

## Analisis Sentimen Pengguna Aplikasi *MyPertamina* Menggunakan Metode *Naïve Bayes* Berbasis Data Ulasan di *Play Store*

Rifkiansyah<sup>1</sup>, Santoso Setiawan<sup>2</sup>, Findi Ayu Sariasih<sup>3</sup>

<sup>1</sup>Universitas Bina Sarana Informatika, Jalan Kramat 98, Kota Jakarta Pusat Kode Pos. 10450, Indonesia

<sup>3</sup>Universitas Bina Sarana Informatika, Jalan Kali Abang Tengah No. 8, Kota Bekasi Kode Pos. 17122, Indonesia

### INFORMASI ARTIKEL

*Sejarah Artikel:*

Diterima Redaksi: 19-07-2025

Revisi Akhir: 21-08-2025

Diterbitkan Online: 10-09-2025

### KATA KUNCI

MyPertamina

Analisis Sentimen

Naïve Bayes

TF-IDF

Evaluasi Model

### KORESPONDENSI

E-mail: [rifkiansyah789@gmail.com](mailto:rifkiansyah789@gmail.com)

### ABSTRACT

*MyPertamina* merupakan aplikasi digital milik PT Pertamina yang dirancang untuk mendukung distribusi bahan bakar bersubsidi secara non-tunai dan tepat sasaran. Meskipun telah diunduh oleh jutaan pengguna, aplikasi ini menerima beragam tanggapan yang terekam dalam bentuk ulasan di *Google Play Store*. Penelitian ini dilakukan untuk mengetahui kecenderungan opini pengguna melalui pendekatan analisis sentimen otomatis. Pengambilan data ulasan dilakukan dengan memanfaatkan library *Google Play Scraper*, lalu diproses melalui tahapan preprocessing seperti normalisasi teks, pembersihan karakter, tokenisasi, penghapusan kata umum, dan stemming. Label sentimen positif dan negatif diberikan dengan bantuan model pra-latih *IndoBERT*. Proses selanjutnya mencakup konversi teks ke bentuk numerik dengan metode *TF-IDF* dan penerapan algoritma *Multinomial Naïve Bayes* untuk klasifikasi. Model diuji menggunakan *confusion matrix* dan validasi silang *10-Fold Cross Validation*. Hasil penelitian menunjukkan bahwa mayoritas ulasan pengguna bersifat negatif (63,65%), dan model klasifikasi berhasil mencapai akurasi sebesar 89,25%, presisi 87,5%, *recall* 82%, serta *F1-score* sebesar 84,69%. Hal ini menunjukkan bahwa pendekatan yang digunakan efektif dalam mengidentifikasi persepsi masyarakat terhadap layanan aplikasi *MyPertamina*.

## 1. PENDAHULUAN

Dalam industri bahan bakar di Indonesia, saat ini terdapat banyak perusahaan ritel yang bergerak, sebut saja Pertamina, Vivo, Shell, dan perusahaan-perusahaan lainnya. Sebagai Badan Usaha Milik Negara (BUMN) terbesar dalam segmen minyak dan gas di Indonesia, Pertamina diperintahkan oleh pemerintah untuk menyalurkan bahan bakar bersubsidi agar tepat sasaran dan sesuai dengan porsi yang telah ditetapkan. Bahan bakar bersubsidi adalah bahan bakar minyak yang memperoleh dukungan biaya dari pemerintah guna meringankan beban masyarakat berpenghasilan rendah. Namun tragisnya, program peruntukan BBM saat ini masih dianggap tidak tepat sasaran dan umumnya dinikmati oleh masyarakat mampu. Pada 1 Juli 2022, PT Pertamina (Persero) telah merealisasikan sebuah instrumen modern agar penyaluran bahan bakar subsidi tetap tepat sasaran sesuai ketentuan dan porsi yang telah dirancang. Melalui instrumen yang belum pernah digunakan sebelumnya ini, para pengguna bensin bersubsidi jenis Peralite dan Biosolar

diwajibkan untuk mendaftarkan kendaraannya di aplikasi *MyPertamina* [1].

PT Pertamina menciptakan platform layanan keuangan digital *MyPertamina*. Platform ini memiliki fungsi yang mirip dengan aplikasi seperti Dana dan OVO. Pengguna dapat membeli produk Pertamina, termasuk bahan bakar minyak (BBM), melalui *MyPertamina* dan menaunkannya ke dompet elektronik LinkAja mereka. Pembelian bahan bakar kemudian dapat dibayar dengan dana yang tersisa di akun LinkAja. Baik *Google Play Store* maupun *App Store* menawarkan aplikasi *MyPertamina* untuk diunduh [2].

Pada 16 April 2025, *MyPertamina* telah menerima lebih dari 10 juta unduhan dan peringkat 4,3 menurut data terbaru dari *Google Play Store*. Jika peringkat *Google Play Store* semakin rendah untuk sebuah aplikasi berarti pengguna masih belum puas dengan kinerjanya. Karena banyaknya ulasan pengguna yang tidak terstruktur dan tidak dikategorikan, maka diperlukan sebuah metode untuk mengumpulkan data evaluasi, yaitu dengan melakukan analisis sentimen pada kumpulan data ulasan pengguna.

Peneliti memutuskan untuk melakukan analisis sentimen setelah menemukan bahwa banyak dari rating yang ada di aplikasi *MyPertamina* tidak sesuai dengan rating yang diberikan. Analisis sentimen merupakan metode otomatis untuk memahami, mengambil, dan mengolah data teks guna mengidentifikasi muatan emosional atau sikap dalam suatu kalimat. Untuk mengetahui dominasi opini positif atau negatif dalam ulasan pengguna terhadap aplikasi *MyPertamina*, diperlukan penerapan teknik analisis opini [3].

## 2. TINJAUAN PUSTAKA

### 2.1. *MyPertamina*

PT Pertamina mengembangkan sebuah platform digital bernama *MyPertamina* untuk memudahkan transaksi pembelian produk mereka, termasuk bahan bakar minyak. Aplikasi ini memiliki fitur serupa dengan layanan dompet digital seperti Dana dan OVO. Pengguna dapat menautkan aplikasi ke akun LinkAja untuk melakukan pembayaran secara non-tunai. *MyPertamina* dapat diakses dan diinstal melalui *Google Play* maupun *App Store*, sehingga mudah diakses oleh masyarakat luas [2].

### 2.2. Analisis Sentimen

*Opinion mining* atau analisis sentimen, merupakan cabang ilmu yang berperan dalam mengenali opini, menilai persepsi, serta menggali ekspresi emosional dan sikap terhadap beragam objek seperti produk, jasa, individu, institusi, peristiwa, hingga karakteristik tertentu. Pada era awal 2000-an, kajian terkait analisis sentimen berkembang dengan cepat dan menempati posisi penting sebagai salah satu fokus utama dalam studi *natural language processing* [4].

### 2.3. *Google Play Scraper*

*Google Play Scraper* merupakan sebuah metode yang digunakan untuk mengekstraksi data dari platform *Google Play Store* tanpa memerlukan dependensi eksternal, dengan menggunakan bahasa pemrograman *Python*. Informasi yang dapat diambil meliputi detail aplikasi seperti nama aplikasi, URL pengembang, kategori, rating secara keseluruhan, ulasan, deskripsi, gambar thumbnail, tingkat rating konten, serta tangkapan layar aplikasi. Selain itu, melalui *API* ini juga dimungkinkan untuk mengumpulkan data ulasan pengguna, termasuk nama pengguna, foto profil, nilai rating, tanggal ulasan, jumlah suka pada komentar, hingga isi komentar yang diberikan [5].

### 2.4. *Python*

*Python* merupakan bahasa pemrograman interpretatif yang dapat dijalankan lintas platform dengan prinsip keterbacaan kode tinggi. Popularitasnya terus meningkat, terutama dalam bidang *Data Science*, *Machine Learning*, dan *Internet of Things (IoT)*, karena kemudahan penggunaan serta dukungan pustaka yang lengkap, bersifat *open source*, dan fleksibel. *Python* banyak dimanfaatkan untuk pembersihan data, visualisasi, hingga pengembangan model, sehingga menjadi pilihan utama bagi pengembang [6].

### 2.5. *Naïve Bayes*

*Naïve Bayes* merupakan metode yang tidak bergantung pada aturan khusus, melainkan memanfaatkan frekuensi atau jumlah kemunculan masing-masing kategori dalam data latih

untuk memperkirakan kemungkinan yang paling tinggi, dengan dasar perhitungan dari teori probabilitas. Algoritma ini termasuk dalam *supervised learning* dan umumnya digunakan untuk membangun basis data yang membutuhkan data latih dengan label. Metode ini digunakan dalam dua jenis pemodelan, yaitu regresi dan klasifikasi, khususnya dalam pengelompokan variabel seperti suhu panas atau dingin, kondisi sakit atau sehat, dan sebagainya. *Naïve Bayes* juga merupakan pendekatan probabilistik yang mengasumsikan bahwa antar fitur dalam suatu data tidak memiliki ketergantungan [7] [10] [11].

### 2.6. *Multinomial Naïve Bayes*

*Multinomial Naïve Bayes* merupakan metode klasifikasi teks yang menggunakan frekuensi kemunculan kata sebagai fitur utama. Model ini bekerja pada data diskrit dengan mengasumsikan dokumen sebagai kumpulan kata tanpa memperhatikan urutannya. Pendekatan ini banyak digunakan dalam analisis sentimen karena keberadaan kata tertentu dapat menjadi indikator arah sentimen, dengan distribusi *Multinomial* digunakan dalam pemrosesan kata [8] [12].

### 2.7. *TF-IDF*

Salah satu metode populer dalam mengubah data berbasis teks ke dalam bentuk numerik adalah melalui pendekatan pembobotan *TF-IDF*. Metode ini terdiri atas dua elemen utama, salah satunya adalah *Term Frequency (TF)*, yang berfungsi untuk menghitung seberapa sering sebuah kata muncul di dalam sebuah dokumen, serta *Inverse Document Frequency (IDF)*, yang menilai seberapa unik atau signifikan kata tersebut di antara seluruh dokumen dalam kumpulan data. Kata-kata yang memiliki relevansi tinggi dalam dokumen akan memperoleh bobot besar, sedangkan kata-kata umum yang kurang bermakna akan diberikan bobot yang lebih kecil [9].

### 2.8. *K-Fold Cross Validation*

*K-fold Cross Validation* merupakan salah satu teknik validasi yang umum digunakan untuk menilai kinerja model dalam *machine learning*. Teknik ini bekerja dengan membagi dataset menjadi  $k$  subset yang berukuran serupa untuk memastikan distribusi yang merata. Setiap iterasi dalam skema *k-fold* melibatkan pemilihan satu subset sebagai data pengujian, sedangkan sisa data dimanfaatkan untuk proses pelatihan model. Sebagai contoh, pada iterasi pertama, subset pertama ( $S_1$ ) digunakan sebagai data uji, sedangkan subset lainnya digunakan sebagai data pelatihan. Pada iterasi berikutnya, subset kedua ( $S_2$ ) menjadi data uji, dan sisanya menjadi data latih. Prosedur ini diulang hingga seluruh subset ( $S_k$ ) telah berperan sebagai data uji satu kali [10] [14] [15].

### 2.9. Penelitian Terdahulu

Penelitian yang dilakukan oleh [10], dilakukan sentimen pengguna terhadap aplikasi *MyPertamina*, sebuah layanan digital dari PT Pertamina untuk keperluan transaksi bahan bakar kendaraan. Ulasan pengguna sebanyak 5.722 ulasan diperoleh melalui proses web scraping dari *Google Play Store*. Proses analisis dilakukan menggunakan *Python* di platform *Google Colab*, dengan penerapan algoritma *Naïve Bayes Classifier (NBC)*. Dataset dibagi dengan proporsi 80% sebagai data pelatihan dan 20% sebagai data pengujian. Berdasarkan hasil evaluasi, mayoritas sentimen pengguna bersifat negatif, dan

performa model menunjukkan akurasi sebesar 87%, presisi 86%, *recall* 90%, serta *F1-score* sebesar 87%.

Penelitian sebelumnya [2], bertujuan untuk menganalisis sentimen pengguna terhadap aplikasi *MyPertamina*, sebuah platform keuangan digital besutan PT Pertamina yang memiliki fungsi serupa dengan aplikasi seperti OVO dan Dana. Metode klasifikasi yang digunakan dalam studi ini adalah *Naïve Bayes* dan *K-Nearest Neighbor (K-NN)*, dengan total 1.500 ulasan dari Google Play Store yang dikelompokkan ke dalam tiga jenis sentimen: positif, netral, dan negatif. Dari hasil pengujian, algoritma *Naïve Bayes* mencatat akurasi sebesar 75%, sedangkan performa *K-NN* bervariasi antara 56% hingga 73% tergantung pada konfigurasi parameter yang digunakan. Temuan ini menunjukkan bahwa *Naïve Bayes* lebih efektif dalam mengidentifikasi sentimen pengguna terhadap aplikasi *MyPertamina*.

Dalam penelitian yang dilakukan oleh [3] analisis sentimen terhadap ulasan pengguna aplikasi *MyPertamina* dilakukan karena ditemukan ketidaksesuaian antara nilai rating dan isi komentar pengguna, serta munculnya berbagai pendapat di media. Studi ini menerapkan pendekatan *text mining* melalui beberapa tahap, yakni *scraping* data, pelabelan sentimen, pembersihan teks, dan *preprocessing* yang mencakup transformasi teks, tokenisasi, serta penyaringan kata. Metode klasifikasi yang digunakan adalah *Naïve Bayes* dengan teknik pembobotan *TF-IDF*, sementara evaluasi kinerja model dilakukan melalui *confusion matrix* dan didukung dengan visualisasi *word cloud*. Penelitian ini memanfaatkan 3.948 ulasan dari *Google Play Store* dan menghasilkan temuan bahwa sebagian besar sentimen bersifat negatif. Model yang dibangun menunjukkan performa tinggi dengan akurasi 91%, presisi 92%, dan *recall* 100%.

### 3. METODOLOGI

Analisis sentimen dalam penelitian ini dilakukan menggunakan algoritma *Multinomial Naïve Bayes*, dengan sumber data berupa ulasan pengguna aplikasi *MyPertamina* yang diperoleh melalui platform *Google Play Store*. Proses penelitian ini dibagi ke dalam beberapa tahapan dan ditampilkan secara visual pada Gambar 1.



Gambar 1. Alur Analisis Data

#### 3.1. Pengumpulan Data

Proses awal dimulai dengan pengambilan data ulasan pengguna melalui pemanfaatan library *Google Play Scraper*. Informasi yang dikumpulkan meliputi tanggal ulasan, nilai rating, isi komentar, serta informasi relevan lainnya. Data ini diambil dari aplikasi *MyPertamina* dengan memperhatikan batas jumlah maksimum ulasan dan rentang waktu tertentu yang didukung oleh *library* tersebut.

#### 3.2. Text Preprocessing

Pada langkah ini data ulasan dibersihkan dan dipersiapkan agar dapat diolah secara optimal oleh algoritma analisis sentimen. Tahapan pra-pemrosesan dilakukan melalui lima proses berikut:

- a) *Case Folding*  
Seluruh teks diubah ke huruf kecil guna menjaga konsistensi dan mencegah perbedaan makna yang disebabkan oleh penggunaan huruf kapital.
- b) *Tokenizing*  
Memecah teks ulasan menjadi unit-unit kata (token) agar lebih mudah dianalisis secara otomatis.
- c) *Cleansing*  
Karakter yang tidak diperlukan seperti angka, simbol, tanda baca, serta spasi berlebih dihapus guna memperoleh teks yang bersih dan konsisten.
- d) *Filtering*  
Menyaring kata-kata yang bersifat umum dan kurang memberikan informasi penting, seperti kata sambung dan kata ganti, menggunakan daftar *stopword* Bahasa Indonesia.
- e) *Stemming*  
Setiap kata dikembalikan ke bentuk dasarnya menggunakan algoritma *stemming* Bahasa Indonesia, misalnya kata “bermain” menjadi “main”.

#### 3.3. Labeling

Dalam penelitian ini, pelabelan sentimen dilakukan secara otomatis menggunakan model pra-latih *w11wo/indonesian-roberta-base-sentiment-classifier*, yang merupakan adaptasi dari *IndoBERT/RoBERTa* untuk klasifikasi sentimen dalam Bahasa Indonesia. Meskipun model ini mendukung tiga kategori sentimen (positif, netral, negatif), penelitian ini hanya menggunakan dua label utama: positif dan negatif, guna menyederhanakan analisis dan memudahkan evaluasi terhadap persepsi pengguna aplikasi *MyPertamina*.

#### 3.4. Pembagian Data

Setelah tahap *preprocessing* dan pelabelan selesai, *dataset* dibagi menjadi dua *subset*: 80% data pelatihan dan 20% data pengujian. Tujuan pembagian ini adalah agar model dapat mempelajari pola dari data latih, lalu diuji menggunakan data yang belum dikenali untuk menilai kemampuan generalisasi model. Proses pembagian data dilakukan secara acak dengan memanfaatkan fungsi *train\_test\_split* dari pustaka *scikit-learn*, dengan mempertimbangkan keseimbangan proporsi kelas dalam kedua *subset*.

#### 3.5. Pembobotan TF-IDF

Setelah proses pembagian data, langkah selanjutnya adalah mengubah data teks menjadi bentuk numerik dengan

menggunakan metode *TF-IDF* (*Term Frequency-Inverse Document Frequency*). Metode ini berfungsi untuk menghitung bobot dari setiap kata dengan mempertimbangkan frekuensi kemunculan kata tersebut dalam satu dokumen dan tingkat kelangkaannya di keseluruhan dokumen. Melalui pendekatan ini, algoritma klasifikasi dapat lebih mudah mengidentifikasi kata-kata yang berkontribusi besar dalam membedakan isi antar dokumen.

### 3.6. Klasifikasi Menggunakan *Naïve Bayes*

Setelah proses pembobotan *TF-IDF*, langkah selanjutnya adalah proses klasifikasi menggunakan algoritma *Naïve Bayes*, khususnya varian *Multinomial Naive Bayes* yang sering digunakan dalam pengolahan data teks. Algoritma ini didasarkan pada *Teorema Bayes* serta asumsi bahwa setiap fitur bersifat saling bebas. Model kemudian memperkirakan probabilitas suatu ulasan tergolong ke dalam kategori "positif" atau "negatif" dengan mempertimbangkan frekuensi kemunculan kata-kata dalam teks ulasan tersebut.

### 3.7. Evaluasi Model

Evaluasi Model dilakukan dengan dua pendekatan utama, yaitu *Confusion Matrix* dan *10-Fold Cross Validation*. *Confusion matrix* digunakan untuk menghitung metrik performa seperti akurasi, presisi, *recall*, dan *F1-score*. Akurasi mengukur proporsi prediksi yang benar, presisi menunjukkan ketepatan prediksi positif, *recall* mengukur kemampuan model mengenali data positif yang sebenarnya, dan *F1-score* merepresentasikan keseimbangan antara presisi dan *recall*. Sementara itu, *10-Fold Cross Validation* digunakan untuk mengurangi bias dalam evaluasi, data dibagi menjadi 10 bagian dan dilakukan pelatihan serta pengujian secara bergiliran. Teknik ini memberikan penilaian yang lebih stabil dan menyeluruh terhadap performa model klasifikasi.

## 4. HASIL DAN PEMBAHASAN

### 4.1. Dataset

Penelitian ini memanfaatkan data berupa sekumpulan tanggapan atau ulasan yang diberikan oleh pengguna aplikasi *MyPertamina*, yang diperoleh secara otomatis melalui proses *web scraping* menggunakan pustaka *google-play-scraper* seperti yang ditunjukkan pada gambar 2. Pengambilan data dilakukan dari halaman aplikasi *MyPertamina* di platform *Google Play Store*, dengan total sebanyak 2.000 data ulasan yang berhasil dikumpulkan.

```

$ pip install google-play-scraper
Collecting google-play-scraper
  Downloading google_play_scraper-1.2.7-py3-none-any.whl.metadata (50 kB)
    Downloading google_play_scraper-1.2.7-py3-none-any.whl (28 kB)
Installing collected packages: google-play-scraper
Successfully installed google-play-scraper-1.2.7
    
```

Gambar 2. Proses instalasi pustaka *google-play-scraper*

```

from google_play_scraper import Sort, reviews
import pandas as pd

app_id = 'com.dafturn.mypertamina'

result = reviews(
    app_id,
    lang='id',
    country='id',
    sort=Sort.NEWEST, # Urut dari yang terbaru
    count=2000 # Jumlah review yang mau diambil (max 200 per call)
)
    
```

Rifikiansyah

Gambar 3. Proses *scraping* ulasan aplikasi *MyPertamina*

Informasi yang diambil dari masing-masing ulasan meliputi tanggal ulasan, nilai rating yang diberikan oleh pengguna, serta isi teks ulasan itu sendiri. Data kemudian disimpan dalam format *Excel* untuk mempermudah proses pengolahan dan analisis pada tahap selanjutnya. Dataset ini menjadi dasar utama dalam penerapan analisis sentimen yang dilakukan pada penelitian ini.

```

# Convert ke DataFrame
df = pd.DataFrame(result)
df = df[['userName', 'score', 'at', 'content']] # Pilih kolom penting
    
```

Gambar 4. Proses pengambilan data dan pemilihan kolom penting

	userName	score	at	content
0	MURTIALI BIN WADMAH	5	2025-05-05 08:51:01	mantap
1	Ahmad Faruk	5	2025-05-05 08:40:41	oke
2	ILWAN ALIANSAH	1	2025-05-05 08:26:05	buat apa ada aplikasinya kalau masih nyuruh di...
3	Soebandee World	5	2025-05-05 08:06:18	kereenn online mantap
4	jam ancher	1	2025-05-05 07:05:02	ga bisa ketik otp
5	pram mikla	1	2025-05-05 06:02:34	bug input koda otp ga bisa
6	Rahmat Harefa	1	2025-05-05 05:53:53	padahal sudah punya akun... mih keterangannya d...
7	Riadi Adi	5	2025-05-05 05:12:40	mantp
8	jayeng rono	1	2025-05-05 05:04:48	error terus
9	murkham irkham	5	2025-05-05 04:21:24	ok
10	Wirman Sahab	5	2025-05-05 03:44:00	mantap jiwaku dan ragaku, selamat ya buat mype...
11	Sobur 1985	1	2025-05-05 02:58:15	menyessal saya mendsontlod aplikasi ini, banyak ...

Gambar 5. Hasil *scraping* ulasan aplikasi *MyPertamina*

### 4.2. *Preprocessing*

Setelah data ulasan terkumpul, tahap berikutnya adalah tahap *preprocessing*. Tujuan dari tahap ini adalah untuk membersihkan teks ulasan agar lebih siap digunakan dalam tahap *labeling*. Langkah-langkah pra-pemrosesan yang diterapkan meliputi *Case Folding*, *Cleansing*, *Tokenizing*, *Filtering*, dan *Stemming*.

```

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

stop_factory = StopwordRemoverFactory()
stopwords = set(stop_factory.get_stop_words())

# Preprocessing
def preprocess_text(text):
    # 1. Lowercase
    lowercased = text.lower()

    # 2. Cleansing (hapus non-huruf, simbol, emoji, angka, dll)
    cleaned = re.sub(r"[^a-zA-Z]", "", lowercased)
    cleaned = re.sub(r"[^\s]", "", cleaned).strip()

    # 3. Tokenizing
    tokens = cleaned.split()

    # 4. Stopword Removal (Filtering)
    filtered_tokens = [token for token in tokens if token not in stopwords]

    # 5. Stemming
    stemmed_tokens = [stemmer.stem(token) for token in filtered_tokens]

    return pd.Series([lowercased, cleaned, tokens, filtered_tokens, stemmed_tokens])

df[['lowercased', 'cleaned', 'tokenized', 'filtered', 'stemmed']] = df['content'].astype(str).apply(preprocess_text)
    
```

Gambar 6. Tahap *Preprocessing*

Adapun contoh dari hasil *preprocessing* sebagai berikut :

Tabel 1. Contoh Penulisan Tabel

Tahap Preprocessing	Data Input	Data Hasil
<i>Case Folding</i>	My Pertamina tetap terbaik	my pertamina tetap terbaik
<i>Cleansing</i>	mantap proses cepat, terimakasih	mantap proses cepat terimakasih
<i>Tokenizing</i>	jaya terus pertamina	['jaya', 'terus', 'pertamina']
<i>Filtering</i>	Aplikasi buruk mempersulit rakyat kecil	['aplikasi', 'buruk', 'mempersulit', 'rakyat', 'kecil']

Stemming Aplikasi terburuk sepanjangan ['aplikasi', 'buruk', 'panjang', 'adab', 'peradaban manusia', 'manusia']

Dapat dilihat pada Tabel 1. di atas menunjukkan contoh hasil dari setiap tahapan *preprocessing* teks yang dilakukan pada data ulasan pengguna aplikasi *MyPertamina*. Dengan penerapan tahapan tersebut, data ulasan menjadi lebih bersih, terstruktur, dan siap digunakan pada proses pelabelan otomatis menggunakan *IndoBERT*.

### 4.3. Labeling

Setelah melewati tahap *preprocessing*, tahap berikutnya adalah proses pelabelan sentimen. *Labeling* dilakukan menggunakan model pra-latih *IndoBERT* (*w11wo/indonesian-roberta-base-sentiment-classifier*) untuk mengkategorikan setiap ulasan ke dalam dua kelas sentimen, yaitu positif dan negatif. Proses pelabelan dilakukan secara otomatis tanpa modifikasi tambahan, guna meningkatkan efisiensi dan memastikan konsistensi dalam pemberian label terhadap seluruh data yang telah melalui tahap *preprocessing*.

```
import pandas as pd
from transformers import pipeline
from tqdm.auto import tqdm
toda.pandas()

# load data
df = pd.read_csv("hasil_preprocessing.csv")

# load model IndoBERT
classifier = pipeline("sentiment-analysis", model="w11wo/indonesian-roberta-base-sentiment-classifier")

# labeling pakai IndoBERT
def map_indobert(label):
    if label == "positive":
        return "positif"
    else:
        return "negatif"

df["label_indobert_raw"] = df["cleaned_text"].astype(str).progress_apply(lambda x: classifier(x)[0]["label"].lower())
df["label_indobert"] = df["label_indobert_raw"].apply(map_indobert)

# hasil labeling
df.to_csv("hasil_dengan_label.csv", index=False)
```

Gambar 7. Proses pelabelan menggunakan *IndoBERT*

	content	cleaned_text	label_indobert
1584	Pertamina koruplorrr	perfamina koruplorrr	negatif
572	jelekssss mobil saya jadi mogok isinya air semua	jelekssss mobil jadi mogok isi air semua	negatif
332	mantap lancar jaya	mantap lancar jaya	positif
1073	bagus	bagus	positif
433	Baik	baik	positif
651	mantap	mantap	positif
1780	Aplikasi jelek, ngajutin subsidi BBM di tolak L...	aplikasi jelek ngajutin subsidi bsm tolak lerus...	negatif
751	sulit mendapatkan kode QR	sulit dapat kode qr	negatif
673	PERTAMAX di SPBU hampir semua kota baikpapan...	perfamax sptu hampir semua kota baikpapan war...	negatif
1260	Pertamina Sarang Koruptor	perfamina sarang koruptor	negatif
1574	Jos	jos	negatif
1772	Kenapa aplikasinya gak bisa login akun dan da...	aplikasi gak login akun daftar jadi guna apak...	negatif
1006	Aplikasi bermanfaat	aplikasi mantaa	positif
730	aplikasi tidak bagus. tidak ada sistem pelaporan	aplikasi bagus sistem lapor	positif
544	Urutak saat ini masih 🙄🙄🙄🙄	NaN	positif
581	bagus ramah	bagus ramah	positif
1049	Awas menggunakan aplikasi ini data saya bocor ...	awas guna aplikasi data bocor dar' web pafutl...	negatif
1636	Mudah pendaftarannya	mudah daftar	positif
1478	pelayanan yg sangat buruk	layan yg sangat buruk	negatif
424	aplikasinya gimana ya aku mau masuk akun subsi...	aplikasi gimana aku mau masuk akun subssi lep...	negatif

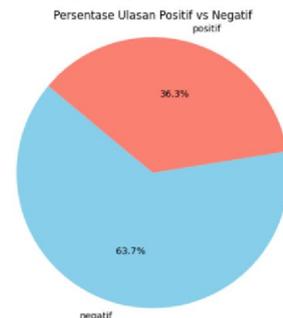
Gambar 8. Hasil pelabelan menggunakan *IndoBERT*

Gambar 8. menunjukkan hasil pelabelan otomatis terhadap data ulasan yang telah melalui tahap *preprocessing*. Proses ini menggunakan model *IndoBERT* melalui *pipeline sentiment analysis*, setiap ulasan diklasifikasikan ke dalam kategori positif atau negatif. Label yang dihasilkan akan menjadi dasar untuk tahap klasifikasi berikutnya. Adapun distribusi jumlah ulasan berdasarkan label sentimen sebagai berikut :

Label	Jumlah Ulasan	Persentase (%)
positif	727	36.35
negatif	1273	63.65

Gambar 9. Jumlah data ulasan

Gambar 9. menampilkan visualisasi sebaran sentimen dari total 2.000 ulasan yang telah diklasifikasikan. Sebanyak 727 ulasan (36,35%) tergolong positif, sementara 1.273 ulasan (63,65%) termasuk dalam sentimen negatif. Data ini menunjukkan bahwa mayoritas pengguna menyampaikan ulasan bernada negatif terhadap aplikasi *MyPertamina* selama periode Mei 2025. Perbandingan proporsi ini akan divisualisasikan lebih lanjut dalam bentuk diagram pie.



Gambar 10. Diagram presentase sentiment pada bulan Mei 2025

### 4.4. Pembagian dan Pembobotan Data

Pada penelitian ini, data ulasan dibagi secara acak menjadi dua subset, yaitu 80% untuk data pelatihan dan 20% untuk data pengujian, menggunakan pustaka *scikit-learn*. Pembagian ini memungkinkan model belajar dari data latihan dan dievaluasi secara objektif menggunakan data uji. Selanjutnya, data teks yang telah diproses dikonversi menjadi bentuk numerik menggunakan metode *TF-IDF*. *Term Frequency (TF)* menghitung frekuensi kemunculan kata dalam suatu dokumen, sedangkan *Inverse Document Frequency (IDF)* mengukur tingkat keunikan kata dalam seluruh korpus. Bobot *TF-IDF* membantu model memfokuskan pada kata-kata yang paling relevan untuk klasifikasi sentimen. Ilustrasi pembagian data ditampilkan pada Gambar 11.

```
# data dibagi jadi data training dan testing dengan test_size = 0.20 dan random_state = 0
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    df['content'],
    df['label_indobert'],
    test_size = 0.20,
    random_state = 0
)

# PERBOBOTAN - TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer()
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
tfidf_test = tfidf_vectorizer.transform(X_test)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)

(1600,)
(1600,)
(400,)
(400,)
```

Gambar 11. Pembagian data *training* dan data *testing*

### 4.5. Evaluasi Model

Evaluasi model dalam penelitian ini difokuskan pada algoritma *Multinomial Naive Bayes*, yang dinilai berdasarkan performanya dalam proses klasifikasi sentimen. Evaluasi dilakukan dalam dua tahap utama, yakni proses pelatihan model dan pengujian terhadap data yang belum pernah dilihat

sebelumnya. Pada tahap pelatihan, model mempelajari pola dari data latih untuk meminimalkan kesalahan prediksi. Selanjutnya, model diuji menggunakan data uji guna mengukur efektivitasnya. *Multinomial Naïve Bayes* dipilih karena umum digunakan untuk klasifikasi teks dan menunjukkan kinerja yang baik dalam menangani data ulasan.

```

import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split

# Turunkan 80% ke data latih
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Cetak jumlah data
print(f"Jumlah data latih: {X_train.shape[0]}")
print(f"Jumlah data uji: {X_test.shape[0]}")

# MultinomialNB
clf = MultinomialNB()
clf.fit(X_train, y_train)

# Test
y_test_pred = clf.predict(X_test)

# Evaluasi
print(f"Accuracy: {accuracy_score(y_test, y_test_pred)}")
print(f"Precision: {precision_score(y_test, y_test_pred, average='binary')}")
print(f"Recall: {recall_score(y_test, y_test_pred, average='binary')}")
print(f"F1-Score: {f1_score(y_test, y_test_pred, average='binary')}")

# Confusion Matrix
print(f"Confusion Matrix: {confusion_matrix(y_test, y_test_pred)}")
print(f"Classification Report: {classification_report(y_test, y_test_pred, zero_division=0)}")
    
```

Gambar 12. Proses evaluasi model *Multinomial Naïve Bayes*

Evaluasi dilakukan dengan menghitung sejumlah metrik seperti akurasi, presisi, *recall*, dan *F1-score*. Tujuan penggunaan metrik-metrik tersebut adalah untuk menyajikan penilaian yang objektif terhadap kinerja model terhadap performa model dalam mengklasifikasikan data dengan tepat.

```

MultinomialNB Accuracy : 0.8925
MultinomialNB Precision: 0.9015151515151515
MultinomialNB Recall : 0.9333333333333333
MultinomialNB f1_score : 0.9171483622350675
confusion matrix:
[[238 17]
 [ 26 119]]
precision    recall  f1-score   support

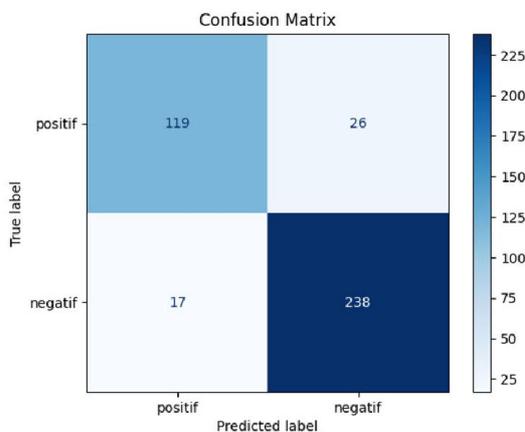
negatif      0.90      0.93      0.92      255
positif      0.88      0.82      0.85      145

accuracy          0.89          400
macro avg         0.89          400
weighted avg      0.89          400
    
```

Gambar 13. Hasil evaluasi model *Multinomial Naïve Bayes*

#### 4.6. Confusion Matrix

Analisis terhadap *confusion matrix* berperan penting dalam mengenali berbagai bentuk kesalahan yang terjadi saat model melakukan klasifikasi. Dengan demikian, *confusion matrix* tidak hanya menampilkan tingkat akurasi, tetapi juga memberikan informasi mendalam terkait presisi dan sensitivitas model. Adapun hasil evaluasi model *Multinomial Naïve Bayes* yang didapatkan dari *confusion matrix* dapat dilihat pada gambar berikut :



Gambar 14. *Confusion Matrix*

Gambar 14. menampilkan visualisasi *confusion matrix* dari evaluasi model klasifikasi menggunakan algoritma *Naïve Bayes*. Dari total data uji, sebanyak 119 data positif berhasil diprediksi dengan benar atau *TP (True Positive)*, 26 data positif salah diklasifikasikan sebagai negatif atau *FN (False Negative)*, 238 data negatif diklasifikasikan dengan benar atau *TN (True Negative)*, dan 17 data negatif salah diklasifikasikan sebagai positif atau *FP (False Positive)*. Hasil ini menunjukkan bahwa model mampu mengklasifikasikan ulasan positif dan negatif dengan cukup akurat, ditandai oleh tingginya nilai *True Negative (TN)* dan *True Positive (TP)*. Berikut rumus dari *confusion matrix* :

- Accuracy*

$$\text{Akurasi} = \frac{119+238}{119+238+17+26} = 0.8925$$
- Precision*

$$\text{Presisi} = \frac{119}{119+17} = 0.875$$
- Recall*

$$\text{Recall} = \frac{119}{119+26} = 0.8206$$
- F1-Score*

$$\text{F1-Score} = 2 \frac{0.875 \times 0.8206}{0.875 + 0.8206} = 0.8469$$

#### 4.7. K-Fold Cross Validation

*K-Fold Cross Validation* digunakan sebagai metode evaluasi tambahan untuk meningkatkan akurasi dan mengurangi bias dalam penilaian performa model. Dalam penelitian ini diterapkan skema *10-Fold Cross Validation*, di mana data dibagi menjadi 10 bagian, dan proses pelatihan serta pengujian dilakukan sebanyak 10 kali secara bergantian. Metode ini menghasilkan evaluasi yang lebih stabil dan membantu mencegah *overfitting* maupun *underfitting*. Dengan pendekatan ini, performa algoritma *Multinomial Naïve Bayes* dapat dinilai secara lebih menyeluruh dan dapat diandalkan dalam klasifikasi sentimen ulasan pengguna aplikasi *MyPertamina*.

Fold ke-	Akurasi
0	0.910000
1	0.905000
2	0.915000
3	0.890000
4	0.865000
5	0.900000
6	0.895000
7	0.915000
8	0.880000
9	0.895000
Rata-rata	0.897000

Gambar 15. Hasil *K-Fold Cross Validation* pada model *Multinomial Naïve Bayes*

### 5. KESIMPULAN DAN SARAN

Hasil analisis sentimen terhadap 2.000 ulasan pengguna aplikasi *MyPertamina* menunjukkan bahwa 63,65% bersentimen negatif dan 36,35% positif, mencerminkan tingkat ketidakpuasan pengguna yang cukup tinggi. Data dikumpulkan melalui *web scraping*, diproses melalui tahapan *preprocessing (case folding, tokenizing, cleansing, filtering, stemming)*, dan dilabeli otomatis menggunakan *IndoBERT*. Transformasi teks dilakukan dengan *TF-IDF*, kemudian data dibagi menjadi 80% latih dan 20% uji. Proses klasifikasi menggunakan *Multinomial Naïve Bayes*,

dengan evaluasi melalui *confusion matrix* dan *10-Fold Cross Validation*. Hasil evaluasi menunjukkan performa baik, dengan akurasi 89,25%, presisi 87,5%, *recall* 82%, dan *F1-score* 84,69%. Pendekatan ini terbukti efektif dalam mengklasifikasikan sentimen dan memberikan gambaran umum terhadap persepsi pengguna aplikasi *MyPertamina*.

Berdasarkan hasil penelitian ini, terdapat beberapa saran yang disampaikan kepada pihak terkait untuk dipertimbangkan:

1. Untuk pengguna, diharapkan memberikan ulasan yang jujur, jelas, dan sesuai pengalaman, agar dapat membantu pengembang memahami permasalahan nyata dan meningkatkan kualitas layanan aplikasi.
2. Untuk peneliti selanjutnya, disarankan untuk memperluas cakupan data dari berbagai periode, mempertimbangkan sentimen netral, serta membandingkan performa algoritma lain seperti *SVM*, *Decision Tree*, atau *deep learning*. Penambahan fitur visualisasi seperti *word cloud* juga direkomendasikan untuk memperkaya analisis.
3. Untuk pengembang aplikasi, hasil analisis ini dapat menjadi bahan evaluasi layanan, khususnya terkait antarmuka, performa teknis, dan pengalaman pengguna. Disarankan pula untuk mengintegrasikan sistem monitoring berbasis sentimen sebagai solusi jangka panjang dalam menjaga kualitas layanan digital.

## DAFTAR PUSTAKA

[1] I. Maulana, W. Apriandari, and A. Pambudi, "Analisis Sentimen Berbasis Aspek Terhadap Ulasan Aplikasi Mypertamina Menggunakan Support Vector Machine," *IDEALIS Indones. J. Inf. Syst.*, vol. 6, no. 2, pp. 172–181, 2023, doi: 10.36080/idealis.v6i2.3022.

[2] H. Taufiqurrahman, F. Tri Anggraeny, M. Muharrom Al Haromainy, and Et.al, "Perbandingan Algoritma Naïve Bayes Dan K-Nearest Neighbor Pada Analisis Sentimen Ulasan Aplikasi Mypertamina," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3934–3939, 2024, doi: 10.36040/jati.v7i6.7801.

[3] G. Darmawan, S. Alam, and M. I. Sulisty, "Analisis Sentimen Berdasarkan Ulasan Pengguna Aplikasi Mypertamina Pada Google Playstore Menggunakan Metode Naïve Bayes," *STORAGE – J. Ilm. Tek. dan Ilmu Komput.*, vol. 2, no. 3, pp. 100–108, 2023.

[4] D. S. Sayogo *et al.*, "ANALISIS SENTIMEN ULASAN INSTAGRAM DI GOOGLE PLAY STORE," vol. 7, no. 6, pp. 3314–3319, 2023.

[5] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," vol. 6, no. 9, pp. 4305–4313, 2022.

[6] N. A. Susanti, M. Walid, and H. et Al., "KLASIFIKASI DATA TWEET UJARAN KEBENCIAN DI MEDIA SOSIAL," vol. 6, no. 2, pp. 538–543, 2022.

[7] M. K. Insan, U. Hayati, and O. Nurdiawan, "ANALISIS SENTIMEN APLIKASI BRIMO PADA ULASAN PENGGUNA DI," vol. 7, no. 1, pp. 478–483, 2023.

[8] M. Raffi, A. Suharso, I. Maulana, and U. S. Karawang, "Analisis sentimen ulasan aplikasi binar pada google play store menggunakan algoritma naïve bayes sentiment analysis of binar application reviews on google play store using naïve bayes algorithm," vol. 6, 2023.

[9] N. F. Arminda *et al.*, "IMPLEMENTASI ALGORITMA

MULTINOMIAL NAIVE BAYES PADA ANALISIS SENTIMEN TERHADAP ULASAN PENGGUNA APLIKASI BRIMO," vol. 7, no. 3, pp. 1817–1822, 2023.

- [10] A. Sabrani, I. G. Putu, W. Wedashwara, and F. Bimantoro, "METODE MULTINOMIAL NAÏVE BAYES UNTUK KLASIFIKASI ARTIKEL ONLINE TENTANG GEMPA DI INDONESIA ( Multinomial Naïve Bayes Method for Classification of Online Article About Earthquake in Indonesia )," vol. 2, no. 1, pp. 89–100, 2020.
- [11] Khairunnisa, K., Dewi, S. K., Rahmawati, D. D., & Sari, A. P. . (2024). Analisis sentimen komentar pada postingan instagram akun "standwithus" menggunakan klasifikasi naïve bayes. *Jurnal ilmiah informatika*, 12(02), 191–199. <https://doi.org/10.33884/jif.v12i02.9263>
- [12] Nur Rahman, R., Rahim, A., & Joko Pranoto, . W. (2025). Analisis sentimen ulasan game efootball 2024 pada playstore menggunakan algoritma naïve bayes. *Jurnal ilmiah informatika*, 13(01), 38–44. <https://doi.org/10.33884/jif.v13i01.9913>
- [13] Ridhoi, R., Verdikha, N. A., & Yulianto, F. (2025). Analisis klasifikasi ulasan aplikasi sirekap 2024 menggunakan ekstraksi fitur distilbert dan metode support vector machine. *Jurnal ilmiah informatika*, 13(01), 26–32. <https://doi.org/10.33884/jif.v13i01.9753>
- [14] Kharisma, N. ., & Pusparini, N. N. . (2025). Analisis kepuasan layanan biro administrasi akademik kemahasiswaan stmik widuri menggunakan algoritma naïve bayes classifier. *jurnal ilmiah informatika*, 13(01), 1–7. <https://doi.org/10.33884/jif.v13i01.9427>
- [15] Huda, D. N. I., Prianto, C., & Awangga, R. M. (2023). Analisis sentimen perbandingan layanan jasa pengiriman kurir pada ulasan play store menggunakan metode decision tree dan random forest. *Jurnal ilmiah informatika*, 11(02), 150–158. <https://doi.org/10.33884/jif.v11i02.7952>

## BIODATA PENULIS



**Rifkiansyah**

merupakan mahasiswa di Universitas Bina Sarana Informatika, Jurusan Informatika



**Santoso Setiawan, M. Kom**

merupakan Dosen di Universitas Bina Sarana Informatika



**Findi Ayu Sariasih, ST., MMSI**

merupakan Dosen di Universitas Bina Sarana Informatika