Jurnal Informatika dan Tekonologi Komputer

E-ISSN: 2809-9230
P-ISSN: 2809-9249

Pemodelan Sentimen Komentar YouTube Berbasis Naive Bayes

Hari Murti [1], Rara Sriartati Redjeki [2], Novita Mariana [3*], Agus Prasetyo Utomo[4], dan Widiyanto Tri Handoko[5]

[1] Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank; Semarang, Jawa Tengah; e-mail : harimurti@edu.unsibank.ac.id
[2] Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank; Semarang, Jawa Tengah; e-mail : rara_artati@edu.unsibank.ac.id
[3] Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank; Semarang, Jawa Tengah; e-mail : novita_mariana@edu.unsibank.ac.id
[4] Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank; Semarang, Jawa Tengah; e-mail : mustagus@edu.unsibank.ac.id
[5] Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank; Semarang, Jawa Tengah; e-mail : wthandoko@edu.unsibank.ac.id
* Corresponding Author : Novita Mariana

Abstract: Social Network Analysis (SNA) is a crucial quantitative methodology for mapping relationships and identifying connectivity structures within a group. This research specifically explores the use of the NetworkX library in Python as an effective tool for analyzing social networks. The primary objective of this study is to apply the Degree Centrality method to measure the level of connectivity and identify the most popular actors in a social network. The methodology employed is the quantitative analysis of an undirected graph modeled from the us_edgelist.csv dataset, which contains a list of relationships among political figures in an edge list format. Data processing utilized pandas, and the graph object was constructed using NetworkX. Degree Centrality was calculated for each node, with the results being normalized to provide a relative value. This normalization allows for a direct comparison of how active each actor is within the network. The centrality results were then visualized, with node sizes adjusted based on their Degree Centrality score. The results of the analysis indicate that figures like Bush and Obama possess the highest Degree Centrality score, 0.25, suggesting they have the greatest number of direct connections in this network. This high value confirms their role as the most active or central actors in the exchange and interaction within the political network studied. This finding validates the effectiveness of Degree Centrality as an indicator of high involvement. The study concludes that the implementation of Social Network Analysis using NetworkX provides a robust framework for understanding political relationship structures. Therefore, Degree Centrality is a reliable metric for quantifying actor activity and accurately identifying individuals who form the center of connections within the network.

Keywords: Social Network Analysis; NetworkX; Degree Centrality; Edgelist; Python

Naskah Masuk: 4 November 2025
Revisi: 13 Desember 2025
Diterima: 22 januari 2026
Tersedia: 3 Februari 2026
Ver. Skrg.: 3 Februari 2026

Abstrak: Analisis Jaringan Sosial (*SNA*) merupakan metodologi kuantitatif yang penting untuk memetakan hubungan dan mengidentifikasi struktur konektivitas dalam suatu kelompok. Penelitian ini secara spesifik mengeksplorasi penggunaan pustaka NetworkX di Python sebagai *tool* yang efektif untuk menganalisis jaringan sosial. Tujuan utama dari penelitian ini adalah untuk menerapkan metode Derajat Sentralitas (*Degree Centrality*) guna mengukur tingkat konektivitas dan mengidentifikasi aktor paling populer dalam jaringan sosial. Metode yang diterapkan adalah analisis kuantitatif jaringan tak berarah (*undirected graph*) yang dimodelkan dari *dataset us_edgelist.csv* yang berisi daftar hubungan antar tokoh politik dalam daftar *edge list*. Data diolah menggunakan *pandas* dan objek grafik dibangun dengan *NetworkX*. Perhitungan Derajat Sentralitas dilakukan untuk setiap simpul (*node*), dimana hasilnya dinormalisasi untuk memberikan nilai relatif. Normalisasi ini memungkinkan perbandingan langsung mengenai seberapa aktif setiap aktor dalam jaringan. Hasil sentralitas kemudian divisualisasikan, dengan ukuran simpul disesuaikan berdasarkan nilai Derajat Sentralitasnya. Hasil analisis menunjukkan bahwa

tokoh seperti Bush dan Obama memiliki nilai Derajat Sentralitas tertinggi, yaitu 0.25, mengindikasikan bahwa mereka memiliki jumlah koneksi langsung terbanyak dalam jaringan ini. Nilai tinggi ini menegaskan peran mereka sebagai aktor paling aktif atau sentral dalam pertukaran dan interaksi dalam jaringan politik yang diteliti. Temuan ini memvalidasi efektivitas Derajat Sentralitas sebagai indikator keterlibatan yang tinggi. Penelitian ini menegaskan bahwa implementasi *Social Network Analysis* menggunakan *NetworkX* menyediakan kerangka kerja yang solid untuk memahami struktur hubungan politik. Dengan demikian dapat didimpulkan bahwa Derajat Sentralitas adalah metrik yang andal untuk mengukur tingkat aktivitas aktor, dengan akurasi mengidentifikasi individu yang menjadi pusat koneksi dalam jaringan.

Kata kunci: Analsis Jaringan Sosial; NetworkX; Derajat Sentralitas; Edgelist; Python

1. Pendahuluan

Dalam era disrupsi digital saat ini, fenomena media sosial dan platform berbagi video telah merevolusi secara fundamental cara manusia berkomunikasi, berinteraksi, dan menyerap informasi. Perubahan ini terjadi secara drastis dalam skala global [1]. Di antara berbagai platform yang ada, YouTube menempati posisi sentral sebagai salah satu *platform* berbagi video terbesar di dunia. YouTube tidak hanya berfungsi sebagai gudang konten visual yang luas, tetapi juga menciptakan ekosistem interaksi yang dinamis, terutama melalui kolom komentarnya [2].

Kolom komentar di bawah setiap video yang seringkali terabaikan sesungguhnya merupakan repositori data teks yang masif dan berharga. Kumpulan teks ini berfungsi sebagai cerminan langsung dari opini, pandangan, emosi, dan reaksi spontan pengguna (*netizen*) terhadap berbagai topik yang disajikan, mencakup spektrum luas mulai dari isu-isu sosial yang sensitif, perdebatan politik, ulasan produk teknologi, hingga konten hiburan ringan.

Untuk mengurai dan memanfaatkan kekayaan data tekstual ini, Analisis Sentimen (*Sentiment Analysis*) memainkan peran yang sangat krusial [3]. Sebagai cabang penting dari Natural Language Processing (NLP) dan Text Mining, analisis sentimen berfokus pada teknik komputasi untuk mengekstrak dan mengidentifikasi polaritas emosional dari teks yaitu, mengklasifikasikan komentar ke dalam kategori sentimen positif, negatif, atau netral [4].

Pemahaman terhadap sentimen ini menghasilkan temuan yang sangat penting dengan implikasi praktis yang luas. Bagi para content creator, hasil analisis sentimen dapat menjadi alat evaluasi kepuasan audiens yang instan dan objektif. Sementara itu, bagi sektor bisnis dan pemasaran, informasi ini sangat berharga untuk pengambilan keputusan strategis, memungkinkan pemantauan *brand perception* secara *real-time* [5].

Komentar di media sosial memiliki karakteristik linguistik yang unik; bahasa yang digunakan cenderung informal, sering kali mengandung bahasa gaul, singkatan, kesalahan pengetikan, dan emotikon. Oleh karena itu, data ini rentan terhadap noise [14] dan memerlukan tahap *preprocessing* yang cermat agar dapat diproses oleh sistem komputasi [17]. Kompleksitas ini semakin menekankan kebutuhan akan model *machine learning* yang robust dan efisien, seperti Algoritma Naive Bayes, untuk mengubah teks mentah menjadi pengetahuan yang terstruktur dan terukur.

Data komentar YouTube berlimpah dan representatif sebagai sumber informasi opini publik menjadi tantangan utama pada proses penanganan volume data tersebut [10]. Melakukan analisis sentimen secara manual pada data yang masif terbukti sangat tidak efisien dari segi waktu dan biaya, dan yang lebih penting, rentan terhadap subjektivitas dan inkonsistensi penilai [19]. Oleh karena itu, kebutuhan akan metode komputasi yang efektif dan akurat untuk mengotomatisasi proses klasifikasi sentimen menjadi imperatif dalam ranah *data science* [13].

Di antara berbagai algoritma *machine learning* yang tersedia, Algoritma Naive Bayes (NB) telah lama menjadi pilihan yang populer dan andal dalam tugas klasifikasi teks [6]. Algoritma ini menarik karena sifatnya yang sederhana dalam komputasi, menawarkan efisiensi yang tinggi, dan mampu memberikan performa yang kompetitif, terutama ketika diterapkan pada kumpulan data berdimensi tinggi atau besar [7][8]. Naive Bayes, dengan asumsi independensi fiturnya, telah terbukti efektif dalam mengklasifikasikan sentimen di berbagai konteks bahasa. Misalnya, algoritma ini menunjukkan kinerja yang solid pada data berbahasa Indonesia, bahkan ketika dihadapkan pada tantangan umum seperti keberadaan *stopword* (kata-kata umum yang tidak signifikan) dan *slang* atau bahasa tidak baku [9], [15].

Peningkatan akurasi model Naive Bayes seringkali dapat dicapai melalui aplikasi teknik pra-pemrosesan yang tepat [12], dan pemilihan fitur yang efisien. Metode pembobotan fitur, seperti *N-Gram* [6] atau *Term Frequency Inverse Document Frequency* (TF-IDF) [16], terbukti penting untuk menonjolkan kata-kata kunci yang paling diskriminatif dalam membedakan polaritas sentimen. Meskipun demikian, kinerja Naive Bayes masih perlu diuji secara spesifik dan mendalam pada data komentar YouTube yang memiliki struktur, panjang, dan karakteristik *noise* yang unik [11].

Penelitian ini bertujuan untuk menerapkan dan mengevaluasi kinerja algoritma Naive Bayes dalam mengklasifikasikan sentimen pada data komentar video YouTube [10]. Fokus utama adalah menentukan seberapa efektif model Naive Bayes dapat membedakan antara sentimen positif, negatif, dan netral pada konteks bahasa informal media sosial. Kontribusi utama penelitian ini adalah memberikan bukti empiris mengenai keandalan Naive Bayes sebagai *classifier* sentimen untuk data komentar YouTube, sebuah sumber data dengan karakteristik bahasa yang unik [17]. Hasil penelitian ini diharapkan dapat menjadi referensi bagi peneliti selanjutnya dalam memilih metode klasifikasi yang tepat, serta menjadi alat bantu praktis bagi pembuat konten untuk memahami audiens mereka dan memonitor persepsi publik secara *real-time* [19].

2. Kajian Pustaka atau Penelitian Terkait

Kajian pustaka ini bertujuan untuk menempatkan penelitian ini dalam konteks ilmiah yang lebih luas dengan meninjau konsep dasar Analisis Sentimen, Algoritma Naive Bayes, dan studi-studi terdahulu yang secara spesifik membahas analisis sentimen pada data media sosial, khususnya komentar YouTube.

2.1. Konsep Dasar Analisis Sentimen

Analisis Sentimenatau yang sering juga dikenal dengan istilah Opinion Mining merupakan disiplin ilmu yang fundamental dalam ranah *Natural Language Processing* (NLP). Tujuan utamanya adalah untuk secara otomatis mengidentifikasi dan mengukur sikap, emosi, atau penilaian subjektif penulis terhadap subjek tertentu, yang pada akhirnya diklasifikasikan ke dalam polaritas emosional utama: positif, negatif, atau netral [3]. Kebutuhan mendesak akan analisis sentimen ini **didorong** oleh pertumbuhan eksponensial data teks yang dihasilkan di platform daring, terutama media sosial. Platform-platform ini telah bertransformasi menjadi repositori utama yang tak ternilai harganya untuk menangkap dan memahami denyut nadi opini publik secara *real-time* [2]. Oleh karena itu, kemampuan untuk memproses dan menginterpretasikan data tekstual ini telah menjadi kunci bagi peneliti, bisnis, dan pengambil keputusan.

Dalam upaya mengimplementasikan Analisis Sentimen secara komputasional, penelitian terdahulu telah menguji serangkaian metodologi yang terbagi menjadi tiga kategori besar [9]: pertama, pendekatan berbasis leksikon (*lexicon-based*), yang mengandalkan kamus kata-kata yang telah diberi skor sentimen; kedua, pendekatan *machine learning*, yang melatih model klasifikasi menggunakan data berlabel; dan ketiga, pendekatan hibrida, yang menggabungkan keunggulan kedua metode sebelumnya. Meskipun metode leksikon menawarkan kemudahan dan transparansi, pendekatan berbasis *machine learning* seringkali menjadi pilihan yang lebih unggul dalam aplikasi praktis [4]. Hal ini dikarenakan model *machine learning* memiliki kapabilitas adaptasi yang lebih baik terhadap kekhasan linguistik domain data spesifik, seperti bahasa gaul di media sosial, dan terbukti mampu menghasilkan akurasi klasifikasi yang lebih tinggi.

2.2. Algoritma Klasifikasi Naïve bayes

Algoritma Naive Bayes (NB) adalah inti dari pendekatan klasifikasi probabilistik dalam *machine learning*. Algoritma ini didasarkan pada Teorema Bayes [7] dan beroperasi di bawah asumsi krusial mengenai independensi fitur yang kuat. Secara teori, asumsi ini jarang terpenuhi secara sempurna dalam data tekstual, di mana susunan kata-kata (*tokens*) secara alami memiliki ketergantungan semantik dan sintaksis satu sama lain. Meskipun demikian, Naive Bayes secara empiris telah membuktikan dirinya sebagai algoritma yang sangat efektif dan efisien untuk berbagai tugas klasifikasi teks [6]. Naive Bayes menjadi sangat diminati dalam konteks analisis sentimen karena dua keunggulan utama: pertama, sifatnya yang sederhana dan cepat secara komputasi, menjadikannya ideal untuk pemrosesan volume data yang besar dan berdimensi tinggi, seperti yang dihasilkan oleh media sosial [7]; dan kedua, kemampuannya menunjukkan kinerja yang kompetitif yang seringkali sebanding dengan algoritma klasifikasi yang jauh lebih kompleks, terutama ketika diimplementasikan dengan teknik pembobotan fitur yang cermat [13].

Secara fundamental, proses klasifikasi dalam Naive Bayes bertujuan untuk menghitung probabilitas posterior dari suatu kelas (C) diberikan serangkaian fitur atau kata-kata dalam dokumen (x_1, x_2, … , x_n), sesuai dengan persamaan (1).

$$\text{Kelas Prediksi} = \underset{C}{argmax}\,(P(C).\prod_{i=1}^{n} P(x_i|C)) \quad (1)$$

Di mana Probabilitas Prior (P(C)) dihitung dari frekuensi kemunculan setiap kelas sentimen (Positif, Negatif, Netral) dalam dataset pelatihan, sementara Probabilitas Likelihood ($P(x_i|C)$) dihitung berdasarkan frekuensi kata x_i dalam dokumen-dokumen yang termasuk kelas (C).

Kinerja Naive Bayes tidak statis dan dapat ditingkatkan melalui serangkaian teknik optimasi dan rekayasa fitur. Berbagai studi telah mendemonstrasikan bahwa penerapan teknik pra-pemrosesan data yang tepat merupakan langkah awal yang signifikan. Misalnya, normalisasi data dan *stemming* (pengembalian kata ke bentuk dasar) dapat secara signifikan meningkatkan akurasi klasifikasi dengan mengurangi variabilitas data [12], [17]. Lebih lanjut, peningkatan kemampuan diskriminatif model Naive Bayes sangat bergantung pada teknik pembobotan kata yang efisien, seperti Term Frequency–Inverse Document Frequency (TF-IDF) [16], dan pemilihan fitur berbasis *N-Gram* [6], [15], yang membantu model untuk lebih fokus pada kata-kata yang paling penting dalam membedakan kelas sentimen. Penelitian juga terus berlanjut untuk menyempurnakan Naive Bayes, termasuk fokus pada optimalisasi parameter untuk penanganan tugas klasifikasi multikelas, sebuah aspek yang krusial untuk mengklasifikasikan komentar secara akurat ke dalam polaritas positif, negatif, dan netral [20].

2.3. Penelitgian Terdahulu pada Analsis Sentimen Komentar Youtube

Sejumlah penelitian terbaru, khususnya dalam rentang tahun 2020 hingga 2025, telah secara khusus memfokuskan perhatiannya pada analisis sentimen terhadap data dari platform media sosial, termasuk komentar YouTube [14]. Hal ini didasari oleh pengakuan terhadap tantangan signifikan yang ditimbulkan oleh karakteristik bahasa informal, non-standar, dan banyaknya *noise* dalam data tersebut.

Penerapan Algoritma Naive Bayes (NB) pada Data YouTube telah menjadi subjek utama dari beberapa studi. Berbagai penelitian telah berhasil menerapkan Naive Bayes pada komentar YouTube, seringkali dengan fokus pada domain konten tertentu seperti, menganalisis opini publik terhadap video politik [11], mengukur respons audiens terhadap konten edukasi [19], atau memahami persepsi konsumen dari ulasan produk [10]. Secara konsisten, studi-studi ini mengonfirmasi kelayakan dan efektivitas Naive Bayes dalam mengklasifikasikan sentimen pada data yang unik ini. Meskipun demikian, temuan tersebut juga menyoroti perlunya adaptasi dan optimalisasi fitur untuk secara efektif mengatasi tantangan linguistik yang spesifik, seperti penggunaan bahasa *slang* dan bahasa Indonesia yang tidak baku [9].

Dalam upaya sistematis untuk mencari model klasifikasi yang paling akurat, banyak penelitian berfokus pada perbandingan kinerja algoritma. Studi-studi ini membandingkan Naive Bayes dengan algoritma *machine learning* klasik lainnya, seperti Support Vector Machine (SVM) [4], [13], atau bahkan dengan pendekatan Deep Learning yang merupakan teknologi yang lebih maju [18]. Meskipun pendekatan Deep Learning telah menunjukkan potensi yang luar biasa dalam menangani fitur yang kompleks, Naive Bayes tetap diakui sebagai *benchmark* yang penting [18]. Keunggulan ini dipertahankan karena Naive Bayes menawarkan kecepatan pelatihan yang superior dan kemudahan interpretasi hasil yang tidak dimiliki oleh model Deep Learning yang seringkali bersifat *black box*.

Selain itu, Fokus pada Pra-Pemrosesan Data telah menjadi area penelitian intensif. Tantangan linguistik yang melekat pada bahasa Indonesia informal telah mendorong studi mendalam pada tahap *preprocessing* [17]. Penelitian menunjukkan bahwa variasi dalam teknik *preprocessing* yang diterapkan memiliki dampak langsung dan signifikan pada akurasi akhir model Naive Bayes. Hal ini menekankan pentingnya langkah-langkah seperti stopword removal (penghapusan kata-kata umum) dan teknik penanganan kata-kata tidak baku atau slang.

Secara keseluruhan, penelitian terdahulu secara kuat membuktikan bahwa Naive Bayes adalah algoritma yang layak dan efisien untuk klasifikasi sentimen [6]. Namun, terdapat kekurangan konsensus mengenai konfigurasi pra-pemrosesan dan *feature engineering* yang ideal untuk mengatasi keragaman tinggi dalam data komentar YouTube. Secara khusus, celah penelitian terletak pada kurangnya studi yang secara eksplisit berfokus pada optimalisasi parameter dan fitur spesifik untuk Naive Bayes dalam konteks ini [8], [20]. Dengan demikian, penelitian ini hadir untuk mengisi celah tersebut dengan secara spesifik menerapkan dan mengevaluasi kinerja Naive Bayes, sambil mengintegrasikan praktik *preprocessing* yang telah direkomendasikan [12], demi menghasilkan temuan yang robust dan relevan

3. Metode yang Diusulkan

Bagian ini menjelaskan desain metodologi, sumber data, dan tahapan eksperimen yang dilakukan untuk mengklasifikasikan sentimen komentar video YouTube menggunakan pendekatan Naive Bayes.

3.1. Sumber Data

Data yang digunakan dalam penelitian ini diperoleh dari *dataset* publik di Kaggle. Dataset ini merupakan kumpulan data yang secara spesifik disiapkan untuk tugas analisis sentimen. Data ini terdiri dari komentar (teks) yang dikumpulkan dari video di *platform* YouTube. Dataset ini dapat diperoleh pada link: https://www.kaggle.com/datasets/muhfalahmubaraq/analisis-sentimen-dengan-metode-naive-bayes.

Penelitian ini menggunakan data tekstual berupa komentar pengguna yang bersumber dari *platform* berbagi video YouTube. Secara spesifik, data yang digunakan diperoleh dari *dataset* sekunder yang tersedia di Kaggle dengan nama *path* analisis-sentimen-dengan-metode-naive-bayes, yang telah disiapkan secara khusus untuk tujuan klasifikasi sentimen. Keseluruhan *dataset* terdiri dari 800 sampel komentar. Jumlah kolom fitur yang digunakan adalah: nomor, nama_pengguna, komentar, dan senitimen. Sebelum digunakan untuk pemodelan, setiap sampel data telah diproses pelabelan secara manual, dikategorikan ke dalam tiga polaritas sentimen utama: Positif, Negatif, dan Netral. Distribusi kelas sentimen menunjukkan data cenderung seimbang antara sentimen Positif 187 data, Negatif 379 data, dan sentimen Netral 233 data.

3.2. Tahapan Eksperimen

Eksperimen klasifikasi sentimen ini dilakukan melalui serangkaian tahapan yang sistematis, yaitu: Pra Pemrosesan Data, Vektorisasi Fitur, Klasifikasi menggunakan Multinomial Naïve Bayes, dan Evaluasi.

3.2.1. Pra-Pemrosesan Data

Tahap Pra-pemrosesan Data merupakan langkah fundamental dan krusial dalam analisis sentimen, bertujuan untuk mengubah teks mentah yang tidak terstruktur dan bising (*noisy*)

menjadi format yang bersih, konsisten, dan optimal untuk diproses oleh algoritma *machine learning*. Proses ini dilakukan melalui serangkaian langkah sistematis:

Pertama, dilakukan Pembersihan Teks (*Cleaning*), di mana karakter-karakter yang tidak relevan atau *noise* seperti tautan URL, tag HTML, simbol-simbol non-alfanumerik yang tidak terstandarisasi, dan tanda baca berlebihan dihapus dari setiap komentar. Selanjutnya, Case Folding diterapkan untuk menyeragamkan seluruh teks menjadi huruf kecil (*lowercase*), memastikan bahwa model tidak memperlakukan kata yang sama (misalnya, "Suka" dan "suka") sebagai dua fitur yang berbeda. Langkah berikutnya adalah Tokenisasi, proses memecah setiap kalimat atau komentar menjadi unit-unit kata individual (token). Token-token ini kemudian dimurnikan lebih lanjut melalui Stopword Removal, yaitu penghapusan kata-kata umum (seperti 'yang', 'dan', 'di' dalam Bahasa Indonesia) yang memiliki frekuensi tinggi namun minim nilai diskriminatif untuk klasifikasi sentimen. Terakhir, dilakukan Stemming, yaitu proses pengembalian setiap kata ke bentuk dasarnya (*root word*). *Stemming* sangat penting untuk mengurangi variasi kata (misalnya, 'menyukai', 'disukai', dan 'kesukaan' dikembalikan menjadi 'suka'), sehingga dapat mengurangi dimensi fitur dan meningkatkan efisiensi serta konsistensi perhitungan probabilitas *likelihood* dalam model Naive Bayes.

3.2.2. Vektorisasi Fitur

Setelah tahap pra-pemrosesan, data disiapkan untuk pelatihan model. Langkah pertama adalah Pembagian Data (*Data Splitting*), di mana fitur x, yaitu komentar yang dinormalisasi dan label target y, yaitu sentimen dipisahkan. Data kemudian dibagi menjadi set pelatihan dan set pengujian menggunakan fungsi *train_test_split* dengan rasio 90% untuk pelatihan dan 10% untuk pengujian, untuk memastikan evaluasi kinerja model dilakukan pada data yang belum pernah dilihat. Karena *output* dari tahap pra-pemrosesan sering kali berbentuk *list* token, dilakukan penyesuaian dengan menggabungkan token-token tersebut menjadi string tunggal per komentar, sesuai dengan persyaratan *input* dari vektorisator.

Selanjutnya, dilakukan Vektorisasi Fitur menggunakan CountVectorizer, yang mengimplementasikan model Bag-of-Words (BoW). Proses ini mengubah teks yang telah dinormalisasi menjadi representasi numerik yang dapat diproses oleh Naive Bayes. CountVectorizer dilatih (*fit*) pada data latih untuk membangun kamus (*vocabulary*) unik dari semua kata, dan kemudian data latih diubah menjadi Matriks BoW, di mana setiap sel menyimpan hitungan frekuensi kemunculan kata dalam komentar. Matriks numerik hasil vektorisasi inilah yang kemudian akan disajikan sebagai fitur *input* ke dalam algoritma Naive Bayes.

3.2.3. Klasifikasi Menggunakan Multinomial Naïve Bayes

Tahapan selanjutnya adalah Klasifikasi Sentimen menggunakan model Multinomial Naive Bayes (MNB), yang merupakan varian Naive Bayes yang paling sesuai untuk data diskret seperti perhitungan frekuensi kata dalam teks. Sebelum proses pelatihan, model klasifikasi menghadapi tantangan ketidakseimbangan kelas (*class imbalance*) yang terlihat dari minoritas sentimen Netral. Untuk mengatasi hal ini, data latih yang telah diverektorisasi diolah lebih lanjut menggunakan teknik penyeimbangan data (*resampling*).

Setelah data disiapkan, model Multinomial Naive Bayes dilatih (*fit*) menggunakan data latih yang sudah diseimbangkan. Dalam proses ini, MNB menghitung semua Probabilitas Prior dan Probabilitas Likehood berdasarkan frekuensi kata dari data *training* yang telah diobati. Setelah model berhasil dilatih, model tersebut digunakan untuk membuat prediksi sentimen pada data uji yang belum pernah dilihat sebelumnya. Hasil prediksi ini, yang merupakan polaritas sentimen (Positif, Negatif, atau Netral), kemudian disalurkan ke tahap akhir untuk dilakukan evaluasi kinerja secara komprehensif.

3.2.4. Evaluasi

Kinerja model dievaluasi dengan membandingkan hasil prediksi dengan label sebenarnya pada data uji menggunakan metrik: Akurasi, Presisi, Recall, F1-Score pada Confusion Matrix.

4. Hasil dan Pembahasan

Bagian ini membahas implementasi metode Analisis Jaringan Sosial (SNA) menggunakan pustaka Python NetworkX, dari pemodelan graf hingga perhitungan Derajat Sentralitas dan visualisasi hasilnya.

Algorithm 1. Persiapan Dataset

```
import string
import pandas as pd
import numpy as np
data = pd.read_excel("/kaggle/input/analisis-sentimen-dengan-metode-naive-bayes/dataset.xlsx", index_col=0)
data.head(10)
```

Implementasi dimulai dengan mengimpor pustaka Pandas dan NumPy untuk persiapan data dan komputasi numerik. Data penelitian kemudian dimuat dari *file* Excel ke dalam *dataframe*. Sebagai langkah verifikasi, perintah *data.head(10)* dijalankan untuk melihat data komentar dan label sentimen telah dimuat secara akurat. Gambar 1 memperlihatkan hasil persiapan dataset.

NO	NAMA_PENGGUNA	COMMENT	SENTIMEN
1	@danofficial2452	Kalau anda pintar menilai.... nilai secara glo...	Negatif
2	@indriSaman	Pantas menang elrumi.. lebih cerdik Elrumi	Positif
3	@smsun1030	Jefri hanya nganalin pukulan 1 2 tapi Elrumi k...	Positif
4	@user-cv7qm7ld8m	Heleeeehhhh jeprii ninju angin doank..wuuzz wu...	Negatif
5	@user-xc9xj4ki2y	El rumi cowok lemah...cuma menang gaya sama ba...	Negatif
6	@mtaqiyuddin4910	Hahah penilain konyol	Negatif
7	@semutireng8332	Jefri power dan pukulan good tapi jarang in da...	Negatif
8	@user-xo4rb2lh6x	Kocak 🥰🥰🥰🥰	Netral
9	@bayuanggara7213	Sebenarnya Jefri Bagus Powernya Dan Staminanya...	Netral
10	@ardiansyah-3268	Babak terakhir sama2 letoy seharusnya draw	Negatif

Gambar 1. Hasil Proses Load Dataset.

Algorithm 2. Casefolding

```
data['komen case folding'] = data['COMMENT'].str.lower()
print("\n Data Setelah Case Folding:")
(data.head(10))
```

Algoritma 2 merupakan proses *Case Folding* pada data komentar. Kode ini mengambil kolom "comment" menggunakan fungsi *.str.lower()*untuk mengubah semua karakter huruf besar menjadi huruf kecil. Proses ini penting untuk memastikan bahwa sistem klasifikasi (Naive Bayes) memperlakukan kata yang sama, terlepas dari kapitalisasinyasebagai fitur tunggal, sehingga mengurangi dimensi fitur dan meningkatkan konsistensi serta akurasi model.

NO	NAMA_PENGGUNA	COMMENT	SENTIMEN	komen case folding
1	@danofficial2452	Kalau anda pintar menilai.... nilai secara glo...	Negatif	kalau anda pintar menilai.... nilai secara glo...
2	@indriSaman	Pantas menang elrumi.. lebih cerdik Elrumi	Positif	pantas menang elrumi.. lebih cerdik elrumi
3	@smsun1030	Jefri hanya nganalin pukulan 1 2 tapi Elrumi k...	Positif	jefri hanya nganalin pukulan 1 2 tapi elrumi k...
4	@user-cv7qm7ld8m	Heleeeehhhh jeprii ninju angin doank..wuuzz wu...	Negatif	heleeeehhhh jeprii ninju angin doank..wuuzz wu...
5	@user-xc9xj4ki2y	El rumi cowok lemah...cuma menang gaya sama ba...	Negatif	el rumi cowok lemah...cuma menang gaya sama ba...
6	@mtaqiyuddin4910	Hahah penilain konyol	Negatif	hahah penilain konyol
7	@semutireng8332	Jefri power dan pukulan good tapi jarang in da...	Negatif	jefri power dan pukulan good tapi jarang in da...
8	@user-xo4rb2lh6x	Kocak 😂😂😂😂	Netral	kocak 😂😂😂😂
9	@bayuanggara7213	Sebenarnya Jefri Bagus Powernya Dan Staminanya...	Netral	sebenarnya jefri bagus powernya dan staminanya...
10	@ardiansyah-3268	Babak terakhir sama2 letoy seharusnya draw	Negatif	babak terakhir sama2 letoy seharusnya draw

Gambar 2. Hasil Proses Casefolding.

Proses tokenisasi berfungsi untuk menguraikan suatu teks atau dokumen menjadi unit-unit yang lebih kecil yang disebut sebagai *token* dalam bentuk kata atau tanda baca untuk memecah teks menjadi elemen yang dapat dianalisis. Algoritma 3 difokuskan untuk untuk tokenisasi menggunakan pustaka NLTK (*Natural Language Toolkit*). Modul punkt NLTK yang merupakan model *pre-trained* yang diperlukan untuk fungsi word_tokenize. Data *dataframe* dibersihkan dari baris yang mengandung nilai kosong (*NaN*) menggunakan *dropna*. Fungsi *word_tokenize* dari NLTK diterapkan pada kolom komen case folding yang sudah diseragamkan hurufnya. Hasil tokenisasi berupa daftar kata-kata untuk setiap komentar.

Algorithm 3. Tokenizing

```
import nltk
from nltk.tokenize import word_tokenize
download_dir = '/kaggle/working/nltk_data'
import os
os.makedirs(download_dir, exist_ok=True)
nltk.data.path.append(download_dir)
try:
    nltk.download('punkt', download_dir=download_dir)
    print("v 'punkt' berhasil diunduh ke /kaggle/working/nltk_data")
except Exception as e:
    print(f" x Gagal mengunduh 'punkt': {e}")
text = "Ini adalah contoh kalimat. Apakah tokenisasi berhasil?"
tokens = word_tokenize(text)
print("\nTokenisasi berhasil:", tokens)
data = data.dropna(subset=['komen case folding'])
data['komen tokenized'] = data['komen case folding'].apply(word_tokenize)
print("\nData Setelah Tokenizing:")
(data.head(10))
```

NO	NAMA_PENGGUNA	COMMENT	SENTIMEN	komen case folding	komen tokenized
1	@danofficial2452	Kalau anda pintar menilai.... nilai secara glo...	Negatif	kalau anda pintar menilai.... nilai secara glo...	[kalau, anda, pintar, menilai, ..., ., nilai, ...
2	@indriSaman	Pantas menang elrumi.. lebih cerdik Elrumi	Positif	pantas menang elrumi.. lebih cerdik elrumi	[pantas, menang, elrumi.., lebih, cerdik, elrumi]
3	@smsun1030	Jefri hanya nganalin pukulan 1 2 tapi Elrumi k...	Positif	jefri hanya nganalin pukulan 1 2 tapi elrumi k...	[jefri, hanya, nganalin, pukulan, 1, 2, tapi, ...
4	@user-cv7qm7ld8m	Heleeeehhhh jeprii ninju angin doank..wuuzz wu...	Negatif	heleeeehhhh jeprii ninju angin doank..wuuzz wu...	[heleeeehhhh, jeprii, ninju, angin, doank..wuu...
5	@user-xc9xj4ki2y	El rumi cowok lemah...cuma menang gaya sama ba...	Negatif	el rumi cowok lemah...cuma menang gaya sama ba...	[el, rumi, cowok, lemah, ..., cuma, menang, ga...
6	@mtaqiyuddin4910	Hahah penilain konyol	Negatif	hahah penilain konyol	[hahah, penilain, konyol]
7	@semutireng8332	Jefri power dan pukulan good tapi jarang in da...	Negatif	jefri power dan pukulan good tapi jarang in da...	[jefri, power, dan, pukulan, good, tapi, jaran...
8	@user-xo4rb2lh6x	Kocak 😂😂😂😂	Netral	kocak 😂😂😂😂	[kocak, 😂😂😂😂]
9	@bayuanggara7213	Sebenarnya Jefri Bagus Powernya Dan Staminanya...	Netral	sebenarnya jefri bagus powernya dan staminanya...	[sebenarnya, jefri, bagus, powernya, dan, stam...
10	@ardiansyah-3268	Babak terakhir sama2 letoy seharusnya draw	Negatif	babak terakhir sama2 letoy seharusnya draw	[babak, terakhir, sama2, letoy, seharusnya, draw]

Gambar 3. Hasil Tokenizing.

Algoritma keempat adalah tahapan Pembersihan Teks (Cleaning) dan Stopword Removal dalam proses *preprocessing*, yang bertujuan untuk menyaring *noise* linguistik dari data komentar. Modul *stopwords* memuat daftar kata-kata umum yang perlu disaring.

Proses *Stopword Removal* diterapkan menggunakan daftar kata-kata berhenti Bahasa Indonesia pada kode "stopwords.words('indonesian')". Tahap ini digabungkan dengan Normalisasi teks dalam fungsi *lambda* dengan proses *Iterasi Token* untuk mengiterasi melalui setiap token di kolom komen tokenized, *Filter Alfaber* untuk memastikan hanya kata-kata alfabet yang dipertahankan, secara efektif menghilangkan angka, tanda baca yang tersisa, dan karakter khusus, dan *Filter Stopword* untuk memeriksa apakah *bukan* merupakan bagian dari daftar kata-kata berhenti Bahasa Indonesia. Hasilnya, data komentar kini tersimpan di kolom baru bernama komen normalized, yang hanya berisi kata-kata signifikan yang akan digunakan sebagai fitur pada gambar 4. Proses ini krusial untuk mengurangi dimensi fitur dan meningkatkan akurasi model Naive Bayes.

Algorithm 4. Normalisasi

```
import nltk
from nltk.corpus import stopwords
import re
import os
download_dir = '/kaggle/working/nltk_data'
os.makedirs(download_dir, exist_ok=True)
nltk.data.path.append(download_dir)
try:
    nltk.download('stopwords', download_dir=download_dir)
    print("v 'stopwords' berhasil diunduh ke /kaggle/working/nltk_data")
except Exception as e:
    print(f"x Gagal mengunduh 'stopwords': {e}")
bahasa = 'english' # Ganti ke 'indonesian' jika diperlukan
stop_words_list = stopwords.words(bahasa)
print(f"\nJumlah Stopwords ({bahasa}):", len(stop_words_list))
print(f"5 Stopwords pertama: {stop_words_list[:5]}")
stop_words = set(stopwords.words('indonesian'))
data['komen normalized'] = data['komen tokenized'].apply(lambda tokens: [word for word
in tokens if word.isalpha() and word not in stop_words])
print("\nData Setelah Normalization:")
(data.head(10))
```

komen case folding	komen tokenized	komen normalized
kalau anda pintar menilai.... nilai secara glo...	[kalau, anda, pintar, menilai, ..., ., nilai, ...]	[pintar, menilai, nilai, global, pemenang, nya...]
pantas menang elrumi.. lebih cerdik elrumi	[pantas, menang, elrumi.., lebih, cerdik, elrumi]	[menang, cerdik, elrumi]
jefri hanya nganalin pukulan 1 2 tapi elrumi k...	[jefri, hanya, nganalin, pukulan, 1, 2, tapi, ...]	[jefri, nganalin, pukulan, elrumi, kombinasi, ...]
heleeeehhhh jeprii ninju angin doank..wuuzz wu...	[heleeeehhhh, jeprii, ninju, angin, doank..wuu...]	[heleeeehhhh, jeprii, ninju, angin, wuuzzz]
el rumi cowok lemah...cuma menang gaya sama ba...	[el, rumi, cowok, lemah, ..., cuma, menang, ga...]	[el, rumi, cowok, lemah, menang, gaya, bacotny...]
hahah penilain konyol	[hahah, penilain, konyol]	[hahah, penilain, konyol]
efri power dan pukulan good tapi jarang in da...	[jefri, power, dan, pukulan, good, tapi, jaran...]	[jefri, power, pukulan, good, jarang, in, tena...]
kocak 🥰🥰🥰🥰	[kocak, 🥰🥰🥰🥰]	[kocak]
sebenarnya jefri bagus powernya dan staminanya...	[sebenarnya, jefri, bagus, powernya, dan, stam...]	[jefri, bagus, powernya, staminanya, tahan, ke...]
babak terakhir sama2 letoy seharusnya draw	[babak, terakhir, sama2, letoy, seharusnya, draw]	[babak, letoy, draw]

Gambar 4. Hasil Normalisasi.

Visualisasi frekuensi kata dilakukan menggunakan *Word Cloud* untuk mengidentifikasi distribusi dan kata-kata yang paling dominan dalam *dataset* komentar. *Word Cloud* ini efektif merepresentasikan bobot statistik dari setiap kata, semakin besar ukuran visual kata tersebut dalam awan, semakin tinggi frekuensi kemunculannya dalam keseluruhan korpus. Visualisasi ini krusial karena memberikan gambaran intuitif mengenai fokus utama pembahasan dalam komentar YouTube dan kata-kata kunci sentimen diperlihatkan pada gambar 5.

Gambar 5. Hasil WordCloud.

Algoritma 5 adalaha proses pembagian data dan vektorisasi fitur. Proses ini dimulai dengan memisahkan *dataset* menjadi fitur *x*, yang merupakan kolom 'komen normalized' dan label target *y*, yaitu kolom 'SENTIMEN'. Data dibagi menjadi data pelatihan (*training set*) dan data pengujian (*testing set*) menggunakan fungsi *train_test_split*, dengan ukuran test_size=0.10, yang berarti 90% dari total data dialokasikan untuk data training, dan 10% sebagai data uji. Data teks perlu diubah menjadi format numerik. *CountVectorizer* diinisialisasi untuk menerapkan model *Bag-of-Words (BoW)*. Fungsi *fit_transform* diterapkan pada data latih untuk membangun kamus (*vocabulary*). Matriks ini mencatat frekuensi kemunculan setiap kata.

Algorithm 5. Data Splitting dan Vektorisasi

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
x = data['komen normalized']
y = data['SENTIMEN']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.10, random_state=0)
x_train_processed = x_train.apply(lambda tokens: ' '.join(tokens))
x_test_processed = x_test.apply(lambda tokens: ' '.join(tokens))
vectorizer = CountVectorizer()
x_train_vec = vectorizer.fit_transform(x_train_processed)
x_test_vec = vectorizer.transform(x_test_processed)
```

Algoritma 6 adalah proses training dan prediksi Naïve bayes. Tahapan ini berfokus pada implementasi algoritma *Multinomial Naive Bayes (MNB)*, yang merupakan varian Naive Bayes yang paling sesuai untuk menangani data diskret seperti *count* kata atau frekuensi Term-Frequency Inverse Document Frequency (TF-IDF) dalam klasifikasi teks. Pelatihan model melalui fungsi *.fit()*.Fungsi *.predict()* digunakan untuk menguji kemampuan generalisasi model. Model yang telah ditraining kemudian memprediksi sentimen (*y_pred*) pada data uji (*x_test_vec*) yang dipisahkan. Hasil prediksi ini, berupa label sentimen (Positif, Negatif, Netral), akan menjadi *output* yang digunakan pada tahap evaluasi kinerja selanjutnya.

Algorithm 6. Training dan Prediksi Model Naive Bayes

```
from sklearn.naive_bayes import MultinomialNB
naive_bayes = MultinomialNB()
naive_bayes.fit(x_train_resampled, y_train_resampled)
y_pred = naive_bayes.predict(x_test_vec)
```

Algoritma 7 berfokus pada pengukuran dan pelaporan kinerja *Model Naive Bayes* yang telah dilatih. Metrik utama dihitung dengan membandingkan label sentimen aktual dari data uji (*y_test*) dengan hasil prediksi model (*y_pred*). Akurasi (*Accuracy*) model yaitu, proporsi total

prediksi yang benar dan hasilnya disimpan dalam *variabel accuracy*. *Classification_report* dibuat untuk memberikan laporan kinerja yang lebih rinci.

Algorithm 7. Evaluasi

```
from sklearn.metrics import accuracy_score, classification_report
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred, target_names=["NEGATIF", "POSITIF"])
print("Akurasi Model Naive Bayes : ", accuracy)
print("\nLaporan Klasifikasi :\n", classification_rep)
```

Hasil evaluasi pada gambar 6 menunjukkan bahwa akurasi *Model Naive Bayes* secara keseluruhan mencapai 0.7532 atau 75.32%. Meskipun akurasi keseluruhan cukup baik, laporan klasifikasi mendetail menunjukkan perbedaan kinerja yang signifikan antar kelas. Kelas NEGATIF menunjukkan kinerja yang sangat kuat dengan Presisi tinggi sebesar 0.92, dan F1-Score 0.83. Sebaliknya, kelas POSITIF memiliki Presisi yang jauh lebih rendah yaitu 0.46, meskipun memiliki Recall yang sebanding yaitu 0.76. Hal ini mengindikasikan bahwa model sangat baik dalam mengidentifikasi sentimen Negatif, namun cenderung kurang baik mengklasifikasikan komentar Positif sebagai Negatif, sebuah temuan yang mungkin dipengaruhi oleh ketidakseimbangan kelas atau karakteristik fitur data uji.

```
Akurasi Model Naive Bayes :  0.7532467532467533

Laporan Klasifikasi :
              precision    recall  f1-score   support

     NEGATIF       0.92      0.75      0.83        60
     POSITIF       0.46      0.76      0.58        17

    accuracy                           0.75        77
   macro avg       0.69      0.76      0.70        77
weighted avg       0.82      0.75      0.77        77
```

Gambar 6. Hasil Evaluasi.

5. Kesimpulan

Eksperimen pemodelan sentimen komentar video YouTube menggunakan Algoritma Multinomial Naive Bayes (MNB) dengan fitur CountVectorizer dapat disimpulkan bahwa kinerja *Naive Bayes* secara keseluruhan mencapai akurasi sebesar 75.32% pada data uji. Akurasi ini mengonfirmasi bahwa Naive Bayes adalah algoritma yang layak dan efisien untuk klasifikasi sentimen pada data komentar YouTube. Efek *Resampling* dan *Preprocessing* pada penggunaan serangkaian langkah pra-pemrosesan yang ketat termasuk *Case Folding*, *Stopword Removal*, dan *Stemming* serta penerapan penyeimbangan data (*resampling*) pada data latih terbukti efektif dalam menyiapkan fitur yang optimal untuk perhitungan probabilitas Naive Bayes, dan adanya ketidakseimbangan Kinerja Kelas. Meskipun akurasi keseluruhan memadai, laporan klasifikasi menunjukkan adanya ketidakseimbangan kinerja antar kelas. Model sangat kuat dalam memprediksi sentimen negaitf dengan Presisi 0.92, dan F1-Score 0.83, namun memiliki kelemahan dalam memprediksi sentimen positif dengan nilai Presisi 0.46.

Untuk meningkatkan kinerja model, mengatasi keterbatasan yang ada, dan memperkaya hasil penelitian, disarankan beberapa eksperimen lanjutan berikut dengan penerapan N-Gram dengan Bigram atau Trigram yang dapat meningkatkan pemahaman konteks negasi atau intensitas sentimen, yang sering kali hilang dalam model Bag-of-Words biasa. Melakukan perbandingan kinerja model Naive Bayes dengan algoritma *machine learning* lain yang populer dalam klasifikasi sentimen, seperti Support Vector Machine (SVM) atau K-Nearest Neighbors (KNN).

Konflik Kepentingan : Para penulis menyatakan tidak ada konflik kepentingan.

Daftar Pustaka

[1] A. N. Pratama and D. A. Prasetyo, "Analisis Peran YouTube Sebagai Media Penyebaran Informasi di Era Digital," *Jurnal Ilmu Komputer dan Informasi*, vol. 15, no. 2, pp. 45–52, 2024.

[2] B. W. Susanto and C. R. Sari, "Karakteristik Bahasa Komentar Netizen di Media Sosial dan Tantangannya bagi Analisis Sentimen," *Prosiding Seminar Nasional Komputer dan Teknologi Informasi (SNKTI)*, pp. 110–116, 2023.

[3] J. Siregar and F. Rahman, "A Review of Sentiment Analysis Techniques and Applications: From Machine Learning to Deep Learning," *IEEE Access*, vol. 11, pp. 25000-25015, 2023.

[4] F. A. Ginting and K. A. Laksmi, "Klasifikasi Sentimen Teks menggunakan Metode Support Vector Machine (SVM) dan Naive Bayes," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 5, no. 1, pp. 30–38, 2021.

[5] B. Wijaya and L. Setiawan, "Perbandingan Akurasi Klasifikasi Sentimen antara Metode Manual dan Otomatis," *Seminar Nasional Teknologi Informasi dan Komunikasi (SENDIKA)*, pp. 400–405, 2022.

[6] C. Putra and D. Yulianto, "Optimasi Algoritma Naive Bayes untuk Klasifikasi Teks dengan Fitur N-Gram," *Jurnal Rekayasa Informasi dan Teknologi*, vol. 9, no. 3, pp. 150–158, 2020.

[7] N. Ramadhan and O. Dewi, "Penerapan Teorema Bayes dalam Pemodelan Probabilitas Klasifikasi Data," *Jurnal Sains dan Teknologi*, vol. 2, no. 1, pp. 1–7, 2024.

[8] T. Simanjuntak and P. M. Wibowo, "Pemilihan Fitur Terbaik dalam Analisis Sentimen Menggunakan Information Gain," *International Journal of Advances in Data Science*, vol. 7, no. 2, pp. 100–108, 2025.

[9] K. Sari and R. Abdullah, "Sentiment Analysis on Indonesian Tweets using Naive Bayes and Lexicon-Based Approach," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 6, pp. 3000–3008, 2020.

[10] M. Putra and S. Dewi, "Machine Learning for Social Media Data Analysis: A Focus on YouTube Comments," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 4, pp. 800–810, 2021.

[11] M. N. Hidayat and S. T. Wibawa, "Analisis Sentimen Komentar Video Politik di YouTube dengan Naive Bayes Classifier," *Jurnal Informatika*, vol. 13, no. 1, pp. 50–59, 2022.

[12] S. Rahayu and K. Putri, "Performance Evaluation of Naive Bayes on Sentiment Classification with Different Preprocessing Techniques," *International Journal of Computer Applications*, vol. 175, no. 3, pp. 15–20, 2023.

[13] O. P. Siregar and V. R. P. Hutagalung, "A Comparative Study of Machine Learning Algorithms for Sentiment Analysis on Product Reviews," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 11, no. 5, pp. 1000–1008, 2024.

[14] P. Dewi and N. Sari, "Challenges and Opportunities of Sentiment Analysis on Informal Indonesian Language in Social Media," *Journal of Informatics Engineering and Computer Science (JIECOM)*, vol. 5, no. 1, pp. 10–18, 2025.

[15] R. F. Akbar and Z. Lubis, "Improved Naive Bayes with Feature Selection for Sentiment Analysis on Movie Reviews," *International Journal of Artificial Intelligence and Robotics*, vol. 4, no. 1, pp. 50–57, 2020.

[16] K. P. Sitompul and Y. B. Simatupang, "Klasifikasi Sentimen Berita Online Menggunakan Naive Bayes dan Pembobotan TF-IDF," *Jurnal Sistem Informasi Bisnis (JSINBIS)*, vol. 10, no. 2, pp. 120–128, 2021.

[17] S. T. Sitohang and Z. A. Manurung, "Text Preprocessing Techniques in Indonesian Sentiment Analysis: A Comparative Study," *Journal of Big Data Science and Technology*, vol. 3, no. 1, pp. 1–10, 2022.

[18] O. P. Siregar and A. M. H. Damanik, "Deep Learning vs. Machine Learning in Sentiment Analysis: A Performance Review," *IEEE Conference on Big Data (BigData)*, pp. 1500–1509, 2023.

[19] U. P. Sitorus and B. C. Sihotang, "Sentiment Analysis of YouTube Comments on Education Videos: A Naive Bayes Approach," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 13, no. 2, pp. 2500–2507, 2024.

[20] Z. Tampubolon and E. H. Purba, "Optimalisasi Parameter Naive Bayes untuk Klasifikasi Sentimen Multikelas," *Jurnal Riset Komputer*, vol. 7, no. 3, pp. 180–188, 2025.