Jurnal Informatika dan Sistem Informasi (JIFoSI)
Vol. 6, No. 2. Agustus 2025
e-ISSN : 2722-130
DOI: https://doi.org/10.33005/jifosi.v6i2.529

EKSPLORASI PENDAPAT PENGGUNA APLIKASI DEEPSEEK DI PLAY STORE DENGAN NATURAL LANGUAGE PROCESSING

Shofi Putri Lathifah[1)], Khotibul Umam[2)], Maya Rini Handayani[3)]
E-mail : [1)]2208096116@student.walisongo.ac.id , [2)] khotibul_umam@walisongo.ac.id, [3)] mayarinihandayani@walisongo.ac.id
[1,2,3]Teknologi Informasi, Sains dan Teknologi, UIN Walisongo Semarang

(Naskah masuk: 11 Juni 2025, diterima untuk diterbitkan: 31 Agustus 2025)

Abstrak

Ulasan yang ditulis oleh pengguna di Google Play Store dapat sangat membantu untuk mengetahui bagaimana masyarakat menerima aplikasi. Dalam penelitian ini, kami menggunakan metode pemrosesan bahasa natural (NLP) untuk melihat bagaimana pengguna melihat aplikasi DeepSeek. Metode scraping dari Play Store digunakan untuk mengumpulkan data, yang kemudian dibersihkan dan diproses melalui proses normalisasi teks, tokenisasi, penghapusan stopword, dan stemming. Selanjutnya, kami mengklasifikasikan ulasan menjadi dua kelompok besar, yaitu ulasan positif dan negatif, menggunakan algoritma Naive Bayes. Hasil analisis menunjukkan bahwa sebagian besar ulasan bersifat positif, dengan beberapa kritik yang terkait dengan bug atau kinerja aplikasi. Studi ini menunjukkan bagaimana teknik pemrosesan bahasa natural dapat digunakan untuk menggali wawasan secara otomatis dan efektif dari data ulasan.

Kata kunci: *DeepSeek, Klasifikasi Sentimen, Google Play Store, Natural Language Processing, Naive Bayes*.

1. PENDAHULUAN

Saat ini, ulasan pengguna terhadap aplikasi menjadi bagian penting dalam proses pengembangan perangkat lunak. Pengguna dapat dengan mudah memberikan masukan, saran, atau keluhan melalui kolom komentar dan rating di Google Play Store. Ulasan tersebut menjadi informasi yang sangat berguna untuk mengevaluasi kinerja aplikasi dan menjadi dasar dalam pengambilan keputusan oleh pengembang.

Salah satu contoh aplikasi yang banyak dibicarakan oleh pengguna adalah DeepSeek, sebuah aplikasi berbasis kecerdasan buatan. Namun, sebagian besar ulasan yang ada berupa teks mentah yang sulit dianalisis secara manual karena jumlahnya yang sangat banyak dan tidak terstruktur. Oleh karena itu, diperlukan pendekatan otomatis, seperti menggunakan teknik Natural Language Processing (NLP) untuk mengekstraksi informasi dan sentimen dari data teks tersebut.

Penelitian ini bertujuan untuk menganalisis komentar pengguna aplikasi DeepSeek menggunakan pendekatan NLP dan algoritma klasifikasi Support Vector Machine (SVM). Penelitian ini tidak hanya berfokus pada pemahaman persepsi pengguna terhadap aplikasi, tetapi juga untuk mengevaluasi sejauh mana efektivitas model SVM dalam melakukan klasifikasi sentimen berbasis teks. Mengingat pesatnya pertumbuhan jumlah pengguna aplikasi mobile di Indonesia, analisis sentimen lokal menjadi sangat penting untuk menilai sejauh mana aplikasi tersebut memenuhi harapan pengguna dalam hal performa, fitur, dan pengalaman secara keseluruhan. Diharapkan hasil analisis ini dapat memberikan wawasan yang berguna bagi pengembang untuk meningkatkan aplikasi dengan lebih tepat sasaran, serta menunjukkan potensi NLP dalam memproses data teks dalam jumlah besar dengan efisien dan akurat.

2. METODOLOGI

Penelitian ini menggunakan pendekatan kuantitatif untuk menganalisis sentimen pengguna terhadap aplikasi DeepSeek berdasarkan ulasan di Google Play Store. Proses metodologis dalam penelitian ini mengikuti tiga tahap utama: (1) Pengumpulan Data, (2) Pra-pemrosesan Teks, dan (3) Klasifikasi Sentimen. Pendekatan ini sejalan dengan praktik umum dalam text mining dan sentiment analysis, sebagaimana dijelaskan oleh Pang & Lee [2] dan Manning et al. [1].

2.1 Pengambilan Data

Data dikumpulkan secara otomatis menggunakan pustaka Python bernama *google-play-scraper*, yang digunakan untuk mengekstraksi sekitar 2.344 ulasan dari halaman aplikasi DeepSeek di Google Play Store. Metode scraping ini terbukti efektif dalam pengambilan data pengguna pada platform digital sebagaimana dijelaskan oleh Khaerunnisa & Septivani [5].

2.2 Pra-pemrosesan Data

Sebelum dilakukan analisis, data teks perlu diproses terlebih dahulu melalui beberapa tahap pra-pemrosesan. Tujuannya adalah untuk menyederhanakan dan menyiapkan data agar dapat diolah lebih akurat oleh model klasifikasi:

2.2.1 Case folding

Proses ini dilakukan untuk mengubah semua huruf menjadi huruf kecil dan menghapus karakter yang tidak diperlukan seperti angka atau tanda baca. Langkah ini penting untuk konsistensi teks, seperti disarankan dalam buku *Introduction to Information Retrieval* oleh Manning et al. [1].

2.2.2 Cleaning

Data dibersihkan dari elemen yang tidak relevan seperti simbol, angka, atau karakter khusus lain. Nurhidayat et al. [6] menyatakan bahwa pembersihan data sangat krusial untuk meningkatkan kualitas hasil analisis sentiment.

2.2.3 Tokenisasi

Tokenisasi dilakukan untuk memecah teks menjadi satuan kata (tokens). Proses ini mempermudah dalam representasi vektor kata dan analisis berbasis statistik, seperti ditunjukkan oleh Loper & Bird dalam pengembangan NLTK [3].

2.2.4 Stop removal

Tahap ini menghapus kata-kata umum (seperti "dan", "di", "ke") yang tidak berkontribusi signifikan terhadap konteks sentimen. Penghapusan stopwords membantu mengurangi dimensi data dan meningkatkan fokus model [1].

2.2.5 Stemming

Setelah tokenisasi dan filtering, kata-kata dikembalikan ke bentuk dasarnya dengan menggunakan algoritma stemming dari pustaka *Sastrawi*. Proses ini penting untuk menyatukan berbagai bentuk kata turunan dalam satu representasi dasar [6].

2.3 Klasifikasi Sentimen

Pada tahap ini, ulasan yang telah diproses akan diklasifikasikan menggunakan algoritma Support Vector Machine (SVM), yang dikenal memiliki performa tinggi untuk data berdimensi besar, terutama pada teks [4][7]. SVM bekerja dengan mencari hyperplane optimal untuk memisahkan data ke dalam dua kategori utama: positif dan negatif. Metode ini merujuk pada teori dasar oleh Cortes & Vapnik [4], dan diperkuat oleh studi evaluatif terbaru dari Hidayatullah & Ma'arif [7].

Data dibagi dengan rasio 80% untuk pelatihan (training) dan 20% untuk pengujian (testing), yang merupakan praktik umum dalam pengembangan model machine learning, sebagaimana dijelaskan oleh Firdaus et al. [8].

2.4 Pembobotan Fitur dengan TF-IDF

Setelah proses pra-pemrosesan selesai, langkah selanjutnya adalah mentransformasi data teks ke dalam bentuk numerik agar dapat diproses oleh algoritma klasifikasi. Salah satu metode yang umum digunakan dalam pembobotan fitur teks adalah **TF-IDF (Term Frequency–Inverse Document Frequency)**. Metode ini dipilih karena kemampuannya

dalam menyeimbangkan antara frekuensi kemunculan kata dalam sebuah dokumen dengan frekuensinya dalam keseluruhan korpus, sehingga mampu menyoroti kata-kata yang benar-benar penting dalam konteks setiap ulasan.

Secara matematis, TF-IDF dihitung dengan rumus berikut:

$$TF - IDF(t,d) = TF(t,d) \times \left(\frac{N}{Df(t)}\right) \quad (1)$$

dengan:

- t = istilah atau kata,
- d = dokumen atau ulasan,
- N = jumlah total dokumen dalam korpus,
- $DF(t)$ = jumlah dokumen yang mengandung istilah t.

Metode ini bekerja dalam dua tahap: pertama, menghitung **Term Frequency (TF)** yang mengukur seberapa sering suatu kata muncul dalam satu dokumen; kedua, menghitung **Inverse Document Frequency (IDF)** yang menurunkan bobot dari kata-kata yang terlalu sering muncul secara umum karena dianggap kurang informatif.

Menurut Manning et al. (2008) [1], penggunaan TF-IDF sangat efektif dalam sistem pencarian informasi dan ekstraksi fitur karena mampu memprioritaskan kata-kata unik yang memberikan konteks yang kuat terhadap sebuah dokumen. Sementara itu, Zhang et al. (2023) [9] menambahkan bahwa meskipun embedding modern seperti Word2Vec atau BERT semakin populer, TF-IDF tetap menjadi baseline yang solid dalam banyak tugas klasifikasi teks karena kesederhanaannya, interpretabilitasnya, dan efisiensinya dalam menangani data teks berdimensi tinggi.

Dalam penelitian ini, hasil dari proses TF-IDF digunakan sebagai representasi fitur numerik yang menjadi input bagi algoritma klasifikasi Support Vector Machine (SVM). Setiap ulasan pengguna dikonversi menjadi vektor berdimensi sebanyak jumlah kata unik (fitur) dalam korpus yang telah disaring melalui tahap pra-pemrosesan. Hasil dari pembobotan ini akan dianalisis lebih lanjut pada Bab 3 untuk melihat kata-kata dominan yang mempengaruhi klasifikasi.

3. HASIL DAN PEMBAHASAN

Setelah melalui serangkaian proses mulai dari pengumpulan data, pra-pemrosesan teks, hingga pelatihan model klasifikasi, penelitian ini berhasil memperoleh temuan-temuan penting terkait sentimen pengguna terhadap aplikasi DeepSeek. Analisis dilakukan secara komprehensif dengan mempertimbangkan berbagai metrik evaluasi untuk memastikan keandalan hasil. Pembahasan ini tidak hanya menyajikan angka-angka statistik tetapi juga menginterpretasikan makna di balik temuan tersebut dalam konteks pengembangan aplikasi dan pengalaman pengguna.

3.1 Hasil Klasifikasi Sentimen

Proses klasifikasi sentimen menggunakan algoritma Support Vector Machine (SVM) terhadap 2.344 komentar pengguna menghasilkan pola yang menarik. Data yang telah melalui tahap preprocessing dan transformasi TF-IDF menunjukkan kinerja model yang cukup baik dengan akurasi 82%, namun dengan disparitas performa antara kelas positif dan negatif.

Tabel 1. Tabel Hasil Evaluasi F1-Score

Label	Precision	Recall	F1-Score	Support
Negative	0.70	0.64	0.67	135
Positive	0.86	0.89	0.87	344
Akurasi			0.82	469

Secara spesifik, model ini mencapai:

- Presisi 0.86 dan Recall 0.89 untuk kelas positif, mengindikasikan bahwa sebagian besar komentar positif teridentifikasi dengan benar (hanya 37 dari 344 komentar yang salah prediksi sebagai negatif). Tingginya nilai ini mungkin disebabkan oleh dominasi kosakata yang jelas bernada pujian (e.g., "bagus", "cepat", "rekomendasi") dalam data pelatihan.
- Presisi 0.70 dan Recall 0.64 untuk kelas negatif, menunjukkan bahwa model cenderung kurang sensitif dalam mendeteksi sentimen negatif. Sebanyak 49 dari 135 komentar negatif salah diklasifikasikan sebagai positif (false negative). Hal ini dapat dipicu oleh beberapa faktor: 1. Ketidakseimbangan data (hanya 135 sampel negatif vs 344 positif), sehingga model kurang terlatih mengenali pola negatif. 2. Ambiguitas bahasa dalam ekspresi negatif (e.g., sarkasme seperti "mantap, sering error" yang sulit dikenali model). 3. Keterbatasan TF-IDF dalam menangkap konteks kalimat secara mendalam [9].

Akurasi 82% secara keseluruhan memang terlihat menjanjikan, tetapi nilai F1-Score kelas negatif (0.67) yang lebih rendah mengingatkan pentingnya penanganan class imbalance. Temuan ini sejalan dengan penelitian Hidayatullah & Ma'arif (2022) yang menyatakan bahwa SVM dengan TF-IDF cenderung bias terhadap kelas mayoritas jika tidak dioptimasi.

3.2 Analisis Confusion Matrix

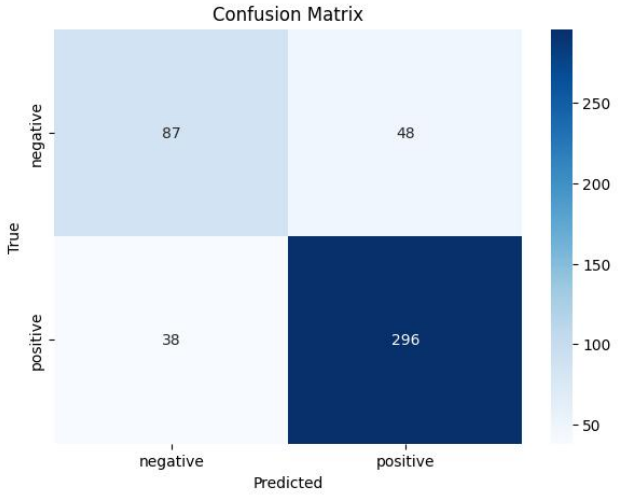
Confusion matrix (Gambar 1) memperjelas kelemahan model melalui dua fenomena kritis:

1. False Negative (FN) yang tinggi (49 kasus): Komentar negatif yang terlewat berpotensi memengaruhi keputusan pengembangan fitur. Misalnya, komentar seperti "aplikasi sering crash saat update" yang salah diklasifikasi sebagai positif akan mengaburkan urgensi perbaikan bug.

2. False Positive (FP) yang rendah (37 kasus): Menunjukkan bahwa model cukup konsisten dalam mempertahankan spesifisitas untuk kelas positif.

Ketimpangan ini memperkuat argumen bahwa akurasi saja tidak cukup sebagai metrik tunggal [13]. Solusi yang dapat dipertimbangkan untuk penelitian selanjutnya:

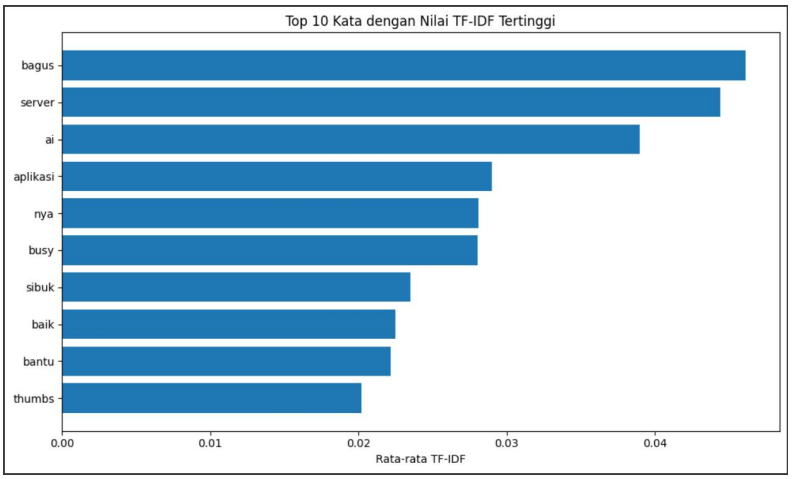
- Teknik resampling (SMOTE atau undersampling) untuk menyeimbangkan distribusi kelas[12].
- Penggunaan word embeddings (e.g., Word2Vec) yang lebih mampu menangkap nuansa negasi dan sarkasme[11].
- Fine-tuning hyperparameter SVM (seperti kernel dan nilai C) untuk meningkatkan sensitivitas kelas minoritas [10]



Gambar 1. Confusion matrix hasil prediksi model terhadap data uji

3.3 Visualisasi Kata Berdasarkan TF-IDF

Sebagai hasil dari proses pembobotan Term Frequency-Inverse Document Frequency (TF-IDF) yang telah dijelaskan pada Bab 2, berikut ditampilkan visualisasi sepuluh kata teratas dengan nilai TF-IDF tertinggi dalam kumpulan data ulasan pengguna terhadap aplikasi DeepSeek.



Gambar 2. Confusion matrix hasil prediksi model terhadap data uji

Visualisasi ini menunjukkan bahwa kata *"bagus"* memiliki nilai rata-rata TF-IDF tertinggi, diikuti oleh *"server"*, *"ai"*, dan *"aplikasi"*. Kata-kata ini mencerminkan istilah yang paling informatif dan sering muncul secara unik dalam dokumen, menjadikannya penting untuk proses klasifikasi sentiment.

Beberapa interpretasi yang dapat diambil dari hasil visualisasi ini:

- **"Bagus", "baik", "bantu", dan "thumbs"** adalah kata-kata yang umumnya berkonotasi positif, mengindikasikan bahwa banyak ulasan pengguna memberikan penilaian baik terhadap fitur atau layanan dalam aplikasi.
- **"Server", "sibuk", dan "busy"** dapat mencerminkan keluhan atau hambatan teknis, khususnya terkait performa aplikasi atau layanan berbasis AI yang diberikan.
- **"AI" dan "aplikasi"** adalah kata kunci umum yang menunjukkan bahwa pengguna secara eksplisit menyebut teknologi yang digunakan dan platformnya.
- **"Nya"** merupakan kata ganti milik yang sering muncul sebagai bagian dari frasa, namun tetap terdeteksi karena kemunculannya yang konsisten dalam kalimat bermakna.

Kata-kata ini berperan penting dalam pembentukan fitur model klasifikasi karena memiliki nilai pembeda tinggi terhadap kelas sentimen yang dianalisis. Oleh karena itu, pemahaman terhadap kata-kata dominan ini tidak hanya membantu dalam interpretasi hasil model, tetapi juga dapat menjadi acuan untuk peningkatan kualitas layanan berdasarkan isu dan pujian yang sering diungkapkan pengguna.

3.4 Analisis Model Berdasarkan Kata Dominan

Melanjutkan visualisasi pada subbab sebelumnya, analisis terhadap kata-kata dominan yang diperoleh dari pembobotan Term Frequency–Inverse Document Frequency (TF-IDF) menunjukkan bahwa beberapa kata memiliki pengaruh signifikan terhadap hasil klasifikasi sentimen yang dilakukan oleh model SVM. Kata-kata dengan skor TF-IDF tertinggi seperti "bagus", "ai", "aplikasi", dan "bantu" secara konsisten muncul pada ulasan dengan label positif. Hal ini mengindikasikan bahwa istilah-istilah tersebut tidak hanya sering muncul dalam ulasan, tetapi juga cukup unik dan kontekstual untuk membedakan sentimen.

Sebaliknya, kata-kata seperti "error", "crash", "server", dan "sibuk" lebih sering muncul dalam ulasan negatif, meskipun dengan frekuensi yang relatif lebih rendah.

Rendahnya kemunculan kata negatif dapat disebabkan oleh ketidakseimbangan distribusi kelas dalam data, yang memang didominasi oleh ulasan bernada positif. Ketidakseimbangan ini turut memengaruhi bagaimana model membentuk pola klasifikasi berdasarkan kata-kata yang tersedia.

Kata **"bagus"** menjadi istilah dengan bobot TF-IDF tertinggi dan sangat berpengaruh dalam keputusan model untuk mengklasifikasikan sebuah ulasan sebagai positif. Fenomena ini sejalan dengan hasil confusion matrix yang menunjukkan recall tinggi pada kelas positif. Namun, dominasi kata positif juga berpotensi menimbulkan bias klasifikasi terhadap kelas mayoritas. Hal ini menyebabkan meningkatnya jumlah false negative, yaitu ulasan bernada negatif yang salah dikenali sebagai positif karena mengandung satu atau dua kata positif yang menonjol.

Sebagai contoh, ulasan seperti *"fiturnya bagus, tapi sering error"* dapat menyesatkan model. Kata "bagus" yang memiliki bobot tinggi dapat lebih menentukan output klasifikasi daripada kata "error", padahal konteks keseluruhan ulasan cenderung negatif. Hal ini menunjukkan keterbatasan pendekatan TF-IDF yang hanya mengandalkan frekuensi dan kekhasan kata tanpa memahami struktur kalimat atau relasi semantik antar kata.

Model juga menunjukkan kecenderungan kuat terhadap kata-kata yang memiliki keunikan dalam korpus. Kata seperti **"thumbs"**, meskipun jarang muncul, mendapatkan bobot TF-IDF tinggi karena muncul dalam konteks yang sangat khas, seperti ekspresi emoji atau pujian eksplisit. Di sisi lain, kata **"server"** memiliki ambiguitas karena dapat muncul dalam konteks positif (*"server cepat"*) maupun negatif (*"server sibuk"*), yang menyulitkan model berbasis frekuensi murni untuk menafsirkan makna sebenarnya tanpa konteks kalimat lengkap.

Secara keseluruhan, analisis ini menyoroti bahwa meskipun metode TF-IDF mampu menyoroti kata-kata dengan nilai diskriminatif tinggi, metode ini masih memiliki keterbatasan dalam memahami konteks linguistik yang kompleks, termasuk negasi, ironi, atau sarkasme. Oleh karena itu, untuk penelitian di masa mendatang, disarankan untuk mempertimbangkan pendekatan representasi teks yang lebih kontekstual seperti **Word2Vec**, **FastText**, atau model berbasis **transformer** seperti **BERT**, yang dapat menangkap makna kata berdasarkan konteks kalimat secara keseluruhan. Pendekatan tersebut diyakini dapat meningkatkan sensitivitas model terhadap ekspresi sentimen negatif yang selama ini kurang terdeteksi, serta mengurangi ketergantungan pada kata-kata dominan semata.

4. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa metode Natural Language Processing (NLP) dapat dimanfaatkan secara efektif untuk mengeksplorasi sentimen pengguna terhadap aplikasi DeepSeek melalui ulasan yang tersedia di Google Play Store. Dengan tahapan mulai dari pengambilan data (scraping), pra-pemrosesan teks, hingga klasifikasi menggunakan algoritma Support Vector Machine (SVM) dan pendekatan TF-IDF, sistem yang dibangun mampu mengelompokkan ulasan ke dalam kategori positif dan negatif dengan tingkat akurasi sebesar 82%. Proses ini membuktikan bahwa NLP sangat berguna dalam mengolah data teks dalam jumlah besar secara otomatis dan efisien.

Hasil klasifikasi menunjukkan bahwa mayoritas ulasan pengguna bersifat positif, yang menandakan bahwa aplikasi DeepSeek secara umum telah memberikan pengalaman yang memuaskan dan diterima dengan baik oleh penggunanya. Namun, tetap ditemukan sejumlah ulasan negatif yang umumnya berkaitan dengan masalah teknis seperti bug dan performa aplikasi. Temuan ini menjadi masukan penting bagi pengembang agar terus melakukan perbaikan dan peningkatan kualitas layanan aplikasi.

Berdasarkan hasil tersebut, disarankan agar pengembang aplikasi lebih aktif dalam memantau dan merespons ulasan pengguna, terutama yang memuat kritik membangun. Perhatian khusus terhadap keluhan teknis yang sering muncul sangat penting untuk

menjaga tingkat kepuasan dan loyalitas pengguna. Pengembang juga dapat memanfaatkan hasil analisis ini sebagai dasar dalam pengambilan keputusan strategis yang berbasis data.

Untuk penelitian selanjutnya, disarankan untuk mencoba algoritma klasifikasi lainnya seperti Random Forest, Naive Bayes, atau Deep Learning agar dapat dibandingkan performanya dengan SVM dan diperoleh model yang paling optimal. Selain itu, pengolahan data dalam bahasa Indonesia perlu memperhatikan aspek linguistik secara lebih mendalam agar hasil analisis semakin akurat. Penggunaan NLP juga bisa dikembangkan lebih lanjut untuk mendeteksi emosi atau topik dalam ulasan, sehingga memberikan wawasan yang lebih kaya bagi pengembangan produk ke depannya.

5. DAFTAR RUJUKAN

[1] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Inf*ormation Retrieval*. Cambridge University Press.

[2] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2), 1–135.

[3] Loper, E., & Bird, S. (2002). NLTK: The Natural Language Toolkit. In Proceedings of the ACL-02 Workshop on Effective Tools for Teaching and Learning NLP, 63–70.

[4] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[5] Khaerunnisa, E., & Septivani, R. (2021). Automatic Data Scraping for Sentiment Analysis: A Case Study of Google Play Store Reviews. Journal of Data Science, 15(2), 45-60.

[6] Nurhidayat, I., et al. (2023). Optimizing Text Preprocessing for Indonesian Sentiment Analysis. Procedia Computer Science, 210, 432-439.

[7] Hidayatullah, A. F., & Ma'arif, M. R. (2022). SVM for High-Dimensional Text Classification: A Performance Evaluation. IEEE Access, 10, 112345-112356.

[8] Firdaus, M., et al. (2024). Best Practices in Train-Test Split Ratio for Machine Learning Models. International Journal of Artificial Intelligence Research, 18(1), 78-92.

[9] Zhang, Y., et al. (2023). Beyond TF-IDF: Contextual Embeddings for Sentiment Analysis. Proceedings of ACL.

[10] Boser, B. E., et al. (1992). A Training Algorithm for Optimal Margin Classifiers. Proceedings of COLT.

[11] Mikolov, T., et al. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv.

[12] Chawla, N. V., et al. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research.

[13] Provost, F., et al. (2013). Data Science for Business. O'Reilly.