Jurnal SimanteC
Vol. 14, No. 1 Desember 2025
P-ISSN : 2088-2130
E-ISSN : 2502-4884

Penerapan analisis sentimen opini masyarakat terhadap ulasan aplikasi *Grab* pada *google play store* menggunakan Algoritma *Naive Bayes*

Application of analysis sentiment of public opinion regarding Grab app reviews on google play store using the Naive Bayes Algorithm

[1]Bella Audrey Zadia, [2] Siti Lailiyah, [3]Tommy Bustomi

[1,2,3]Teknik Informatika, STMIK Widya Cipta Dharma,
Jl. M. Yamin No.25, Gn. Kelua, Kec. Samarinda Ulu, Kota Samarinda,
Kalimantan Timur 75123, Indonesia
e-mail: 2143024@wicida.ac.id

Abstrak

Penelitian ini bertujuan untuk menganalisis sentimen opini masyarakat terhadap ulasan aplikasi Grab di Google Play Store menggunakan algoritma Multinomial Naive Bayes. Proses penelitian mengikuti tahapan KDD, mulai dari seleksi data, pre-processing, pelabelan sentimen, pembagian dataset, pelatihan model, hingga evaluasi kinerja. Sebanyak 5.000 data ulasan dikumpulkan melalui teknik web scraping dan diproses secara bertahap menggunakan metode cleaning, normalisasi, tokenizing, stopword removal, dan stemming. Model Multinomial Naive Bayes yang dibangun mampu mengklasifikasikan sentimen ulasan ke dalam kategori positif, negatif, dan netral dengan akurasi sebesar 69,6%. Precision dan recall untuk sentimen negatif dan positif tergolong tinggi, namun performa pada sentimen netral masih rendah akibat distribusi data yang tidak seimbang. Hasil penelitian ini memberikan insight penting bagi pengembang Grab dalam memahami persepsi pengguna dan dapat dijadikan dasar pengambilan keputusan strategis untuk peningkatan kualitas layanan aplikasi. Dengan demikian, Multinomial Naive Bayes terbukti efektif dan efisien untuk analisis sentimen ulasan aplikasi Grab di Google Play Store.

Kata Kunci: *Analisis Sentimen, Multinomial Naive Bayes, Grab, Google Play Store, KDD*

Abstract

This study aims to analyze public sentiment regarding reviews of the Grab application on the Google Play Store using the Multinomial Naive Bayes algorithm. The research process follows the KDD stages, starting from data selection, pre-processing, sentiment labeling, dataset splitting, model training, to performance evaluation. A total of 5,000 review data were collected through web scraping techniques and processed step by step using cleaning, normalization, tokenizing, stopword removal, and stemming methods. The developed Multinomial Naive Bayes model was able to classify review sentiments into positive, negative, and neutral categories with an accuracy of 69.6%. Precision and recall for negative and positive sentiments are relatively high, but the performance for neutral sentiment remains low due to imbalanced data distribution. The results of this study provide valuable insights for Grab developers to understand user perceptions and can serve as a basis for strategic decision-making to improve application service quality. Thus, Multinomial Naive Bayes has proven to be effective and efficient for sentiment analysis of Grab app reviews on the Google Play Store.

Keywords*: Sentiment Analysis, Multinomial Naive Bayes, Grab, Google Play Store, KDD*

1 PENDAHULUAN

Perkembangan teknologi digital yang pesat telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam sektor transportasi. Aplikasi berbasis daring seperti *Grab* telah menjadi solusi utama dalam memenuhi kebutuhan mobilitas masyarakat di Indonesia

dan Asia Tenggara [1],[2]. *Grab* tidak hanya menyediakan layanan transportasi, tetapi juga layanan pengiriman makanan, pembayaran digital, dan berbagai layanan lainnya yang terintegrasi dalam satu *platform*. Dengan semakin banyaknya pengguna, kualitas layanan *Grab* menjadi fokus utama agar dapat mempertahankan dan meningkatkan kepuasan pelanggan [1].

Ulasan pengguna di *Google Play Store* merupakan sumber data penting yang mencerminkan opini dan pengalaman nyata masyarakat terhadap aplikasi *Grab* [3],[4],[5]. Data ulasan ini mengandung informasi berharga yang dapat digunakan untuk mengidentifikasi kelebihan, kekurangan, serta aspek layanan yang perlu diperbaiki. Namun, jumlah ulasan yang sangat besar dan beragam menjadikan analisis manual tidak efektif dan efisien. Oleh karena itu, diperlukan metode komputasi yang mampu mengolah dan mengklasifikasikan opini pengguna secara otomatis [6], [7].

Analisis sentimen merupakan pendekatan yang tepat untuk mengkategorikan opini masyarakat ke dalam sentimen positif, negatif, atau netral berdasarkan teks ulasan. Metode ini telah banyak digunakan dalam berbagai penelitian untuk memahami persepsi publik terhadap produk atau layanan [2],[8],[9]. Dalam konteks aplikasi *Grab*, analisis sentimen dapat membantu perusahaan dalam mengambil keputusan strategis untuk meningkatkan kualitas layanan berdasarkan umpan balik pengguna [10].

Namun, analisis sentimen terhadap ulasan aplikasi menghadapi beberapa tantangan, seperti penggunaan bahasa informal, singkatan, dan variasi ekspresi yang kompleks [4]. Penelitian oleh Bustomi dan Fahmi menegaskan pentingnya proses *preprocessing* data yang tepat, seperti tokenisasi dan penghilangan *stopwords*, untuk meningkatkan akurasi klasifikasi sentimen [1]. Selain itu, ketidakseimbangan data antara ulasan positif dan negatif juga menjadi faktor yang perlu diperhatikan dalam pemilihan algoritma [11],[5].

Kebaruan/*novelty* penelitian ini adalah penerapan algoritma *Multinomial Naive Bayes* untuk menganalisis sentimen ulasan aplikasi *Grab* di *Google Play Store* yang lebih jarang dijumpai dalam penelitian sebelumnya, terutama pada *platform Google Play Store*, yang belum banyak dibahas dalam literatur yang ada. Penelitian ini juga memberikan pemahaman lebih dalam mengenai tantangan dalam mengklasifikasikan sentimen netral yang cenderung lebih ambigu.

Algoritma *Naive Bayes* dipilih dalam penelitian ini karena kemampuannya dalam menangani data teks berdimensi tinggi dengan kecepatan komputasi yang relatif cepat dan akurasi yang cukup baik [12],[7]. Beberapa studi sebelumnya, termasuk penelitian yang dilakukan oleh Bustomi, menunjukkan bahwa *Naive Bayes* efektif dalam klasifikasi sentimen opini publik pada berbagai domain, termasuk aplikasi dan media sosial [13]. Dengan menggunakan algoritma ini, diharapkan proses klasifikasi ulasan *Grab* dapat berjalan optimal [14].

Meskipun telah banyak penelitian terkait analisis sentimen, sebagian besar fokus pada *platform* media sosial seperti Twitter atau Facebook, sementara ulasan aplikasi di *Google Play Store* kurang mendapat perhatian khusus [15],[16],[17]. Penelitian ini berusaha mengisi kekosongan tersebut dengan mengaplikasikan algoritma *Naive Bayes* pada data ulasan *Grab* di *Google Play Store*, sehingga dapat memberikan gambaran yang lebih spesifik dan relevan mengenai opini masyarakat terhadap aplikasi ini [18],[19],[20].

Tujuan dari penelitian ini adalah untuk menganalisis sentimen opini masyarakat terhadap aplikasi *Grab* berdasarkan ulasan di *Google Play Store* menggunakan algoritma *Naive Bayes*. Penelitian ini dapat memberikan kontribusi dalam pengembangan metode analisis sentimen serta memberikan rekomendasi perbaikan layanan bagi pengembang *Grab*. Dengan demikian, penelitian ini memiliki manfaat praktis dan akademis dalam bidang sistem informasi dan analisis data.

2 TINJAUAN PUSTAKA

2.1. Algoritma *Naive Bayes*

Naive Bayes adalah metode klasifikasi probabilistik berbasis teorema *Bayes* dengan asumsi independensi antar fitur. Rumus utamanya didefinisikan sebagai:

$$P(c|d) = \frac{P(d|c) \cdot P(c)}{P(d)} \quad (1)$$

Pada rumus (1), P(c|d) adalah probabilitas dokumen d termasuk dalam kelas sentimen c, yang dihitung menggunakan probabilitas prior kelas P(c) dan probabilitas kemunculan kata-kata dalam dokumen pada kelas tersebut P(d|c) [1], [2]. Untuk menghitung probabilitas kemunculan kata-kata pada kelas c, digunakan rumus:

$$P(d|c) = \prod_{i=1}^{n} P(w_i|c) \quad (2)$$

Pada rumus (2), $P(wi|c)$ adalah probabilitas kemunculan kata wi pada kelas c, yang dihitung dengan menggunakan data pelatihan yang telah dilabeli sentimen [3]. Untuk menghindari probabilitas nol, digunakan Laplace smoothing, yang didefinisikan oleh rumus berikut:

$$P(w_i|c) = \frac{\text{freq}(w_i,c)+1}{\sum_w(\text{freq}(w,c)+1)} \quad (3)$$

Pada rumus (3), $count(wi,c)$ adalah jumlah kemunculan kata wi pada kelas c, dan $|V|$ adalah jumlah kata unik dalam *vocabulary* yang digunakan dalam model [4].

2.2. Pra-Proses Data

Tahap ini meliputi berbagai proses untuk membersihkan dan menyiapkan data sebelum digunakan dalam analisis sentimen. Proses pertama adalah tokenisasi, yang memecah teks menjadi unit kata seperti berikut:

Tokenisasi: “aplikasi bagus”→[“$aplikasi$”,”$bagus$”]

Pada tahap tokenisasi, teks yang telah dibagi menjadi kata-kata atau token yang terpisah [5]. Selanjutnya, tahap *stopword removal* digunakan untuk menghilangkan kata yang tidak memberikan informasi penting dalam analisis, seperti kata sambung atau preposisi [6]. Rumus untuk *stopword removal* adalah:

Stopword Removal:”yang”, ”dan”, ”di”→Dihapus

Pada *stopword removal*, kata-kata yang sering muncul namun tidak memiliki makna penting untuk analisis sentimen dihilangkan dari dataset [7]. Setelah itu, proses *stemming* dilakukan untuk mengubah kata berimbuhan menjadi bentuk dasarnya, seperti pada rumus berikut:

Stemming:”dipesan”→”pesan”

Pada *stemming*, kata yang berimbuhan diubah menjadi bentuk dasar untuk mengurangi variasi kata yang sama [8].

2.3. Pembobotan TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) digunakan untuk menilai pentingnya kata dalam dokumen. Rumus TF (*Term Frequency*) didefinisikan sebagai berikut:

$$\text{TF}(t,d) = \frac{\text{freq}(t,d)}{\text{total kata dalam } d} \quad (4)$$

Pada rumus (4), TF(t) mengukur seberapa sering kata t muncul dalam suatu dokumen, yang berfungsi untuk menilai relevansi kata dalam dokumen tersebut [9]. Kemudian, IDF (*Inverse Document Frequency*) dihitung dengan rumus berikut:

$$\text{IDF}(t) = \log\left(\frac{N}{\text{DF}(t)}\right) \quad (5)$$

Pada rumus (5), N adalah jumlah total dokumen, dan DF(t) adalah jumlah dokumen yang mengandung kata t. IDF mengurangi bobot kata-kata yang sering muncul dalam banyak dokumen dan meningkatkan bobot kata-kata yang jarang muncul [10]. Pembobotan TF-IDF akhirnya dihitung dengan mengalikan nilai TF dan IDF untuk setiap kata dalam dokumen:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \cdot \text{IDF}(t) \quad (6)$$

Pada rumus (6), TF−IDF(t) memberikan bobot untuk setiap kata berdasarkan seberapa penting kata tersebut dalam dokumen tertentu dibandingkan dengan dokumen lainnya [11].

2.4. Evaluasi Kinerja Model

Kinerja model dievaluasi dengan menggunakan beberapa metrik, antara lain akurasi, *precision*, *recall*, dan F1-*score*. Akurasi dihitung dengan rumus:

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

Pada rumus (7), akurasi mengukur seberapa banyak prediksi yang benar dibandingkan dengan jumlah keseluruhan *data testing* [12]. *Precision* diukur dengan rumus berikut:

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (8)$$

Pada rumus (8), TP adalah *true positive*, yaitu jumlah prediksi benar untuk kelas positif, dan FP adalah *false positive*, yaitu jumlah prediksi salah untuk kelas positif [13]. *Recall* dihitung dengan rumus:

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

Pada rumus (9), FN adalah *false negative*, yaitu jumlah prediksi salah untuk kelas negatif [14]. F1-*score* adalah *harmonic mean* dari *precision* dan *recall*, yang dihitung dengan rumus:

$$\text{F1} = 2 \cdot \frac{\text{Presisi} \cdot \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (10)$$

Pada rumus (10), F1-*score* memberikan gambaran keseluruhan kinerja model dalam hal keseimbangan antara *precision* dan *recall* [15].

3 METODE PENELITIAN

Penelitian ini mengadopsi metodologi *Knowledge Discovery in Database* (KDD), yang merupakan salah satu pendekatan umum dalam analisis data besar. KDD terdiri dari beberapa tahapan yang saling terkait untuk mengolah dan menemukan pola dalam data. Proses ini diikuti dalam penelitian untuk menganalisis sentimen ulasan aplikasi *Grab* di *Google Play Store* menggunakan algoritma *Naive Bayes*.

3.1. Seleksi Data

Tahap pertama dalam proses KDD adalah seleksi data, yang bertujuan untuk mengumpulkan data ulasan aplikasi *Grab* yang relevan. Data yang digunakan dalam penelitian ini diperoleh melalui teknik *web scraping* menggunakan *library* Python *google-play-scraper*. Data yang dikumpulkan terdiri dari 5.000 ulasan aplikasi *Grab* yang mencakup atribut seperti *reviewId*, *content*, *score*, dan *timestamp*. Kriteria seleksi data mencakup:

1) Ulasan yang ditulis dalam bahasa Indonesia dan Inggris.
2) Ulasan yang berasal dari periode 1 Januari 2020 hingga 31 Desember 2025.
3) Data lengkap yang mencakup informasi terkait *ReviewId*, konten ulasan, skor, dan *timestamp*.

3.2. *Pre-Processing* Data

Tahap ini adalah inti dari pembersihan data yang meliputi beberapa sub-proses [8],[10], sebagai berikut:

1) *Cleaning*: Menghapus karakter khusus, simbol, emoji, dan elemen lain yang tidak relevan untuk analisis sentimen.
2) *Case Folding*: Mengonversi seluruh teks menjadi huruf kecil untuk menghindari duplikasi kata.
3) Normalisasi Kata: Mengubah kata slang atau singkatan menjadi bentuk baku, seperti mengubah kata “gabut” menjadi “tidak puas”.
4) *Tokenizing*: Memecah kalimat atau paragraf ulasan menjadi unit kata tunggal.
5) *Stopword Removal/Filtering*: Menghapus kata-kata umum yang tidak memiliki makna penting, seperti “yang”, “dan”, “di”.
6) *Stemming*: Mengubah kata berimbuhan menjadi bentuk dasar, seperti mengubah “dipesan” menjadi “pesan”.

3.3. Pelabelan Data (*Lexicon Based*)

Setelah data dibersihkan, tahap selanjutnya adalah pelabelan sentimen. Ulasan diberi label sentimen menggunakan pendekatan *lexicon-based*. Klasifikasi dilakukan ke dalam tiga kategori:

1) Positif: Ulasan yang mengandung kata atau ekspresi yang menunjukkan kepuasan atau pengalaman baik terhadap aplikasi.
2) Negatif: Ulasan yang mengandung kata atau ekspresi yang menunjukkan kekecewaan atau pengalaman buruk.
3) Netral: Ulasan yang tidak secara jelas menunjukkan sentimen positif atau negatif.

Tahap ini menggunakan gabungan antara skor rating pengguna dan analisis kata kunci untuk memastikan akurasi dalam pelabelan sentimen [16].

3.4. *Splitting Dataset*

Dataset yang telah diberi label kemudian dibagi menjadi dua bagian, yaitu:

1) *Data Training* (80%): Digunakan untuk melatih model algoritma Naive Bayes.
2) *Data Testing* (20%): Digunakan untuk menguji performa model yang telah dilatih. Pembagian data dilakukan secara acak untuk memastikan representasi yang adil dari setiap kelas sentimen pada kedua subset data.

3.5. Algoritma *Naive Bayes*

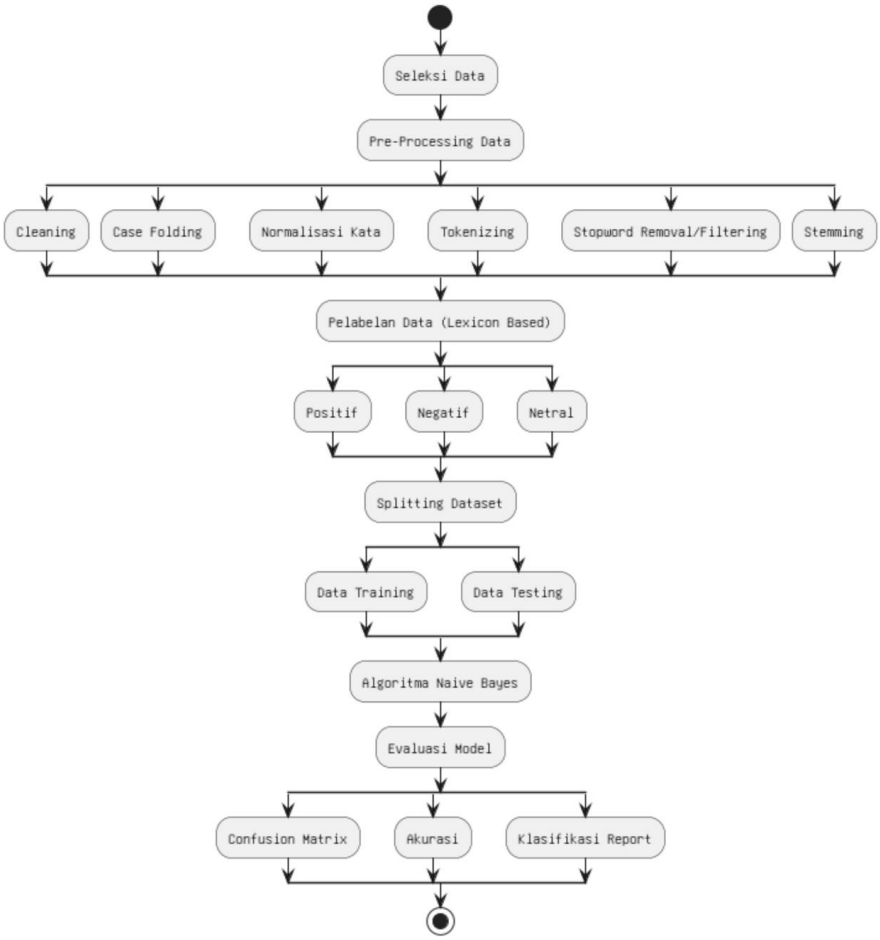
Pada tahap ini, model klasifikasi sentimen dibangun menggunakan algoritma *Multinomial Naive Bayes*. Algoritma ini dipilih karena kemampuannya untuk menangani data teks berdimensi tinggi dan performa komputasi yang efisien [20]. Model ini mempelajari probabilitas kemunculan kata pada setiap kelas sentimen berdasarkan data pelatihan. Untuk menghindari probabilitas nol, digunakan *Laplace smoothing*. Selain itu, validasi dilakukan menggunakan 10-*fold cross-validation* untuk memastikan konsistensi model dalam klasifikasi sentimen [9].

3.6. Evaluasi Model

Setelah model dilatih, tahap terakhir adalah evaluasi kinerja model. Evaluasi dilakukan menggunakan tiga metrik, yaitu:

1) *Confusion Matrix*: Matriks yang menunjukkan jumlah prediksi yang benar dan salah untuk masing-masing kelas sentimen (positif, negatif, netral).
2) Akurasi: Persentase prediksi yang benar dibandingkan dengan jumlah keseluruhan *data testing.*
3) Klasifikasi Report: Laporan yang menunjukkan nilai *precision*, *recall*, dan F1-*score* untuk setiap kelas sentimen.

Sebagai contoh, akurasi model yang diperoleh dalam penelitian ini adalah 69,6%, dengan *precision* dan *recall* yang cukup baik untuk sentimen negatif dan positif, namun rendah untuk sentimen netral. Hal ini disebabkan oleh distribusi data yang tidak seimbang antara kelas positif, negatif, dan netral [10].



Gambar 1. Tahapan KDD

Gambar 1 menggambarkan tahapan-tahapan yang diterapkan dalam penelitian ini. Tahapan ini mengikuti proses KDD yang terdiri dari seleksi data, *pre-processing*, pelabelan data, pembagian dataset, pelatihan model, dan evaluasi model. Setiap tahapan saling berhubungan dan sangat penting dalam menghasilkan model yang dapat mengklasifikasikan sentimen ulasan dengan akurat.

4 HASIL DAN PEMBAHASAN

4.1. Seleksi Data

Proses seleksi data dilakukan dengan mengumpulkan ulasan aplikasi *Grab* dari *Google Play Store* untuk periode 2020 hingga 2025 menggunakan teknik *web scraping* dengan bantuan *library* Python *google-play-scraper*. Data yang berhasil dikumpulkan terdiri dari beberapa atribut penting seperti *reviewId*, *content*, *score*, dan *timestamp*. Setelah seleksi data, ulasan di*filter* agar hanya memuat data yang berbahasa Indonesia dan Inggris serta memiliki kelengkapan atribut yang diperlukan. Hasil seleksi menghasilkan dataset yang siap untuk tahap *pre-processing*.

Tabel 1 menunjukkan contoh data ulasan yang berhasil dikumpulkan setelah proses seleksi. Tabel ini berisi informasi terkait tanggal, waktu, dan konten ulasan.

Tabel 1. Hasil seleksi Data

Tanggal	Waktu	Ulasan
2025-04-19	11:06:08	tolong perbaiki Maps pada aplikasi. belum belok sudah disuruh belok...
2025-04-27	07:14:28	saya pesan *Grab*car plus yang dapat kenapa Bluebird kali cancel...

Pada Tabel 1, dijelaskan bahwa data yang dikumpulkan mencakup atribut tanggal, waktu, dan konten ulasan yang menjadi dasar untuk analisis lebih lanjut.

4.2. Pre-Processing Data

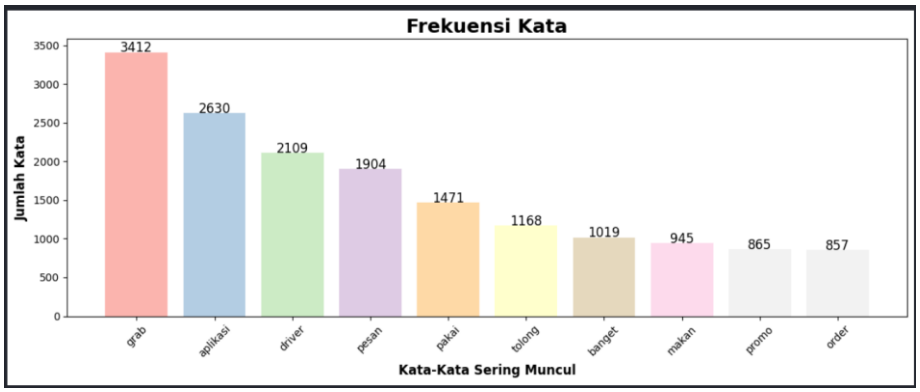
Tahapan *pre-processing* dilakukan untuk membersihkan dan menyiapkan data sebelum analisis sentimen dilakukan. Setiap tahapan dalam *pre-processing* menghasilkan perubahan signifikan pada *dataset*. Proses yang dilakukan adalah *cleaning*, *case folding*, normalisasi kata, *tokenizing*, *stopword removal*, dan *stemming*. Semua langkah tersebut bertujuan untuk memastikan data yang akan dianalisis bebas dari gangguan yang bisa mempengaruhi hasil analisis.

Tabel 2 menunjukkan hasil *pre-processing* pada beberapa data ulasan. Dalam tabel ini, setiap langkah *pre-processing* ditampilkan dalam kolom terpisah untuk memperlihatkan perubahan yang terjadi pada setiap data ulasan setelah diproses.

Tabel 2. Hasil *Pre-processing Data*

Ulasan	***Cleaning***	***Case_folding***	***Normalisasi***	***Tokenize***	***Stopword removal***	***Stemming* data**
tolong perbaiki Maps pada aplikasi...	tolong perbaiki Maps pada aplikasi...	tolong perbaiki maps pada aplikasi...	tolong perbaiki maps pada aplikasi...	['tolong', 'perbaiki', 'maps', ...]	['tolong', 'perbaiki', 'maps', ...]	tolong baik maps aplikasi belok suruh...

Pada Tabel 2, dijelaskan hasil dari tahap *pre-processing* yang melibatkan pembersihan data dan normalisasi teks untuk memudahkan analisis.



Gambar 2. Distribusi Frekuensi Kata

Gambar 2 di atas adalah hasil frekuensi kata yang paling sering muncul setelah proses *pre-processing*

4.3. Pelabelan Data

Setelah tahap *pre-processing* selesai, data diberi label sentimen menggunakan pendekatan *lexicon-based*. Label sentimen yang digunakan dalam penelitian ini adalah positif, negatif, dan netral. Proses pelabelan ini menggunakan kombinasi antara rating pengguna dan analisis kata kunci dalam setiap ulasan. Hasil pelabelan ini kemudian digunakan untuk melatih model klasifikasi.

Tabel 3 menunjukkan hasil pelabelan sentimen untuk beberapa ulasan yang telah diproses. Setiap ulasan diberi label berdasarkan sentimen yang terkandung dalam kontennya.

Tabel 3. Hasil Pelabelan Data

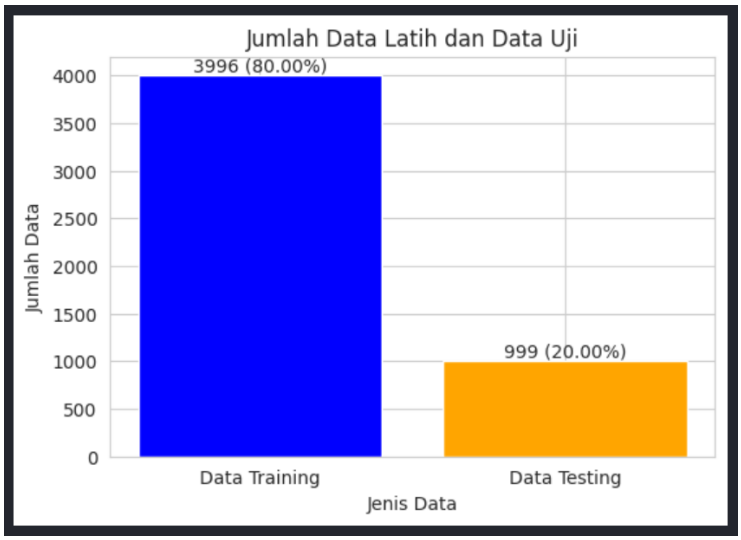
Ulasan	***Score***	***Sentiment***
tolong baik maps aplikasi belok suruh...	0	Netral
pesan *Grab*car plus bluebird kali *cancel*...	3	Positif
driver taxi keluh tetang bayar harga ...	-1	Negatif

Pada Tabel 3, dijelaskan bagaimana setiap ulasan diberi label sentimen berdasarkan konten dan skor yang diberikan oleh pengguna.

4.4. *Splitting Dataset*

Dataset yang telah diberi label kemudian dibagi menjadi dua bagian: *data training* dan *data testing*. Pembagian ini dilakukan dengan rasio 80:20, di mana 80% digunakan untuk melatih model dan 20% sisanya digunakan untuk menguji model yang telah dilatih. Pembagian ini dilakukan secara acak untuk memastikan distribusi yang adil dari masing-masing kelas sentimen.

Gambar 3 menunjukkan distribusi *data training* dan *data testing*. Diagram ini memberikan gambaran jelas tentang proporsi data yang digunakan dalam tahap pelatihan dan pengujian.



Gambar 3. Distribusi *Data Training* dan *Data Testing*

Pada Gambar 3, dapat dilihat bahwa *data training* terdiri dari 80% *dataset* yang digunakan untuk melatih model, sementara 20% sisanya digunakan untuk evaluasi kinerja model.

4.5. Evaluasi Model

Setelah model *Naive Bayes* dilatih menggunakan *data training*, tahap selanjutnya adalah evaluasi kinerja model. Evaluasi dilakukan untuk menilai seberapa baik model dalam mengklasifikasikan sentimen ulasan aplikasi *Grab* ke dalam kategori positif, negatif, dan netral. Proses evaluasi ini melibatkan beberapa metrik, seperti *Confusion Matrix*, Akurasi, dan Klasifikasi *Report*.

1) Confusion Matrix

Tabel 4 menunjukkan hasil *Confusion Matrix* dari model yang telah dilatih. *Confusion matrix* memberikan gambaran tentang jumlah prediksi yang benar dan salah untuk setiap kelas sentimen yang ada. Berdasarkan matriks ini, kita dapat melihat seberapa baik model dalam memprediksi masing-masing kategori sentimen.

Tabel 4. Hasil *Confusion Matrix*

	Prediksi Negatif	**Prediksi Netral**	**Prediksi Positif**
Aktual Negatif	309	2	71
Aktual Netral	72	5	95
Aktual Positif	61	3	381

Pada Tabel 4, dapat dilihat bahwa:

1) Negatif: Model mampu mengklasifikasikan 309 data negatif dengan benar, namun ada 71 kasus yang salah diklasifikasikan sebagai positif dan 2 sebagai netral.
2) Netral: Model mengalami kesulitan dalam mengklasifikasikan sentimen netral, hanya berhasil mengklasifikasikan 5 data netral dengan benar, sedangkan 72 data netral salah diklasifikasikan sebagai negatif dan 95 sebagai positif.
3) Positif: Model berhasil mengklasifikasikan 381 data positif dengan benar, namun 61 data positif salah diklasifikasikan sebagai negatif dan 3 sebagai netral.

2) *Precision*, *Recall* dan F1-*Score*

Selanjutnya, Tabel 5 menunjukkan hasil Klasifikasi *Report* yang memuat nilai *precision*, *recall*, dan F1-*score* untuk masing-masing kelas sentimen. *Precision* mengukur seberapa tepat prediksi yang dilakukan model, *recall* mengukur seberapa banyak data yang benar-benar terdeteksi oleh model, dan F1-*score* adalah rata-rata harmonis dari *precision* dan *recall*.

Tabel 5. Hasil Klasifikasi

Kelas	Precision	Recall	F1-score	Support
Negatif	0.70	0.81	0.75	382
Netral	0.50	0.03	0.05	172
Positif	0.70	0.86	0.77	445
Accuracy			0.70	999
Macro avg	0.63	0.56	0.52	999
Weighted avg	0.66	0.70	0.64	999

Pada Tabel 5, dijelaskan bahwa:

1) *Precision* untuk sentimen negatif dan positif masing-masing mencapai 0.70, yang menunjukkan bahwa model dapat memprediksi sentimen negatif dan positif dengan cukup baik. Namun, untuk sentimen netral, *precision* sangat rendah (0.50).
2) *Recall* untuk sentimen positif cukup tinggi (0.86), tetapi sangat rendah untuk sentimen netral (0.03), yang mengindikasikan bahwa model kesulitan dalam mendeteksi sentimen netral.
3) F1-*score* untuk sentimen negatif dan positif cukup baik, tetapi sangat rendah untuk sentimen netral (0.05), yang menunjukkan bahwa model belum optimal dalam menangani sentimen netral.

5 KESIMPULAN

Berdasarkan hasil penelitian, penerapan algoritma *Multinomial Naive Bayes* pada analisis sentimen opini masyarakat terhadap ulasan aplikasi *Grab* di *Google Play Store* periode 2020–2025 mampu mengklasifikasikan sentimen positif, negatif, dan netral dengan akurasi 69,6%, di mana model menunjukkan performa yang baik pada sentimen positif dan negatif namun masih menghadapi tantangan pada sentimen netral, sehingga hasil analisis ini memberikan kontribusi penting bagi pengembang *Grab* dalam memahami persepsi pengguna dan dapat dijadikan dasar pengambilan keputusan strategis untuk peningkatan kualitas layanan aplikasi di masa mendatang.

DAFTAR PUSTAKA

[1] A. Parisi et al., “Impact of COVID-19 pandemic on vaccine hesitancy and sentiment changes: A survey of healthcare workers in 12 countries,” Public Health, vol. 238, pp. 188–196, 2025, doi: https://doi.org/10.1016/j.puhe.2024.11.016.

[2] M. Malik, D. A. Pangestu, and M. R. Pribadi, “Analisis Sentimen Hasil Pertandingan Sepakbola Timnas Indo- nesia di Piala Asia U-23 pada Platform Youtube menggunakan Algoritma Suport Vector Machine (SVM),” Appl. Inf. Technol. Comput. Sci., vol. 3, no. 1, pp. 38-45, 2024. https://doi.org/10.58466/aicoms.v3i1.1528

[3] P. Chinnasamy, V. Suresh, K. Ramprathap, B. J. A. Jebamani, K. Srinivas Rao, and M. Shiva Kranthi, “COVID-19 vaccine sentiment analysis using public opinions on Twitter,” Mater. Today Proc., vol. 64, pp. 448–451, 2022, doi: https://doi.org/10.1016/j.matpr.2022.04.809.

[4] Z. Hanif and U. Surapati, “Analisis Sentimen terhadap Perpanjangan Masa Jabatan Presiden: Lexicon Based dan Naive Bayes pada Analisis Sentimen Pengguna Twitter,” J. JTIK, vol. 9, no. 1, pp. 204–209, 2025. [Online]. Available: https://journal.lembagakita.org/index.php/jtik Doi: https://doi.org/10.35870/jtik.v9i1.3080

[5] J. R. Jim, M. A. R. Talukder, P. Malakar, M. M. Kabir, K. Nur, and M. F. Mridha, “Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review,” Nat. Lang. Process. J., vol. 6, p. 100059, 2024, doi: https://doi.org/10.1016/j.nlp.2024.100059.

[6] M. Badjrie, A. S. P. Sari, and R. S. Dewi, “Analisis Sentimen Berbasis Aspek pada Review Female Daily Menggunakan TF-IDF dan Naïve Bayes,” J. Media Inform. Budidarma, vol. 5, no. 2, pp. 422–430, 2021. [Online]. Available: https://ejurnal.stmik-budidarma.ac.id/index.php/mib Doi: https://doi.org/10.30865/mib.v5i2.2845

[7] P. A. Henríquez and F. Alessandri, "Analyzing digital societal interactions and sentiment classification in Twitter (X) during critical events in Chile," Heliyon, vol. 10, no. 12, p. e32572, 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e32572.

[8] M. Isnan, G. N. Elwirehardja, and B. Pardamean, "Sentiment Analysis for TikTok Review Using VADER Sentiment and SVM Model," Procedia Comput. Sci., vol. 227, pp. 168-175, 2023, doi: https://doi.org/10.1016/j.procs.2023.10.514.

[9] H. T. Wijaya, A. Info, and N. Bayes, "Sentiment Analysis of Twitter Users Towards Kartu Prakerja Program Using the Naive Bayes Method," Int. J. Adv. Data Inf. Syst., vol. 5, no. 2, pp. 242-252, 2024, doi: 10.59395/ijadis.v5i2.1342 https://doi.org/10.59395/ijadis.v5i2.1342

[10] R. Jufri, E. Yuliana, and E. S. Sari, "Analysis of Public Sentiment on Google Play Store Tije Application Users Using Naïve Bayes Classifier Method," JUTIF, vol. 5, no. 1, pp. 243–251, 2024. [Online]. Available: https://jutif.if.unsoed.ac.id/index.php/jurnal Doi: https://doi.org/10.52436/1.jutif.2024.5.1.1648

[11] T. W. Putra and A. , Agung Triayudi, "Analisis Sentimen Pembelajaran Daring menggunakan Metode Naïve Bayes , KNN , dan Decision Tree," J. Teknol. Inf. dan Komun., vol. 6, no. 1, 2022, doi: https://doi.org/10.35870/jtik.v6i1.368.

[12] R. P. Tanjung, A. Purwoko, and R. A. Lubis, "Naives Bayes Algorithm for Twitter Sentiment Analysis," J. Phys. Conf. Ser., vol. 1933, no. 012019, pp. 1-6, 2021, doi: 10.1088/1742-6596/1933/1/012019. https://doi.org/10.1088/1742-6596/1933/1/012019

[13] N. A. Putri, A. Srirahayu, and N. A. Sudibyo, "Analisis Sentimen Terhadap Aplikasi KitaLulus Menggunakan Metode Naive Bayes dari Ulasan Google Play Store Sentiment Analysis of the KitaLulus Application Using the Naive Bayes Method from Google Play Store Reviews," Smart Comp Jurnalnya Orang Pint. Komput., vol. 14, no. 105, pp. 269-279, 2025, doi: 10.30591/smartcomp.v14i2.7230. https://doi.org/10.30591/smartcomp.v14i2.7230

[14] D. Rizki, S. Pratama, T. A. Munandar, and K. Fadhilla, "Multinomial Naive Bayes Algorithm for Indonesian language Sentiment Classification Related to Jakarta International Stadium (JIS)," Int. J. Inf. Technol. Comput. Sci. Appl., vol. 2, no. 1, pp. 12–22, 2024, doi: https://doi.org/10.58776/ijitcsa.v2i1.118.

[15] Y. Chen, H. Wang, and L. Zhang, "Sentiment Analysis of Public Opinion on Presidential Advisory Appointments Using Naive Bayes Classification," JIITUJ, vol. 8, no. 2, pp. 110–120, 2024. [Online]. Available: https://jiitujournal.com/index.php/jiituj Doi: https://doi.org/10.22437/jiituj.v8i2.35254

[16] I. Gustina and A. Yudhistira, "Analisis Sentimen Program Coding Anak SD Menggunakan Metode Naive Bayes Sistem Informasi , Fakultas Teknik dan Ilmu Komputer , Universitas Teknokrat Indonesia , Indonesia Sentiment Analysis of Elementary School Children ' s Coding Program Using the Naive," J. Pendidik. dan Teknol. Indones., vol. 5, no. 2, pp. 505-514, 2025, doi: https://doi.org/10.52436/1.jpti.668.

[17] C. Dewi, R.-C. Chen, H. J. Christanto, and F. Cauteruccio, "Multinomial Naïve Bayes Classifier for Sentiment Analysis of Internet Movie Database," Vietnam J. Comput. Sci., vol. 10, no. 04, pp. 485-498, 2023, doi: 10.1142/S2196888823500100. https://doi.org/10.1142/S2196888823500100

[18] N. Umar and M. A. Nur, "Application of Naïve Bayes Algorithm Variations on Indonesian General Election Analysis Dataset for Sentiment Analysis," J. RESTI, vol. 6, no. 4, pp. 585–590, 2022. [Online]. Available: https://jresti.ikmi.ac.id/index.php/jresti Doi: https://doi.org/10.29207/resti.v6i4.4179

[19] A. Falasari and M. A. Muslim, "Optimize Naïve Bayes Classifier Using Chi Square and Term Frequency Inverse Document Frequency For Amazon Review Sentiment Analysis," J. Soft Comput. Explor., vol. 3, no. 1, pp. 31–36, 2022. [Online]. Available: https://jurnal.iaii.or.id/index.php/SCE Doi: https://doi.org/10.52465/joscex.v3i1.68

[20] W. B. Zulfikar, A. R. Atmadja, and S. F. Pratama, "Sentiment Analysis on Social Media Against Public Policy Using Multinomial Naive Bayes," Sci. J. Informatics, vol. 10, no. 1, pp. 25-34, 2023, doi: 10.15294/sji.v10i1.39952. https://doi.org/10.15294/sji.v10i1.39952