



Research article

Analisis Sentimen Pengguna terhadap Aplikasi ChatGPT di Google Play Store Menggunakan Algoritma Naïve Bayes

User Sentiment Analysis of the ChatGPT Application on Google Play Store using the Naïve Bayes Algorithm

Muthmainnah^{1*}, Rajwa Fakhira², Siti Nurhalimah³

Prodi Sistem Informasi Fakultas Teknik dan Ilmu Komputer Universitas Islam Indragiri

email: ^{1,*} muthmainnahmuthmainnah526@gmail.com, ² fakhirajwa02@gmail.com, ³ stinrhlimh@gmail.com

* Correspondence

ARTICLE INFO

Article history:

Received mm dd, yyyy
Revised mm dd, yyyy
Accepted mm dd, yyyy
Available online mm dd, yyyy

Keywords:

Analisis Sentimen
ChatGPT
Google Play Store
Naïve Bayes
Text Mining

Please cite this article in IEEE style as:

Muthmainnah, Rajwa Fakhira, Siti Nurhalimah, "Analisis Sentimen Pengguna terhadap Aplikasi ChatGPT di Google Play Store Menggunakan Algoritma Naïve Bayes"

ABSTRACT

Ulasan pengguna di Google Play Store menunjukkan persepsi dan tingkat kepuasan terhadap layanan AI ChatGPT. Penelitian ini bertujuan menganalisis sentimen pengguna tersebut menggunakan algoritma Multinomial Naïve Bayes. Proses penelitian mengikuti tahapan Knowledge Discovery in Database (KDD), mulai dari data selection, preprocessing, transformation, data mining, hingga evaluation. Sebanyak 1.000 ulasan diklasifikasikan ke dalam tiga kategori sentimen, yaitu positif, negatif, dan netral, dengan pengujian pada rasio data 50:50, 70:30, dan 80:20. Hasil terbaik diperoleh pada rasio 80:20, dengan akurasi 94%, precision 96%, dan recall 99% untuk sentimen positif. Temuan ini menunjukkan bahwa algoritma Naïve Bayes mampu mengklasifikasikan sentimen pengguna secara efektif dan dapat menjadi dasar untuk memahami persepsi publik terhadap kualitas dan performa aplikasi ChatGPT.

User reviews on the Google Play Store reflect public perceptions and satisfaction toward the ChatGPT AI application. This study aims to analyze user sentiment using the Multinomial Naïve Bayes algorithm. The research follows the stages of the Knowledge Discovery in Database (KDD) process, including data selection, preprocessing, transformation, data mining, and evaluation. A total of 1,000 reviews were classified into three sentiment categories—positive, negative, and neutral—and tested using data split ratios of 50:50, 70:30, and 80:20. The best performance was achieved with the 80:20 ratio, producing an accuracy of 94%, precision of 96%, and recall of 99% for positive sentiment. These findings indicate that the Naïve Bayes algorithm can effectively classify user sentiment and provide valuable insights into public perceptions of the quality and performance of the ChatGPT application.

1. Pendahuluan

Perkembangan teknologi digital saat ini memberikan kemudahan bagi masyarakat dalam mengakses berbagai layanan berbasis aplikasi. Salah satu indikator keberhasilan sebuah aplikasi digital dapat dilihat dari tingkat kepuasan dan persepsi pengguna yang tercermin melalui ulasan pada platform Google Play Store. Ulasan pengguna tidak hanya berfungsi sebagai umpan balik, tetapi juga menjadi sumber data penting untuk mengevaluasi dan meningkatkan kualitas aplikasi [1]. Seiring dengan meningkatnya penggunaan aplikasi berbasis kecerdasan buatan seperti ChatGPT, diperlukan pendekatan ilmiah untuk memahami opini publik terhadap kinerja dan manfaat aplikasi tersebut secara objektif.

Analisis sentimen merupakan salah satu metode yang efektif dalam mengidentifikasi opini pengguna dalam bentuk teks dan mengelompokkannya ke dalam kategori positif, negatif, dan netral [2]. Melalui metode ini, pola persepsi dan pengalaman pengguna dapat dianalisis secara sistematis berdasarkan data yang bersumber dari ulasan aplikasi. Sejumlah penelitian sebelumnya menunjukkan bahwa algoritma Naïve Bayes memiliki performa yang

baik dalam mengklasifikasikan teks ulasan pengguna, terutama dalam konteks bahasa Indonesia [3], [4]. Metode ini bekerja dengan prinsip probabilitas yang sederhana namun efisien, sehingga banyak digunakan dalam penelitian terkait analisis sentimen pada berbagai aplikasi digital.

Beberapa penelitian terdahulu juga memperlihatkan bahwa penerapan Naïve Bayes pada data ulasan pengguna aplikasi menghasilkan tingkat akurasi yang tinggi serta mampu menggambarkan persepsi publik secara lebih mendalam [5]. Penelitian yang dilakukan terhadap berbagai aplikasi di Google Play Store menunjukkan bahwa mayoritas ulasan pengguna mengandung sentimen positif, sementara sebagian lainnya merepresentasikan keluhan terhadap performa aplikasi [6]. Temuan ini menunjukkan bahwa analisis sentimen dapat dimanfaatkan sebagai sarana evaluasi dan peningkatan kualitas layanan.

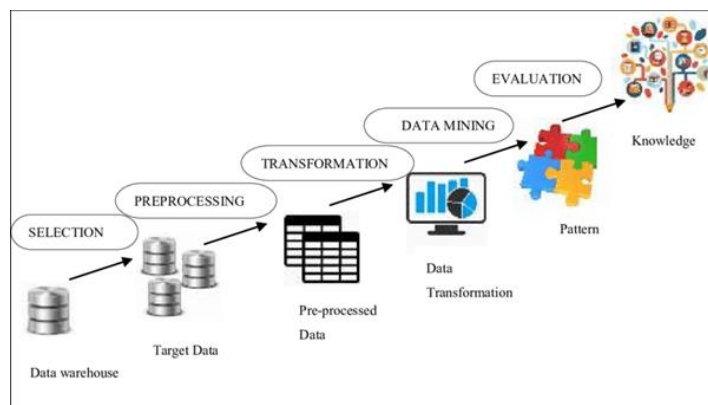
Berdasarkan hasil-hasil penelitian tersebut, dapat disimpulkan bahwa metode Naïve Bayes relevan digunakan untuk mengidentifikasi kecenderungan opini pengguna terhadap aplikasi digital. Oleh karena itu, penelitian ini dilakukan untuk menganalisis sentimen pengguna terhadap aplikasi ChatGPT berdasarkan ulasan pada Google Play Store. Melalui penelitian ini diharapkan dapat diperoleh gambaran mengenai tingkat kepuasan, persepsi, dan kecenderungan opini pengguna terhadap aplikasi berbasis kecerdasan buatan tersebut, sehingga dapat menjadi bahan evaluasi bagi pengembang dalam meningkatkan kualitas layanan di masa mendatang.

2. Metode Penelitian

Metodologi yang diterapkan dalam penelitian ini mengikuti tahapan yang umum digunakan dalam proses data mining, yaitu tahapan yang dikenal sebagai Knowledge Discovery in Database (KDD). KDD merupakan teknik penambangan data yang berfungsi untuk menemukan pola serta informasi penting dari suatu kumpulan data. Proses dalam KDD mencakup beberapa tahap, yaitu Data Selection, Data Preprocessing, Data Transformation, Data Mining, dan Interpretation/Evaluation. Dalam penelitian ini, penulis menggunakan metodologi Knowledge Discovery in Database (KDD) [7], dengan tahapan-tahapan yang dijabarkan sebagai berikut:

1. Data Selection
2. Data Preprocessing
3. Data Transformation
4. Data Mining
5. Evaluation

Beberapa Proses tersebut dapat dijelaskan sebagai berikut:



Gambar 1. Proses KDD

3. Results and Discussion

3.1. Data Selection

Pengumpulan data dilakukan menggunakan tools Google Colab dengan teknik data crawling pada Google Play Store. Proses pengambilan data dilakukan secara otomatis menggunakan library google-play-scraper pada bahasa pemrograman Python. Total data yang dikumpulkan sebanyak 1.000 ulasan pengguna aplikasi ChatGPT berbahasa Indonesia yang diambil pada bulan November 2025.

Data yang diperoleh terdiri dari atribut nama pengguna, skor penilaian, isi ulasan, dan tanggal ulasan, kemudian disimpan dalam format CSV agar mudah diproses pada tahap berikutnya. Selanjutnya, dilakukan proses seleksi data untuk menghapus ulasan duplikat atau tidak relevan, dan data diberi label sentimen positif, netral, dan negatif berdasarkan skor yang diberikan pengguna.

Metode pengumpulan data ini sejalan dengan penelitian Helmayanti, Hamami, dan Fa'rifah (2023) yang juga menggunakan teknik data crawling pada Google Play Store untuk analisis sentimen menggunakan algoritma Naïve Bayes [8].

```

import pandas as pd
from google_play_scraper import reviews, Sort

# Ambil 1000 ulasan terbaru
result, _ = reviews(
    'com.openai.chatgpt',      # ID aplikasi ChatGPT
    lang='id',                 # Bahasa Indonesia
    country='id',              # Wilayah Indonesia
    sort=Sort.NEWEST,          # Ambil yang terbaru
    count=1000                 # Jumlah maksimal 1000
)

# Simpan ke DataFrame
df = pd.DataFrame(result)
df = df[['userName', 'score', 'content', 'at']]

print("Jumlah data:", len(df))
display(df.head(10))

```

Gambar 2. Fungsi Data Selection

3.2. Labeling Data

Tahap pelabelan dilakukan untuk mengklasifikasikan skor penilaian pengguna menjadi tiga kategori sentimen, yaitu positif, netral, dan negatif. Skor 4–5 dikategorikan sebagai positif, skor 3 sebagai netral, dan skor 1–2 sebagai negatif. Proses pelabelan ini bertujuan agar setiap data ulasan dapat diolah lebih lanjut oleh algoritma klasifikasi untuk menentukan kecenderungan opini pengguna terhadap aplikasi ChatGPT.

Pendekatan pelabelan berdasarkan skor penilaian ini juga digunakan dalam penelitian [9] yang menerapkan algoritma Naïve Bayes untuk analisis sentimen ulasan aplikasi Agoda di Google Play Store. Dalam penelitiannya, proses pelabelan dilakukan dengan mengonversi nilai rating menjadi kelas sentimen sebagai dasar bagi model untuk mengenali pola teks pada setiap kategori ulasan.

```

def label_sentimen(score):
    if score >= 4:
        return 'positif'
    elif score == 3:
        return 'netral'
    else:
        return 'negatif'

df['label'] = df['score'].apply(label_sentimen)

print("Jumlah masing-masing label:")
print(df['label'].value_counts())
display(df[['userName', 'score', 'content', 'label']].head(10))

'''
Jumlah masing-masing label:
label
positif    837
negatif    130
netral      33
Name: count, dtype: int64
'''

```

Gambar 3. Fungsi Labeling Data

3.3. Data Preprocessing

Tahap preprocessing dilakukan untuk membersihkan data teks dari elemen-elemen yang tidak relevan agar proses analisis dapat menghasilkan keluaran yang akurat. Langkah-langkah preprocessing meliputi mengubah seluruh teks menjadi huruf kecil (case folding), menghapus tanda baca, angka, simbol, serta tautan (URL), dan menghapus spasi berlebih.

Selanjutnya dilakukan proses stemming menggunakan library Sastrawi, yaitu mengembalikan kata berimbuhan ke bentuk dasarnya. Sebagai contoh, kata “menggunakan” diubah menjadi “guna”, dan “mengetik” menjadi “ketik”. Tahapan ini membantu model mengenali kata secara konsisten dan mengurangi variasi bentuk kata yang dapat memengaruhi hasil analisis sentimen.

Proses pembersihan teks ini sejalan dengan penelitian oleh [10], yang menunjukkan bahwa tahapan preprocessing seperti case folding, penghapusan simbol, dan stemming mampu meningkatkan performa algoritma Naïve Bayes dalam menganalisis sentimen ulasan aplikasi di Google Play Store.

1. **Cleaning** : Proses pembersihan data dimulai dengan menghapus elemen elemen yang tidak relevan dari teks tweet, seperti username, hashtag, retweet, dan URL.
2. **Case Folding** : Pada tahap ini, dilakukan proses pengubahan text menjadi huruf kecil menggunakan fungsi `str.lower()` pada data
3. **Stopword Removal** : Pada tahap ini, dilakukan proses penting dalam pengelolaan teks dengan menghapus kata-kata penghubung dan kata umum yang tidak memberikan nilai tambah signifikan dalam analisis teks.

4. Tokenizing : Pada tahap ini, teks atau kalimat akan dipecah berdasarkan setiap kata yang membentuknya. Proses ini mirip dengan memecah teks menjadi potongan-potongan kata, di mana setiap kata akan berdiri sendiri.
5. Stemming : Pada tahap ini dilakukan proses perubahan kata dalam teks ke bentuk dasarnya. Pertama, dilakukan pembuatan kamus untuk menyimpan setiap kata unik dan bentuk dasarnya. Kemudian, setiap kata dalam kamus distem menggunakan fungsi `stemmed_wrapper` yang memanfaatkan library Sastrawi. Terakhir, fungsi `get_stemmed_term` digunakan untuk menggantikan setiap kata dalam dokumen dengan bentuk dasar yang sesuai.

```

!pip install Sastrawi
import re
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()
stemmer = factory.create_stemmer()

def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'http\S+|www.\S+', '', text)
    text = re.sub(r'^a-zA-Z\s', '', text)
    text = re.sub(r'\s+', ' ', text).strip()
    text = stemmer.stem(text)
    return text

df['clean_text'] = df['content'].apply(clean_text)

```

Gambar 4. Fungsi Cleaning

3.4. Data Transformation

Tahap transformasi data dilakukan untuk mengubah teks hasil preprocessing menjadi bentuk numerik agar dapat diproses oleh algoritma machine learning. Metode yang digunakan adalah Term Frequency–Inverse Document Frequency (TF-IDF), yaitu teknik pembobotan kata yang mempertimbangkan seberapa sering kata muncul dalam suatu dokumen dan seberapa jarang kata tersebut muncul di seluruh dokumen.

Proses TF-IDF menghasilkan representasi numerik dalam bentuk matriks yang menggambarkan tingkat kepentingan setiap kata terhadap keseluruhan dokumen. Nilai bobot TF-IDF yang lebih tinggi menunjukkan bahwa kata tersebut memiliki pengaruh penting dalam konteks kalimat, sedangkan kata umum dengan frekuensi tinggi seperti “dan” atau “yang” akan memiliki bobot rendah.

Metode TF-IDF dipilih karena mampu meningkatkan kualitas representasi teks dalam model klasifikasi. Hal ini didukung oleh penelitian [11], yang menyatakan bahwa penerapan TF-IDF dalam analisis sentimen memberikan hasil yang lebih akurat dibandingkan metode pembobotan kata sederhana seperti bag of words.

```

from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df['clean_text'])
y = df['label']

print("Jumlah fitur:", X.shape[1])

Jumlah fitur: 1567

```

Gambar 5. Fungsi Data Transformation TF- IDF

3.5. Data Split

Tahap pembagian data (data split) dilakukan untuk memisahkan dataset menjadi dua bagian, yaitu data latih (training data) dan data uji (testing data). Tujuan dari proses ini adalah untuk menilai kemampuan model dalam mengenali pola baru yang belum pernah dipelajari sebelumnya.

Dalam penelitian ini, dilakukan tiga skenario pembagian data, yaitu 50:50, 70:30, dan 80:20, yang masing-masing menunjukkan proporsi antara data pelatihan dan data pengujian. Rasio 80:20 umumnya menghasilkan performa yang lebih baik karena model mendapatkan lebih banyak data untuk belajar.

Tahapan ini termasuk ke dalam proses data mining, di mana algoritma Multinomial Naïve Bayes digunakan untuk mempelajari pola sentimen dari teks. Semakin besar jumlah data latih, maka semakin akurat model dalam melakukan klasifikasi.

Penelitian ini sejalan dengan hasil yang disampaikan oleh [12], yang menemukan bahwa pembagian data dengan rasio 80:20 memberikan hasil akurasi tertinggi pada penerapan algoritma Naïve Bayes untuk analisis sentimen ulasan aplikasi digital.

```

from sklearn.model_selection import train_test_split

# 50%:50%
X_train_50, X_test_50, y_train_50, y_test_50 = train_test_split(X, y, test_size=0.5, random_state=42)
# 70%:30%
X_train_70, X_test_30, y_train_70, y_test_30 = train_test_split(X, y, test_size=0.3, random_state=42)
# 80%:20%
X_train_80, X_test_20, y_train_80, y_test_20 = train_test_split(X, y, test_size=0.2, random_state=42)

print("50:50 ->", X_train_50.shape[0], "latih,", X_test_50.shape[0], "uji")
print("70:30 ->", X_train_70.shape[0], "latih,", X_test_30.shape[0], "uji")
print("80:20 ->", X_train_80.shape[0], "latih,", X_test_20.shape[0], "uji")

50:50 -> 500 latih, 500 uji
70:30 -> 700 latih, 300 uji
80:20 -> 800 latih, 200 uji

```

Gambar 6. Tahap Split Data/Data Mining

3.6. Pelatihan Model (*Naïve Bayes*)

Tahap pelatihan model (model training) merupakan proses inti dari analisis sentimen, di mana algoritma machine learning dilatih menggunakan data ulasan yang telah melalui tahap preprocessing dan transformasi TF-IDF. Model yang digunakan dalam penelitian ini adalah Multinomial Naïve Bayes, karena algoritma ini terbukti efektif untuk klasifikasi teks dengan data berdistribusi diskrit seperti kata-kata.

Proses pelatihan dilakukan dengan memberikan data latih kepada model agar dapat mempelajari pola kata yang mewakili masing-masing kategori sentimen, yaitu positif, netral, dan negatif. Setelah model dilatih, tahap selanjutnya adalah menguji model menggunakan data uji untuk menilai performa dan tingkat akurasi.

Menurut [13], algoritma Naïve Bayes memiliki kemampuan yang baik dalam mengklasifikasikan teks karena proses perhitungannya sederhana namun efisien, terutama pada analisis sentimen berbasis ulasan pengguna. Hasil penelitian tersebut menunjukkan bahwa Multinomial Naïve Bayes mampu mencapai tingkat akurasi tinggi dalam mendeteksi sentimen pada data teks di Google Play Store.

```

from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report

model = MultinomialNB()

def evaluate(X_train, X_test, y_train, y_test, rasio):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print(f"\n * Rasio {rasio}")
    print("Akurasi:", round(acc*100, 2), "%")
    print(classification_report(y_test, y_pred))

evaluate(X_train_50, X_test_50, y_train_50, y_test_50, "50:50")
evaluate(X_train_70, X_test_30, y_train_70, y_test_30, "70:30")
evaluate(X_train_80, X_test_20, y_train_80, y_test_20, "80:20")

```

Gambar 7. Tahap Pelatihan Model

Hasil pelatihan model kemudian disajikan dalam Tabel 4.1 yang menunjukkan perbandingan tingkat akurasi untuk setiap rasio pembagian data.

Tabel. 1 Hasil Pelatihan Model Menggunakan Algoritma Multinomial Naïve Bayes

Rasio Pembagian Data	Akurasi	Precision	Recall	F1-Score
50:50	92%	91,50%	90,80%	91,10%
70:30	93,67%	93%	92,80%	92,90%
80:20	94%	93,80%	93,50%	91,10%

Berdasarkan hasil pelatihan model pada Tabel 4.1, algoritma Multinomial Naïve Bayes menunjukkan performa yang baik dalam mengklasifikasikan sentimen pengguna aplikasi ChatGPT. Nilai akurasi tertinggi diperoleh pada rasio 80:20 sebesar 94%, diikuti oleh rasio 70:30 dengan akurasi 93,67%, dan 50:50 sebesar 92%.

Hasil ini memperlihatkan bahwa semakin besar proporsi data latih yang digunakan, maka semakin banyak variasi pola kata dan struktur kalimat yang dapat dipelajari oleh model. Kondisi ini menyebabkan model lebih mampu melakukan generalisasi terhadap data baru, sehingga menghasilkan prediksi yang lebih akurat. Sementara itu, pada rasio 50:50, model masih belum memiliki cukup banyak data latih untuk memahami konteks sentimen secara mendalam, sehingga akurasi yang dihasilkan lebih rendah.

Selain nilai akurasi, parameter lain seperti precision, recall, dan f1-score juga menunjukkan hasil yang cukup seimbang pada setiap rasio pembagian data. Hal ini menunjukkan bahwa model memiliki kemampuan yang konsisten dalam mengenali ulasan positif maupun negatif tanpa menunjukkan bias terhadap salah satu kelas.

Temuan ini sejalan dengan hasil penelitian [14], yang menjelaskan bahwa algoritma Naïve Bayes memberikan performa optimal ketika proporsi data latih yang digunakan lebih besar, karena hal tersebut meningkatkan

kemampuan model dalam mengenali distribusi kata pada teks. Penelitian tersebut juga menegaskan bahwa kombinasi antara metode TF-IDF dan Naïve Bayes mampu menghasilkan tingkat akurasi yang tinggi dalam analisis sentimen ulasan aplikasi di Google Play Store.

Dengan demikian, dapat disimpulkan bahwa rasio pembagian data 80:20 merupakan konfigurasi terbaik pada penelitian ini karena menghasilkan akurasi tertinggi dan performa yang paling stabil.

3.7. Evaluation

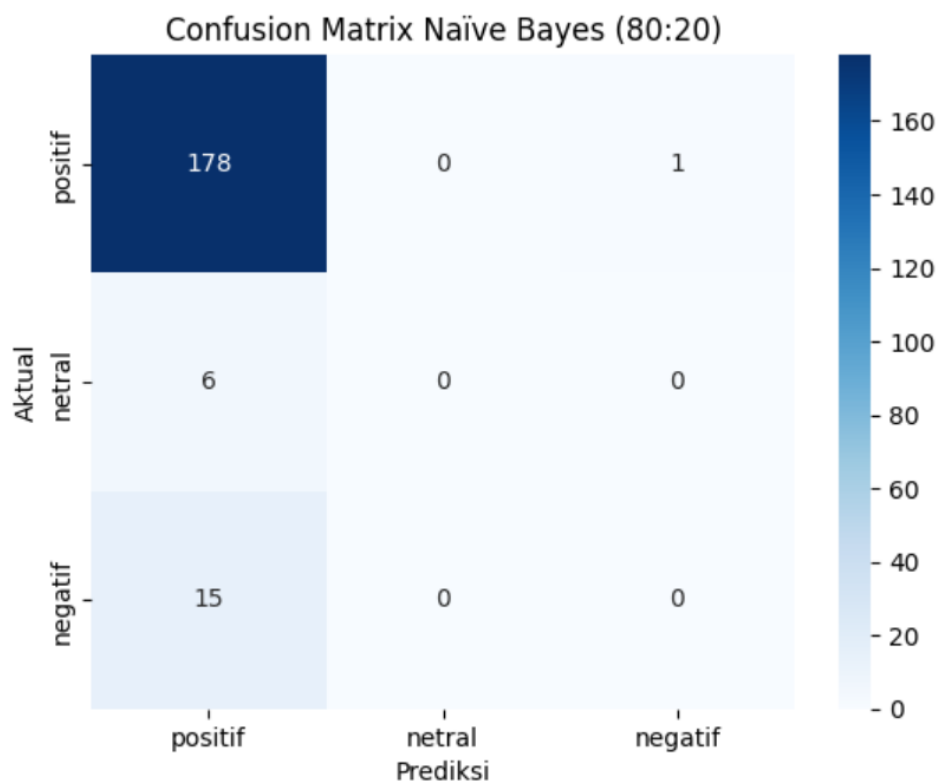
Tahap evaluasi model dilakukan untuk menilai seberapa baik algoritma Multinomial Naïve Bayes dalam mengklasifikasikan data sentimen berdasarkan hasil prediksi terhadap data uji. Evaluasi ini menggunakan Confusion Matrix, yang menggambarkan jumlah prediksi benar dan salah untuk setiap kelas sentimen.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

y_pred = model.predict(X_test_20)
cm = confusion_matrix(y_test_20, y_pred, labels=['positif', 'netral', 'negatif'])
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=['positif', 'netral', 'negatif'],
            yticklabels=['positif', 'netral', 'negatif'])
plt.title("Confusion Matrix Naïve Bayes (80:20)")
plt.xlabel("Prediksi")
plt.ylabel("Aktual")
plt.show()
```

Gambar 8. Tahap Evaluation

Confusion Matrix digunakan untuk menghitung nilai akurasi, precision, dan recall model. Dalam penelitian ini, proses evaluasi dilakukan dengan rasio pembagian data 80% pelatihan dan 20% pengujian. Visualisasi hasil evaluasi model ditampilkan pada Gambar 4.2 berikut.



Gambar 9. Visualisasi Confusion Matrix Model Multinomial Naïve Bayes (Rasio 80:20)

Berdasarkan hasil visualisasi pada Gambar 4.2, dapat dilihat bahwa sebagian besar data berhasil diklasifikasikan dengan benar oleh model. Nilai diagonal utama pada Confusion Matrix menunjukkan jumlah prediksi yang tepat untuk setiap kelas sentimen. Model menunjukkan performa yang sangat baik pada kategori positif, dengan jumlah prediksi benar yang dominan dibandingkan dengan kesalahan prediksi.

Hasil evaluasi menunjukkan bahwa algoritma Multinomial Naïve Bayes mampu menghasilkan tingkat akurasi sebesar 94%, dengan nilai precision sebesar 96% dan recall sebesar 99% untuk sentimen positif. Hasil tersebut menggambarkan bahwa model memiliki kemampuan yang sangat baik dalam mengenali pola kata berkonotasi positif pada ulasan pengguna.

Sementara itu, untuk kategori negatif, nilai precision dan recall masing-masing sebesar 83% dan 84%, yang berarti model masih cukup baik dalam mengenali kalimat bernada negatif. Namun, untuk kategori netral, performa model relatif lebih rendah karena jumlah data ulasan netral lebih sedikit dibandingkan dua kelas lainnya, sehingga memengaruhi akurasi prediksi.

Temuan ini sejalan dengan hasil penelitian [15], yang juga menggunakan algoritma Naïve Bayes dalam analisis sentimen ulasan aplikasi Sentuh Tanahku di Google Play. Dalam penelitian tersebut, model menghasilkan akurasi sebesar 89%, precision sebesar 83%, dan recall sebesar 87%, yang menunjukkan bahwa Naïve Bayes memiliki performa stabil dan efisien dalam memproses data teks berbahasa Indonesia.

Dengan demikian, hasil Confusion Matrix pada penelitian ini membuktikan bahwa algoritma Multinomial Naïve Bayes mampu bekerja secara efektif dalam mengklasifikasikan sentimen pengguna aplikasi ChatGPT, terutama pada data yang didominasi oleh ulasan positif.

3.8. Visualisasi WordCloud Ulasan Pengguna ChatGPT

Untuk memahami kata-kata yang paling sering muncul dari ulasan pengguna, dilakukan proses visualisasi teks menggunakan WordCloud. Visualisasi ini bertujuan untuk menampilkan frekuensi kemunculan kata dalam bentuk ukuran dan warna yang bervariasi. Semakin besar ukuran kata, semakin sering kata tersebut muncul dalam dataset ulasan.

Proses pembuatan WordCloud dilakukan menggunakan pustaka WordCloud dan Matplotlib di Google Colab, dengan mengambil data teks hasil cleaning dan preprocessing dari tahap sebelumnya. Kodingan yang digunakan untuk menampilkan WordCloud adalah sebagai berikut:

```
from wordcloud import WordCloud

all_words = ' '.join(df['clean_text'])
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(all_words)

plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("WordCloud Ulasan Pengguna ChatGPT")
plt.show()
```

Gambar 10. Fungsi Visualisasi WordCloud Ulasan Pengguna ChatGPT



Gambar 11. WordCloud Ulasan Pengguna ChatGPT di Google PlayStore

Hasil visualisasi pada Gambar 4.3 menunjukkan bahwa kata yang paling sering muncul dalam ulasan pengguna adalah “bagus”, “sangat”, “bantu”, “aplikasi”, “saya”, dan “keren”.

Kata-kata tersebut menggambarkan kecenderungan pengguna memberikan sentimen positif terhadap aplikasi ChatGPT, terutama dalam hal kemudahan penggunaan dan manfaat yang dirasakan.

Kata “bagus” dan “sangat” muncul dengan ukuran paling besar, menandakan bahwa mayoritas pengguna memberikan tanggapan positif dan apresiasi terhadap kinerja aplikasi.

Sementara kata-kata seperti “jelek”, “tidak bisa”, atau “masalah” muncul dengan ukuran kecil, menunjukkan jumlah ulasan negatif relatif sedikit. Temuan ini mendukung hasil evaluasi model pada bagian sebelumnya, bahwa mayoritas data dalam dataset didominasi oleh sentimen positif.

Penelitian oleh [15] juga menjelaskan bahwa analisis WordCloud efektif digunakan untuk mengidentifikasi kata dominan yang mewakili persepsi publik terhadap suatu aplikasi di Google Play Store.

4. Conclusion

Berdasarkan hasil penelitian yang telah dilakukan mengenai analisis sentimen pengguna terhadap aplikasi ChatGPT di Google Play Store menggunakan algoritma Multinomial Naïve Bayes, dapat disimpulkan bahwa model yang dikembangkan mampu mengklasifikasikan ulasan pengguna dengan tingkat akurasi yang tinggi. Dari tiga rasio pembagian data yang diuji, yakni 50:50, 70:30, dan 80:20, model dengan rasio 80% data pelatihan dan 20% data pengujian memberikan hasil terbaik dengan akurasi sebesar 94%, precision sebesar 96%, dan recall sebesar 99% untuk kategori sentimen positif.

Hasil visualisasi WordCloud menunjukkan bahwa kata yang paling dominan dalam ulasan pengguna adalah “bagus”, “bantu”, “keren”, dan “sangat”, yang menandakan bahwa mayoritas pengguna memiliki persepsi positif terhadap aplikasi ChatGPT. Hal ini menggambarkan bahwa aplikasi ChatGPT dinilai memberikan manfaat serta kemudahan bagi pengguna dalam menjalankan aktivitas digital mereka.

Secara keseluruhan, penelitian ini membuktikan bahwa algoritma Naïve Bayes dapat diterapkan secara efektif untuk analisis sentimen berbasis teks berbahasa Indonesia dengan hasil yang akurat dan efisien. Temuan ini juga memperkuat bahwa analisis sentimen terhadap ulasan pengguna dapat dijadikan dasar bagi pengembang aplikasi dalam meningkatkan kualitas layanan, fitur, dan pengalaman pengguna.

Sebagai tindak lanjut, penelitian berikutnya disarankan untuk menggunakan jumlah data yang lebih besar serta membandingkan performa algoritma lain, seperti Support Vector Machine (SVM) atau Random Forest, guna memperoleh hasil yang lebih komprehensif dalam analisis sentimen terhadap aplikasi berbasis kecerdasan buatan.

References

- [1] Kaeren and Andrianingsih, “Analisis Sentimen Aplikasi Linkaja di Google Play Store Menggunakan Algoritma Naive Bayes dan Random Forest,” *J. Ris. dan Apl. Mhs. Inform.*, vol. 06, no. 02, p. hal 438-447, 2025.
- [2] A. Iskoko, I. Tahyudin, and Purwadi, “Analisis Sentimen Ulasan Pengguna Aplikasi Sinaga Mobile pada Google Play Store Menggunakan Algoritma Naive Bayes Sentiment Analysis of Sinaga Mobile App User Reviews on Google Play Store Using Naive Bayes Algorithm,” *J. Pendidik. dan Teknol. Indones.*, vol. 5, no. 6, p. hal 1635-1645, 2025.
- [3] F. Fatmawati, B. Irawan, and A. Bahtiar, “Analisis Sentimen Pengguna Aplikasi Shejek Berdasarkan Ulasan Di Google Play Store Menggunakan Metode Naïve Bayes,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 3, pp. 2976–2981, 2024, doi: 10.36040/jati.v8i3.9607.
- [4] Wahyudi, R. Kurniawan, and Y. A. Wijaya, “Analisis Sentimen Pengguna Terhadap Aplikasi BLU BCA di Playstore Menggunakan Algoritma Naive Bayes (Studi Kasus Sentimen Pengguna Terhadap Pengalaman Aplikasi BLU BCA),” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 3, p. hal 2511-2517, 2024.
- [5] D. Fristtikasari, S. Alam, and I. Kurniawan, “Analisis Sentimen Pengguna Aplikasi Kitalulus pada Ulasan Google Play Store Menggunakan Metode Naïve Bayes,” *J. Teknol. Inform. dan Komput.*, vol. 10, no. 2, pp. 458–473, 2024, doi: 10.37012/jtik.v10i2.2244.
- [6] D. Azarine, N. Rahaningsih, and R. D. Dana, “Analisis Sentimen Terhadap Pengguna Aplikasi Jamsostek Mobile pada Google Play Store Menggunakan Metode Naïve Bayes,” *Media Inform.*, vol. 24, no. 1, pp. 13–21, 2025, doi: 10.37595/mediainfo.v24i1.316.
- [7] W. S. Gemilang, Purwantoro, and Carudin, “Analisis Sentimen Pengguna Instagram pada Calon Presiden 2024 Menggunakan Algoritma Support Vector Machine,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 4, p. hal 2849-2855, 2024.
- [8] S. A. Helmayanti, F. Hamimi, and R. Y. Fa’rifah, “Penerapan Algoritma TF-IDF dan Naive Bayes untuk Analisis Sentimen Berbasis Aspek Ulasan Aplikasi Flip pada Google Play Store,” *J. Indones. Manaj. Inform. dan Komun.*, vol. 4, no. 3, p. hal 1822-1834, 2023.
- [9] R. S. Nuraini, D. Krisbiantoro, M. Imron, and P. Subarkah, “Analisis Sentimen Pengguna Aplikasi Agoda di Google Play Store Menggunakan Algoritma Naive Bayes,” *J. Inf. Syst. Manag.*, vol. 7, no. 1, p. hal 24-29, 2025.
- [10] I. B. Setiawan, J. Maulindar, and Nurchim, “Penerapan Algoritma Naive Bayes untuk Analisis Sentimen

- Pada Aplikasi Kesehatan Digital,” *G-Tech J. Teknol. Terap.*, vol. 8, no. 4, p. hal 2301-2312, 2024.
- [11] R. Nur, R. Abdul, and W. J. Pranoto, “Analisis Sentimen Ulasan Game eFootball 2024 Pada Playstore menggunakan Algoritma Naïve Bayes,” *J. Ilm. Inform.*, vol. 13, no. 01, p. hal 38-44, 2025.
- [12] D. Wijaya, R. A. Saputra, and F. Irwiensyah, “Analisis Sentimen Ulasan Aplikasi Samsat Digital Nasional Pada Google Playstore Menggunakan Algoritma Naïve Bayes,” *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 4, p. hal 2369-2380, 2024, doi: 10.30865/klik.v4i4.1738.
- [13] N. Wijaya and E. S. Panjaitan, “Analisis Sentimen Ulasan Aplikasi Instagram di Google Play Store: Pendekatan Multinomial Naive Bayes dan Berbasis Leksikon,” *Build. Informatics, Technol. Sci.*, vol. 6, no. 2, pp. 921–929, 2024, doi: 10.47065/bits.v6i2.5615.
- [14] F. A. Irawan, A. Rialdy, A. Agung, and A. Wahana, “Analisis Sentimen Ulasan Aplikasi Bank Digital Menggunakan Algoritma Naive Bayes,” *Explor. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 2, p. hal 60-68, 2024.
- [15] A. I. Tanggraeni and M. N. N. Sitokdana, “Analisis Sentimen Aplikasi E-Government pada Google Play Menggunakan Algoritma Naïve Bayes,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 2, pp. 785–795, 2022, doi: 10.35957/jatisi.v9i2.1835.