukan hyperplane terbaik dengan memaksimalkan margin, yang secara matematis setara dengan meminimalkan nilai $\|w\|^2$, dengan syarat bahwa seluruh data diklasifikasikan dengan benar berdasarkan persamaan berikut:

$$y_i (w \cdot x_i - b) \geq 1 \quad (5)$$

Dalam implementasinya, SVM menggunakan kernel linear, yang terbukti efektif untuk data teks berdimensi tinggi seperti hasil ekstraksi fitur dengan TF-IDF. Untuk menangani masalah ketidakseimbangan kelas dalam dataset, parameter `class_weight` disetel ke 'balanced', yang secara otomatis memberikan bobot lebih besar pada kelas minoritas. Hal ini bertujuan untuk mengurangi kecenderungan model terhadap bias. Setelah konfigurasi dilakukan, model SVM dilatih menggunakan data pelatihan untuk menentukan hyperplane pemisah yang paling optimal.

2.6 Evaluasi

Tahap evaluasi bertujuan untuk mengukur performa model SVM yang telah dilatih secara objektif. Evaluasi dilakukan dengan menggunakan data uji (*testing data*) sebesar 20% dari total dataset, yang berisi data yang belum pernah dilihat oleh model selama proses pelatihan. Kinerja model diukur berdasarkan kemampuannya dalam memprediksi sentimen dengan benar.

Dasar dari semua perhitungan metrik evaluasi adalah Confusion Matrix. *Confusion Matrix* merupakan sebuah tabel yang merangkum hasil prediksi model dengan membandingkannya terhadap label aktual. Matriks ini terdiri dari empat komponen utama yaitu True Positive (TP) adalah jumlah data positif yang diprediksi secara akurat sebagai positif, sedangkan True Negative (TN) adalah jumlah data negatif yang diprediksi secara akurat sebagai negatif. False Positive (FP) adalah jumlah data negatif yang diprediksi secara salah sebagai positif. False Negative (FN): Sebuah jumlah data positif yang diprediksi negatif secara salah.

Empat metrik utama: akurasi, presisi, recall, dan F1-score, digunakan untuk mengukur kinerja model dari masing-masing komponen Confusion Matrix tersebut.

- a. Akurasi (*Accuracy*) Mengukur tingkat ketepatan model secara keseluruhan.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

- b. Presisi (*Precision*) Mengukur tingkat keakuratan dari prediksi positif yang dibuat oleh model.

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

- c. Recall (Sensitivity) Mengukur kemampuan model dalam menemukan kembali semua data yang seharusnya berlabel positif.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

- d. F1-Score yang merupakan rata-rata harmonik dari Presisi dan *Recall*, memberikan gambaran tentang performa yang seimbang, terutama dalam situasi di mana data tidak seimbang.

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (9)$$

Dalam implementasinya, semua nilai metrik ini dihitung secara otomatis menggunakan fungsi *accuracy_score* dan *classification_report* dari *library Scikit-learn*.

3. Hasil dan Diskusi

3.1. Text Preprocessing

Penelitian ini dilakukan dengan memanfaatkan platform Google Colab serta sejumlah pustaka (*library*) Python. Proses yang diterapkan dalam metode penelitian ini diawali dengan pengumpulan data sebanyak 9.032 entri. Selanjutnya, dilakukan tahap pra-pemrosesan teks guna menyamakan format data serta meningkatkan kinerja model klasifikasi. Hasil dari tahapan preprocessing tersebut dapat dilihat pada Tabel 1.

Tabel 1. Preprocessing

No	Preprocessing	Teks
1	Lowercasing	aplikasi yang sangat mantap untuk menonton film semoga selalu menjadi yg terbaik.
2	Cleaning	aplikasi yang sangat mantap untuk menonton film semoga selalu menjadi yg terbaik
3	Tokenisasi (Split)	['aplikasi', 'yang', 'sangat', 'mantap', 'untuk', 'menonton', 'film', 'semoga', 'selalu', 'menjadi', 'yg', 'terbaik']
4	Stemming	['aplikasi', 'yang', 'sangat', 'mantap', 'untuk', 'tonton', 'film', 'moga', 'selalu', 'jadi', 'yg', 'baik']
5	Stopword Removal	aplikasi sangat mantap tonton film moga selalu jadi yg baik

Setelah proses pra-pemrosesan selesai, setiap ulasan berhasil dikonversi menjadi kumpulan token yang bersih dan terstruktur. Data tersebut kemudian diberi label sentimen berdasarkan nilai rating, di mana skor 1 hingga 3 diklasifikasikan sebagai 'Negatif', sedangkan skor 4 hingga 5 sebagai 'Positif'. Data yang telah dibersihkan dan diberi label ini selanjutnya siap digunakan dalam tahap ekstraksi fitur dan pembuatan model klasifikasi. Adapun hasil akhir dari tahapan text preprocessing dan pemetaan label ditampilkan pada Gambar 3 sebagai ilustrasi dari proses yang telah dilakukan.

[illegible]

Gambar 3. Pelabelan Data

3.2. Ekstraksi Fitur

Dalam tahap ekstraksi fitur, tujuan dari proses pembersihan adalah untuk mengubah teks yang telah dibersihkan menjadi bentuk data terstruktur yang dapat diolah oleh algoritma. Dalam penelitian ini, metode TF-IDF (Term Frequency-Inverse Document Frequency) digunakan untuk mengukur tingkat kepentingan suatu kata dalam sebuah dokumen tertentu. Implementasinya dilakukan menggunakan *TfidfVectorizer* dari pustaka *scikit-learn*, yang dikonfigurasi untuk menghasilkan kamus fitur berisi 5.000 kata yang paling sering muncul dalam korpus. Karena proses tokenisasi sudah dilakukan sebelumnya, vectorizer disesuaikan agar dapat langsung menerima input berupa token. Output dari tahap ini berupa sebuah matriks TF-IDF yang menggambarkan bobot masing-masing kata dalam setiap ulasan, dan siap digunakan dalam proses pemodelan selanjutnya.

```

1 X = df['cleaned_content_tokens']
2 y = df['sentimen']
3 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
4
5 def dummy_fun(doc):
6     return doc
7
8 tfidf = TfidfVectorizer(
9     max_features=5000,
10    tokenizer=dummy_fun,
11    preprocessor=dummy_fun,
12    token_pattern=None
13 )
14
15 X_train_vec = tfidf.fit_transform(X_train)
16 X_test_vec = tfidf.transform(X_test)

```

Gambar 4. Implementasi Ekstraksi Fitur TF-IDF

3.3. Pemodelan Support Vector Machine (SVM)

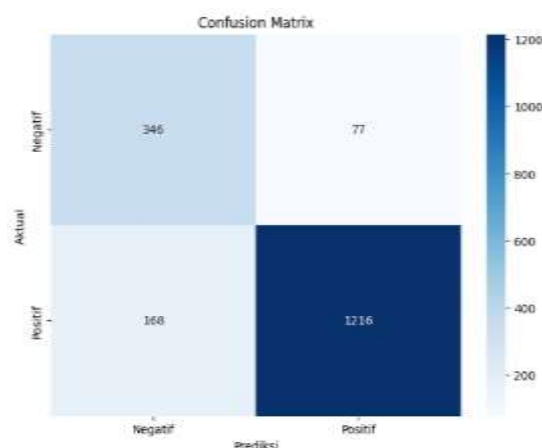
Model Support Vector Machine (SVM) dibangun dengan menggunakan bantuan *library* scikit-learn dalam bahasa pemrograman Python. Model ini dikonfigurasi dengan beberapa *hyperparameter*, yaitu menggunakan *kernel*='linear' dan *class_weight*='balanced' untuk menangani data tidak seimbang. Model kemudian dilatih menggunakan data *training* yang telah melalui proses ekstraksi fitur TF-IDF.

```

1 model = SVC(kernel='linear', class_weight='balanced')
2 model.fit(X_train_vec, y_train)
3
4 y_pred = model.predict(X_test_vec)
5
6 accuracy = accuracy_score(y_test, y_pred)
7 report = classification_report(y_test, y_pred)
8
9 print(f"Akurasi Model: {accuracy * 100:.2f}%\n")
10 print("Laporan Klasifikasi:")
11 print(report)

```

Gambar 4. Implementasi Support Vector Machine (SVM)



Gambar 4. Confusion Matrix

Selanjutnya, model yang telah dilatih diuji performanya menggunakan data uji. Hasil dari pengujian model SVM ini ditunjukkan oleh *confusion matrix* pada Gambar 4.

Tabel 2. Hasil Evaluasi

Kategori	Precision	Recall	F1-core	Akurasi
Negatif	67,00%	82,00%	74,00%	-
Positif	94,00%	88,00%	91,00%	-
Total (weighted Avg)	88,00%	86,00%	87,00%	86,00%

Berdasarkan Tabel 1, model *Support Vector Machine* (SVM) yang digunakan menunjukkan akurasi keseluruhan sebesar 86,00%, dengan performa yang sangat baik dalam mengklasifikasikan sentimen positif, ditunjukkan oleh nilai *precision* sebesar 94,00% dan *F1-score* sebesar 91,00%; sementara itu, pada klasifikasi sentimen negatif, performa model cenderung lebih rendah dengan *precision* sebesar 67,00% dan *F1-score* sebesar 74,00%, yang disebabkan oleh kondisi *imbalanced data* pada dataset, di mana jumlah ulasan positif jauh lebih banyak dibandingkan ulasan negatif, dan secara keseluruhan nilai *weighted average F1-score* tercatat sebesar 87,00%.

4. Kesimpulan

Hasil penelitian menunjukkan bahwa model *Support Vector Machine* (SVM) dengan ekstraksi fitur menggunakan metode *TF-IDF* berhasil diterapkan untuk analisis sentimen pada ulasan pengguna aplikasi Loklok. Selain itu, model ini menunjukkan kinerja yang cukup baik dengan akurasi sebesar 86,00%, dengan nilai *precision (weighted average)* sebesar 88,00%, *recall (weighted average)* sebesar 86,00%, dan *F1-score (weighted average)* sebesar 87,00%. Untuk pengembangan lebih lanjut, performa model ini masih dapat ditingkatkan melalui beberapa pendekatan, seperti melakukan *hyperparameter tuning* pada algoritma SVM, menerapkan teknik *resampling* seperti *SMOTE* untuk menangani permasalahan *imbalanced data*, serta membandingkannya dengan algoritma lain, baik dari kategori *machine learning* maupun *deep learning*.

Daftar Pustaka

- [1] Y. Zhang, Y. Du, and Y. Li, "Entertainment apps, limited attention and investment performance," *Front Psychol*, vol. 14, 2023, doi: 10.3389/fpsyg.2023.1118797.
- [2] "Review of the Lazada application on Google Play Store: sentiment analysis," *Journal of Computing Research and Innovation*, vol. 9, no. 1, Sep. 2024, doi: 10.24191/jcrinn.v9i1.412.
- [3] S. Azhari, N. Rahaningsih, R. Danar Dana, and S. Tinggi Manajemen Informatika Dan Komputer, "Peningkatan Akurasi Analisis Sentimen Pada Aplikasi Loklok Dengan Metode Naïve Bayes," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 1, pp. 2830–7062, doi: 10.23960/jitet.v12i3.5848.
- [4] M. Maulidah, A. Ardiansyah, L. Putri Gemilang, and N. Fitria Indriarti, "Analisis Sentimen Pada Ulasan Aplikasi Superbank Dengan Metode Support Vector Machine Dan Naive Bayes," *Indonesian Journal on Software Engineering (IJSE)*, vol. 10, no. 2, pp. 147–155, 2024, [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ijse>
- [5] J. A. Putra, A. Dharmawan, and J. Gondohanindijo, "Sentimen Analysis Digitalent Mobile Application Using Naïve Bayes And Svm With Tf-Idf Fitur Extraction," *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 7, no. 4, 2024.
- [6] K. Yuni Suratri *et al.*, "Analisis Sentimen Ulasan Traveloka Menggunakan Metode Naïve Bayes Classifier dan Information Gain," *JNATIA*, vol. 3, no. 1, 2024.
- [7] V. Fitriyana *et al.*, "Analisis Sentimen Ulasan Aplikasi Jamsostek Mobile Menggunakan Metode Support Vector Machine," 2023.
- [8] P. Studi Informatika, F. Matematika dan Ilmu Pengetahuan Alam, J. Raya Kampus UNUD, B. Jimbaran, and K. Selatan, "Analisis Sentimen Ulasan Aplikasi M-Paspor Menggunakan TF-IDF dan Support Vector Machine Ni Luh Putu Happy Nirmala a1 , Ida Bagus Gede Dwidasmara a2," *JNATIA*, vol. 3, no. 2, 2025.
- [9] E. Eskiyyaturofikoh and R. R. Suryono, "Analisis Sentimen Aplikasi X Pada Google Play

Store Menggunakan Algoritma Naïve Bayes Dan Support Vector Machine (SVM),” *JIP/ (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 3, pp. 1408–1419, Aug. 2024, doi: 10.29100/jipi.v9i3.5392.

Halaman ini sengaja dibiarkan kosong