

# Penerapan Metode Naïve Bayes untuk Analisis Sentimen pada Ulasan Pengguna Aplikasi ChatGPT di Google Play Store

Tri Ramadhani Putra Hermawan, Akhmad Rizal Dzikrillah\*

Fakultas Teknik, Teknik Informatika, Universitas Muhammadiyah Prof. Dr. Hamka, Jakarta, Indonesia

Email: <sup>1</sup>triramadhani.putrahermawan26@gmail.com, <sup>2,\*</sup>ahmadrizaldzikrillah@uhamka.ac.id

Email Penulis Korespondensi: ahmadrizaldzikrillah@uhamka.ac.id

Submitted: 23/06/2024; Accepted: 30/06/2024; Published: 30/06/2024

**Abstrak**—ChatGPT merupakan sebuah aplikasi chatbot yang dikembangkan oleh OpenAI. Aplikasi ini telah menarik banyak pengguna dalam waktu singkat. Komentar pengguna dikategorikan menjadi positif dan negatif, yang menunjukkan sentimen mereka pada penggunaan aplikasi ini. Meskipun ChatGPT memberikan kemudahan-kemudahan dari berbagai fiturnya, ChatGPT sendiri juga memiliki sisi negatifnya apabila disalahgunakan. Beberapa orang beranggapan bahwa nantinya masyarakat akan bergantung pada informasi yang diberikan oleh chatbot ini dan mengurangi keinginan untuk mencari tahu informasinya sendiri, karena informasi dari ChatGPT masih menggunakan model generasi lama. Dari keresahan akan hal tersebut dibuatlah penelitian lebih dalam tentang analisis sentimen masyarakat yang telah menggunakan aplikasi ChatGPT. Diharapkan melalui penelitian ini akan mendapatkan kesimpulan tentang tanggapan dari ulasan masyarakat terhadap penggunaan aplikasi ChatGPT. Data penelitian akan diambil dari ulasan para pengguna aplikasi ChatGPT yang ada di Play Store. Google Colaboratory bersama Google Play Scraper akan digunakan ketika proses mengumpulkan data. Data yang digunakan berjumlah 1430 data yang akan melalui tahap preprocessing untuk dibersihkan. Setelah data berhasil dibersihkan, maka data akan melalui proses labeling data positif dan negatif, dan akan diklasifikasikan melalui metode Naïve Bayes. Hasil studi menunjukkan bahwa penerapan metode Naïve Bayes mampu mengklasifikasikan sentimen pengguna menggunakan Confusion Matrix dengan persentase nilai akurasi sebesar 94,05%, persentase nilai presisi sebesar 95% untuk positif dan 81,25% untuk negatif. Lalu persentase nilai recall untuk positif 98,84%, dan untuk negatif 48%.

**Kata Kunci:** Naïve Bayes; Confusion Matrix; ChatGPT; Play Store; Analisis Sentimen

**Abstract**—ChatGPT is a chatbot application developed by OpenAI. It has attracted a large number of users in a short period of time. User comments are categorized into positive and negative, indicating their sentiments on using this app. Although ChatGPT provides convenience from its various features, it also has its downside if misused. Some people think that people will depend on the information provided by this chatbot and reduce the desire to find out the information themselves, because the information from ChatGPT still uses the old generation model. From this concern, a deeper research on sentiment analysis of people who have used the ChatGPT application is made. It is hoped that this research will be able to collect data on public responses to ChatGPT, both pros and cons. Research data will be taken from reviews of ChatGPT application users in the Play Store. Google Collaboratory with Google Play Scraper will be used during the data collection process. The data that has been obtained will go through a preprocessing stage to be cleaned. After the data is successfully cleaned, the data will go through the process of labeling positive and negative data, and will be classified through the Naïve Bayes method. The study results show that the application of the Naïve Bayes method is able to classify user sentiment using Confusion Matrix with a percentage accuracy value of 94.05%, a percentage precision value of 95% for positive and 81.25% for negative. Then the percentage of recall value for positive is 98.84%, and for negative is 48%.

**Keywords:** Naïve Bayes; Confusion Matrix; ChatGPT; Play Store; Analysis Sentiment

## 1. PENDAHULUAN

Di era di mana teknologi digital mengalami pertumbuhan yang pesat, terdapat peningkatan para pengguna aplikasi *mobile*. Hal ini mendorong pentingnya pemahaman terhadap tingkat kepuasan dan keluhan pengguna dalam memakai sebuah aplikasi. Google menghadirkan Play Store, *platform* dengan berbagai fitur dan jenis konten, seperti *game*, film, buku, dan berbagai aplikasi, dengan variasi kategori sesuai keinginan *user*. Satu fitur yang paling menonjol di Play Store adalah sistem *review* dan rating, dimana pengguna dapat memberikan ulasan tanggapan mereka terhadap aplikasi yang mereka gunakan [1].

Aplikasi populer yang saat ini sering digunakan adalah ChatGPT, Aplikasi ini merupakan program komputer yang dirancang untuk bertukar pesan teks dan suara dengan manusia. Dilengkapi AI (*Artificial Intelligence*) dan NLP (*Natural Language Processing*), aplikasi ini menjelma menjadi asisten pintar yang mampu menjawab berbagai pertanyaan pengguna [2]. ChatGPT, keluaran OpenAI yang revolusioner, merupakan perpaduan teknologi informasi, komunikasi, dan kecerdasan buatan yang luar biasa. Didasari oleh teknologi pemrosesan bahasa alami (NLP), ChatGPT mampu berdialog layaknya manusia, menghasilkan tulisan baku atau ilmiah dengan kualitas yang baik, dan mengikuti panduan yang diberikan dengan presisi dan efektif [3].

Akademisi memiliki perdebatan tentang kecerdasan chatGPT. Karena ChatGPT mudah untuk membuat jurnal dan artikel ilmiah yang bagus, ada kekhawatiran tentang kemungkinan ketergantungan berlebihan dan penurunan minat menulis karya ilmiah. Untuk memahami sentimen masyarakat terhadap teknologi ini, penelitian tentang analisis sentimen masyarakat diperlukan. Analisis sentimen merupakan proses yang digunakan ketika memahami, mengekstrak, dan memproses data berbasis *text*, umumnya berfungsi untuk mendapatkan informasi terkait hasil sentimen yang terkandung dalam suatu opini [4]. Tujuan mengenai analisis sentimen ini untuk mengetahui evaluasi terhadap suatu isu dari perspektif orang lain, umumnya terkait dengan apakah mereka memiliki

pandangan positif atau negatif terhadap suatu permasalahan [5]. Teknik klasifikasi yang melibatkan persiapan model untuk menggambarkan data dan mengklasifikasikannya ke dalam kategori yang tersedia, menjadi kunci dalam analisis ini. Naïve Bayes, salah satu algoritma klasifikasi data, dipilih karena keunggulannya dalam kecepatan dan akurasi, menjadikannya salah satu metode terbaik untuk klasifikasi data dengan tingkat akurasi dan juga perhitungannya yang tinggi [6].

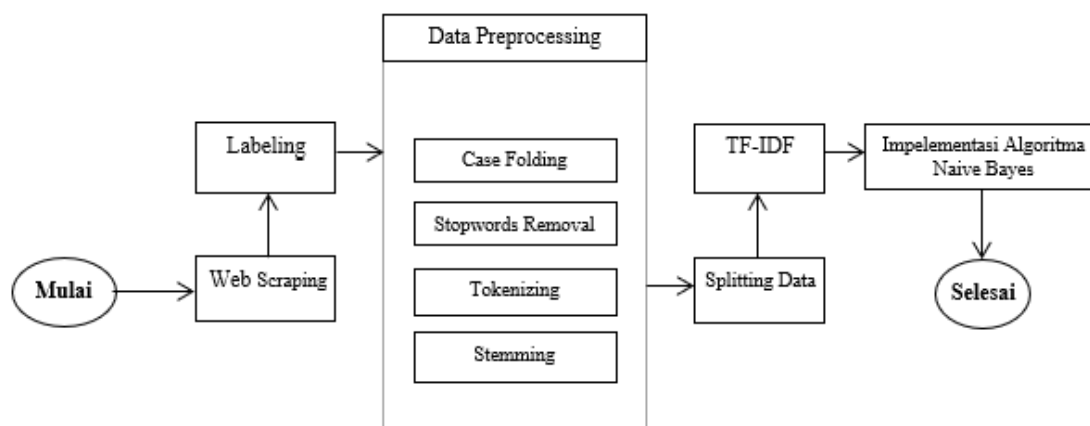
Studi sebelumnya yang dilakukan oleh M. I. Rifaldi, dkk (2023) dengan melakukan analisis sentimen pengguna Twitter pada aplikasi ChatGPT menggunakan naïve bayes, mengumpulkan sebanyak 1000 data menggunakan *API Twitter* RapidMiner dan menghasilkan 301 komentar dengan sentimen positif dan negatif. Akurasi yang didapatkan sebesar 80%, ketepatan sebesar 80,95%, dan nilai *recall* sebesar 89,47% dihasilkan dari pengujian penelitian tersebut menggunakan *Confusion Matrix* [7]. Selain itu, penelitian lain oleh D. Transiska, dkk (2024) yang mengumpulkan 300 data dari Twitter mengenai penggunaan aplikasi ChatGPT dengan menggunakan RapidMiner. Data tersebut kemudian dibagi 80:20 untuk data uji dan data latih. Dengan menggunakan *Confusion Matrix*, penelitian tersebut menemukan persentase akurasi 96,55%, persentase presisi 89,19%, dan persentase *recall* 95,18% [8]. Selanjutnya, penelitian lain oleh N. Agustina, dkk (2022) yang menggunakan metode *scrapping web* pada Google Colab untuk aplikasi Shopee, menghasilkan 1000 data ulasan terbaru pada tahun 2021. Selanjutnya, melakukan pembagian data sebesar 80:20 untuk data pelatihan dan uji coba, menghasilkan hasil ulasan yang cenderung positif. Setelah penerapan Naïve Bayes didapatkan bahwa persentase nilai akurasi dengan metode *Hold-Out* sebesar 83%, persentase nilai ketepatan sebesar 83%, persentase nilai *recall* sebesar 100%, dan persentase nilai *f1-score* sebesar 91% [9]. Studi selanjutnya oleh E. Daryfayi, dkk (2020) mengambil sebanyak 1500 ulasan dari sepuluh aplikasi di Play Store, yaitu Lazada, Ruang Guru, Gojek, Grab, Spotify, Tokopedia, Linkaja, Kredivo, LINE Webtoon, dan Bukalapak. Persentase nilai hasil penelitian menggunakan algoritma MNB (Multinomial Naive Bayes) adalah 78,9% [10]. Yang terakhir, penelitian oleh Santoso dan Wibowo (2022) dengan mengambil ulasan pengguna aplikasi Buzzbreak dengan membagi data sebanyak 3436 ulasan menjadi data pelatihan dan data pengujian, masing-masing dengan perbandingan 90:10, 80:20, dan 70:30. Hasil penelitian menunjukkan bahwa klasifikasi sentimen menggunakan metode *naïve bayes classifier* pada data pelatihan memiliki *split ratio* 70%:30% dengan persentase nilai AUC sebesar 76,52% dan pada data testing *split ratio* 90%:10% nilai AUC sebesar 79,34% dengan kesimpulan bahwa tingkat akurasinya sedang [11]. Dari hasil-hasil penelitian di atas menjelaskan bahwa penerapan metode Naïve Bayes menghasilkan tingkat akurasi dan presisi yang cukup tinggi. Oleh karena itu peneliti nantinya akan menggunakan metode tersebut dalam analisis sentimen pada ulasan pengguna.

Dari contoh penelitian-penelitian sebelumnya yang memiliki objek dan metode pengambilan data yang beragam, seperti Twitter, RapidMiner, aplikasi Buzzbreak, peneliti akan menggunakan metode algoritma Multinomial Naïve Bayes dengan memanfaatkan Google Colab yang akan digunakan dalam penelitian ini pada proses analisis sentimen untuk aplikasi *chatbot* ChatGPT. Tujuan penelitian ini diharapkan dapat memberikan informasi tentang sentimen masyarakat, khususnya dari pengguna aplikasi ChatGPT, dengan mengumpulkan data dari ulasan yang diberikan pada kolom ulasan aplikasi.

## 2. METODOLOGI PENELITIAN

### 2.1 Alur Penelitian

Penelitian ini menggunakan data yang diperoleh dari Google Play Store menggunakan Google Colab dengan *web scrapping* menggunakan *python* pada aplikasi ChatGPT. Data yang diambil akan diklasifikasikan menggunakan nilai sentimen dengan algoritma *Naïve Bayes*. Berikut Gambar 1 yang merupakan kerangka alur penelitian.



**Gambar 1.** Tahapan Penelitian

Berikut adalah penjelasan lebih lanjut tentang proses penelitian yang terdapat pada Gambar 1, tahapan-tahapan penelitian ini antara lain:

**a. Web Scraping**

Merupakan sebuah proses mengumpulkan data dalam bentuk ulasan pada kolom *review* di Google Play Store. Proses ini dilakukan dengan menggunakan bahasa pemrograman Python.

**b. Labeling**

*Labeling* merupakan pemberian label pada data yang telah dikumpulkan. Label pada penelitian ini berperan penting dalam mengkategorikan data berdasarkan sentimen yang terkandung di dalamnya. Dua kategori label yang digunakan adalah label bersifat positif untuk data yang mengandung sentimen positif dan label bersifat negatif untuk data yang mengandung sentimen negatif [12].

**c. Data Preprocessing**

*Data Preprocessing* merupakan tahapan proses persiapan dalam mengolah data, nantinya data tersebut akan melalui beberapa tahapan lain seperti: *cleaning*, *case folding*, *stopword removal*, *tokenize*, dan *stemming*. *Data Preprocessing* pada penelitian ini akan menggunakan python pada Google Colab.

**d. Splitting Data**

*Splitting Data* merupakan tahapan membagi data ke dalam bentuk data latih dan data uji. Pada penelitian ini data dibagi dengan menggunakan rasio perbandingan 80%:20%.

**e. Transformation (TF-IDF)**

Merupakan sebuah proses transformasi data yang dikenal sebagai *Term Frequency-Inverse Document Frequency (TF-IDF)* dengan menghitung bobot dari setiap kata.

**f. Implementasi Algoritma Naïve Bayes**

Metode *Naïve Bayes* akan menghitung probabilitas suatu kelas berdasarkan nilai fitur-fiturnya, nantinya model ini akan menggunakan *Confusion Matrix* dalam mengevaluasi performa model yang digunakan.

**2.2 Web Scraping**

*Web scraping* merupakan sebuah teknik yang digunakan dalam mengumpulkan data dari internet. Salah satu penerapannya adalah dalam proses pengumpulan ulasan pengguna dari Google Play Store menggunakan *Python*. Pengambilan ulasan pada Google Play Store akan menggunakan *google-play-scraper* yang pertama-tama harus terinstal menggunakan *pip* pada *python*. Data diambil sebanyak 1430 ulasan yang *most relevant* pada aplikasi ChatGPT. Ulasan yang diambil semuanya menggunakan bahasa Indonesia.

**2.3 Data Preprocessing**

*Preprocessing* adalah tahap dimana data diproses sehingga nantinya akan siap untuk diproses lebih lanjut pada pengolahan berikutnya. Tahap ini membersihkan data dari kata-kata yang tidak memiliki arti [13]. Adapun tahapan yang ada di dalam prosesnya yaitu:

- Case Folding* adalah proses mengubah semua *text* atau *sentences* dalam data menjadi huruf kecil secara keseluruhan (*lower case*). Nantinya karakter huruf besar dalam data menjadi karakter kecil dari 'a' hingga 'z'[14]. Proses bertujuan memudahkan ketika nantinya menyesuaikan dokumen dengan ukuran huruf yang sama. [6]
- Stopword removal* atau *filtering* merupakan proses dalam menghapus kata-kata yang dianggap tidak penting. [15]. Proses ini bertujuan untuk menghapus data yang tidak utuh, data yang mengandung kesalahan, dan kata-kata yang tidak relevan [16].
- Tokenizing*, juga dikenal sebagai 'tokenisasi', adalah sebuah teknik untuk memecah kalimat menjadi kata per kata atau memotong urutan string menjadi potongan kata yang jauh lebih sederhana [17].
- Stemming* merupakan sebuah proses dengan bertujuan mengurangi kata berimbuhan menjadi ke bentuk dasarnya dengan menghilangkan imbuhan [16] Algoritma *stemming* yang digunakan berbeda-beda untuk setiap bahasa [18].

**2.4 TF-IDF**

Setelah proses *preprocessing text*, langkah selanjutnya adalah pembobotan teks untuk mendapatkan nilai yang dapat digunakan dalam klasifikasi [19]. Metode TF-IDF (Term Frequency-Inverse Document Frequency) nantinya akan digunakan untuk melakukan penelitian ini dengan memberikan nilai bobot pada setiap kata yang ada di dalam dokumen [20]. Penggunaan TF-IDF akan membantu meningkatkan akurasi klasifikasi teks dengan memberikan nilai bobot yang lebih tinggi pada kata-kata yang relevan dan informatif, sementara nilai bobot yang lebih rendah pada kata-kata umum yang tidak memiliki makna yang signifikan [4].

**2.5 Naïve Bayes**

Dengan asumsi dasar kemandirian antar setiap peristiwa atau kondisi, algoritma *Naive Bayes* merupakan salah satu algoritma yang populer dalam proses *text mining*. Asumsi ini memungkinkan *Naive Bayes* untuk bekerja secara efisien dan efektif dalam mengklasifikasikan data teks, terutama ketika data tersebut memiliki banyak fitur atau kategori [21]. Penelitian ini akan menggunakan *Multinomial Naive Bayes (MNB)* sebagai metode klasifikasi yang

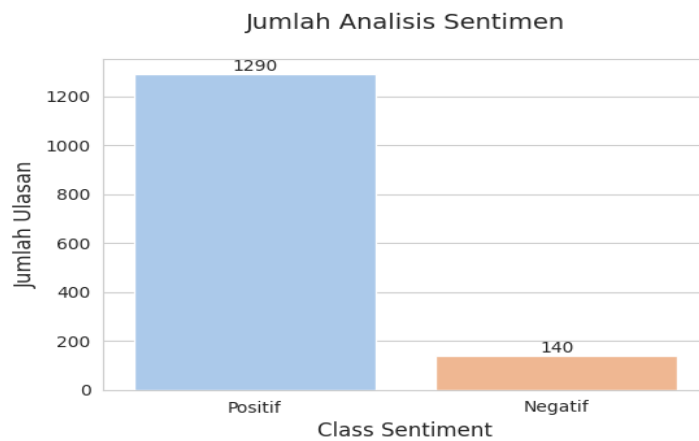


Setelah *code* pada Gambar 3 berhasil dijalankan, output selanjutnya dapat dilihat pada Tabel 2 yang akan menunjukkan hasil pelabelan sentimen sebagai berikut:

Tabel 2. Hasil Labeling

Content	Score	Label
ChatGPT versi app jelek amat, gak sebaik versi web. Jawaban di aplikasi cenderung pada kesimpulan dan jawaban singkat, gk sedetail dan seinformatif versi web.	1	Negatif
Sangat membantu dibikin aplikasi seperti ini, jadi tidak perlu repot-repot untuk buka browser. Sayangnya di aplikasi ini, dalam memberi jawaban tidak disertai enter	4	Positif

Dari hasil *labeling* yang dilakukan, hasil sentiment dari ulasan-ulasan tersebut menghasilkan total data 1430 yang dapat divisualisasikan ke dalam gambar perbandingan antara ulasan negatif dan positif menggunakan *wordcloud*. Hasil dari visualisasi perbandingan tersebut dapat dilihat pada Gambar 4 sebagai berikut:



Gambar 4. Visualisasi Hasil Labeling

Dapat terlihat dari Gambar 4, total data berjumlah 1430 dengan terdapat 1290 ulasan yang memiliki nilai positif dan hanya 140 ulasan yang bernilai negatif. Terdapat 70 total data lain yang kemungkinan masih bernilai *null* pada penelitian ini, nilai bintang 3 masih dikatakan rancu. Data tersebut dapat mengarah ke positif dan bisa juga ke arah sebaliknya.

### 3.3 Data Preprocessing

Tahapan selanjutnya adalah *text processing* dimana data kemudian akan diproses untuk menghilangkan kata-kata yang tidak diperlukan, adapun tahapan dari *preprocessing* tersebut sebagai berikut:

#### 3.3.1 Case Folding

Mengubah bentuk huruf secara keseluruhan menjadi huruf yang lebih kecil dikenal sebagai *case folding*. Contoh hasil penerapannya dapat dilihat pada Tabel 3 sebagai berikut:

Tabel 3. Hasil Penerapan Case Folding

Content	Score	Label	text_clean
ChatGPT versi app jelek amat, gak sebaik versi web. Jawaban di aplikasi cenderung pada kesimpulan dan jawaban singkat, gk sedetail dan seinformatif versi web.	1	Negatif	chatgpt versi app jelek amat, gak sebaik versi web. Jawaban di aplikasi cenderung pada kesimpulan dan jawaban singkat, gk sedetail dan seinformatif versi web.
Sangat membantu dibikin aplikasi seperti ini, jadi tidak perlu repot-repot untuk buka browser. Sayangnya di aplikasi ini, dalam memberi jawaban tidak disertai enter	4	Positif	sangat membantu dibikin aplikasi seperti ini jadi tidak perlu repot-repot untuk buka browser sayangnya di aplikasi ini dalam memberi jawaban tidak disertai enter

#### 3.3.2 Stopwords

Untuk menjamin kualitas data yang digunakan, proses penting dalam analisis data untuk tahapan selanjutnya setelah *case folding* pada penelitian ini merupakan proses *filtering* atau *stopword removal*. Tujuannya untuk menghapus kata-kata yang mengandung kesalahan, dan kata-kata yang tidak relevan. Contoh penerapan proses *filtering* tersebut dapat dilihat pada Tabel 4 sebagai berikut:

Tabel 4. Hasil Penerapan Filtering

text_clean	text_StopWord
chatgpt versi app jelek amat, gak sebaik versi web. Jawaban di aplikasi cenderung pada kesimpulan dan jawaban singkat, gk sedetail dan seinformatif versi web. sangat membantu dibikin aplikasi seperti ini jadi tidak perlu repot-repot untuk buka browser sayangnya di aplikasi ini dalam memberi jawaban tidak disertai enter	chatgpt versi app jelek gak versi web aplikasi cenderung kesimpulan singkat gk sedetail seinformatif versi web sangat membantu dibikin aplikasi ini jadi tidak repot repot buka browser sayangnya aplikasi tidak disertai enter

### 3.3.3 Tokenizing

Proses tokenisasi merupakan langkah penting dalam pengolahan data teks. Pada proses ini, teks dipecah menjadi bagian-bagian kecil yang disebut token, yaitu unit-unit terkecil yang bermakna, seperti kata, angka, atau tanda baca. Hasil dari proses tokenisasi dapat digambarkan dalam tabel seperti Tabel 5 berikut:

Tabel 5. Hasil Penerapan Stopword Removal

text_StopWord	text_tokens
chatgpt versi app jelek gak versi web aplikasi cenderung kesimpulan singkat gk sedetail seinformatif versi web sangat membantu dibikin aplikasi ini jadi tidak repot repot buka browser sayangnya aplikasi tidak disertai enter	chatgpt,versi,app,jelek,gak,versi,web,aplikasi,cenderung, kesimpulan, singkat, gk, sedetail, seinformatif, versi, web sangat, membantu, dibikin, aplikasi, jadi, tidak, repot, repot, buka, browser, sayangnya, aplikasi, tidak, disertai, enter

### 3.3.4 Stemming

Tujuan dari *stemming* adalah untuk menemukan kata dasar atau awal dari kata-kata yang memiliki *imbuhan*. Proses ini dilakukan dengan menghilangkan imbuhan, seperti prefiks, sufiks, dan infiks, sehingga menghasilkan kata dasar yang memiliki makna yang sama. Hasil dari proses *stemming* pada penelitian ini menggunakan *library* Sastrawi pada teks berbahasa Indonesia, adapun contohnya dapat dilihat pada Tabel 6 sebagai berikut:

Tabel 6. Hasil Penerapan Stemming

text_tokens	text_stemindo
chatgpt, versi, app, jelek, gak, versi, web, aplikasi, cenderung, kesimpulan, singkat, gk, sedetail, seinformatif, versi, web sangat, membantu, dibikin, aplikasi, jadi, tidak, repot, repot, buka, browser, sayangnya, aplikasi, tidak, disertai, enter	chatgpt versi app jelek gak versi web aplikasi cenderung simpul singkat gk detail informatif versi web chatgpt membantu jadi tidak repot repot buka browser sayang aplikasi tidak serta enter

## 3.4 Splitting Data

```

1 #membagi data menjadi data training dan testing dengan test_size = 0.20 dan random state nya 0
2 from sklearn.model_selection import train_test_split
3 X_train, X_test, y_train, y_test = train_test_split(data_clean['content'], data_clean['Label'],
4                                               test_size = 0.20,
5                                               random_state = 0)
    
```

Gambar 5. Tampilan Code Tahapan Splitting Data

Tahapan selanjutnya ialah membagi data ke dalam bentuk data latih dan data uji. Proses tersebut menggunakan bahasa pemrograman *Python* dengan menggunakan fungsi dari *train\_test\_split* pada *library sklearn.model\_selection* untuk membagi data menjadi data latih dan data uji. Berikut penjelasan rincinya dari gambar 5 di atas:

- from sklearn.model\_selection import train\_test\_split: Baris ini mengimpor *train\_test\_split* dari *library sklearn*. Fungsinya biasa digunakan dalam *machine learning* ketika memisahkan kumpulan data ke bentuk *set* pelatihan dan *set* pengujian.
- X\_train, X\_test, y\_train, y\_test = train\_test\_split(data\_clean['konten'], data\_clean['Label'], test\_size=0.20, random\_state=0): Baris ini melakukan pembagian data yang sebenarnya.
- data\_clean['konten']: Ini mewakili fitur atau variabel independen dari kumpulan data.
- data\_clean['Label']: Ini mewakili variabel target atau label dalam kumpulan data.
- test\_size=0.20: Argumen ini menentukan berapa banyak data yang akan dibagi ke dalam *set* pengujian. Dalam hal ini, dua puluh persen, atau 0,20 dari data akan digunakan untuk pengujian, dan sisa delapan puluh persen akan digunakan untuk pelatihan.
- random\_state=0: Argumen ini menetapkan *seed* untuk *generator* angka acak yang digunakan untuk mengacak data sebelum dibagi. Menetapkan value dari *seed* akan memastikan pembagian data dapat direproduksi jika kode dijalankan beberapa kali.

### 3.5 TF-IDF

Penelitian ini menerapkan metode TF-IDF dalam menentukan bobot nilai untuk kata-kata pada dokumen, kemudian menghasilkan vektor yang merepresentasikan setiap kata dalam dokumen. Dalam penerapannya terdapat dua langkah yang digunakan, seperti TF-IDF *vectorizer* dan *count vectorizer*.

#### 3.5.1 TF-IDF Vectorizer

Suatu metode transformasi yang dikenal sebagai *Term Frequency-Inverse Document Frequency* (TF-IDF) merupakan metode yang akan memberikan nilai pada setiap kata yang berkaitan dengan dokumen. Berikut tampilan *code* yang digunakan dalam tahapan TF-IDF pada penelitian ini yang dapat dilihat pada Gambar 6 sebagai berikut:

```

1 from sklearn.feature_extraction.text import TfidfVectorizer
2
3 tfidf_vectorizer = TfidfVectorizer()
4 tfidf_train = tfidf_vectorizer.fit_transform(X_train)
5 tfidf_test = tfidf_vectorizer.transform(X_test)
6
7 print(X_train.shape)
8 print(y_train.shape)
9 print(X_test.shape)
10 print(y_test.shape)

```

**Gambar 6.** Tampilan Code Tahapan TF-IDF

Setelah *code* pada Gambar 6 berhasil dijalankan, output keseluruhan rincian tahapan TF-IDF dapat dilihat sebagai berikut:

- Impor: Memuat class `TfidfVectorizer` yang digunakan untuk transformasi TF-IDF.
- Inisialisasi TF-IDF Vectorizer: Membuat *instance* dari class `TfidfVectorizer` untuk menyiapkan "alat" transformasi.
- Transformasi Data Training:
  - Melatih model TF-IDF vectorizer menggunakan *data training* `X_train`. Ini termasuk menghitung bobot TF-IDF pada setiap kata.
  - Mengubah data training `X_train` menjadi representasi numerik menggunakan model TF-IDF yang sudah dilatih. Representasi ini disimpan dalam variabel `tfidf_train`.
- Transformasi Data Testing: Mengubah *data testing* `X_test` menjadi representasi numerik menggunakan model TF-IDF yang sudah dilatih pada *data training*. Representasi ini disimpan dalam variabel `tfidf_test`.
- Mencetak Bentuk Data: Mencetak bentuk dari *data training* dan *testing* sebelum dan sesudah transformasi, serta bentuk dari label *training* dan *testing*.

Dengan menggunakan transformasi TF-IDF, kode ini berfungsi dalam menyiapkan data teks agar dapat diproses dan dianalisis lebih lanjut pada tahap selanjutnya.

#### 3.5.2 Count Vectorizer

```

1 from sklearn.feature_extraction.text import CountVectorizer
2
3 vectorizer = CountVectorizer()
4 vectorizer.fit(X_train)
5
6 X_train = vectorizer.transform(X_train)
7 X_test = vectorizer.transform(X_test)

```

**Gambar 7.** Tampilan Code Count Vectorizer

*Code* pada Gambar 7 di atas merupakan proses yang disebut *Vector Bag-of-Words* (BOW) pada data teks dengan menggunakan class `CountVectorizer` dari library `sklearn.feature_extraction.text`. Berikut penjelasan detailnya dari Gambar 7 sebagai berikut:

- Impor: `from sklearn.feature_extraction.text import CountVectorizer`: Baris ini mengimpor class `CountVectorizer`, yang secara khusus dirancang untuk membuat representasi *bag-of-words* dari data teks.
- Inisialisasi `CountVectorizer`: `vectorizer = CountVectorizer()`: Baris ini membuat *instance* dari class `CountVectorizer`. Parameter yang digunakan dalam penelitian adalah parameter *default*.
- Pelatihan Vectorizer (Fit): `vectorizer.fit(X_train)`: Baris ini melatih (fit) `CountVectorizer` pada data training `X_train`. Selama proses ini, `vectorizer` menganalisis kosakata kata yang ada dalam *data training*. Ini pada dasarnya membangun *dictionary* berisi kata-kata unik yang ditemukan dalam kumpulan *data training*.
- Transformasi Data Training: `X_train = vectorizer.transform(X_train)`: Baris ini melakukan transformasi *bag-of-words* yang sebenarnya pada *data training* `X_train` menggunakan `vectorizer` yang sudah dilatih.

Potongan *code* dari Gambar 7 akan berfungsi dalam menyiapkan data teks dengan mengubahnya menjadi representasi numerik berdasarkan jumlah kata yang ada.

### 3.6 Algoritma Naïve Bayes

Pengklasifikasian yang merupakan tahapan selanjutnya setelah tahap transformasi TF-IDF adalah dengan melakukan penerapan algoritma Naïve Bayes. Proses ini menggunakan pembagian 80% data pelatihan dan 20% data pengujian. Data yang telah dibagi akan diuji menggunakan *confusion matrix* menggunakan *python*. *Code* dibuat dengan melakukan impor menggunakan beberapa fitur seperti: *TfidfVectorizer* yang digunakan ketika mengubah data teks menggunakan proses TF-IDF, *MultinomialNB* untuk mengimplementasikan algoritma *Multinomial Naïve Bayes*, kemudian *accuracy\_score*, *precision\_score*, *recall\_score*, *f1\_score* yang digunakan untuk mengevaluasi kinerja model, dan *confusion matrix*. Berikut tampilan *code* dari penerapan algoritma *Naïve Bayes* yang dapat dilihat pada Gambar 8 sebagai berikut:

```

1 import pandas as pd
2 from sklearn.feature_extraction.text import TfidfVectorizer
3 from sklearn.naive_bayes import MultinomialNB
4 from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
5 from sklearn.model_selection import train_test_split
6 from sklearn.metrics import classification_report
7 from sklearn.metrics import confusion_matrix
8
9 clf = MultinomialNB()
10 clf.fit(X_train, y_train)
11 predicted = clf.predict(X_test)
12
13 print("MultinomialNB Accuracy:", accuracy_score(y_test,predicted))
14 print("MultinomialNB Precision:", precision_score(y_test,predicted, average="binary", pos_label="Negatif"))
15 print("MultinomialNB Recall:", recall_score(y_test,predicted, average="binary", pos_label="Positif"))
16 print("MultinomialNB f1_score:", f1_score(y_test,predicted, average="binary", pos_label="Negatif"))
17
18 print(f'confusion_matrix:\n {confusion_matrix(y_test, predicted)}')
19 print('-----\n')
20 print(classification_report(y_test, predicted, zero_division=0))
21
22 # Load dataset
23 data_clean = pd.read_csv('hasil_TextPreProcessing_chatgpt.csv')
24
25 #

```

Gambar 8. Tampilan Code Penerapan Algoritma Naïve Bayes

Setelah *code* pada Gambar 8 berhasil dijalankan, penjelasan rinci penggunaan *code* tersebut dapat dilihat sebagai berikut:

- Inisialisasi Model: `clf = MultinomialNB()`: Membuat *instance* dari klasifikasi *Multinomial Naïve Bayes*.
- Pelatihan Model: `clf.fit(X_train, y_train)`: Melatih klasifikasi pada *data training X\_train* (fitur) dan *y\_train* (label).
- Prediksi: `predicted = clf.predict(X_test)`: Menggunakan model terlatih untuk memprediksi label untuk *data testing* yang belum dilihat *X\_test*.
- Evaluasi:
  - `accuracy_score`: Akurasi keseluruhan model.
  - `precision_score`: Mengukur proporsi positif benar di antara prediksi positif (mempertimbangkan kelas "Negatif").
  - `recall_score`: Mengukur proporsi positif aktual yang diidentifikasi model dengan benar (mempertimbangkan kelas "Positif").
  - `f1_score`: Skor rata-rata harmonis untuk *recall* dan presisi.
  - Mencetak *confusion matrix* dalam memvisualisasikan distribusi prediksi yang benar dan salah untuk setiap kelas.
  - Mencetak laporan klasifikasi, memberikan informasi detail tentang kinerja model untuk setiap kelas.
- Memuat Dataset: Bagian yang dikomentari (`# Load dataset`) menunjukkan dalam memuat data *preprocessing* pada file CSV bernama "hasil\_TextPreProcessing\_chatgpt.csv" menggunakan *pandas*.

Hasil *output* ketika *code* pada Gambar 8 berhasil dijalankan akan ditunjukkan keseluruhannya pada Gambar 9 sebagai berikut :

```

MultinomialNB Accuracy: 0.9405594405594405
MultinomialNB Precision: 0.8125
MultinomialNB Recall: 0.9884169884169884
MultinomialNB f1_score: 0.6046511627906976
confusion_matrix:
[[ 13  14]
 [   3 256]]
-----
              precision    recall  f1-score   support

 Negatif      0.81      0.48      0.60         27
 Positif      0.95      0.99      0.97        259

 accuracy          0.94         286
 macro avg         0.88         286
 weighted avg      0.94         286

```

Gambar 9. Hasil Akurasi, Presisi, Recall, dan f1-score



Berdasarkan Gambar 9 tersebut, hasil klasifikasi menggunakan algoritma Naive Bayes membagi data menjadi rasio 80:20, dengan 80 persen data latih dan 20 persen data uji. Secara keseluruhan, ini menghasilkan persentase tingkat akurasi sebesar 94%, dengan nilai precision sebesar 81% untuk hasil negatif, dan nilai presisi sebesar 95% untuk hasil positif. Hasil persentase nilai *recall* untuk negatif sebesar 48% dan bernilai 99% untuk hasil positif. Lalu persentase nilai *f1-score* untuk negatif sebesar 60% dan bernilai 97% untuk hasil positif.

#### 4. KESIMPULAN

Dari hasil penelitian dengan menggunakan data bersih sebanyak 1430 ulasan dari bulan Juli 2023 hingga bulan Juni 2024 pada aplikasi ChatGPT di Google Play Store dengan teknik *web scraping* dan melalui proses *preprocessing* dalam mengolah data tersebut menggunakan proses *case folding*, *stopword removal* atau *filtering*, *tokenizing*, dan proses terakhir *stemming* menggunakan bahasa pemrograman *python*. Penelitian ini membagi data ke dalam bentuk pelatihan dan pengujian dengan rasio perbandingan 80 persen untuk data latih dan 20 persen untuk data uji. Setelah itu melakukan *labeling* dengan hasil berupa 1290 ulasan bersifat positif dan sebanyak 140 bersifat negatif. Dari hasil ini, dapat diketahui bahwa sentimen masyarakat terhadap penggunaan ChatGPT di Google Play Store bernilai positif. Dari hasil pengujian didapatkan hasil tingkat akurasi sebesar 94% dengan nilai presisi 81% untuk negatif dan 95% untuk positif dengan menggunakan algoritma Naive Bayes. Hasil nilai *recall* sebesar 48% untuk negatif dan untuk positif sebesar 99%. Lalu nilai *f1-score* untuk negatif sebesar 60% dan 97% untuk positif. Tingkat nilai akurasi yang diperoleh menunjukkan bahwa algoritma *Naive Bayes* sangat akurat, meskipun nilai *recall* dan nilai *f1-score* cukup rendah dibandingkan dengan nilai untuk data positif. Sebagai saran untuk penelitian selanjutnya, algoritma yang digunakan dapat diubah seperti menggunakan *Support Vector Machines (SVM)* atau *Random Forest* untuk melihat perbandingan algoritma yang paling baik. Data yang digunakan juga dapat ditingkatkan agar model analisis dapat lebih presisi.

#### REFERENCES

- [1] R. Wahyudi and G. Kusumawardana, "Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine," *J. Inform.*, vol. 8, no. 2, pp. 200–207, 2021.
- [2] mukrodin mukrodin and N. Mega Sasmitha, "Artificial Intelligence Dalam Aplikasi Chatbot Sebagai Helpdesk Obyek Wisata Dengan Permodelan Natural Language Processing (Studi Kasus: Kabupaten Cilacap)," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 10, no. 1, pp. 7–14, 2021.
- [3] A. Setiawan and U. K. Luthfiyani, "Penggunaan ChatGPT Untuk Pendidikan di Era Education 4.0: Usulan Inovasi Meningkatkan Keterampilan Menulis," *J. PETISI*, vol. 04, no. 01, pp. 49–58, 2023.
- [4] E. Y. Hidayat, R. W. Hardiansyah, and A. Affandy, "Analisis Sentimen Twitter untuk Menilai Opini Terhadap Perusahaan Publik Menggunakan Algoritma Deep Neural Network," *J. Nas. Teknol. dan Sist. Inf.*, vol. 7, no. 2, pp. 108–118, 2021.
- [5] B. Nugraha, "Metode Klasifikasi Analisis Sentimen pada Media Sosial," *Syntax J. Inform.*, vol. 9, no. 2, pp. 119–123, 2020.
- [6] E. Undamayanti *et al.*, "Analisis Sentimen Menggunakan Metode Naive Bayes Berbasis Particle Swarm Optimization Terhadap Pelaksanaan Program Merdeka Belajar Kampus Merdeka," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 6, no. 2, pp. 916–930, 2022.
- [7] M. I. Rifaldi, Y. R. Ramadhan, and I. Jaelani, "Analisis Sentimen Terhadap Aplikasi Chatgpt Pada Twitter Menggunakan Algoritma Naive Bayes," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 7, no. 2, pp. 802–814, 2023.
- [8] D. Transiska, D. Febriawan, and F. N. Hasan, "Analisis Sentimen Terhadap Penggunaan Chatgpt Berdasarkan Twitter Menggunakan Algoritma Naive Bayes," *MIB Media Inform. Budidarma*, vol. 8, no. April, pp. 1077–1086, 2024.
- [9] N. Agustina, D. H. Citra, W. Purnama, C. Nisa, and A. R. Kurnia, "Implementasi Algoritma Naive Bayes untuk Analisis Sentimen Ulasan Shopee pada Google Play Store," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 2, no. 1, pp. 47–54, 2022.
- [10] E. Daryfayi, P. Daulay, and I. Asror, "Sentimen Analisis pada Ulasan Google Play Store Menggunakan Metode Naive Bayes," *e-Proceeding Eng.*, vol. 7, no. 2, pp. 8400–8410, 2020.
- [11] D. P. Santoso and W. Wibowo, "Analisis Sentimen Ulasan Aplikasi Buzzbreak Menggunakan Metode Naive Bayes Classifier pada Situs Google Play Store," *J. Sains dan Seni ITS*, vol. 11, no. 2, 2022.
- [12] A. Nabillah, S. Alam, and M. G. Resmi, "Twitter User Sentiment Analysis Of TIX ID Applications Using Support Vector Machine Algorithm," *RISTEC Res. Inf. Syst. Technol.*, vol. 3, no. 1, pp. 14–27, 2022.
- [13] E. S. Basryah, A. Erfina, and C. Warman, "Analisis Sentimen Aplikasi Dompot Digital Di Era 4.0 Pada Masa Pandemi Covid-19 Di Play Store," *SISMATIK (Seminar Nas. Sist. Inf. dan Manaj. Inform.)*, pp. 189–196, 2021.
- [14] A. Rozaqi, A. Triayudi, and R. T. Aldisa, "Analisis Sentimen Vaksinasi Booster Berdasarkan Twitter Menggunakan Algoritma Naive Bayes dan K-NN," *J. Sist. Komput. dan Inform.*, vol. 4, no. 1, p. 184, 2022.
- [15] Y. Nurdiansyah, F. Rahman, and P. Pandunata, "Analisis Sentimen Opini Publik Terhadap Undang-Undang Cipta Kerja pada Twitter Menggunakan Metode Naive Bayes Classifier," *Pros. Semin. Nas. Sains Teknol. dan Inov. Indones.*, vol. 3, no. November, pp. 201–212, 2021.
- [16] F. F. Irfani, "Analisis Sentimen Review Aplikasi Ruangguru Menggunakan Algoritma Support Vector Machine," *JBMI (Jurnal Bisnis, Manajemen, dan Inform.)*, vol. 16, no. 3, pp. 258–266, 2020.
- [17] A. Riyani, M. Z. Naf'an, and A. Buhanudidin, "Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen," *J. Linguist. Komputasional 2019*, vol. 2, no. 1, pp. 49–54, 2019.
- [18] N. Herlinawati, Y. Yuliani, S. Faizah, W. Gata, and S. Samudi, "Analisis Sentimen Zoom Cloud Meetings di Play Store Menggunakan Naive Bayes dan Support Vector Machine," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 5, no. 2, p. 293,



- 2020.
- [19] D. Musfiroh, U. Khaira, P. Eko, P. Utomo, and T. Suratno, "Sentiment Analysis of Online Lectures in Indonesia from Twitter Dataset Using InSet Lexicon Analisis Sentimen terhadap Perkuliahan Daring di Indonesia dari Twitter Dataset Menggunakan InSet Lexicon," vol. 1, no. April, pp. 24–33, 2021.
  - [20] D. A. Wulandari, R. R. Saedudin, and R. Andreswari, "Analisis Sentimen Media Sosial Twitter Terhadap Reaksi Masyarakat Pada Ruu Cipta Kerja Menggunakan Metode Klasifikasi Algoritma Naive Bayes," *e-Proceeding Eng.*, vol. 8, no. 5, pp. 9007–9016, 2021.
  - [21] N. Q. Rizkina and F. N. Hasan, "Analisis Sentimen Komentar Netizen Terhadap Pembubaran Konser NCT 127 Menggunakan Metode Naive Bayes," *J. Inf. Syst. Res.*, vol. 4, no. 4, pp. 1136–1144, 2023.