

PERFORMANCE COMPARISON OF SVM, NAIVE BAYES, AND LOGISTIC REGRESSION CLASSIFICATION ALGORITHMS IN ANALYZING NOICE APP USER REVIEWS

Ahmad Bahar^{*1}, Tri Astuti², Primandani Arsi³

^{1,2,3}Informatics, Faculty of Computer Science, Universitas Amikom Purwokerto, Indonesia
Email: ¹ahmadbahar911@gmail.com, ²tri_astuti@amikompurwokerto.ac.id,
³ukhti.prima@amikompurwokerto.ac.id

(Article received: May 7, 2024; Revision: June 14, 2024; published: August 09, 2024)

Abstract

In the rapidly growing digital era, user reviews on distribution platforms such as the Google Play Store are a key indicator in assessing the popularity, quality, and user satisfaction of applications. This study aims to compare the performance of SVM, Naive Bayes, and Logistic Regression classification algorithms in analyzing user reviews of the Noice app, an audio content platform. The research involves steps such as data collection, data pre-processing, word embedding, modeling, model evaluation, and sentiment analysis. Testing was conducted using 1877 data. The data from the reviews were divided into scenarios, with training and testing data divided in ratios of 90:10, 80:20, and 70:30. The results showed that the SVM algorithm achieved the highest accuracy rate (80%) in the 90:10 data split scenario. However, Naive Bayes also showed competitive results with 78% accuracy in the same scenario. Meanwhile, Logistic Regression achieved 78% accuracy when the data was split in an 80:20 ratio. Evaluation was done using metrics such as accuracy, precision, recall, and F1-score. Sentiment analysis showed a positive trend with 1194 positive data compared to 683 negative data. From the comparison of data sharing scenarios and algorithms, SVM at 90:10 data sharing gave the best results.

Keywords: *Logistic Regression, Naive Bayes, Noice, SVM.*

PERBANDINGAN KINERJA ALGORITMA KLASIFIKASI SVM, NAIVE BAYES, DAN LOGISTIC REGRESSION DALAM ANALISIS ULASAN PENGGUNA TERHADAP APLIKASI NOICE

Abstrak

Dalam era digital yang berkembang pesat, ulasan pengguna di platform distribusi seperti Google Play Store menjadi indikator utama dalam menilai popularitas, kualitas, dan kepuasan pengguna terhadap aplikasi. Penelitian ini bertujuan untuk membandingkan kinerja algoritma klasifikasi SVM, Naive Bayes, dan Logistic Regression dalam menganalisis ulasan pengguna terhadap aplikasi Noice, sebuah platform konten audio. Penelitian ini melibatkan langkah-langkah seperti pengumpulan data, pra-pemrosesan data, embedding kata, pemodelan, evaluasi model, dan analisis sentimen. Pengujian dilakukan dengan menggunakan 1877 data. Data dari ulasan tersebut dibagi menjadi beberapa skenario, dengan data pelatihan dan pengujian dibagi dalam rasio 90:10, 80:20, dan 70:30. Hasil penelitian menunjukkan bahwa algoritma SVM mencapai tingkat akurasi tertinggi (80%) dalam skenario pembagian data 90:10. Namun, Naive Bayes juga menunjukkan hasil yang kompetitif dengan akurasi 78% dalam skenario yang sama. Sedangkan, Logistic Regression mencapai akurasi 78% ketika data dibagi dalam rasio 80:20. Evaluasi dilakukan dengan menggunakan metrik seperti akurasi, presisi, recall, dan F1-score. Analisis sentimen menunjukkan kecenderungan positif dengan 1194 data positif dibandingkan dengan 683 data negatif. Dari perbandingan skenario pembagian data dan algoritma, SVM pada pembagian data 90:10 memberikan hasil terbaik.

Kata kunci: *Logistic Regression, Naive Bayes, Noice, SVM.*

1. PENDAHULUAN

Dalam era digital yang berkembang pesat, ulasan pengguna di platform distribusi seperti Google

Play Store menjadi salah satu indikator utama dalam menilai popularitas, kualitas, dan kepuasan pengguna terhadap sebuah aplikasi. Jutaan ulasan yang diposting setiap hari memberikan gambaran yang

berharga tentang bagaimana pengguna menanggapi dan mempersepsikan berbagai fitur, layanan, dan pengalaman yang ditawarkan oleh aplikasi.

Aplikasi Noice, sebuah platform konten audio yang berfokus pada podcast, radio, audiobook, dan audioseries, juga tidak luput dari pengaruh ulasan pengguna di Google Play Store. Pada tahun 2018, Noice memasuki panggung sebagai salah satu penyedia audio terkemuka di Indonesia, menawarkan berbagai macam konten audio berkualitas. Hingga kini, lebih dari 2 Juta pengguna telah menikmati beragam podcast, audiobook, dan serial orisinal yang ditawarkan oleh platform ini [1]. Pada tahun 2023, Noice, platform streaming konten audio lokal, diakui oleh Google Play sebagai salah satu aplikasi lokal terkemuka. Penghargaan ini menegaskan komitmen Noice sebagai platform lokal yang berdedikasi pada pengembangan ekosistem konten audio di Indonesia. Mereka fokus pada kualitas, kreativitas, dan relevansi konten yang mencerminkan kekhasan budaya lokal [2]. Selanjutnya, Noice telah menambahkan fitur tambahan, yakni kemampuan untuk membuat dan menayangkan video podcast dan livestream. Fitur ini dirancang untuk membantu para kreator memperkuat hubungan dengan para pendengar mereka. Selain itu, ada juga fitur VIP dan Premium di Noice, yang merupakan opsi berlangganan yang bebas iklan. Dengan berlangganan ini, pengguna dapat menikmati konten-konten khusus yang tersedia di platform Noice, seperti dari Musuh Masyarakat, Berizik, Ruang 28, dan Trio Kurnia. Dengan penambahan fitur baru dan peningkatan jumlah pengguna yang terus meningkat, penting bagi pengembang untuk memahami dengan cermat sentimen dan pendapat pengguna terhadap layanan yang mereka tawarkan. Untuk memastikan kualitas layanan yang disediakan tetap terjaga, ulasan dari pengguna aplikasi bisa memberikan wawasan berharga tentang pengalaman mereka saat menggunakan aplikasi tersebut.

Jumlah ulasan yang melimpah di internet menyulitkan proses ekstraksi informasi jika dilakukan secara manual dengan membaca seluruh ulasan satu per satu. Maka, diperlukan pendekatan berbeda untuk mendapatkan informasi dari ulasan-ulasan yang ada. Analisis sentimen adalah metode yang digunakan untuk memahami opini yang disampaikan oleh pengguna melalui platform media sosial. Pengguna memiliki kebebasan untuk menyatakan pandangan, opini, dan perasaan mereka tentang berbagai topik populer dan peristiwa yang sedang trending. Analisis sentimen membantu dalam mengekstraksi informasi yang terkandung dalam postingan yang dipublikasikan. Proses ini melibatkan pemahaman dan pengukuran emosi yang tercermin dalam teks yang ditulis oleh pengguna. Media sosial memfasilitasi interaksi antar pengguna di seluruh dunia dan memungkinkan mereka untuk berbagi pendapat tentang topik yang relevan. Analisis sentimen dapat dimanfaatkan untuk meningkatkan layanan pelanggan, strategi pemasaran, serta sebagai

indikator kinerja media sosial. Dalam beberapa tahun terakhir, peran media sosial dalam kehidupan sehari-hari semakin signifikan, bahkan menjadi sumber informasi tentang peristiwa besar maupun kecil. Pengguna tidak hanya menyampaikan konten tentang acara yang mereka ikuti, tetapi juga mengekspresikan perasaan yang mereka alami terkait konten tersebut [3].

Penelitian yang dilakukan memberikan ikhtisar terhadap beberapa penelitian sebelumnya yang telah dilakukan dalam domain analisis sentimen. Menurut penelitian sebelumnya yang berjudul analisis sentimen terhadap aplikasi ruangguru menggunakan algoritma naive bayes, random forest dan support vector machine menerapkan analisis sentimen pada aplikasi ruangguru sebagai salah satu platform pembelajaran daring yang populer, terutama di kalangan siswa Indonesia. Penelitian ini telah menghasilkan hasil bahwa dari model klasifikasi Random Forest dengan menggunakan Cross Validation, mencapai tingkat akurasi sebesar 97,16% dan skor AUC sebesar 0,996. Kemudian, akurasi dengan model klasifikasi Support Vector Machine mencapai tingkat akurasi sebesar 96,01% dengan nilai AUC sebesar 0,543, dan akurasi pada pengujian model klasifikasi Naive Bayes mencapai 94,16% dengan skor AUC sebesar 0,999 [4]. Pada penelitian yang berjudul perbandingan algoritma klasifikasi support vector machine, random forest dan logistic regression pada ulasan shopee penelitian ini dilaksanakan dengan tujuan menganalisis pendapat pengguna aplikasi shopee di Indonesia. Hasil pengujian menunjukkan bahwa tingkat akurasi algoritma Logistic Regression sebesar 86%, diikuti oleh Support Vector Machine (SVM) dengan 91%, dan Random Forest mencapai 94% [5].

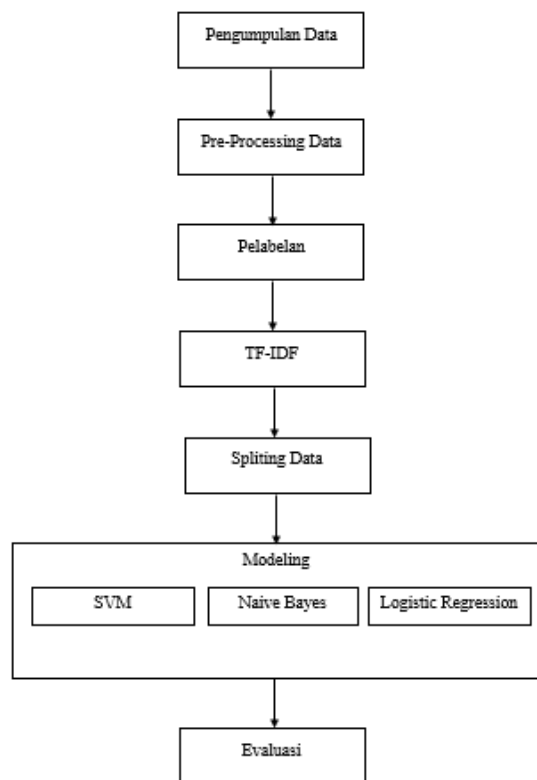
Penelitian terdahulu yang telah dilakukan tentang analisis sentimen pada ulasan aplikasi Gojek di Google Play Store [6]. Berdasarkan hasil penelitian, performa dari empat algoritma yang diuji yaitu Logistic Regression, MultinomialNB, SVM, dan K-NN. Di antara algoritma tersebut, Logistic Regression menunjukkan performa paling unggul secara keseluruhan. Skor performa akurasi, presisi, recall, dan F1-Score dari Logistic Regression secara berurutan adalah 82,45%, 82,49%, 82,45%, dan 82,43%. Selain itu, penelitian analisis sentimen juga telah dilakukan pada Twitter terkait aplikasi PeduliLindungi menggunakan algoritma Naïve Bayes dan Support Vector Machine [7]. Hasil klasifikasi menggunakan metode Naïve Bayes dan Support Vector Machine menunjukkan bahwa Support Vector Machine memiliki akurasi lebih tinggi, yaitu 91%, sedangkan Naïve Bayes memiliki akurasi sebesar 90%.

Berdasarkan literatur sebelumnya, dalam penelitian ini akan menerapkan beberapa algoritma machine learning seperti Support Vector Machine, Logistic Regresion, serta Naïve Bayes. Support Vector Machine merupakan metode machine learning

yang bertujuan untuk menemukan hyperplane optimal yang dapat memisahkan dua kelas dalam ruang input [8]. Keunggulan Logistic Regression adalah kemampuan interpretasi model yang tinggi, memungkinkan pemahaman yang jelas tentang hubungan antara fitur dan probabilitas kelas [9]. Naive Bayes adalah metode klasifikasi yang menggunakan perhitungan probabilitas berdasarkan teorema Bayes untuk menentukan kelas yang optimal dari setiap kelompok atribut [10].

Penelitian ini bertujuan untuk menggabungkan machine learning dengan menggunakan algoritma SVM, Naive Bayes, dan *Logistic Regression* berdasarkan tingkat akurasi model dalam berbagai skenario pembagian data untuk memperoleh informasi mengenai pengalaman pengguna dalam menggunakan aplikasi Noice. Dengan menerapkan analisis sentimen, ulasan dapat diklasifikasikan menjadi ulasan dengan sentimen positif dan negatif. Aplikasi yang terpilih dalam penelitian ini dipilih berdasarkan kriteria yang telah ditetapkan oleh peneliti, dengan harapan aplikasi tersebut dapat memberikan kontribusi dalam pengembangan aplikasi yang lebih baik. Selain itu, algoritma yang menunjukkan hasil terbaik dalam penelitian ini diharapkan dapat menjadi sumber inovasi untuk penelitian mendatang yang akan dilakukan.

2. METODE PENELITIAN



Gambar 1. Metode Penelitian

Penelitian ini melibatkan beberapa langkah proses. Data dan informasi obyektif akan menjadi landasan dalam penelitian ini, diharapkan penelitian yang dihasilkan akan memiliki kualitas yang baik.

gambar 1 menggambarkan langkah-langkah proses seperti pengumpulan data, pra-pemrosesan data, embedding kata, pemodelan, evaluasi model, dan analisis sentimen.

2.1. Pengumpulan data

Google Play adalah platform multiplatform yang dapat diakses baik melalui perangkat seluler maupun situs web. Dalam penelitian ini, data ulasan aplikasi "Noice" diambil melalui situs web Google Play. Teknik yang diterapkan dalam mengambil data ulasan aplikasi "Noice" adalah teknik web scraping. Data yang diambil melalui proses web scraping berasal dari halaman web dengan tool yang disediakan Google Colab. Setelah berhasil diambil, data akan disimpan dalam format CSV.

2.2. Pre-Processing Data

Tahap Pre-Processing diperlukan untuk menghilangkan data yang tidak perlu, sehingga memungkinkan metode klasifikasi SVM, Naive Bayes dan Logistic Regression bekerja lebih optimal dalam perhitungannya. Tahap ini melibatkan pengenalan konten dan struktur teks yang disesuaikan dengan pengelompokannya [11]. Berikut adalah langkah-langkah pra-pemrosesan:

- *Cleaning*. Ini adalah langkah untuk membersihkan kata-kata yang tidak diperlukan guna mengurangi gangguan. Kata-kata yang dihilangkan termasuk URL, tanda pagar (#), nama pengguna (@username), dan alamat surel, serta tanda baca.
- *Case folding*. Ini adalah tahap mengubah bentuk kata menjadi bentuk yang sama, baik itu seluruhnya huruf kecil atau huruf besar.
- *Tokenizing*. Proses ini bertujuan untuk memahami kata-kata dalam teks dengan membaginya menjadi urutan-urutan yang terpisah oleh spasi atau karakter khusus.
- *Filtering*. Tahap ini berfungsi untuk menghilangkan kata-kata umum yang sering muncul dan kurang relevan dengan teks. Kata-kata yang akan dihapus telah ditetapkan dalam *stop word list*.
- *Stemming*. Ini adalah langkah untuk mengembalikan kata-kata yang telah diubah menjadi bentuk asalnya.

2.3. Pelabelan

Langkah berikutnya adalah memberi label pada data. Dalam penelitian ini, proses pelabelan didasarkan pada rating, di mana rating dengan skor 1-3 mengindikasikan ketidakpuasan pengguna dan diberi label "negatif", sedangkan rating dengan skor 4-5 menandakan kepuasan pengguna dan diberi label "positif" [12].

2.4. TF-IDF

TF-IDF (*Term Frequency–Inverse Document Frequency*) adalah teknik pembobotan yang sering digunakan dalam pencarian informasi dan penambangan data. TF atau frekuensi term merujuk pada seberapa sering sebuah kata muncul dalam sebuah teks, sementara IDF merujuk pada indeks frekuensi teks yang terbalik. Konsep dasar di balik TF-IDF adalah bahwa kata-kata yang muncul lebih sering dalam satu dokumen dan lebih jarang dalam dokumen lain seharusnya lebih penting karena lebih berguna untuk klasifikasi [13]. Sehingga, metode ini sering digunakan dalam mengekstraksi kata kunci, membandingkan kesamaan teks, dan mengklasifikasikan topik. Algoritma TF-IDF digunakan untuk menghitung nilai bobot setiap kata dalam setiap teks dokumen. Algoritma ini memperhitungkan probabilitas kemunculan kata dalam satu teks (TF) dan bobot kata tersebut dalam seluruh dokumen (IDF) [14]. Pembobotan TF-IDF terbentuk berdasarkan persamaan 1.

$$TF \cdot IDF = TF_{i,j} \times IDF_{i,j} = TF_{i,j} \times \log \frac{N}{DF_j} \quad (1)$$

Dimana,

N = Jumlah dokumen yang terdapat dalam koleksi

TF = term frequency

IDF = inverse document frequency.

2.5. Splitting Data

Setelah melakukan word embedding dengan menggunakan TF-IDF, langkah berikutnya adalah membagi data menjadi data pelatihan dan data uji. Pembagian ini memungkinkan kerangka kerja untuk menginspeksi kumpulan data terlebih dahulu, diikuti dengan pengujian informasi untuk memastikan tingkat akurasi. Dengan menggunakan pembagian data seperti 70:30, 80:20 dan 90:10. Data uji adalah data yang sebelumnya tidak digunakan dalam penelitian namun digunakan untuk mengevaluasi keberhasilan penelitian, sementara data pelatihan adalah data yang akan digunakan dalam penelitian.

2.6. SVM

Support Vector Machine (SVM) merupakan metode *supervised learning* untuk mengklasifikasikan data. SVM memiliki konsep matematis yang lebih terperinci dibandingkan dengan teknik klasifikasi lainnya. Metode ini dapat menyelesaikan masalah klasifikasi linear dan non-linear. Ini merupakan sistem pembelajaran yang memakai ruang hipotesis dengan representasi berupa fungsi linear pada fitur-fitur berdimensi tinggi, yang dilatih dengan algoritma pembelajaran berdasarkan pada prinsip optimisasi [15]. SVM beroperasi dengan mencari *hyperplane* yang optimal untuk memisahkan dua kelas data, yang ditentukan oleh jarak margin terbesar antara *hyperplane* dan vektor pendukung (*support vector*) [16]. Persamaan untuk menghitung

hyperplane pada klasifikasi SVM linear dapat dilihat pada persamaan 2.

$$f(x) = w \cdot x + b \quad (2)$$

Keterangan:

w: Vektor bobot

x: Vektor data

b: bias.

2.7. Naive Bayes

Metode Naïve Bayes adalah salah satu algoritma yang terkenal dalam analisis sentimen untuk mengklasifikasikan teks atau data berdasarkan pada sentimen yang terkandung di dalamnya. Algoritma ini bekerja berdasarkan prinsip probabilitas. Sebagai algoritma pembelajaran berbasis pengawasan, Naïve Bayes mengadopsi asumsi sederhana bahwa setiap pasangan variabel kelas adalah independen. Naïve Bayes menerapkan teorema Bayes [17]. Naïve Bayes hanya memerlukan sedikit data pelatihan dan pada atribut yang tidak relevan, Naïve Bayes tetap dapat diandalkan [18], seperti yang terlihat dalam persamaan 3.

$$P(A|B) = P(B|A)P(A)/P(B) \quad (3)$$

Keterangan:

$P(A|B)$: Probabilitas terjadinya A dengan syarat bahwa B telah terjadi

$P(B|A)$: Probabilitas terjadinya B dengan syarat bahwa A telah terjadi

$P(A)$: Probabilitas A terjadi

$P(B)$: Probabilitas B terjadi

2.8. Logistic Regression

Logistic Regression merupakan algoritma machine learning yang berguna dalam analisis sentimen. Dalam analisis sentimen Regresi Logistik menggunakan metode statistik untuk memprediksi probabilitas sentimen positif dan negatif berdasarkan variabel input. Sebagai algoritma machine learning *Logistic Regression* dapat mengelompokkan data ke dalam dua kategori yaitu kelompok sentimen negatif dan positif [19]. Logistic regression memiliki peran dalam memodelkan beberapa fungsi linier yang dapat dilihat pada persamaan 4.

$$\text{Logit}(P) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots \beta_k X_k \quad (4)$$

Keterangan

P= Peluang fitur

X_1, X_2, \dots, X_k = nilai prediktor

$\beta_1, \beta_2, \dots, \beta_k$ = intersepsi model.

2.9. Evaluasi

Proses evaluasi melibatkan penggunaan alat seperti Confusion Matrix dan word cloud. *Confusion Matrix* adalah alat evaluasi umum dalam klasifikasi dan analisis data yang menunjukkan jumlah data yang

diklasifikasikan dengan benar dan yang salah dalam bentuk tabel. Dengan menggunakan *Confusion Matrix*, kita dapat menghitung metrik evaluasi seperti akurasi, presisi, recall, dan F1-Score. *True positive*, *false positive*, *true negative*, dan *false negative* adalah istilah yang digunakan dalam evaluasi klasifikasi model melalui Confusion Matrix. Presisi mengukur proporsi positif yang benar-benar positif dari semua hasil yang diklasifikasikan sebagai positif, sedangkan recall mengukur proporsi positif yang berhasil diidentifikasi dari semua data yang seharusnya positif. F1-Score adalah metrik yang menggabungkan presisi dan recall dengan memberikan bobot yang seimbang. Akurasi mencerminkan seberapa dekat hasil klasifikasi dengan klasifikasi aktual secara keseluruhan. Dengan menggunakan metrik-metrik ini, kita dapat memahami kinerja model klasifikasi dalam mengidentifikasi kelas-kelas yang berbeda [20]. Word cloud pada dasarnya merupakan visualisasi dari teks yang digunakan dalam berbagai konteks, biasanya dibuat dengan merangkum isi teks. Umumnya, ini berguna dalam menganalisis teks yang berasal dari sejumlah sumber, membantu dalam mengevaluasi kesamaan informasi untuk tujuan penelitian tertentu. Dengan menyajikan ringkasan kata-kata individual tanpa mempertimbangkan signifikansi linguistik atau asosiasi, *word cloud* bertujuan untuk keperluan statistik, meskipun memiliki keterbatasan atau bahkan tidak menawarkan kemampuan interaktif [21].

3. HASIL DAN PEMBAHASAN

Bab ini akan menjelaskan satu per satu hasil penelitian yang telah dilakukan mulai dari Pengumpulan Data, Pra-pemrosesan Data, Pelabelan Data, TF-IDF, Splitting Data, Klasifikasi, dan Evaluasi Hasil Klasifikasi.

3.1. Pengumpulan Data

Dengan memanfaatkan teknik Web Scraping, penelitian ini berhasil mengumpulkan 1877 data rating dan ulasan dari pengguna aplikasi Noice di Situs Web Google Play Store.

Tabel 1. Data Hasil Web Scraping

Komentar	Rating
Ini app noice tiap kali lagi dengerin podcast sering bgt force close, tiba2 mati sendiri dan harus diulang lagi dari awal. Tolong lah diperbaiki ini appnya padahal udah sering diupdate tetap aja begitu mulu masalahnya cuma bikin kesel doang tiap mau dengerin podcast di noice. App gak guna!!	1
Menurut ku ini Aplikasi luar biasa untuk konten audio.Konten nya bagus bagus ada yang lucu ada yang berpikir banyak lah jenis nya.Kalo saya paling suka musmas dan ruang 28.Makan tidak perlu,koin noice nomor 1👍 Cuman saran saya untuk auto rotasi di video nya dibagusin lah.Mutar mutar terus hp ku kalo nonton yang ada videonya.Udah di setting dari hp(kunci rotasi) masih tetep muter muter. Semoga noice bisa lebih baik...Aamiin	4

Tabel I menunjukkan hasil dari proses pengumpulan data menggunakan web scraping, yang menghasilkan data komentar dan rating dari aplikasi Noice. Penelitian ini hanya menggunakan dua jenis data tersebut dari pengguna Noice di Google Play Store setelah proses web scraping, dengan melakukan filter agar hanya data komentar dan rating yang tetap ada.

3.2. Pra-pemrosesan Data

Seperti yang dijelaskan dalam metode penelitian, langkah-langkah pra-pemrosesan dalam penelitian ini meliputi beberapa tahap, seperti Case folding, Cleaning data, melakukan tokenisasi, Filtering, dan melakukan stemming.

Tabel 2. Hasil Cleaning dan Case folding

Komentar	Hasil Cleaning dan Case folding
Ini app noice tiap kali lagi dengerin podcast sering bgt force close, tiba2 mati sendiri dan harus diulang lagi dari awal. Tolong lah diperbaiki ini appnya padahal udah sering diupdate tetap aja begitu mulu masalahnya cuma bikin kesel doang tiap mau dengerin podcast di noice. App gak guna!!	ini app noice tiap kali lagi dengerin podcast sering bgt force close tiba2 mati sendiri dan harus diulang lagi dari awal. tolong lah diperbaiki ini appnya padahal udah sering diupdate tetap aja begitu mulu masalahnya cuma bikin kesel doang tiap mau dengerin podcast di noice app gak guna
Menurut ku ini Aplikasi luar biasa untuk konten audio.Konten nya bagus bagus ada yang lucu ada yang berpikir banyak lah jenis nya.Kalo saya paling suka musmas dan ruang 28.Makan tidak perlu,koin noice nomor 1👍 Cuman saran saya untuk auto rotasi di video nya dibagusin lah.Mutar mutar terus hp ku kalo nonton yang ada videonya.Udah di setting dari hp(kunci rotasi) masih tetep muter muter. Semoga noice bisa lebih baik...Aamiin	menurut ku ini aplikasi luar biasa untuk konten audio konten nya bagus ada yang lucu ada yang berpikir banyak lah jenis nya.kalo saya paling suka musmas dan ruang 28 makan tidak perlu koin noice nomor 1 cuman saran saya untuk auto rotasi di video nya dibagusin lah.mutar terus hp ku kalo nonton yang ada videonya udah di setting dari hp kunci rotasi masih tetep muter. semoga noice bisa lebih baik aamiin

Tabel 2 menampilkan hasil dari proses pembersihan data. Dalam tabel tersebut, terlihat bahwa data telah melewati tahap pembersihan teks, di mana semua huruf besar telah diubah menjadi huruf kecil dan emoji serta tanda baca juga telah dihilangkan.

Tabel 3. Hasil Normalisasi

Hasil Cleaning dan Case folding	Hasil Normalisasi
ini app noice tiap kali lagi dengerin podcast sering bgt force close tiba2 mati sendiri dan harus diulang lagi dari awal. tolong lah diperbaiki ini appnya padahal udah sering diupdate tetap aja begitu mulu masalahnya cuma bikin kesel doang tiap mau	ini aplikasi noice tiap kali lagi dengerin podcast sering banget force close tiba2 mati sendiri dan harus diulang lagi dari awal tolong lah di perbaiki ini appnya padahal sudah sering di perbarui tetap saja begitu mulu masalahnya cuma bikin kesel saja tiap mau dengerin podcast di noice aplikasi tidak guna

dengerin podcast di noice app gak guna	menurut saya ini aplikasi luar biasa untuk konten audio konten nya bagus ada yang lucu ada yang berpikir banyak lah jenis nya.kalo saya paling suka musmas dan ruang 28 makan tidak perlu koin noice nomor 1 cuman saran saya untuk auto rotasi di video nya dibagusin lah.mutar terus hp ku kalo nonton yang ada videonya udah di setting dari handphone kunci rotasi masih tetap muter. semoga noice bisa lebih baik aamiin
	menurut saya ini aplikasi luar biasa untuk konten audio.konten nya bagus ada yang lucu ada yang berpikir banyak lah jenis nya.kalo saya paling suka musmas dan ruang 28.makan tidak perlu koin noice nomor 1 cuman saran saya untuk otomatis rotasi di video nya dibagusin lah.mutar terus handphome saya kalau nonton yang ada videonya.udah di setting dari handphome kunci rotasi masih tetap memutar semoga noice bisa lebih baik aamiin

Tabel 3 memperlihatkan output dari proses normalisasi. Dari tabel ini, terlihat bahwa data telah melalui tahap normalisasi, di mana kalimat yang disingkat telah diubah menjadi bentuk standar dan sejenisnya.

Tabel 4. Hasil Tokenisasi

Hasil Normalisasi	Hasil Tokenisasi
ini aplikasi noice tiap kali lagi dengerin podcast sering banget force close tiba2 mati sendiri dan harus diulang lagi dari awal tolong lah di perbaiki ini appnya padahal sudah sering di perbarui tetap saja begitu mulu masalahnya cuma bikin kesel saja tiap mau dengerin podcast di noice aplikasi tidak guna	'aplikasi', 'noice', 'tiap', 'kali', 'lagi', 'dengerin', 'podcast', 'sering', 'banget', 'force', 'close', 'tiba2', 'mati', 'sendiri', 'harus', 'diulang', 'lagi', 'dari', 'awal', 'tolong', 'perbaiki', 'appnya', 'padahal', 'sudah', 'sering', 'perbarui', 'tetap', 'saja', 'begitu', 'mulu', 'masalahnya', 'cuma', 'bikin', 'kesel', 'saja', 'tiap', 'dengerin', 'podcast', 'noice', 'aplikasi', 'tidak', 'guna'
menurut saya ini aplikasi luar biasa untuk konten audio.konten nya bagus ada yang lucu ada yang berpikir banyak lah jenis nya.kalo saya paling suka musmas dan ruang 28.makan tidak perlu koin noice nomor 1 cuman saran saya untuk otomatis rotasi di video nya dibagusin lah.mutar terus handphome saya kalau nonton yang ada videonya.udah di setting dari handphone kunci rotasi masih tetap memutar semoga noice bisa lebih baik aamiin	'menurut', 'saya', 'aplikasi', 'luar', 'biasa', 'untuk', 'konten', 'audio.konten', 'bagus', 'yang', 'lucu', 'yang', 'berpikir', 'banyak', 'jenis', 'kalo', 'saya', 'paling', 'suka', 'musmas', 'ruang', 'makan', 'tidak', 'perlu', 'koin', 'noice', 'nomor', 'cuman', 'saran', 'saya', 'untuk', 'otomatis', 'rotasi', 'video', 'dibagusin', 'mutar', 'terus', 'handphome', 'saya', 'kalau', 'nonton', 'yang', 'videonya.udah', 'setting', 'dari', 'handphome', 'kunci', 'rotasi', 'masih', 'tetap', 'memutar', 'semoga', 'noice', 'bisa', 'lebih', 'baik', 'aamiin'

Tabel 4 menunjukkan hasil dari proses tokenisasi. Pada tahap ini, penelitian berhasil membagi kalimat-kalimat panjang menjadi kata-kata atau yang biasa disebut sebagai token. Setiap token berupa kata, dan dipisahkan oleh koma (,).

Tabel 5. Hasil Stopword

Hasil Tokenisasi	Hasil Stopword
'aplikasi', 'noice', 'tiap', 'kali', 'lagi', 'dengerin', 'podcast', 'sering', 'banget', 'force', 'close', 'tiba2', 'mati', 'sendiri', 'harus', 'diulang', 'lagi', 'dari', 'awal', 'tolong', 'perbaiki', 'appnya', 'padahal', 'sudah', 'sering',	'aplikasi', 'noice', 'kali', 'dengerin', 'podcast', 'banget', 'force', 'close', 'tiba2', 'mati', 'diulang', 'tolong', 'perbaiki', 'appnya', 'perbarui', 'mulu', 'aplikasi', 'konten', 'audio.konten', 'bagus', 'yang', 'lucu', 'yang', 'berpikir', 'jenis', 'kalo', 'suka', 'musmas', 'ruang', 'makan', 'koin', 'noice', 'nomor', 'cuman', 'saran', 'otomatis', 'rotasi', 'video', 'dibagusin', 'mutar', 'terus', 'handphome', 'saya', 'kalau', 'nonton', 'yang', 'videonya.udah', 'setting', 'dari', 'handphome', 'kunci', 'rotasi', 'masih', 'tetap', 'memutar', 'semoga', 'noice', 'bisa', 'lebih', 'baik', 'aamiin'

'perbarui', 'tetap', 'saja', 'begitu', 'mulu', 'masalahnya', 'cuma', 'bikin', 'kesel', 'saja', 'tiap', 'dengerin', 'podcast', 'noice', 'aplikasi', 'tidak', 'guna'	'bikin', 'kesel', 'dengerin', 'podcast', 'noice', 'aplikasi'
'menurut', 'saya', 'aplikasi', 'luar', 'biasa', 'untuk', 'konten', 'audio.konten', 'bagus', 'yang', 'lucu', 'yang', 'berpikir', 'banyak', 'jenis', 'kalo', 'saya', 'paling', 'suka', 'musmas', 'ruang', 'makan', 'koin', 'noice', 'nomor', 'cuman', 'saran', 'otomatis', 'rotasi', 'video', 'dibagusin', 'mutar', 'terus', 'handphome', 'saya', 'kalau', 'nonton', 'yang', 'videonya.udah', 'setting', 'dari', 'handphome', 'kunci', 'rotasi', 'masih', 'tetap', 'memutar', 'semoga', 'noice', 'bisa', 'lebih', 'baik', 'aamiin'	'aplikasi', 'konten', 'audio.konten', 'bagus', 'lucu', 'berpikir', 'jenis', 'kalo', 'suka', 'musmas', 'ruang', 'makan', 'koin', 'noice', 'nomor', 'cuman', 'saran', 'otomatis', 'rotasi', 'video', 'dibagusin', 'mutar', 'handphome', 'nonton', 'videonya.udah', 'setting', 'handphome', 'kunci', 'rotasi', 'semoga', 'noice', 'aamiin'

Tabel 5 menggambarkan output dari proses stopword. Pada tahap ini, diperlihatkan hasil dari proses stopword yang dilakukan dengan membandingkan kata-kata hasil tokenisasi dengan daftar kata-kata stopword, di mana jika tidak ada kesamaan, kata-kata tersebut akan dihapus.

Tabel 6. Hasil Stemming

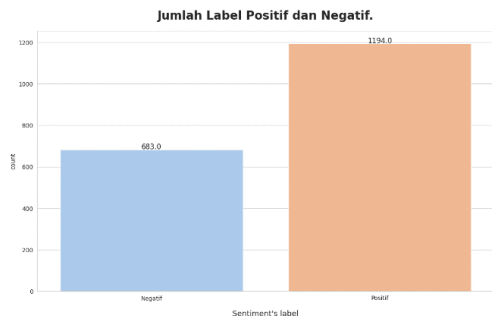
Hasil Stopword	Hasil Stemming
'aplikasi', 'noice', 'kali', 'dengerin', 'podcast', 'banget', 'force', 'close', 'tiba2', 'mati', 'diulang', 'tolong', 'perbaiki', 'appnya', 'perbarui', 'mulu', 'bikin', 'kesel', 'dengerin', 'podcast', 'noice', 'aplikasi'	aplikasi noice kali dengerin podcast banget force close tiba2 mati ulang tolong baik appnya baru mulu bikin kesel dengerin podcast noice aplikasi
'aplikasi', 'konten', 'audio.konten', 'bagus', 'lucu', 'berpikir', 'jenis', 'kalo', 'suka', 'musmas', 'ruang', 'makan', 'koin', 'noice', 'nomor', 'cuman', 'saran', 'otomatis', 'rotasi', 'video', 'dibagusin', 'mutar', 'handphome', 'nonton', 'videonya.udah', 'setting', 'handphome', 'kunci', 'rotasi', 'memutar', 'semoga', 'noice', 'aamiin'	aplikasi konten audio konten bagus lucu pikir jenis kalo suka musmas ruang makan koin noice nomor cuman saran otomatis rotasi video dibagusin mutar handphome nonton video udah setting handphome kunci rotasi putar moga noice aamiin

Tabel 6 menampilkan hasil dari proses stemming dengan melakukan penyesuaian pada kata-kata dalam data untuk menghindari kebingungan yang mungkin terjadi pada sistem, termasuk penanganan kata-kata seperti "di", "me-", "meng-", "kan", "nya", dan sejenisnya.

3.3. Pelabelan

Setelah melalui tahap preprocessing, dataset kemudian diberi label sentimen, terutama positif dan negatif. Dalam proses pelabelan ini, data akan diolah secara otomatis, terutama dalam menghitung nilai poin. Penentuan apakah data termasuk dalam kelompok sentimen positif atau negatif didasarkan pada nilai rating. Jika nilai rating kalimat > 3, maka kalimat tersebut dikategorikan sebagai sentimen positif. Sebaliknya, jika nilai rating kalimat ≤ 3, maka kalimat tersebut dikategorikan sebagai sentimen

negatif. Untuk hasil pelabelan dapat dilihat pada gambar 2.



Gambar 2. Hasil Pelabelan Data

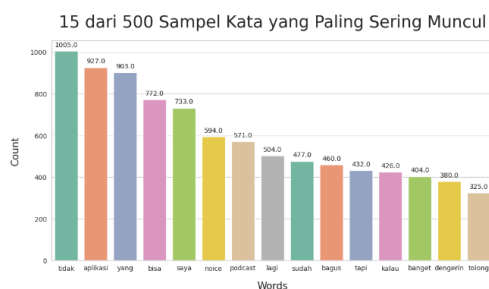
3.4. TF-IDF

Pada langkah ini, upaya dilakukan untuk meningkatkan kinerja klasifikasi guna memperbaiki akurasi dan mempercepat waktu komputasi. Untuk tujuan ini, fitur yang dimanfaatkan adalah TF-IDF. Tabel 7 menampilkan output dari proses pembobotan kata menggunakan metode TF-IDF dengan Python.

Tabel 7. Hasil TF-IDF

Indeks Baris	Nomor Kata	Skor TF-IDF
0	2736	0.2900410323314745
0	2727	0.11010941837781357
0	2052	0.20896898419071927
0	1836	0.09425851550192099
0	1737	0.2788250522908901
0	1352	0.12515297079348084
0	1239	0.21236133276253727
0	1199	0.18623361857882967
0	959	0.25700707992164684
0	476	0.611698136675617
0	431	0.1942592578256201
0	391	0.14021122573178327
0	258	0.42472266552507454

Dari tabel 7, dapat disimpulkan bahwa angka Indeks baris mewakili indeks baris dari setiap data yang telah diproses. Nomor kata adalah nomor integer yang unik untuk setiap kata yang terdapat dalam baris tersebut. Sementara skor TF-IDF merupakan hasil pembobotan (skor) yang dihitung menggunakan metode TF-IDF. Setelah melalui proses preprocessing dan pembobotan TF-IDF, dapat dilihat kata-kata yang paling umum atau kata kunci utama ditampilkan dalam bentuk diagram plot, Seperti yang ditampilkan dalam ilustrasi pada gambar 3.



Gambar 3. Top Word

3.5. Klasifikasi dan Evaluasi Hasil Klasifikasi

Dalam penilaian model, teknik confusion matrix diterapkan di berbagai pengaturan yang berbeda di mana data pelatihan dan pengujian dibagi dalam rasio 90:10, 80:20, dan 70:30. Hasilnya menunjukkan *accuracy*, *precision*, *F1-Score* dan *recall* dari model yang dievaluasi. Mengevaluasi model di berbagai skenario pembagian data memberikan wawasan tentang kinerjanya dalam kondisi yang beragam.

Tabel 8. Hasil Klasifikasi Algoritma SVM

Skenario	SVM			
	Accuracy	Precision	Recall	F1-Score
90:10	0.80	0.81	0.80	0.80
80:20	0.78	0.79	0.78	0.78
70:30	0.79	0.79	0.79	0.79

Berdasarkan tabel 8, dapat disimpulkan bahwa klasifikasi algoritma SVM menggunakan pembagian data 90:10 menghasilkan tingkat akurasi 80%, F1-score 80%, recall 80%, dan presisi 81% yang lebih tinggi dibandingkan dengan pembagian data menjadi 70:30 dan 80:20. Hal ini disebabkan oleh fakta bahwa pembagian data 90:10 memberikan lebih banyak data latih untuk algoritma klasifikasi belajar, sementara data uji cenderung lebih kecil dibandingkan dengan dua skenario lainnya.

Tabel 9. Hasil Klasifikasi Algoritma Naïve Bayes

Skenario	Naïve Bayes			
	Accuracy	Precision	Recall	F1-Score
90:10	0.78	0.80	0.78	0.78
80:20	0.77	0.78	0.77	0.77
70:30	0.77	0.78	0.77	0.76

Berdasarkan tabel 9, Naïve Bayes menunjukkan hasil akurasi 78%, F1-score 78%, recall 78%, dan presisi 80% yang lebih unggul dalam pembagian data 90:10, yang sebanding dengan klasifikasi yang dilakukan oleh algoritma SVM.

Tabel 10. Hasil Klasifikasi Algoritma Logistic Regression

Skenario	Logistic Regression			
	Accuracy	Precision	Recall	F1-Score
90:10	0.77	0.77	0.77	0.77
80:20	0.78	0.78	0.78	0.78
70:30	0.77	0.77	0.77	0.77

Dari tabel 10 menunjukkan perbedaan hasil antara algoritma SVM, Naïve Bayes, dan Logistic Regression. Ditemukan bahwa dalam algoritma Logistic Regression menghasilkan akurasi 78%, Precision 78%, Recall 78%, dan F1-Score 78%, lebih baik ketika data dibagi dengan rasio 80:20.

3.2. Visualisasi Wordcloud

Wordcloud dalam penelitian ini mencerminkan seberapa kerap atau pentingnya kata-kata tersebut. Semakin besar ukuran kata, semakin terus menerus kemunculan kata tersebut dalam dokumen. Di bawah ini adalah awan kata hasil dari data yang telah melalui tahap pra-pemrosesan.



Gambar 4. Word Cloud Positif

Gambar 4 adalah word cloud positif dari data yang telah melalui tahap pra-pemrosesan. Kata-kata yang kerap keluar dalam ulasan positif adalah aplikasi, noise, podcast, dan bagus.



Gambar 5. Word Cloud Negatif

Gambar 6 adalah word cloud negatif dari data yang telah melalui tahap pra-pemrosesan. Kata-kata yang kerap keluar dalam ulasan negatif adalah aplikasi, noise, iklan, unduh, dan banget.

4. DISKUSI

Berikut ini adalah beberapa penelitian terdahulu yang membahas mengenai kepuasan menggunakan metode klasifikasi SVM, Naïve Bayes, dan Logistic Regression. Penelitian awal yang dilakukan oleh Ratna Andini Husen dan rekan-rekannya meneliti kepuasan layanan di Bank BSI. Hasil eksperimen menunjukkan bahwa SVM mencapai akurasi sebesar 0,88%, Naive Bayes mencapai akurasi sebesar 0,76%, sementara logistic regression mencapai akurasi sebesar 0,86% [22]. Penelitian kedua, yang dilakukan oleh Rifky Aziz Fauzianto dan Supatman, mengeksplorasi pendapat masyarakat tentang Tech Winter di Twitter. Mereka menggunakan algoritma logistic regression, SVM, dan Naive Bayes. Hasilnya menunjukkan bahwa model logistic regression mencapai akurasi sebesar 83,33%, model SVM mencapai akurasi yang sangat tinggi yaitu 94,44%, dan model naive Bayes juga menunjukkan akurasi yang tinggi, sebanding dengan SVM, yaitu 94,44% [23]. Kedua penelitian ini mengonfirmasi bahwa algoritma logistic regression, SVM, dan Naive Bayes efektif dalam melakukan klasifikasi analisis sentimen. Temuan ini menjadi alasan utama bagi peneliti untuk memilih algoritma logistic regression, SVM, dan Naive Bayes dalam penelitian ini.

5. KESIMPULAN

Hasil analisis terhadap aplikasi Noice berdasarkan ulasan yang paling relevan dan terbaru menggunakan tiga algoritma klasifikasi, yaitu SVM, Naive Bayes, dan Logistic Regression, serta tiga

skenario pembagian data (70:30, 80:20, dan 90:10). Hasilnya menunjukkan kecenderungan arah sentimen menuju positif yaitu 1194 data sementara itu, sentimen negatif memiliki proporsi sebanyak 683 data. Dari perbandingan menggunakan tiga skenario pembagian data (70:30, 80:20, dan 90:10) dengan algoritma SVM, Logistic Regression, dan Naïve Bayes, hasilnya menunjukkan akurasi tertinggi diperoleh menggunakan algoritma SVM dengan skema pembagian data 90:10, nilai akurasi mencapai 80%.

Peneliti selanjutnya disarankan untuk lebih mengoptimalkan proses penyiapan data, termasuk pengembangan kamus kata, serta memperhatikan aspek seperti kapitalisasi teks, tanda baca, dan tanda lainnya untuk memahami ulasan dengan lebih baik. Penerjemahan emoji harus dilakukan dengan memperhatikan konteks ulasan untuk menggambarkan sentimen pengguna aplikasi secara akurat. Metode seleksi fitur yang efisien perlu dipertimbangkan untuk mengurangi waktu komputasi, dengan memilih metode yang dapat menyeleksi fitur tanpa kehilangan banyak informasi. Selain itu, penelitian selanjutnya dapat menguji dan membandingkan kinerja metode lain, seperti algoritma *deep learning* atau analisis teks berbasis aturan, untuk memastikan penggunaan pendekatan yang paling efektif dalam menganalisis sentimen publik.

DAFTAR PUSTAKA

- [1] Noice, "Tentang Noice." <https://www.noice.id/tentang-noice/> (accessed Mar. 25, 2024).
- [2] C. Fikri, "Noice Dinobatkan Google Play Jadi Aplikasi Lokal Terkemuka," *Berita Satu*, 2023. <https://www.beritasatu.com/ototekno/1062758/noice-dinobatkan-google-play-jadi-aplikasi-lokal-terkemuka> (accessed Mar. 25, 2024).
- [3] S. Vashishtha and S. Susan, "Fuzzy rule based unsupervised sentiment analysis from social media posts," *Expert Syst. Appl.*, vol. 138, p. 112834, 2019, doi: 10.1016/j.eswa.2019.112834.
- [4] E. Fitri, "Sentiment Analysis of the Ruanguru Application Using Naive Bayes, Random Forest and Support Vector Machine Algorithms," *J. Transform.*, vol. 18, no. 1, p. 71, 2020.
- [5] A. Saepudin, A. Faqih, and G. Dwilestari, "Perbandingan Algoritma Klasifikasi Support Vector Machine, Random Forest dan Logistic Regression Pada Ulasan Shopee," *J. TEKNO KOMPAK*, vol. 18, no. 1, pp. 178–192, 2024.
- [6] A. Maulana, Inayah Khasnaputri Afifah, Asghafi Mubarrak, Kiagus Rachmat Fauzan,

- Ardhan Dwintara, and B. P. Zen, "Comparison of Logistic Regression, Multinomialnb, Svm, and K-Nn Methods on Sentiment Analysis of Gojek App Reviews on the Google Play Store," *J. Tek. Inform.*, vol. 4, no. 6, pp. 1487–1494, 2023, doi: 10.52436/1.jutif.2023.4.6.863.
- [7] I. Yunanto and S. Yulianto, "Twitter Sentiment Analysis Pedulilindungi Application Using Naïve Bayes and Support Vector Machine," *J. Tek. Inform.*, vol. 3, no. 4, pp. 807–814, 2022, doi: 10.20884/1.jutif.2022.3.4.292.
- [8] E. Suryati, Styawati, and A. Ari Aldino, "Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM)," *J. Teknol. dan Sist. Inf.*, vol. 4, no. 1, pp. 96–106, 2023.
- [9] M. Prasetya, M. Wulandari, and S. A. Nikmah, "Implementasi NLP (Natural Language Processing) Dasar pada Analisis Sentiment Review Spotify," *Stain. (Seminar Nas. Teknol. Sains)*, vol. 3, no. 1, pp. 145–153, 2024.
- [10] A. Lattu and D. Novira, "Analisis Sentimen Masyarakat Di Instagram Mengenai Kereta Cepat Jakarta-Bandung Menggunakan Algoritma Naïve Bayes Classifier," *Sismatik*, vol. 1, pp. 1–8, 2023.
- [11] S. Wahyu Handani, D. Intan Surya Saputra, Hasirun, R. Mega Arino, and G. Fiza Asyrofi Ramadhan, "Sentiment analysis for go-jek on google play store," *J. Phys. Conf. Ser.*, vol. 1196, no. 1, 2019, doi: 10.1088/1742-6596/1196/1/012032.
- [12] Primandani Arsi, Pungkas Subarkah, and Bagus Adhi Kusuma, "Analisis Sentimen Game Genshin Impact pada Play Store Menggunakan Naïve Bayes Clasifier," *J. Ilm. Tek. Mesin, Elektro dan Komput.*, vol. 3, no. 1, pp. 161–170, 2023, doi: 10.51903/juritek.v3i1.1962.
- [13] H. Liu, X. Chen, and X. Liu, "A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis," *IEEE Access*, vol. 10, pp. 32280–32289, 2022, doi: 10.1109/ACCESS.2022.3160172.
- [14] Imamah and F. H. Rachman, "Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regression," *Proceeding - 6th Inf. Technol. Int. Semin. ITIS 2020*, pp. 238–242, 2020, doi: 10.1109/ITIS50118.2020.9320958.
- [15] R. Irwanda *et al.*, "Sentiment analysis of Aceh tourism and its culture from Twitter data using support vector machine (SVM), naive Bayesian and k-nearest neighbour (KNN)," *AIP Conf. Proc.*, vol. 3082, no. 1, 2024, doi: 10.1063/5.0202198.
- [16] I. Maulana, W. Apriandari, and A. Pambudi, "Analisis Sentimen Berbasis Aspek Terhadap Ulasan Aplikasi My Pertamina Menggunakan Support Vector Machine," *IDEALIS Indones. J. Inf. Syst.*, vol. 6, no. 2, pp. 172–181, 2023, doi: 10.36080/idealis.v6i2.3022.
- [17] S. H. Dinar Ajeng Kristiyanti, "Sentiment Analysis of Public Acceptance of Covid-19 Vaccines Types in Indonesia using Naïve Bayes, Support Vector Machine, and Long Short- Term Memory (LSTM)," vol. 5, no. 158, pp. 2–6, 2023.
- [18] A. Harun and D. Putri Ananda, "Analisa Sentimen Opini Publik Tentang Vaksinasi Covid-19 di Indonesia Menggunakan Naïve bayes dan Decision Tree," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 1, no. 1, pp. 58–64, 2021, doi: 10.57152/malcom.v1i1.63.
- [19] K. Hasanah, "Comparison of Sentiment Analysis Model for Shopee Comments on Google Play Store," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 13, no. 1, pp. 21–30, 2024, doi: 10.32736/sisfokom.v13i1.1916.
- [20] R. I. Borman, Y. Fernando, and Y. Egi Pratama Yudoutomo, "Identification of Vehicle Types Using Learning Vector Quantization Algorithm with Morphological Features," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 2, pp. 339–345, 2022, doi: 10.29207/resti.v6i2.3954.
- [21] A. I. KABIR, K. AHMED, and R. KARIM, "Word Cloud and Sentiment Analysis of Amazon Earphones Reviews with R Programming Language," *Inform. Econ.*, vol. 24, no. 4/2020, pp. 55–71, 2020, doi: 10.24818/issn14531305/24.4.2020.05.
- [22] R. A. Husen, R. Astuti, L. Marlia, R. Rahmaddeni, and L. Efrizoni, "Analisis Sentimen Opini Publik pada Twitter Terhadap Bank BSI Menggunakan Algoritma Machine Learning," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 3, no. 2, pp. 211–218, 2023, doi: 10.57152/malcom.v3i2.901.
- [23] R. A. Fauzianto *et al.*, "Analisis Sentimen Opini Masyarakat Terhadap Tech Winter Pada Twitter," vol. 4, no. 9, pp. 1577–1585, 2023, doi: 10.46799/jsa.v3i9.909.