JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER

p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

Implementasi Metode Naive Bayes Berbasis Python Untuk Analisis Sentimen Aplikasi Google Gemini

Ikhwan[1*], Yudhi Septianto[2]

[1,2]Fakultas Ilmu Komputer, Program Studi Sistem Informasi, Universitas Pamulang, Serang, Indonesia
Email: [1*]dosen03358@unpam.ac.id, [2]dosen03465@unpam.ac.id
(*Email Corresponding Author: dosen03358@unpam.ac.id)
Received: August 18,2026 | Revision: September 7,2026 | Accepted: September 8,2026

Abstrak

Artificial Intelligence (AI) mengalami perkembangan dalam beberapa tahun terakhir, salah satunya adalah Google Gemini. Tingginya penggunaan layanan digital ini menghasilkan ulasan pengguna yang sangat besar dan tersimpan informasi penting terkait kepuasan, pengalaman, serta pandangan pengguna dalam menilai kualitas layanan. Kondisi ini terlihat dari jumlah unduhan 1 M+, 4,7 bintang dari 5, dan 41,8 juta lebih ulasan. Untuk menganalisis sentimen, peneliti menggunakan tahapan, di antaranya pengumpulan data yang dilakukan dari Maret hingga Juni 2026 dengan *scraping* di Google Play Store dan mendapatkan 20.000 ulasan. Data *preprocessing* di antaranya *cleaning*, *stopword removal*, *case folding*, *stemming* dan tokenisasi dengan hasil pemrosesan data 11.035. Hasil labeling didapat ulasan positif 8.197, negatif 2.323, dan netral 515. Klasifikasi hanya pada kelas positif dan negatif, untuk netral tidak digunakan. Algoritma menggunakan *Multinomial* Naive Bayes dengan pemberian bobot pada kata menggunakan TF-IDF, serta *confusion matrix* untuk evaluasi yang terdiri akurasi, recall, presisi, F1-Score dan juga visualisasi data *word cloud.* Hasil performa yang didapat yang cukup baik. Pada rasio 80:20, data dibagi dalam data *training* dan *testing* dengan akurasi 0.8826 atau 88,26%, presisi 0.8968 atau 89,68%, recall 0.9605 atau 96,05%, dan F1-score 0.9276 atau 92,76%. Tujuan penelitian untuk menganalisis sentimen oleh pengguna dan mengklasifikasi data sentimen, serta evaluasi performa pada model menggunakan algoritma Naive Bayes berbasis Python. Harapan pada penelitian ini dapat menjadi bahan evaluasi bagi pengembang agar lebih memaksimalkan aplikasi dan pelayanannya.

Kata Kunci: Naive Bayes, Analisis Sentimen, Google Gemini, Python, TF-IDF

Abstract

Artificial Intelligence (AI) has seen significant advancements in recent years, one of which is Google Gemini. The high usage of these digital services has generated a massive volume of user reviews that contain important information regarding user satisfaction, experiences, and perspectives on service quality. This is evident from the 1 million+ downloads, a 4.7-star rating out of 5, and over 41.8 million reviews. To analyze sentiment, the researchers followed several steps, including data collection conducted from March through June 2026 by scraping the Google Play Store, which yielded 20,000 reviews. Data preprocessing included cleaning, stopword removal, case folding, stemming, and tokenization, resulting in 11,035 processed data points. The labeling results yielded 8,197 positive reviews, 2,323 negative reviews, and 515 neutral reviews. Classification was performed only for the positive and negative classes; the neutral class was not used. The algorithm uses the Multinomial Naive Bayes model with TF-IDF-weighted words, as well as a confusion matrix for evaluation—which includes accuracy, recall, precision, and F1-score—and data visualization via a word cloud. The performance results were quite good. With an 80:20 split, the data was divided into training and testing sets, yielding an accuracy of 0.8826 (88.26%), precision of 0.8968 (89.68%), recall of 0.9605 (96.05%), and an F1-score of 0.9276 (92.76%). The purpose of this study is to analyze user sentiment and classify sentiment data, as well as to evaluate the performance of a model using a Python-based Naive Bayes algorithm. It is hoped that this study will serve as a basis for evaluation by developers to further optimize their applications and services.

Keywords*: Naive Bayes, Sentiment Analysis, Google Gemini, Python, TF-IDF*

1. PENDAHULUAN

Di era berkembangnya teknologi informasi dan komunikasi, khususnya pada bidang *artificial inteligence* (AI) telah mengalami perkembangan dalam beberapa tahun terakhir. Perkembangan tersebut menghadirkan kecerdasan buatan dengan teknologi AI model *Large Language Model* (LLM) [1]. Salah satu platform yang digagas oleh Google adalah Mobile Gemini AI. Kehadiran Google Gemini telah berevolusi pada sudut pandang manusia terkait bagaimana cara manusia berinteraksi dengan teknologi dan pemanfaatannya. Platform ini dikembangkan agar dapat membantu pengguna dalam berbagai aktivitas dengan interaksi berbasis bahasa alami [2]. Pemanfaatan Google Gemini secara signifikan mengalami peningkatan. Hal ini dapat terlihat di web *Google Play Store* dengan unduhan mencapai 1 M+, 4,7 dari bintang 5, dan 41,8 juta ulasan dari pengguna.

Intensitas yang tinggi pada penggunaan aplikasi menghasilkan sentimen atau ulasan pengguna yang cukup banyak. Sentiman tersebut menjadi data yang cukup penting karena terdapat penilaian, pengalaman, pendapat pengguna terhadap layananan aplikasi. Untuk menguji dan mengukur data yang cukup banyak dan besar pada sentimen tersebut tentu tidak bisa dilakukan secara manual mengingat data yang begitu kompleks. Dengan demikian pemanfaatan teknologi dengan pendekatan komputasional dan implementasi algoritma dapat membantu dalam analisis sentimen pengguna tersebut. Analisis sentimen sendiri merupakan proses atau teknik yang penting untuk digunakan untuk memproses data teks dalam mengetahui komentar *user* mengenai suatu produk atau aplikasi [3]. Analisis sentimen

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

merupakan metode yang digunakan pada *natural language preprocessing* (NLP) yang dikenal dengan pemrosesan dengan model bahasa alami dengan *split* teks menjadi kategori positif, negatif, dan netral [4].

Untuk melihat referensi dan perbandingan dari penelitian sebelumnya terkait memanfaatkan analisis sentimen pada studi kasus, dapat kita lihat pada penelitian sebelumnya yaitu. Pada penelitian [5], Model Naive Bayes dan *Term Frequency-Inverse Document Frequency* digunakan dalam membangun model dengan kesesuaian artikel jurnal sinta dengan berdasar pada metadata. Penelitian ini menyebutkan adanya keterbatasan dalam pembahasan terkait penentuan kesesuaian artikel pada jurnal yang terindeks di sinta, sehingga penelitian lebih lanjut perlu dilakukan untuk mengisi keterbatasan tersebut. *Dataset* menggunakan data sekitar 1.200 metadata artikel yang didapat dari *crawling* manual pada portal sinta. Selain TF-IDF, penerapan algoritma diiringi dengan skema *cross-validation*. Hasil evaluasi didapat akurasi 0,7058, presisi 0,6977, recall 0,7133 dan F1-score 0,7065. Metode Naive Bayes dengan TF-IDF sudah cukup baik digunakan, akan tetapi terdapat ruang yang perlu ditingkatkan untuk penelitian selanjutnya. Meskipun Naïve Bayes efisien dan cepat dalam proses pelatihan, namun performanya dipengaruhi oleh asumsi independensi kata dannya ketidakseimbangan kelas [5].

Begitu juga dengan penelitian analisis sentimen pada aplikasi pemilu, penelitian ini menggunakan Naive Bayes. Penelitian dilakukan pada aplikasi Komisi Pemilihan Umum (KPU) dengan tujuan menganalisis perasaan pengguna terkait aplikasi. *Dataset* didapat dari ulasan di *Google Play Store* sebanyak 100 *review* dan data tersebut displit jadi 160 data pelatihan dan 40 data pengujian. *Preprocessing* meliputi tahapan pembersihan, tokenisasi dan vektorisasi. Hasil tersebut menunjukkan bahwa mayoritas pengguna mengungkapkan kata positif. Akurasi yang didapat yaitu 82,5% untuk sentimen positif dan 75% untuk sentimen negatif [4]. Selanjutnya, penelitian yang dilakukan oleh [6] berisi analisis sentimen pada makan bergizi gratis (MBG) yang menggunakan model Naive Bayes. Tujuan penelitian untuk menganalisis komentar positif dan negatif pada youtube dengan menggunakan bahasa Python. Pelabelan menggunakan teknik *sentiment polarity* yang dilakukan secara otomatis. 1.470 *dataset* komentar yang terdiri dari 1.403 positif dan 67 negatif digunakan pada penelitian. Akurasi didapat 87,35% negatif, dan recall 56,76%. Hasil pada penelitian ini didapat pemodelan klasifikasi dengan komentar negatif.

Selanjutnya, pada penelitian [7] analisis sentimen dilakukan diplatform chatgpt di web *Google Play Store* dengan algoritma SVM dan Naive bayes. Tujuan dari penelitian untuk analisis ulasan pengguna dan melihat perbandingan dengan dua algoritma tersebut menggunakan bahasa Python. *Dataset* diambil dari *scraping* sehingga diperoleh 9,879 ulasan, data tersebut finalisasi dari pembersihan, *preprocessing* yang meliputi *stopword removal*, *case folding*, *stemming* dan *tokenizing*. Pelabelan menggunakan teknik *lexicon-based* dan pembentukan fitur menggunakan TF-IDF. Rasio pembagi data 80:20 dengan hasil akurasi yang didapat sebesar 90% [7]. Selain itu, terdapat juga analisis sentimen pada genreative AI menggunakan metode Naive Bayes dengan bahasa Python. Penelitian ini menganalisis sentimen ulasan pengguna terhadap aplikasi Generative AI. *Dataset* yang digunakan sebanyak 50.000 ulasan yang terdiri dari 5 aplikasi yaitu Google Gemini, ChatGPT, Claude, Microsoft Copilot dan Perplexity [8]. Tahapannya antara lain *preprocessing* teks yang terdiri atas *case folding, cleaning, tokenizing, stopword removal*, serta transformasi fitur menggunakan metode TF-IDF dengan representasi *unigram* dan *bigram*. Klasifikasi dalam tiga kategori yaitu positif, negatif dan netral. Hasil menunjukkan distribusi sentimen sebesar 33.695 atau 67,4% data positif, 13.275 atau 26,6% data negatif, dan 3.030 atau 6,1% data netral. Hasil evaluasi model menunjukan performa yang baik dengan tingkat akurasi sebesar 83,07%, presisi 77,79%, *recall* 83,07%, dan F1-*score* 80,29%.

Meskipun ada metode lain yang mungkin memiliki akurasi lebih tinggi, metode Naive Bayes masih tetap menjadi pilihan yang tepat karena lebih fleksibel dan tidak terlalu bergantung pada banyak fitur rumit [9]. *Multinomial* Naive Bayes adalah salah satu dari metode *probabilistic reasoning.* Metode tersebut memanfaatkan nilai probabilitas dari setiap data yang digunakan [10]. Metode Naive Bayes merupakan salah satu algoritma yang cukup populer karena sederhana dan efektif dalam memproses teks pada pemrosesan bahasa alami [11]. Algoritma ini membangun model untuk mengklasifikasi sentimen berdasarkan data yang sudah diproses sebelum membangun model. Model ini berjalan dengan menggunakan probabilitas pada teks termasuk dalam kategori tertentu berdasarkan probabilitas pada teorema Bayes [12].

Pada penelitian ini, TF-IDF dipilih karena lebih ringan, mudah diinterpretasikan, dan efektif untuk data teks berskala besar [8]. Meskipun terdapat metode yang lebih modern seperti *Word2Vec* dan *FastText* yang dapat mampu memahami hubungan semantik yang lebih kompleks, namun kebutuhan sumber daya pada komputasi yang lebih besar menjadi alasan tersendiri dan umumnya model *deep learning* yang cocok untuk digunakan. Meskipun pada penelitian terdahulu banyak yang membahas aspek analisis sentimen, namun terdapat kesenjangan terhadap data, akurasi, model analisis sentimen dan tujuan analisis yang diinginkan. Selain itu, prilaku kategori netral yang dianggap tidak penting namun tetap digunakan sebagai bahan analisis, padahal hal ini dapat membuat akurasi dalam model yang dibangun akan menjadi terganggu. Kesenjangan ini menjadi alasan yang cukup kuat untuk penelitian lebih lanjut dalam menentukan pola sentimen yang lebih baik. Kontribusi dan kebaruan pada penelitian ini bukan hanya terfokus pada hasil evaluasi model dan akurasi akan tetapi terfokus pada data terbaru sebagai dataset, evaluasi performa pada model klasifikasi, analisis dominan kata terhadap ulasan positif dan negatif melalui *wordcloud*.

Tahapan penelitian ini yang dilakukan terdiri dari pengumpulan data dengan teknik web *scraping*, pemrosesan data melalui proses *case folding*, *cleaning*, *stopword removal*, *tokenizing* dan *stemming*. Selanjutnya dilakukan pelabelan dan transformasi fitur dengan menggunakan metode TF-IDF. Klasifikasi data ke dalam kategori positif, negatif dan netral menggunakan algoritma *Multinomial* Naive Bayes. Paramter yang digunakan pada klasifikasi yaitu

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

firut *TfidfVectorizer* dengan parameter *max_features*=5000, *ngram_range*=(1,1) dan *min_df*=2. dan juga evaluasi menggunakan *confusion matrix*, serta visualisasi *word cloud* untuk menentukan sentimen positif dan negatif yang dominan. Pendekatan pembobotan kata dengan TF-IDF dapat memberikan nilai pada setiap kata dengan menentukan frekuensi kemunculan istilah pada kata tersebut, serta efektif dalam menentukan tingkat kepentingan suatu kata dalam dokumen [13].

Tujuan dari penelitian ini yaitu untuk menganalisis sentimen ulasan pengguna pada Google Gemini AI di Google Play Store dengan Naive Bayes. Harapan dari hasil penelitian ini yaitu dapat memberi wawasan lebih yang mendalam terkait pola sentimen. Serta memberikan rekomendasi bagi pengembang aplikasi dan sistem pelayanannya dalam meningkatkan layanan.

2. METODOLOGI PENELITIAN

Metodologi pada penelitian terdiri dari tahapan pengumpulan data, pemrosesan data yang terdiri dari pembersihan data, *case folding*, *stopword removal*, tokenisasi, dan *stemming*. Berikutnya pemberian bobot kata dengan teknik TF-IDF, pemodelan Naive Bayes dan proses evaluasi. Gambar 1 berikut ini merupakan tahapan penelitian.

Gambar 1. Tahapan penelitian

2.1 Pengumpulan data

Pengumpulan data dengan teknik *scraping* di web Google Play Store pada ulasan pengguna Google Gemini. Data yang berhasil didapat sejumlah 20.000 ulasan dengan rentang data dari bulan Juni hingga Agustus 2026. Teknik *scraping* dengan memanfaatkan *Google Colaboratory* menggunakan bahasa pemrograman Python v3 dengann *library* dari *google-play-scraper* versi 1.2.7, *scikit-learn* versi 1.6.1, dan *sastrawi* versi 1.0.1, yang dilakukan pada tanggal 12 Agustus 2026. Tujuan *scraping* ini adalah untuk mendukung efisiensi waktu pengambilan data dengan jumlah yang banyak, sehingga dapat mempercepat perolehan data.

Parameter yang digunakan dalam scraping ini meliputi parameter app.bpjs.mobile, country='id', lang='id', sort=Sort.NEWEST dan count=20000. Parameter tersebut bekerja sesuai dengan fungsinya, app.bpjs.mobile merupakan identitas atau objek scraping, lang='id' digunakan untuk memilih bahasa, country='id' digunakan untuk mengambil ulasan pada wilayah negara Indonesia, dan yang parameter sort=Sort.NEWEST digunakan untuk mengurutkan secara *descending* data hasil *scraping*. Selanjutnya, count=20000 digunakan sebagai parameter untuk menentukan batas maksimal ulasan yang akan diambil.

2.2 *Data Preprocessing*

Langkah ini dilakukan dengan tahapan, di antaranya *data cleaning*, *stopword removal, case folding, tokenizing* dan *stemming*. Berikut ini penjelasan dari setiap item tersebut.

a. Data Cleaning

Pada tahapan ini dilakukan penghapusan pada data yang kosong atau *missing value,* data yang duplikat, dan karakter tertentu yang tidak sesuai dengan kriteria analisis. Data ulasan yang didapat bisa jadi terdapat tanda baca yang tidak sesuai, emoji, simbol, angka, link dan kata lainnya yang tidak sesuai, sehingga hal tersebut perlu dibersihkan agar menjadi lebih rapi dan siap diproses untuk proses berikutnya[14]. Dalam klaisifikasi, proses *cleaning* ini memiliki tujuan penting dimana proses ini akan mengurangi hambatan akibat data, informasi atau teks yang tidak membawa makna penting dalam proses klasifikasi [14].

b. Case folding

Proses ini digunakan untuk mengkonversi seluruh huruf pada teks menjadi huruf kecil, tujuannya adalah untuk membantu mengurangi variasi yang tidak diperlukan pada teks oleh model, sehingga proses oleh model yang dibangun menjadi lebih mudah dan teks menjadi lebih seragam [12].

c. Stopword Removal

Kata umum yang tidak mempunyai peran dalam membedakan antarkelas akan dihapus dengan menggunakan proses ini, seperti kata sambung dan kata depan, contohnya "dan", "yang", atau "di". Proses ini digunakan untuk mengurangi gangguan yang disebabkan oleh kata-kata yang sering muncul namun kurang memiliki kontribusi dalam konteks atau makna pada teks [11].

d. Tokenisasi

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

Tokenisasi merupakan proses untuk memecah teks menjadi unit kata atau token tersendiri. Proses ini penting untuk mempersiapkan data teks untuk perhitungan frekuensi kata, pengenalan pola, atau implementasi model algoritma [11]. Dengan proses ini, setiap kata secara individual dapat dianalisis oleh model dan memudahkan identifikasi pola atau makna dalam suatu data.

e. Stemming

Proses ini digunakan untuk konversi kata menjadi bentuk dasar atau aslinya dengan menghapus awalan dan akhiran pada kata [15]. Stemming diimplementasikan agar dapat konversi setiap kata ke bentuk dasarnya. Proses ini membuat variasi kata seperti "berjalan", "berjalanlah", dan "berjalanannya" dapat dibentuk menjadi satu kata dasar, yaitu "jalan", sehingga diharapkan hasil analisis dapat menjadi lebih efisien dan akurat.[16]

2.3 TF-IDF

Tahapan ini dipakai untuk memberi bobot pada kata, proses ini dilakukan untuk dapat mengkonversi data menjadi bentuk angka atau numerik, dengan demikian dapat diproses pemodelan oleh algoritma [17]. TF-IDF atau Term Frequency-Inverse Document Frequency bertujuan untuk mengetahui tingkat kepentingan kata pada suatu data yang dinilai berdasarkan tingkat kemunculannya. Proses ini menggunakan fitur *TfidfVectorizer* dengan menggunakan parameter *max_features, ngram_range,* dan *min_df. max_features*=5000 digunakan untuk membatasi fitur sebanyak 5000 kata dengan bobot yang sesuai, *ngram_range*=(1,1) merupakan fitur *unigram* yang digunakan, dan *min_df*=2 merupakan kata yang muncul minimal pada dua dokumen yang dipertahankan.

Pemberian bobot kata berdasarkan tingkat kemunculan pada data atau dokumen yang disebut *term frequency*. Selain itu, seberapa jarang kata akan muncul dalam data atau dokumen yang disebut *inverse document frequency*. TF-IDF merupakan teknik representasi fitur yang dapat memberikan bobot pada setiap kata berdasarkan dua komponen yaitu *term frequency* dan *inverse document frequency* [15]. Bobot nilai diberikan berdasar pada tingkat frekuensi kata tersebut muncul pada suatu data dengan menekankan kata-kata yang sesuai untuk dianalisis [12].

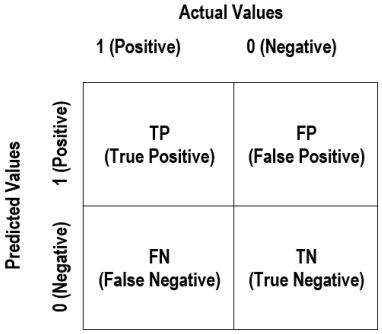
2.4 Pemodelan Naive Bayes

Pemodelan pada algoritma Naive Bayes dilakukan dengan melakukan split data menjadi dua yaitu data pelatihan dan data pengujian menggunakan teknik *train_test_split*. Dimana model ini dilakukan dengan membagi melalui rasio 80:20 yang menghasilkan data training 80% dan data testing 20%. Pemilihan rasio 80:20 umum dilakukan pada kebanyakan praktik umum terhadap penelitian algoritma *machine learning*.

Tujuan dari pemilihan ini ialah memastikan model mendapatkan data pembelajaran agar dapat mengenali pola-pola dengan baik. Akan tetapi penyisaan terhadap data pengujian juga memberikan proporsi yang cuku agar hasil yang didapat tidak bernilai objektif. Setiap model dibuat dan dioptimalkan untuk mempelajari pola sentimen berdasarkan hasil pembobotan kata atau ekstraksi fitur dengan TF-IDF. Setelah data pelatihan dilakukan pemodelan, selanjutnya model diuji dengan data pengujian untuk mengetahui hasil performa klasifikasi algoritma terhadap data baru, dan kemudian dilanjutkan evaluasi dengan metrik akurasi, *recall*, presisi, dan F1-score [18].

2.5 Evaluasi

Proses evaluasi dilakukan untuk menunjukan perbandingan hasil prediksi pada data uji terhadap label asli. Kemudian dibuat visualisasi ke model *confusion matrix,* dengan menampilkan data positif dan negatif atau prediksi benar dan salah pada masing-masing kategori [19]. Pada *confusion matrix* terdapat empat komponen utama yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative.* Gambar berikut ini menjelaskan terkait *confusion matrix*.



Gambar 2. *Confusion Matrix*

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

Teknik ini digunakan untuk mengetahui performa dari algoritma klasifikasi sentimen ulasan pengguna baik kelas positif ataupun negatif. Proses evaluasi menggunakan *confusion matrix* terdiri proses akurasi, *recall*, presisi, f1-score. Pengukuran *confusion matrix* dapat dilihat dengan persamaan sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

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

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$F1-Score = 2\ x\ \frac{\text{precision x recall}}{\text{precision + recall}} \quad (4)$$

Selain itu, digunakan *wordcloud* untuk memvisualisasikan kata kerap muncul dalam data sentimen, baik kelas positif atau negatif. *Word cloud* menampilkan data yang dominan pada ulasan pengguna berdasarkan kelas sentimen, sehingga memudahkan identifikasi kata- kata yang mempunyai frekuensi pada tingkat kemunculan sering digunakan dalam kategori sentimen yang berbeda, atau teknik untuk memahami kata-kata yang kerap muncul pada teks ulasan *dataset* dan istilah yang kerap keluar pada tiap jenis sentimen [11],[3].

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Proses dilakukan dengan teknik *scraping* di web *Google Play Store* menggunakan bahasa Python v3. Library yang digunakan diantaranya Scikit-learn versi 1.6.1, *Google Play Scrapper* versi 1.2.7, dan sastrawi versi 1.0.1. Gambar 3 berikut adalah kode program untuk melakukan *scraping*.

```
# Scraping ulasan aplikasi Google Gemini
result, continuation_token = reviews(
    'com.google.android.apps.bard',
    lang='id',
    country='id',
    sort=Sort.NEWEST,
    count=20000
)
```

Gambar 3. *Scraping* dengan Python

Hasil dari proses scraping didapatkan data 20000 ulasan dalam rentang waktu bulan Juni hingga agustus 2026. Ulasan berfokus pada wilayah dan ulasan berbahasa indonesia. Data tersebut akan tersimpan ke dalam file format csv agar dapat diperoses pada tahapan selanjutnya. Kolom yang digunakan untuk keperluan klasifikasi pada proses selanjutnya yaitu pada kolom rating dan ulasan, sedangkan kolom username dan tanggal tidak digunakan. Hasil pemilihan kolom dapat dilihat pada gambar 4 berikut ini.

Rating	Ulasan
5	baik
4	coba dulu
5	ok
5	Mantap
5	sangat membantu
5	sangat puas
5	Hasil gambarnya selalu bagus dan berkualitas, memuaskan.
5	sangat bagus
5	good, sangat memuaskan
4	bagus dan berguna untuk sehari hari tapi kalau di update bisa makin bagus
4	sangat membantu
5	bagus memudahkan berkreasi
5	keren
5	sangat pintar dan membantu, manta
4	baik
5	good
5	Bagus membantu kita
5	solusi tercepat dan terbaik untuk saat ini
1	kadang jawaban gak jelas di tanya apa di jawab gak nyambung
1	aplikasi yang sangat buruk, jangan download lebih baik download ai seperti dola dll
5	bagus
5	coba

Gambar 4. Hasil *scraping*

3.2 Pemrosesan Data

a. Data Cleaning

Sebelum ke tahap selanjutnya pada pemrosesan data, teks menjalani pembersihan data terlebih dahulu. Pembersihan data dilakukan untuk menghilangkan elemen asing, menghapus data kosong (*missing value*),

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

duplikat data, karakter yang tidak sesuai, dan sentimen pengguna yang tidak memenuhi kriteria untuk analisis. Gambar berikut memperlihatkan coding python untuk pembersihan data.

Gambar 5. *Coding data cleaning*

Ulasan yang didapat pada dari google play store ada kemungkinan berisi tanda baca, simbol, angka, link dan karakter lainnya, sehingga perlu dibersihkan agar tidak menimbulkan ambigu data dan mempengaruhi kualitas analisis. Pada coding diatas, pembersihan data dilakukan dengan membersihkan URL, mention, hastag, emoji, angka, tanda baca, spasi berlebih dan karakter lainnya selain huruf dan spasi menggunakan *library def cleaning_text(text)* python. Setelah pembersihan didapatkan data sebanyak 11,035 dari 20.000 data ulasan.

b. Case Folding

Proses ini dipakai untuk merubah teks dari semua huruf kapital menjadi huruf kecil, sehingga ulasan menjadi lebih konsisten. Pada gambar 6 berikut menunjukan hasil proses tahapan *case folding*.

Rating	Ulasan	Ulasan_Cleaning
5	baik	baik
4	coba dulu	coba dulu
5	ok	ok
5	Mantap	mantap
5	sangat membantu	sangat membantu
5	sangat puas	sangat puas
5	Hasil gambarnya selalu bagus dan berkualitas, mem	hasil gambarnya selalu bagus dan berkualitas memuaskan
5	good, sangat memuaskan	good sangat memuaskan
4	bagus dan berguna untuk sehari hari tapi kalau di upc	bagus dan berguna untuk sehari hari tapi kalau di update bisa makin bagus
5	bagus memudahkan berkreasi	bagus memudahkan berkreasi
5	keren	keren
5	sangat pintar dan membantu, manta	sangat pintar dan membantu manta
5	good	good
5	Bagus membantu kita	bagus membantu kita
5	solusi tercepat dan terbaik untuk saat ini	solusi tercepat dan terbaik untuk saat ini
1	kadang jawaban gak jelas di tanya apa di jawab gak n	kadang jawaban gak jelas di tanya apa di jawab gak nyambung

Gambar 6. Hasil proses *case folding*

c. Stopword Removal

Proses ini digunakan untuk mengolah bahasa alami yang berada pada kata – kata yang akan dianalisis. Kata tersebut dianggap tidak mempunyai makna penting dalam konteks data. Teks tersebut ada kalanya diabaikan pada kasus pemrosesan data, karena teks tersebut dinilai tidak memiliki informasi yang cukup signifikan sebagai tujuan analisis.

1	mantap	mantap
5	mantulll	mantulll
5	ok banget	banget
4	good lah	good lah
5	baik	baik
4	coba dulu	coba dulu
5	mantap	mantap
5	sangat membantu	sangat membantu
5	sangat puas	sangat puas
5	hasil gambarnya selalu bagus dan berkualitas memuaskan	hasil gambarnya selalu bagus berkualitas, memuaskan.
5	good sangat memuaskan	good, sangat memuaskan
4	bagus dan berguna untuk sehari hari tapi kalau di update bisa makin bagus	bagus berguna sehari hari kalau update makin bagus
5	bagus memudahkan berkreasi	bagus memudahkan berkreasi
5	keren	keren
5	sangat pintar dan membantu manta	sangat pintar membantu, manta

Gambar 7. Hasil *stopword removal*

d. Tokenisasi

Proses ini digunakan untuk membentuk teks atau kalimat menjadi kata-kata yang lebih kecil. Hasil pada proses tokenisasi bisa dilihat pada gambar 8 berikut.

mantap	['mantap']
sangat membantu	['sangat', 'membantu']
sangat puas	['sangat', 'puas']
hasil gambarnya selalu bagus dan berkualitas memuaskan	['hasil', 'gambarnya', 'selalu', 'bagus', 'berkualitas,', 'memuaskan.']
good sangat memuaskan	['good,', 'sangat', 'memuaskan']
bagus dan berguna untuk sehari hari tapi kalau di update bisa makin bagus	['bagus', 'berguna', 'sehari', 'hari', 'kalau', 'update', 'makin', 'bagus']
bagus memudahkan berkreasi	['bagus', 'memudahkan', 'berkreasi']
keren	['keren']
sangat pintar dan membantu manta	['sangat', 'pintar', 'membantu,', 'manta']
good	['good']
bagus membantu kita	['bagus', 'membantu']
solusi tercepat dan terbaik untuk saat ini	['solusi', 'tercepat', 'terbaik']
kadang jawaban gak jelas di tanya apa di jawab gak nyambung	['kadang', 'jawaban', 'gak', 'jelas', 'tanya', 'apa', 'jawab', 'gak', 'nyambung']
aplikasi yang sangat buruk jangan download lebih baik download ai seperti dola dll	['aplikasi', 'sangat', 'buruk,', 'jangan', 'download', 'lebih', 'baik', 'download', 'ai', 'dola']

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

Gambar 8. Hasil Tokenisasi

e. Stemming

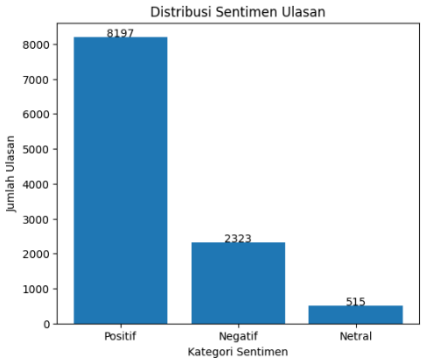
Dalam pengolahan bahasa alami stemming menjadi salah satu cara yang digunakan untuk menyederhanakan teks menjadi bentuk kata dasarnya. Proses ini melibatkan *library Sastrawi* pada python agar dapat merubah kata – kata menjadi bentuk dasarnya. Proses ini akan menghapus awalan dan akhiran dari kata dasar pada teks. Pada gambar 9 berikut menunjukan hasil *stemming*.

bagusss	bagusss
bagus banget	bagus banget
mantap gratis sesukamu	mantap gratis suka
sangat baik	sangat baik
sangat bagus	sangat bagus
mantap	mantap
mantullll	mantullll
ok banget	banget
good lah	good lah
baik	baik
coba dulu	coba dulu
mantap	mantap
sangat membantu	sangat bantu
sangat puas	sangat puas
hasil gambarnya selalu bagus dan berkualitas memuaskan	hasil gambar selalu bagus kualitas muas
good sangat memuaskan	good sangat muas

Gambar 9. Hasil *stemming*

3.3 Pelabelan data

Labelling atau pelabelan data dibagai menjadi 3 kelas berdasarkan rating pada dataset. Kelas tersebut meliputi positif, negatif, dan netral. Label yang digunakan hanya pada kelas positif dan negatif, sedanngkan kelas netral tidak digunakan. Hal ini dikarenakan agar tingkat akurasi yang dihasilkan menjadi lebih baik, dikarenakan tidak terbawanya data yang membuat ambigu tersebut. Hasil distribusi sentimen ditunjukan pada gambar 10 berikut ini.



Gambar 10. Distribusi pelabelan data

Dari gambar tersebut ditunjukan bahwa terdapat total data 11,035, dengan jumlah 81,97 sebagai kelas positif, 23,23 sebagai kelas negatif dan sisanya kelas netral yaitu 515.

3.4 TF-IDF

Ekstraksi fitur menggunakan TF-IDF. *Library fit_transform()* digunakan untuk membantu proses TF-IDF pada data latih. Pada proses pelatihan, yang digunakan hanya pada proses fit saja. Proses pelatihan model dikonfigurasi dengan parameter *ngram_range, max_features, dan min_df*. Sedangkan data uji menggunakan fungsi transform() yang diperoleh dari data latih. Setiap kata akan memiliki bobot angka setelah dilakukan proses extraksi fitur TF-IDF. Pada semua ulasan, bobot setiap *term* akan dihitung berdasarkan kata-kata yang muncul pada ulasan. *Library TfidfVectorizer* pada pemrograman python digunakan pada penelitian ini. Bagan berikut ini merupakan potongan kode program *TfidfVectorizer*.

```
tfidf = TfidfVectorizer(
    max_features=5000,
    ngram_range=(1,1),
    min_df=2
)
```

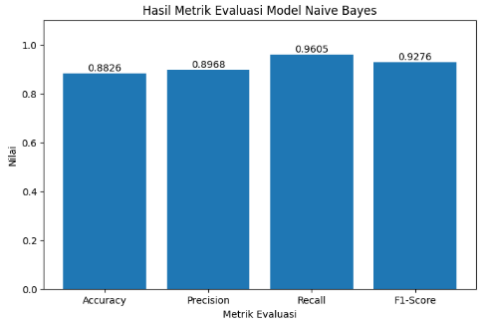
Gambar 11. TF-IDF

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

Setelah proses konversi teks menjadi format numerik, memungkinkan model *machine learning* dapat berfungsi secara efektif. Gambar diatas menunjukan penggunaan fungsi atau parameter dari *ngram_range, max_feature, dan min_df*. Parameter *ngram_range* digunakan untuk pembentukan n-gram yang pada kasus ini diisi dengan nilai (1,1). Parameter *max_features* digunakan untuk membatasi jumlah fitur maksimum pada saat proses klasifikasi dengan nilai diisi "5000". Selanjutnya, parameter *min_df* digunakan untuk mengabaikan kata yang muncul hanya pada satu dokumen, sehingga dapat mengurangi fitur yang sangat jarang, nilai "2" diberikan pada parameter ini.

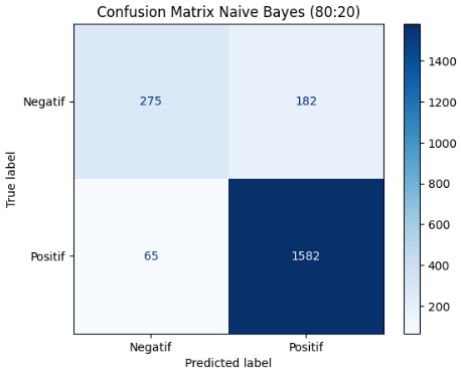
3.5 Pemodelan Naive Bayes

Pemodelan pada data latih menggunakan metode train_split_test, yang artinya membagi data yaitu data pelatihan dan pengujian. Rasio perbandingan menggunakan rasio 80:20 yang berarti 80% data pelatihan dan 20% data pengujian. Penelitian hanya menggunakan kelas positif dan negatif dengan jumlah 11,035 data. Untuk itu pada training akan menghasilkan data 8,828 dari perhitungan 11,035 x 80%, sedangkan untuk data testing akan menghasilkan data 2,207 dari perhitungan 11,035 x 20%. Selanjutnya, hasil pemodelan pada data training di evaluasi dan disajikan ke dalam bentuk *confusion matrix* dengan presisi, akurasi, *recall* dan f1-*score* berdasarkan kinerja algoritma. Gambar 12 berikut ini ditunjukan hasil metrik evaluasi model Naive Bayes.



Gambar 12. Evaluasi model

Pada gambar tersebut menunjukkan data akurasi sebesar 0,8826, presisi 0,8968, recall 0,9605 dan F1-score 0,9276. Pada nilai akurasi tersebut menunjukan prediksi cukup baik. Nilai *precision* yang dihasilkan sebesar 0,8968, angka tersebut menunjukkan bahwa prediksi menghasilkan prediksi sebagian besar benar. Pada hasil *recall* didapat data sebesar 0,9605, yang menunjukkan bahwa model sebagian besar mampu mengidentifikasi data yang benar dari kelas positif. Selain itu, F1-score sebanyak 0,9276 menunjukkan adanya keseimbangan baik antara *precision* dan *recall*. Selanjutnya, hasil data testing yaitu 2.207 data, disajikan ke dalam bentuk *confusion matrix*. Gambar 13 berikut confusion matrix pada data uji.



Gambar 13. *Confusion matrix* Naive Bayes

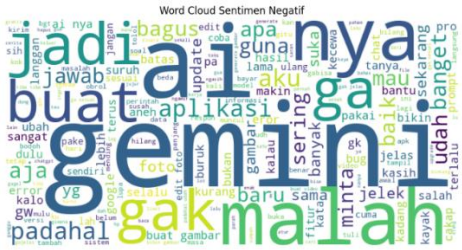
Gambar tersebut menunjukan bahwa true negative, label negatif mempunyai prediksi benar sebanyak 182 data sebagai label negatif. Pada *true positive*, label positif mempunyai prediksi benar 1.582 pada label positif . pada *false negative*, label positif mempunyai prediksi salah sebanyak 182 data sebagai label negatif. Sedangkan pada *false positive*, label negatif mempunyai prediksi salah 65 data sebagai label positif. Dari penjelasan tersebut, secara umum model sudah mampu mengklasifikasikan data uji dengan baik.

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

3.6 *Word Cloud*

Word cloud digunakan dalam penelitian ini untuk menampilkan atau visualisasi sentimen positif dan negatif kata yang dominan atau kata yang kerap muncul pada data ulasan. Gambar berikut ini adalah visualisasi dari *word cloud* ulasan positif dan negatif.

Gambar 14. *Word Cloud Positive*



Gambar 15. *Word Cloud Negative*

Dari gambar 14 dan 15 ditunjukan bahwa kata yang kerap muncul pada ulasan positif seperti kata bagus, baik, sangat, bantu. Selain itu *word cloud* negatif dapat dilihat pada ulasan negatif, kata tersebut diantaranya gak, malah, gak, banget, padahal.

4. KESIMPULAN

Hasil pada penelitian ini secara keseluruhan model menunjukan performa yang cukup baik. Metode penelitian menggunakan algoritma Naive Bayes dengan model pembagian data *test_train_split* berbasis python. Tahapan yang dilakukan diantaranya pengumpulan data dengan teknik scraping menggunakan python dan berhasil dikumpulkan *dataset* sekitar 20.000 ulasan. Parameter *scraping* menggunakan parameter app.bpjs.mobile, country='id', lang='id', sort=Sort.NEWEST dan count=20000. Selanjutnya tahapan pemrosesan data meliputi *data cleaning*, *stopword removal*, *case folding, stemming* dan tokenisasi. Pemberian label dibagi jadi 3 kategori yaitu positif, negatif dan netral, akan tetapi kategori netral tidak digunakan agar pemodelan menjadi lebih baik tanpa adanya ambigu data. Hasil pelabelan didapat data positif 8.197, negatif 2.323, dan netral 515. Pemberian bobot pada kata menggunakan TF-IDF dan pemodelan Naive Bayes. Pemodelan menggunakan fitur *TfidfVectorizer* dengan parameter *max_features*, *ngram_range*, dan *min_df*. *max_features*=10000 digunakan untuk membatasi fitur sebanyak 10000 kata dengan bobot yang sesuai, ngram_range=(1,1). Rasio pembagian pada model dilakukan dengan data pelatihan dan pengujian yaitu rasio 80:20, yang berarti data pelatihan sebanyak 80% dan sisnaya data pengujian yaitu 20%. Selanjutnya proses evaluasi dan pengukuran menggunakan *confusion matrix* dan visualisasi *word cloud* dilakukan. Hasil evaluasi model didapat nilai *accuracy* 0.8826 atau 88,26%, *precision* 0.8968 atau 89,68%, *recall* 0.9605 atau 96,05% dan F1-*score* 0.9276 atau 92,76%. *Word cloud* menampilkan kata dominan kerap muncul pada data sentimen positif maupun negatif. Tujuan penelitian yaitu analisis sentimen terhadap ulasan *user* Google Gemini AI pada *web* Google Play Store dengan pemodelan Naive Bayes. Harapan dari hasil penelitian yaitu dapat memberi wawasan lebih yang mendalam terkait pola sentimen atau ulasan pengguna. Serta memberikan rekomendasi bagi pembuat aplikasi dan sistem pelayanannya agar dapat meningkatkan layanan.

REFERENCES

[1] J. Jin, "Harnessing the Power of Large Language Models for Real- World Sentiment Classification on Social Media," *Int. Conf. Bus. Policy Stud.*, vol. 156, pp. 214–224, 2025, doi: 10.54254/2754-1169/156/2025.20662.

[2] M. D. Alifayoezra, A. Ibrahim, Y. Utama, E. L. Ruskan, and D. R. Indah, "Comparative study of machine learning models for classifying sentiment in google gemini app reviews," *RABIT J. Teknol. dan Sist. Inf.*

JURNAL KOMPUTER TEKNOLOGI INFORMASI SISTEM KOMPUTER
p-ISSN: 2962-3022 | e-ISSN: 2963-7104
Volume 5 No 2 September 2026
Hal: 2210-2218
http://ejurnal.lkpkaryaprima.id/index.php/juktisi
DOI: https://doi.org/10.62712/juktisi.v5i2.1531

Univrab, vol. 11, no. 1, pp. 988–1000, 2026, doi: 10.36341/rabit.v11i1.7201.

[3] M. Z. Mubarok, S. Rizqi, R. M. Rizal, A. B. N. Syalim, and N. Iksan, "Klasifikasi Sentimen Ulasan Aplikasi Claude Menggunakan Naive Bayes dan Support Vector Machine," *JUSIFOR J. Sist. Inf. dan Inform.*, vol. 4, no. 1, pp. 110–119, 2025, doi: 10.70609/jusifor.v4i1.7054.

[4] S. Yanah, W. Purbaratri, S. Paylina, A. N. I. Safitri, and N. K. Tachjar, "Analisis sentimen aplikasi pemilu menggunakan algoritma Naive Bayes," *J. ELEKTRO Inform. SWADHARMA*, vol. 05, no. 01, pp. 131–139, 2025, doi: 10.56486/jeis.vol5no1.684.

[5] Ainunna'imah, I. Riadi, and H. Yuliansyah, "Model Klasifikasi Kesesuaian Artikel Pada Jurnal SINTA Berdasarkan Metadata Menggunakan Term Frequency–Inverse Document Frequency dan Naïve Bayes," *semanTIK*, vol. 11, no. 1, pp. 1–12, 2026, doi: 10.55679/semantik.v12i1.270.

[6] B. Rahmatullah, S. A. Saputra, P. Budiono, and D. P. Wigandi, "Sentimen Analisis Makan Bergizi Gratis Menggunakan Algoritma Naive Bayes," *JIFOTECH (JOURNAL Inf. Technol.*, vol. 05, no. 01, pp. 259–264, 2025, doi: 10.46229/jifotech.v5i1.978.

[7] M. Muzaki, R. Kurniawan, I. Ali, Mulyawan, and S. Anwar, "Analisis sentimen ulasan pengguna aplikasi chatgpt menggunakan algoritma support vector machine dan Naive Bayes berbasis Python," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 10, no. 2, pp. 3087–3094, 2026, doi: 10.36040/jati.v10i2.17833.

[8] R. Nurhasanah and V. A. Elyakim, "Analisis Sentimen Ulasan Pengguna Generative AI Menggunakan Naïve Bayes Berbasis Python," *INSOLOGI J. Sains dan Teknol.*, vol. 5, no. 3, pp. 875–890, 2026, doi: 10.55123/insologi.v5i3.8050.

[9] Fathoni, A. F. Ansori, I. N. Ramadhani, C. R. Anissa, and S. A. Putri, "Analisis sentimen masyarakat indonesia di Twitter terhadap sistem perpajakan Coretax menggunakan metode Naïve Bayes," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 4, pp. 6749–6753, 2025, doi: 10.36040/jati.v9i4.14214.

[10] F. Nurpandi, F. S. Sulaeman, and A. Hermawan, "Analisis Sentimen Terhadap Kinerja Kepolisian Indonesia Menggunakan Metode Multinomial Naive Bayes, Long Short-Term Memory, dan Lexicon-Based," *Media J. Inform.*, vol. 16, no. 1, pp. 1–10, 2024, doi: 10.35194/mji.v16i1.4165.

[11] Chulyatunni'mah, R. Kurniawan, and S. Anwar, "Analisis Sentimen Penggemar Treasure Di Karnaval Mandiri Menggunakan Naïve Bayes," *J. Sist. Inf. TGD*, vol. 4, no. 1, pp. 203–213, 2025, doi: 10.53513/jursi.v4i1.10606.

[12] M. R. Maulana, A. Y. Imran, and Fathoni, "Analisis sentimen ulasan Deepseek AI di Google Playstore menggunakan Naïve Bayes," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 4, pp. 6473–6478, 2025, doi: 10.36040/jati.v9i4.14070.

[13] R. A. Sitorus and I. Zufria, "Application of the Naïve Bayes Algorithm in Sentiment Analysis of Using the Shopee Application on the Play Store," *J. Teknol. Inf. dan Komun.*, vol. 15, no. 01, pp. 53–66, 2024, doi: 10.31849/digitalzone.v15i1.19828.

[14] Y. H. Agustin, N. C. Mulyani, and W. S. Prasetya, "Analisis Sentimen Opini Publik Menggunakan Algoritma Naive Bayes dan TF-IDF," *J. Algoritm.*, vol. 22, no. 2, pp. 1373–1384, 2025, doi: 10.33364/algoritma/v.22-2.2671.

[15] Ainunna'imah, H. Yuliansyah, and I. Riadi, "Subject Area Classification of Journal Articles Based on Metadata Using Bag of Words and Naïve Bayes," *Eng. Sci. Lett.*, vol. 5, no. 02, pp. 72–80, 2026, doi: 10.56741/IISTR.esl.002041.

[16] T. R. Ariantoro, A. Setiadi, Cecilia, Fadhliazis, and N. S. Handini, "Klasifikasi Emosi Publik di Instagram Terhadap Uji Coba Jalan Satu Arah di Palembang dengan Naive Bayes," *J. Algoritm.*, vol. 6, no. 1, 2025, doi: 10.35957/algoritme.v5i3.11498.

[17] H. P. Jelita, M. I. Saad, and Wahyuni, "Penerapan algoritma Naive Bayes dalam analisis sentimen masyarakat terhadap STMIK Widya Cipta Dharma," *Bull. Inf. Technol.*, vol. 6, no. 2, pp. 148–160, 2025, doi: 10.47065/bit.v5i2.2029.

[18] S. E. Lourensia, N. M. S. Iswari, and E. M. Dharma, "Analysis of Student Sentiments towards the Use of Artificial Intelligence in Education Using the Naïve Bayes Algorithm," *J. Teknlogi Inform. dan Komput. MH. Thamrin*, vol. 12, no. 1, pp. 944–958, 2026, doi: 10.37012/jtik.v12i1.3606.

[19] A. T. Bisono and A. Zulherry, "Analisis Sentimen Game Genshin Impact untuk Mengetahui Reaksi dan Harapan Pemain Menggunakan Metode Naïve Bayes," *J. Tek. Inform.*, vol. 4, no. 2, pp. 184–193, 2025, doi: 10.56211/sudo.v4i2.1131.