



FAKULTAS SAINS, TEKNOLOGI, DAN KESEHATAN UNIVERSITAS SAHID SURAKARTA

Jl. Adi Sucipto No. 154, Jajar, Solo 57144

Tel. (0271) 743493, 743494, Fax. (0271) 742047

email: Fstk@usahidsolo.ac.id , website: www.usahidsolo.ac.id

Kepada Yth,
Pimpinan PT Kereta Api Indonesia (KAI)
Perihal : Rekomendasi Perbaikan Aplikasi 'Access by KAI'

Dengan Hormat,
Saya, Moh Andi Setyo Nugroho, Mahasiswa Program Studi Informatika Universitas Sahid Surakarta, mengajukan rekomendasi terkait perbaikan aplikasi 'Access by KAI'. Rekomendasi ini diberikan berdasarkan hasil penelitian saya yang berjudul "Analisis Sentimen Ulasan Pengguna Aplikasi 'Access by KAI' Menggunakan Algoritma Machine Learning".

Penelitian ini dilaksanakan dibawah bimbingan,
Dosen Pembimbing 1 : Ir Dahlan Susilo, M.Kom
Dosen Pembimbing 2 : Dwi Retnoningsih, S.T, M,T

Jurnal Penerbit : <https://jurnal.murnisadar.ac.id/index.php/Tekinkom/article/view/1854>

Berdasarkan hasil dari penelitian, pengguna aplikasi 'Access by KAI' memberikan opini yang beragam terkait dengan pengalaman dalam penggunaan aplikasi 'Access by KAI'. Hasil penelitian analisis sentimen pada penelitian ini menghasilkan sebanyak 44,3% pengguna memberikan ulasan positif, 26,3% ulasan netral dan 29,4% ulasan negatif. Ulasan Positif dari pengguna mendominasi namun banyak pengguna yang memberikan masukan konstruktif yang menunjukkan adanya kebutuhan untuk perbaikan aplikasi. Rekomendasi yang dapat diberikan berdasarkan hasil penelitian kepada PT KAI sebagai berikut.

1. Penerapan Software Quality Assurance (SQA): Lakukan pengecekan aplikasi untuk mendalami masalah yang ada
2. Perbaikan Fitur Aplikasi: Memperbaiki serta meningkatkan kemudahan dalam pemesanan tiket dan proses pembayaran.
3. Peningkatan Kinerja Aplikasi: Fokus pada stabilitas dan responsivitas aplikasi untuk mengurangi error.
4. Peningkatan Komunikasi: Sediakan platform atau saluran komunikasi yang lebih responsif, seperti chat langsung atau hotline, untuk menangani keluhan pengguna terkait pengalaman perjalanan, sehingga pengguna transportasi kereta yang mengalami kendala pada saat perjalanan tidak menuliskan ulasan negatif pada platform aplikasi

Demikian rekomendasi perbaikan ini saya sampaikan dengan harapan dapat memberikan kontribusi positif bagi pengembangan aplikasi 'Access by KAI' dan meningkatkan pengalaman pengguna. Saya percaya bahwa dengan menerapkan rekomendasi ini, PT KAI dapat lebih baik dalam memenuhi kebutuhan dan harapan pengguna, atas perhatian dan kerjasamanya, saya ucapkan terima kas

E-ISSN: 2621-3079 _ P-ISSN: 2621-1556

Jurnal Teknik Informasi Dan Komputer

TEKINKOM-Politeknik Bisnis Indonesia

Sekretariat : Lembaga Penelitian Dan Pengabdian Kepada Masyarakat (LPPM)
Jl. Sriwijaya No. C-E Pematangsiantar – Sumatera Utara (21144)
<https://jurnal.murnisadar.ac.id/index.php/Tekinkom>

SURAT KETERANGAN NASKAH DITERIMA

(Letter of Acceptance)

No. 049/TEKINKOM/LL/XII/2024

Dengan ini, Dewan Redaksi Jurnal Tekinkom (Teknik Informasi dan Komputer) memberitahukan bahwa naskah bapak/ibu dengan identitas :

Judul : ANALISIS SENTIMEN ULASAN APLIKASI "ACCESS BY KAI"
MENGGUNAKAN ALGORITMA MACHINE LEARNING

Penulis : Moh Andi Setyo Nugroho, Dahlan Susilo, Dwi Retnoningsih

DOI : 10.37600/tekinkom.v7i2.1854

Telah memenuhi kriteria publikasi di Jurnal Tekinkom (Teknik Informasi dan Komputer) dan dinyatakan DITERIMA sebagai bahan naskah untuk penerbitan Jurnal TEKINKOM Volume 7 No. 2 Tahun 2024, dalam versi cetak maupun elektronik. Melalui surat keterangan ini, penulis tunduk kepada ketentuan hak cipta pada Jurnal TEKINKOM.

Untuk menghindari adanya duplikasi terbitan dan pelanggaran etika publikasi ilmiah, kami menghimbau agar naskah/artikel tersebut tidak dikirimkan dan dipublikasikan ke penerbitan jurnal/majalah lain.

Demikian surat ini disampaikan, atas partisipasi dan kerjasamanya kami ucapkan terima kasih.

Pematangsiantar, 31 Desember 2024
Hormat kami,



Victor M.M. Siregar, M.Kom
Editor In Chief Jurnal Tekinkom

LEMBAR KONSULTASI SKRIPSI / TA



FAKULTAS SAINS, TEKNOLOGI dan KESEHATAN UNIVERSITAS SAHID SURAKARTA

Jl. Adi Sucipto 154, Solo 57144, Indonesia
Telp. +62 - (0)271-743493, 743494, Fax. +62 - (0)271-742047

NAMA	: Moh Andi Setyo Nugroho
NIM	: 2021061007
PROGRAM STUDI	: Informatika
JUDUL SKRIPSI / TA	: Analisis Sentimen Ulasan Pengguna Aplikasi 'Access by KAI' Menggunakan Algoritma Machine Learning
PEMBIMBING I	: Ir. Dahlan Susilo, M.Kom.

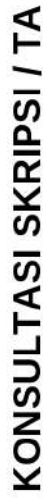
LEMBAR KONSULTASI SKRIPSI / TA



FAKULTAS SAINS, TEKNOLOGI dan KESEHATAN UNIVERSITAS SAHID SURAKARTA

Jl. Adi Sucipto 154, Solo 57144, Indonesia
Telp. +62 - (0)271-743493, 743494, Fax. +62 - (0)271-742047

NAMA	: Moh Andi Setyo Nugroho
NIM	: 2021061007
PROGRAM STUDI	: Informatika
JUDUL SKRIPSI / TA	: Analisis Sentimen Ulasan Pengguna Aplikasi 'Access by KAI' Menggunakan Algoritma Machine Learning
PEMBIMBING II	: Dwi Retnoningsih, S.T., M.T.

[illegible]

```
In [1]: 1 import pandas as pd
2 df = pd.read_csv('dataset.csv')
3 df
```

Out[1]:

	reviewId	userName	userImage	content	score	tl
0	159a7195-b0fc-4051-bb7d-964054fa9248	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Mantap praktis dan sangat membantu	5	
1	8dd9356f-ed0-4ef2-8628-f9a8c0a10b30	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Instalnya lama banget, butuh 1 jam diulang2 ga...	1	
2	3bf7bcd5-cd3d-444f-af97-72804de35a00	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Gak bisa pesen tiket, aplikasi lemot	1	
3	32a10093-bf5d-4430-93dc-de41eea257d6	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Sumpah kecewa banget, aku bela-belain begadang...	1	
4	269c38b0-61cb-45db-a82e-9cad2dd3fe6e	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	kenapa aplikasi pemerintah pada kurang semua s...	1	
...	...	...	...	...	...	...
9119	10606636-aee0-439e-9e75-4c174aecacf9	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Sudah melakukan pembayaran melalui KAI Pay,sa...	3	
9120	8cb6d3c3-0cff-4b27-869d-f73cb7c57760	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Daftar akunnya bikin emosi, liat otp di email,...	1	
9121	a86eaa0a-941f-432a-8431-c8ee77144ef7	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Pelayanan KAI commuterline dhoho (401) kurang ...	1	
9122	c88868c0-e6d7-4ba1-bd7d-616084727229	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Sangat mengecewakan. Jauh2 hari saya mau refun...	1	
9123	02cdf5cd-795a-40d5-b949-1183bf8079f7	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	aplikasi ribet daftar aja sulit,perbaiki dong ...	1	

9124 rows × 11 columns


```
In [2]: 1 df = df[['content']]
        2 df
```

```
Out[2]:
```

	content
0	Mantap praktis dan sangat membantu
1	Instalnya lama banget, butuh 1 jam diulang2 ga...
2	Gak bisa pesen tiket, aplikasi lemot
3	Sumpah kecewa banget, aku bela-belain begadang...
4	kenapa aplikasi pemerintah pada kurang semua s...
...	...
9119	Sudah melakukan pembayaran melalui KAI Pay ,sa...
9120	Daftar akunnya bikin emosi, liat otp di email,...
9121	Pelayanan KAI commuterline dhoho (401) kurang ...
9122	Sangat mengecewakan. Jauh2 hari saya mau refun...
9123	aplikasi ribet daftar aja sulit,perbaiki dong ...
9124	rows × 1 columns

```
In [3]: 1 import re
        2 def clean_text(text):
        3     # 1. Lowercasing
        4     text = text.lower()
        5
        6     # 2. Menghilangkan tanda baca
        7     text = re.sub(r'^\w\s', '', text)
        8
        9     # 3. Menghilangkan angka
        10    text = re.sub(r'\d+', '', text)
        11
        12    # 4. Menghilangkan emoticon (ini bisa bervariasi, tetapi kita bisa
        13    text = re.sub(r'^\w\s', '', text)
        14
        15    # 5. Menghilangkan huruf ganda
        16    text = re.sub(r'(\.)\1+', r'\1', text)
        17
        18    return text
```


In [4]:

```
1 # Terapkan fungsi pembersihan ke kolom 'content'
2 df['content'] = df['content'].apply(clean_text)
3 df
```

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19740\1589648253.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

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

Out[4]:

	content
0	mantap praktis dan sangat membantu
1	instalnya lama banget butuh jam diulang ga bis...
2	gak bisa pesen tiket aplikasi lemot
3	sumpah kecewa banget aku belabelain begadang n...
4	kenapa aplikasi pemerintah pada kurang semua s...
...	...
9119	sudah melakukan pembayaran melalui kai pay sal...
9120	daftar akunya bikin emosi liat otp di email pa...
9121	pelayanan kai comuterline dhoho kurang ramah y...
9122	sangat mengecewakan jauh hari saya mau refund ...
9123	aplikasi ribet daftar aja sulitperbaiki dong s...

9124 rows × 1 columns

In [5]:

```
1 import json
2 import re
3
4 # Membaca file JSON yang berisi kamus normalisasi
5 with open('normalize.json', 'r', encoding='utf-8') as file:
6     normalize_dict = json.load(file)
```

```
In [6]: 1 def normalize_text(text, normalize_dict):
2         # Jika input bukan string, konversi ke string
3         if not isinstance(text, str):
4             text = str(text)
5
6         # Fungsi bantuan untuk mengganti kata dengan normalisasi
7         def replace(match):
8             return normalize_dict[match.group(0)]
9
10        # Membuat pola regex dari kata-kata yang ada di kamus normalisasi
11        pattern = re.compile(r'\b(' + '|'.join(map(re.escape, normalize_dict.keys())) + ')')
12
13        # Mengganti kata-kata dalam teks dengan normalisasi yang sesuai
14        normalized_text = pattern.sub(replace, text)
15
16        return normalized_text
```

```
In [7]: 1 # Memastikan kolom 'content' ada dalam DataFrame
2 if 'content' in df.columns:
3     # Menerapkan fungsi normalisasi pada kolom 'content'
4     df['content'] = df['content'].apply(lambda x: normalize_text(x, normalize_dict))
5 else:
6     print("Kolom 'content' tidak ditemukan dalam DataFrame.")
7
8 # Menampilkan DataFrame yang sudah dinormalisasi
9 df.head()
10
```

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19740\3201629591.py:4: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df['content'] = df['content'].apply(lambda x: normalize_text(x, normalize_dict))
```

Out[7]:

	content
0	mantap praktis dan sangat membantu
1	instalnya lama banget butuh jam diulang tidak ...
2	tidak bisa pesen tiket aplikasi lambat
3	sumpah kecewa banget aku belabelain begadang t...
4	kenapa aplikasi pemerintah pada kurang semua s...

```
In [8]: 1 # Membaca file JSON yang berisi kamus normalisasi
2 with open('new-normalize.json', 'r', encoding='utf-8') as file:
3     normalize_dict = json.load(file)
```

```
In [9]: 1 def normalize_text(text, normalize_dict):
2         # Jika input bukan string, konversi ke string
3         if not isinstance(text, str):
4             text = str(text)
5
6         # Fungsi bantuan untuk mengganti kata dengan normalisasi
7         def replace(match):
8             return normalize_dict[match.group(0)]
9
10        # Membuat pola regex dari kata-kata yang ada di kamus normalisasi
11        pattern = re.compile(r'\b(' + '|'.join(map(re.escape, normalize_dict.keys())) + ')')
12
13        # Mengganti kata-kata dalam teks dengan normalisasi yang sesuai
14        normalized_text = pattern.sub(replace, text)
15
16        return normalized_text
```

```
In [10]: 1 # Memastikan kolom 'content' ada dalam DataFrame
2         if 'content' in df.columns:
3             # Menerapkan fungsi normalisasi pada kolom 'content'
4             df['content'] = df['content'].apply(lambda x: normalize_text(x, normalize_dict))
5         else:
6             print("Kolom 'content' tidak ditemukan dalam DataFrame.")
7
8         # Menampilkan DataFrame yang sudah dinormalisasi
9         df.head()
```

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19740\1413012.py:4: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df['content'] = df['content'].apply(lambda x: normalize_text(x, normalize_dict))
```

```
Out[10]:
```

	content
0	mantap praktis dan sangat membantu
1	instalnya lama banget butuh jam diulang tidak ...
2	tidak bisa pesan tiket aplikasi lambat
3	sumpah kecewa banget aku belabelain begadang t...
4	kenapa aplikasi pemerintah pada kurang semua ...

```

In [11]: 1 import nltk
2 from nltk.tokenize import word_tokenize
3
4 # Pastikan Anda telah mengunduh tokenizer
5 nltk.download('punkt')
6
7 # Fungsi untuk melakukan tokenisasi
8 def tokenize_text(text):
9     # Memeriksa apakah teks bukan NaN
10    if pd.isna(text):
11        return []
12    # Melakukan tokenisasi
13    tokens = word_tokenize(text)
14    return tokens
15
16 # Menerapkan tokenisasi pada kolom 'content' dan menyimpan hasilnya di
17 df['tokens'] = df['content'].apply(tokenize_text)
18 df
19

```

[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\LENOVO\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19740\4176072268.py:17: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
df['tokens'] = df['content'].apply(tokenize_text)

```

Out[11]:

```

	content	tokens
0	mantap praktis dan sangat membantu	[mantap, praktis, dan, sangat, membantu]
1	instalnya lama banget butuh jam diulang tidak ...	[instalnya, lama, banget, butuh, jam, diulang,...
2	tidak bisa pesan tiket aplikasi lambat	[tidak, bisa, pesan, tiket, aplikasi, lambat]
3	sumpah kecewa banget aku belabelain begadang t...	[sumpah, kecewa, banget, aku, belabelain, bega...
4	kenapa aplikasi pemerintah pada kurang semua ...	[kenapa, aplikasi, pemerintah, pada, kurang, S...
...	...	...
9119	sudah melakukan pembayaran melalui kaya pay sa...	[sudah, melakukan, pembayaran, melalui, kaya, ...
9120	daftar akunya buat emosi lihat otp di email pa...	[daftar, akunya, buat, emosi, lihat, otp, di, ...
9121	pelayanan kaya komuter line dhoho kurang ramah...	[pelayanan, kaya, komuter, line, dhoho, kurang...
9122	sangat mengecewakan jauh hari saya mau refund ...	[sangat, mengecewakan, jauh, hari, saya, mau, ...
9123	aplikasi ribet daftar saja sulitperbaiki dong ...	[aplikasi, ribet, daftar, saja, sulitperbaiki,...

9124 rows × 2 columns


```
In [12]: 1 from nltk.corpus import stopwords
2 stop_words = stopwords.words('indonesian')
```

```
In [13]: 1 def remove_stopwords(text):
2         return [word for word in text if word not in stop_words]
3
4 df['filtering_stopwords'] = df['tokens'].apply(lambda x: remove_stopwor
5
6 df.head()
```

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19740\746740486.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df['filtering_stopwords'] = df['tokens'].apply(lambda x: remove_stopword
s(x))
```

```
Out[13]:
```

	content	tokens	filtering_stopwords
0	mantap praktis dan sangat membantu	[mantap, praktis, dan, sangat, membantu]	[mantap, praktis, membantu]
1	instalnya lama banget butuh jam diulang tidak ...	[instalnya, lama, banget, butuh, jam, diulang,...]	[instalnya, banget, butuh, jam, diulang, masuk]
2	tidak bisa pesan tiket aplikasi lambat	[tidak, bisa, pesan, tiket, aplikasi, lambat]	[pesan, tiket, aplikasi, lambat]
3	sumpah kecewa banget aku belabelain begadang t...	[sumpah, kecewa, banget, aku, belabelain, bega...]	[sumpah, kecewa, banget, belabelain, begadang,...]
4	kenapa aplikasi pemerintah pada kurang semua ...	[kenapa, aplikasi, pemerintah, pada, kurang, s...]	[aplikasi, pemerintah, niat, aplikasi, memperm...]

```
In [14]: 1 from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
2 from nltk.stem import PorterStemmer
3 from nltk.stem.snowball import SnowballStemmer
```

```
In [15]: 1 factory = StemmerFactory()
2 stemmer = factory.create_stemmer()
3
4 def stem_text(text):
5     return [stemmer.stem(word) for word in text]
6
7 df['stemming'] = df['filtering_stopwords'].apply(lambda x: ' '.join(stem
8 df.head()
```

C:\Users\LENOVO\AppData\Local\Temp\ipykernel_19740\2362321403.py:7: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df['stemming'] = df['filtering_stopwords'].apply(lambda x: ' '.join(stem_text(x)))
```

```
Out[15]:
```

	content	tokens	filtering_stopwords	stemming
0	mantap praktis dan sangat membantu	[mantap, praktis, dan, sangat, membantu]	[mantap, praktis, membantu]	mantap praktis bantu
1	instalnya lama banget butuh jam diulang tidak ...	[instalnya, lama, banget, butuh, jam, diulang,...]	[instalnya, banget, butuh, jam, diulang, masuk]	instalnya banget butuh jam ulang masuk
2	tidak bisa pesan tiket aplikasi lambat	[tidak, bisa, pesan, tiket, aplikasi, lambat]	[pesan, tiket, aplikasi, lambat]	pesan tiket aplikasi lambat
3	sumpah kecewa banget aku belabelain begadang t...	[sumpah, kecewa, banget, aku, belabelain, bega...]	[sumpah, kecewa, banget, belabelain, begadang,...]	sumpah kecewa banget belabelain begadang tungg...
4	kenapa aplikasi pemerintah pada kurang semua ...	[kenapa, aplikasi, pemerintah, pada, kurang, s...]	[aplikasi, pemerintah, niat, aplikasi, memperm...]	aplikasi perintah niat aplikasi mudah sulit su...

```
In [16]: 1 df = df.dropna(subset=['stemming'])
2 len(df)
```

Out[16]: 9124

```
In [17]: 1 df = df.drop_duplicates(subset=['stemming'])
2 len(df)
```

Out[17]: 8245

```
In [18]: 1 df.to_csv('preprocessing_teks.csv', index=False)
```

In [19]:

```
1 import pandas as pd
2 df = pd.read_csv('preprocessing_teks.csv')
3 df
```

Out[19]:

	content	tokens	filtering_stopwords	stemming
0	mantap praktis dan sangat membantu	['mantap', 'praktis', 'dan', 'sangat', 'membantu']	['mantap', 'praktis', 'membantu']	mantap praktis bantu
1	instalnya lama banget butuh jam diulang tidak ...	['instalnya', 'lama', 'banget', 'butuh', 'jam', 'diulang', 'tidak']	['instalnya', 'banget', 'butuh', 'jam', 'diulang']	instalnya banget butuh jam ulang masuk
2	tidak bisa pesan tiket aplikasi lambat	['tidak', 'bisa', 'pesan', 'tiket', 'aplikasi']	['pesan', 'tiket', 'aplikasi', 'lambat']	pesan tiket aplikasi lambat
3	sumpah kecewa banget aku belabelain begadang t...	['sumpah', 'kecewa', 'banget', 'aku', 'belabelain']	['sumpah', 'kecewa', 'banget', 'belabelain']	sumpah kecewa banget belabelain begadang tunggu...
4	kenapa aplikasi pemerintah pada kurang semua ...	['kenapa', 'aplikasi', 'pemerintah', 'pada', 'kurang', 'semua']	['aplikasi', 'pemerintah', 'niat', 'aplikasi']	aplikasi perintah niat aplikasi mudah sulit su...
...	...	...	...	...
8240	sudah melakukan pembayaran melalui kaya pay sa...	['sudah', 'melakukan', 'pembayaran', 'melalui']	['pembayaran', 'kaya', 'pay', 'saldo', 'kefoto']	bayar kaya pay saldo potong batas habis tiket ...
8241	daftar akunya buat emosi lihat otp di email pa...	['daftar', 'akunya', 'buat', 'emosi', 'lihat', 'otp', 'di', 'email', 'pa']	['daftar', 'akunya', 'emosi', 'lihat', 'otp']	daftar aku emosi lihat otp email pas lihat lan...
8242	pelayanan kaya komuter line dhoho kurang ramah...	['pelayanan', 'kaya', 'komuter', 'line', 'dhoho', 'kurang', 'ramah']	['pelayanan', 'kaya', 'komuter', 'line', 'dhoho']	layan kaya komuter line dhoho ramah yuliantose...
8243	sangat mengecewakan jauh hari saya mau refund ...	['sangat', 'mengecewakan', 'jauh', 'hari', 'saya', 'mau', 'refund']	['mengecewakan', 'refund', 'tiket', 'kereta']	kecewa refund tiket kereta pesan istri anak ri...
8244	aplikasi ribet daftar saja sulitperbaiki dong ...	['aplikasi', 'ribet', 'daftar', 'saja', 'sulitperbaiki', 'dong']	['aplikasi', 'ribet', 'daftar', 'sulitperbaiki']	aplikasi ribet daftar sulitperbaiki sistem top...

8245 rows × 4 columns

```
In [20]: 1 df = df[['stemming']]
2 df
```

```
Out[20]:
```

	stemming
0	mantap praktis bantu
1	instalnya banget butuh jam ulang masuk
2	pesan tiket aplikasi lambat
3	sumpah kecewa banget belabelain begadang tungg...
4	aplikasi perintah niat aplikasi mudah sulit su...
...	...
8240	bayar kaya pay saldo potong batas habis tiket ...
8241	daftar aku emosi lihat otp email pas lihat lan...
8242	layan kaya komuter line dhoho ramah yuliantose...
8243	kecewa refund tiket kereta pesan istri anak ri...
8244	aplikasi ribet daftar sulitperbaiki sistem top...

8245 rows × 1 columns

```
In [21]: 1 df.to_csv('translate.csv', index=False)
```



```
In [2]: 1 import pandas as pd
2 df = pd.read_csv('translated.csv')
3 df.head()
```

Out[2]:

	stemming	translate
0	mantap praktis bantu	great practical help
1	instalnya banget butuh jam ulang masuk	The installation really needs to be re-entered
2	pesan tiket aplikasi lambat	order slow application ticket
3	sumpah kecewa banget bela belain begadang tung...	I swear, I was really disappointed, I stayed u...
4	aplikasi perintah niat aplikasi mudah sulit su...	command application intention easy application...

```
In [3]: 1 import re
2 def clean_text(text):
3     # 1. Lowercasing
4     text = text.lower()
5
6     # 2. Menghilangkan tanda baca
7     text = re.sub(r'^\w\s', '', text)
8
9     # 3. Menghilangkan angka
10    text = re.sub(r'\d+', '', text)
11
12    # 4. Menghilangkan emoticon (ini bisa bervariasi, tetapi kita bisa
13    text = re.sub(r'^\w\s', '', text)
14
15    # 5. Menghilangkan huruf ganda
16    text = re.sub(r'(\.|\s)\1+', r'\1', text)
17
18    return text
```

```
In [4]: 1 # Terapkan fungsi pembersihan ke kolom 'content'
2 df['translate'] = df['translate'].apply(clean_text)
3 df
```

Out[4]:

		stemming	translate
0		mantap praktis bantu	great practical help
1		instalnya banget butuh jam ulang masuk	the instalation realy neds to be rentered
2		pesan tiket aplikasi lambat	order slow aplication ticket
3		sumpah kecewa banget bela belain begadang tung...	i swear i was realy disapointed i stayed up la...
4		aplikasi perintah niat aplikasi mudah sulit su...	comand aplication intention easy aplication di...
...		...	...
8219		bayar kaya pay saldo potong batas habis tiket ...	pay like pay balance cut the limit after the t...
8220		daftar aku emosi lihat otp email pas lihat lan...	register im emotional when i se the email otp ...
8221		layan kaya komuter line dhoho ramah yuliantose...	serve rich comuters line dhoho friendly yulian...
8222		kecewa refund tiket kereta pesan istri anak ri...	disapointed refund train ticket ordered by wif...
8223		aplikasi ribet daftar sulitperbaiki sistem top...	complicated registration aplication dificult t...
8224 rows × 2 columns			

```
In [5]: 1 df = df.dropna(subset=['stemming'])
2 len(df)
```

Out[5]: 8224

```
In [6]: 1 df = df.drop_duplicates(subset=['stemming'])
2 len(df)
```

Out[6]: 8221

```
In [7]: 1 from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
2 analyser = SentimentIntensityAnalyzer()
```

```
In [8]: 1 scores = [analyser.polarity_scores(sentence) for sentence in df['transl
2 print(scores)
3 df['compound_Score'] = [x['compound'] for x in scores]
```

```
[{'neg': 0.0, 'neu': 0.128, 'pos': 0.872, 'compound': 0.7783}, {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}, {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}, {'neg': 0.044, 'neu': 0.956, 'pos': 0.0, 'compound': -0.0516}, {'neg': 0.0, 'neu': 0.734, 'pos': 0.266, 'compound': 0.4404}, {'neg': 0.067, 'neu': 0.746, 'pos': 0.187, 'compound': 0.4939}, {'neg': 0.0, 'neu': 0.526, 'pos': 0.474, 'compound': 0.2023}, {'neg': 0.592, 'neu': 0.408, 'pos': 0.0, 'compound': -0.4404}, {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}, {'neg': 0.0, 'neu': 0.583, 'pos': 0.417, 'compound': 0.5106}, {'neg': 0.412, 'neu': 0.588, 'pos': 0.0, 'compound': -0.5423}, {'neg': 0.333, 'neu': 0.667, 'pos': 0.0, 'compound': -0.3612}, {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}, {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}, {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}, {'neg': 0.0, 'neu': 0.258, 'pos': 0.742, 'compound': 0.9081}, {'neg': 0.425, 'neu': 0.438, 'pos': 0.138, 'compound': -0.6808}, {'neg': 0.483, 'neu': 0.517, 'pos': 0.0, 'compound': -0.2023}, {'neg': 0.156, 'neu': 0.782, 'pos': 0.061, 'compound': -0.4215}, {'neg': 0.152, 'neu': 0.728, 'pos': 0.119, 'compound': -0.128}, {'neg': 0.351, 'neu': 0.649, 'pos': 0.0, 'compound': -0.4019}, {'neg': 0.122, 'neu': 0.541, 'pos': 0.338, 'compound': 0.4939}, {'neg': 0.068, 'neu': 0.685, 'pos': 0.246, 'compound': 0.7101}, {'neg': 0.0, 'neu': 0.225, 'pos': 0.775, 'compound': 0.0}]]
```

```
In [9]: 1 df.head()
```

Out[9]:

	stemming	translate	compound_Score
0	mantap praktis bantu	great practical help	0.7783
1	instalnya banget butuh jam ulang masuk	the instalation realy neds to be reentered	0.0000
2	pesan tiket aplikasi lambat	order slow aplication ticket	0.0000
3	sumpah kecewa banget bela belain begadang tung...	i swear i was realy disapointed i stayed up la...	-0.0516
4	aplikasi perintah niat aplikasi mudah sulit su...	comand aplication intention easy application di...	0.4404

```
In [10]: 1 df.loc[df['compound_Score'] > 0, 'sentiment'] = 'positive'
2 df.loc[df['compound_Score'] == 0, 'sentiment'] = 'neutral'
3 df.loc[df['compound_Score'] < 0, 'sentiment'] = 'negative'
4 df.head()
```

Out[10]:

	stemming	translate	compound_Score	sentiment
0	mantap praktis bantu	great practical help	0.7783	positive
1	instalnya banget butuh jam ulang masuk	the instalation realy neds to be reentered	0.0000	neutral
2	pesan tiket aplikasi lambat	order slow aplication ticket	0.0000	neutral
3	sumpah kecewa banget bela belain begadang tung...	i swear i was realy disapointed i stayed up la...	-0.0516	negative
4	aplikasi perintah niat aplikasi mudah sulit su...	comand aplication intention easy aplication di...	0.4404	positive

```
In [11]: 1 df.to_csv('labeling.csv')
```

```
In [ ]: 1
```



```
In [1]: 1 import pandas as pd
2 df = pd.read_csv('labeling.csv')
3 df.head(15)
```

Out[1]:

	Unnamed: 0	stemming	translate	compound_Score	sentiment
0	0	mantap praktis bantu	great practical help	0.7783	positive
1	1	instalnya banget butuh jam ulang masuk	the instalation realy neds to be rentered	0.0000	neutral
2	2	pesan tiket aplikasi lambat	order slow aplication ticket	0.0000	neutral
3	3	sumpah kecewa banget bela belain begadang tung...	i swear i was realy disapointed i stayed up la...	-0.0516	negative
4	4	aplikasi perintah niat aplikasi mudah sulit su...	comand aplication intention easy aplication di...	0.4404	positive
5	5	kak kereta jam sore malam jurus krian jombang ...	sis the train at midnight krian jombang is emp...	0.4939	positive
6	6	top banget kak	realy top sis	0.2023	positive
7	7	aplikasi cacat bug	bug defective aplication	-0.4404	negative
8	8	akurat	acurate	0.0000	neutral
9	9	ok bagus nti kumat bintang	ok thats god then the stars wil recur	0.5106	positive
10	10	aplikasi buruk daftar susah banget	bad aplication registering is realy difficult	-0.5423	negative
11	11	aplikasi fungsi pesan tiket payah	the aps ticket boking function sucks	-0.3612	negative
12	12	update daftar stasiun aplikasi	update the aplications station list	0.0000	neutral
13	13	aplikasi publik rumit suruh baharu lambat	complicated public aplications tel updates to ...	0.0000	neutral
14	14	update sulit login	update is dificult to log in to	0.0000	neutral

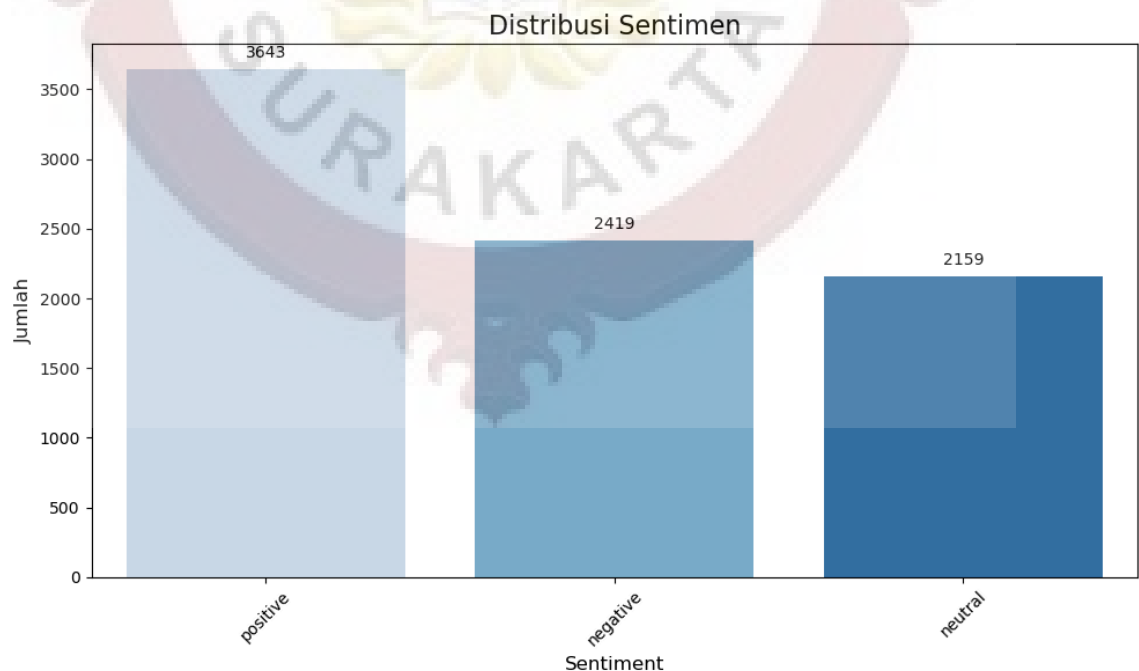
```
In [2]: 1 df = df[['stemming', 'compound_Score', 'sentiment']]
2 df.head()
```

Out[2]:

	stemming	compound_Score	sentiment
0	mantap praktis bantu	0.7783	positive
1	instalnya banget butuh jam ulang masuk	0.0000	neutral
2	pesan tiket aplikasi lambat	0.0000	neutral
3	sumpah kecewa banget bela belain begadang tung...	-0.0516	negative
4	aplikasi perintah niat aplikasi mudah sulit su...	0.4404	positive

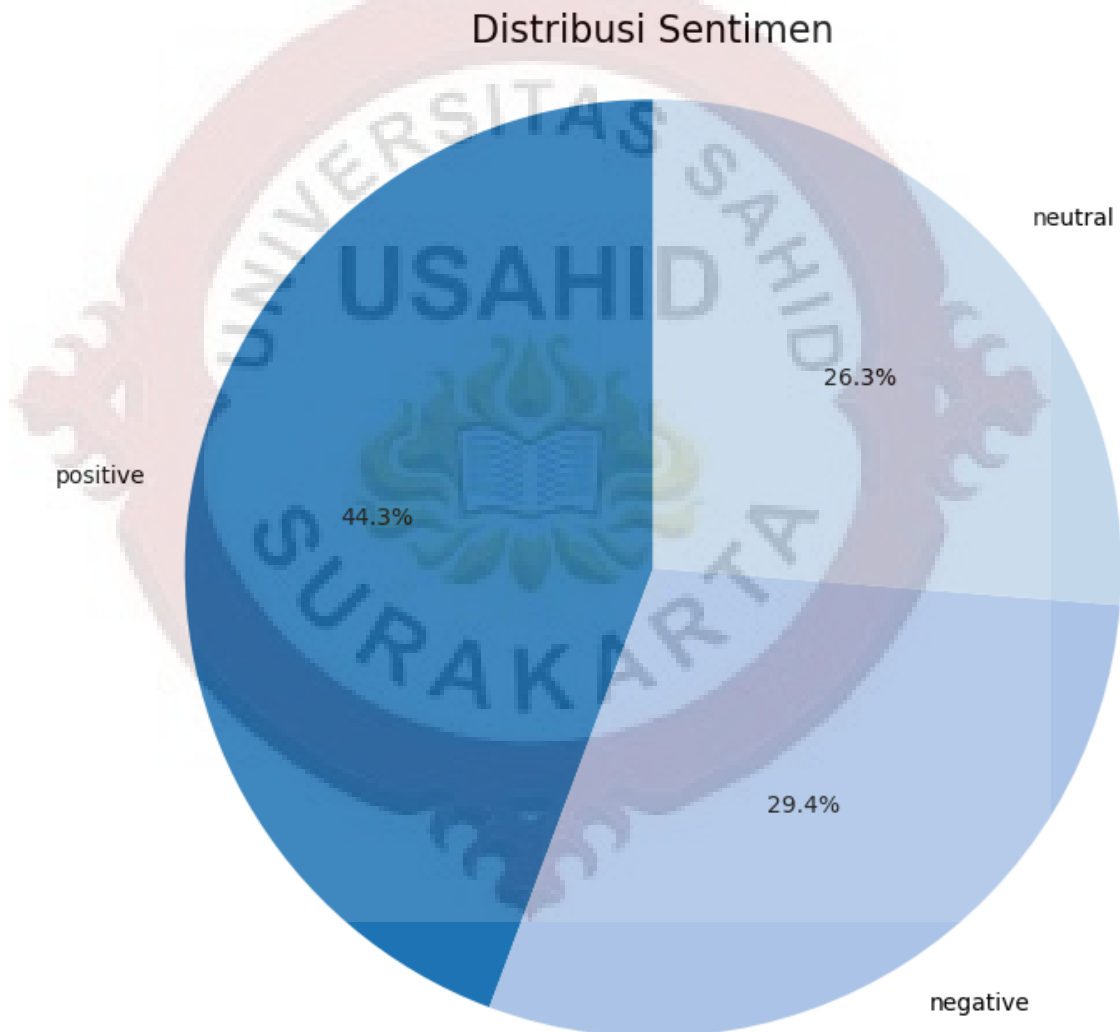

```
In [3]: 1 import matplotlib.pyplot as plt
        2 import seaborn as sns
```

```
In [4]: 1 sentiment_counts = df['sentiment'].value_counts().reset_index()
        2 sentiment_counts.columns = ['sentiment', 'count']
        3
        4 # Membuat figure dengan ukuran yang diinginkan
        5 plt.figure(figsize=(10, 6))
        6
        7 # Menggunakan seaborn untuk countplot
        8 ax = sns.barplot(data=sentiment_counts, x='sentiment', y='count', palette=
        9
        10 # Menambahkan judul dan label
        11 plt.title('Distribusi Sentimen', fontsize=16)
        12 plt.xlabel('Sentiment', fontsize=12)
        13 plt.ylabel('Jumlah', fontsize=12)
        14
        15 # Menambahkan anotasi pada setiap batang
        16 for p in ax.patches:
        17     ax.annotate(f'{int(p.get_height())}',
        18               (p.get_x() + p.get_width() / 2., p.get_height()),
        19               ha='center', va='center',
        20               xytext=(0, 10),
        21               textcoords='offset points')
        22
        23 plt.xticks(rotation=45) # Memutar label sentiment agar mudah dibaca
        24 plt.tight_layout() # Menyesuaikan plot agar semua elemen pas tanpa tum
        25 plt.show()
```



In [5]:

```
1 # Menghitung distribusi sentimen
2 sentiment_counts = df['sentiment'].value_counts()
3
4 # Membuat pie chart dengan warna pastel
5 pastel_colors = ['#1f77b4', '#aec7e8', '#c6dbef'] # Contoh warna pastel
6 plt.figure(figsize=(8, 8))
7 plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct='%1.1f%%')
8
9 # Menambahkan judul
10 plt.title('Distribusi Sentimen', fontsize=16)
11 plt.axis('equal') # Equal aspect ratio ensures that pie chart is circular
12
13 # Menampilkan pie chart
14 plt.show()
```



```

In [7]: 1 # Periksa apakah DataFrame memiliki kolom 'sentiment' dan 'stemming'
2 required_columns = ['sentiment', 'stemming']
3 missing_columns = [col for col in required_columns if col not in df.col
4 if missing_columns:
5     raise ValueError(f"DataFrame 'df' tidak memiliki kolom: {'', '.join(
6
7 # Dapatkan daftar unik sentimen dari DataFrame
8 sentiments = df['sentiment'].unique()
9
10 # Buat WordCloud untuk setiap kategori sentimen
11 for sentiment in sentiments:
12     try:
13         # Filter data berdasarkan kategori sentimen
14         filtered_df = df[df['sentiment'] == sentiment]
15
16         # Periksa apakah ada data yang tersedia untuk kategori sentimen
17         if filtered_df.empty:
18             print(f"Tidak ada data untuk sentimen: {sentiment.capitalize()}")
19             continue
20
21         # Gabungkan semua teks menjadi satu string
22         text = ' '.join(filtered_df['stemming'])
23
24         # Buat objek WordCloud
25         wordcloud = WordCloud(width=800, height=400, background_color='
26
27         # Tampilkan WordCloud
28         plt.figure(figsize=(10, 5))
29         plt.imshow(wordcloud, interpolation='bilinear')
30         plt.axis('off') # Matikan sumbu
31         plt.title(f'WordCloud untuk Sentimen: {sentiment.capitalize()}')
32         plt.show()
33     except Exception as e:
34         print(f"Terjadi kesalahan saat membuat WordCloud untuk sentimen
35

```

WordCloud untuk Sentimen: Positive



WordCloud untuk Sentimen: Neutral



WordCloud untuk Sentimen: Negative




```
In [11]: 1 positive = df[(df['sentiment'] == 'neutral') &
2           (df['stemming'].str.contains('pesan', case=False))]
3 text = ' '.join(positive['stemming'])
4
5 # Membuat WordCloud
6 wordcloud = WordCloud(width=400, height=200, background_color='white').
7
8 # Menampilkan WordCloud
9 plt.figure(figsize=(8, 6))
10 plt.imshow(wordcloud, interpolation='bilinear')
11 plt.axis('off') # Menyembunyikan sumbu
12 plt.title('WordCloud untuk Ulasan netral terbanyak', fontsize=16)
13 plt.show()
```



In [8]:

```
1 import re
2 from collections import Counter
3
4 # Fungsi untuk mendapatkan tiga kata teratas
5 def get_top_words(sentiment):
6     # Menggabungkan semua ulasan untuk kategori ini
7     text = ' '.join(df[df['sentiment'] == sentiment]['stemming'])
8
9     # Menghapus tanda baca dan memecah menjadi kata-kata
10    words = re.findall(r'\w+', text.lower())
11
12    # Menghitung frekuensi kata
13    word_counts = Counter(words)
14
15    # Mendapatkan tiga kata teratas
16    top_words = word_counts.most_common(10)
17
18    return top_words
19
20 # Mencetak tiga kata teratas untuk setiap kategori
21 sentiments = df['sentiment'].unique()
22 for sentiment in sentiments:
23     top_words = get_top_words(sentiment)
24     print(f' kata teratas untuk sentimen "{sentiment}":')
25     for word, count in top_words:
26         print(f' - {word}: {count}')
27     print() # Baris kosong untuk pemisah
```

kata teratas untuk sentimen "positive":

- aplikasi: 1851
- tiket: 1622
- kaya: 1467
- nya: 978
- tolong: 798
- masuk: 736
- baik: 637
- pesan: 597
- bayar: 578
- saldo: 552

kata teratas untuk sentimen "neutral":

- aplikasi: 758
- tiket: 677
- pesan: 306
- nya: 286
- error: 272
- update: 190
- masuk: 177
- daftar: 169
- kereta: 164
- susah: 162

kata teratas untuk sentimen "negative":

- aplikasi: 1331
- tiket: 1167
- nya: 569
- bayar: 562
- pesan: 387
- banget: 379
- masuk: 374
- saldo: 303
- beli: 297
- kaya: 294

```
In [ ]: 1 df.to_csv('final_dataset.csv', index=False)
```

```
In [ ]: 1
```

```
In [1]: 1 import pandas as pd
2 df = pd.read_csv('final_dataset.csv')
3 df.head()
```

```
Out[1]:
```

	stemming	compound_Score	sentiment
0	mantap praktis bantu	0.7783	positive
1	instalnya banget butuh jam ulang masuk	0.0000	neutral
2	pesan tiket aplikasi lambat	0.0000	neutral
3	sumpah kecewa banget bela belain begadang tung...	-0.0516	negative
4	aplikasi perintah niat aplikasi mudah sulit su...	0.4404	positive

```
In [2]: 1 from sklearn.model_selection import train_test_split
2 from sklearn.feature_extraction.text import TfidfVectorizer
3 from sklearn.naive_bayes import MultinomialNB
4 from sklearn.linear_model import LogisticRegression
5 from sklearn.ensemble import RandomForestClassifier
6 from sklearn.metrics import classification_report, accuracy_score, conf
7 import numpy as np
```

```
In [3]: 1 import seaborn as sns
2 import matplotlib.pyplot as plt
3 from sklearn.metrics import confusion_matrix
```

```
In [4]: 1 X = df['stemming']
2 y = df['sentiment']
```

```
In [5]: 1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
In [6]: 1 sentiment_counts_train = y_train.value_counts()
2 sentiment_counts_test = y_test.value_counts()
3
4 print("Jumlah data latih berdasarkan sentimen:")
5 print(sentiment_counts_train)
6
7 print("\nJumlah data uji berdasarkan sentimen:")
8 print(sentiment_counts_test)
```

```
Jumlah data latih berdasarkan sentimen:
sentiment
positive    2914
negative    1917
neutral     1745
Name: count, dtype: int64
```

```
Jumlah data uji berdasarkan sentimen:
sentiment
positive     729
negative     502
neutral      414
Name: count, dtype: int64
```

```
In [7]: 1 tfidf = TfidfVectorizer()
2 X_train_tfidf = tfidf.fit_transform(X_train)
3 X_test_tfidf = tfidf.transform(X_test)
```

```
In [8]: 1 nb_model = MultinomialNB()
2 nb_model.fit(X_train_tfidf, y_train)
3 y_pred_nb = nb_model.predict(X_test_tfidf)
4
5 print("Naive Bayes Performance:")
6 print(classification_report(y_test, y_pred_nb, zero_division=0))
7 print("Accuracy:", accuracy_score(y_test, y_pred_nb))
```

Naive Bayes Performance:

	precision	recall	f1-score	support
negative	0.81	0.34	0.48	502
neutral	0.81	0.21	0.33	414
positive	0.54	0.97	0.69	729
accuracy			0.59	1645
macro avg	0.72	0.51	0.50	1645
weighted avg	0.69	0.59	0.54	1645

Accuracy: 0.5890577507598784

```
In [9]: 1 lr_model = LogisticRegression(max_iter=1000)
2 lr_model.fit(X_train_tfidf, y_train)
3 y_pred_lr = lr_model.predict(X_test_tfidf)
4
5 print("Logistic Regression Performance:")
6 print(classification_report(y_test, y_pred_lr, zero_division=0))
7 print("Accuracy:", accuracy_score(y_test, y_pred_lr))
```

Logistic Regression Performance:

	precision	recall	f1-score	support
negative	0.82	0.68	0.74	502
neutral	0.73	0.80	0.76	414
positive	0.83	0.88	0.85	729
accuracy			0.80	1645
macro avg	0.79	0.79	0.79	1645
weighted avg	0.80	0.80	0.80	1645

Accuracy: 0.7987841945288754

```
In [10]: 1 rf_model = RandomForestClassifier()
2 rf_model.fit(X_train_tfidf, y_train)
3 y_pred_rf = rf_model.predict(X_test_tfidf)
4
5 print("Random Forest Performance:")
6 print(classification_report(y_test, y_pred_rf, zero_division=0))
7 print("Accuracy:", accuracy_score(y_test, y_pred_rf))
```

Random Forest Performance:

	precision	recall	f1-score	support
negative	0.84	0.62	0.71	502
neutral	0.72	0.86	0.79	414
positive	0.80	0.86	0.83	729
accuracy			0.79	1645
macro avg	0.79	0.78	0.78	1645
weighted avg	0.79	0.79	0.78	1645

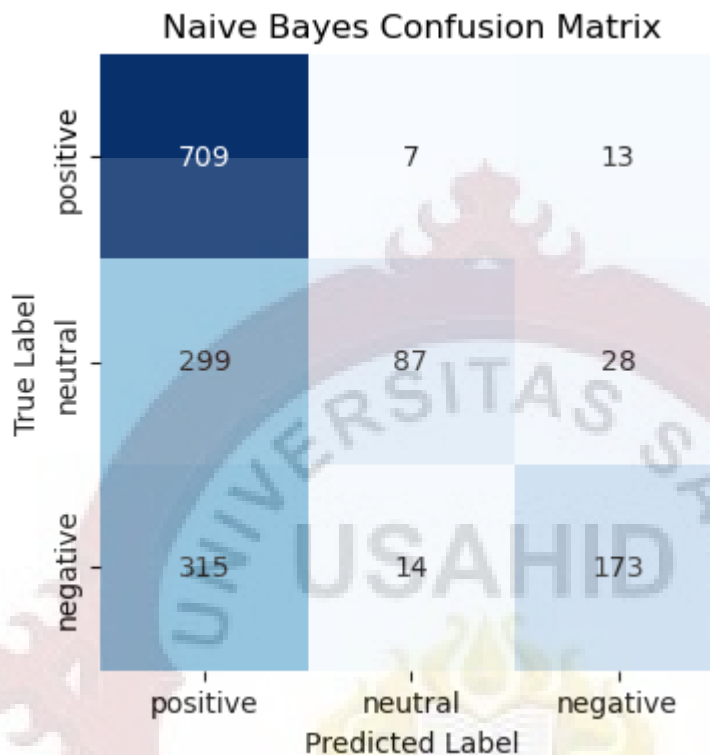
Accuracy: 0.7872340425531915

```
In [11]: 1 def plot_confusion_matrix(y_true, y_pred, classes, title='Confusion Mat
2 cm = confusion_matrix(y_true, y_pred, labels=classes)
3 cm_df = pd.DataFrame(cm, index=classes, columns=classes)
4
5 plt.figure(figsize=(4, 4))
6 sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues', cbar=False,
7             xticklabels=classes, yticklabels=classes)
8 plt.title(title)
9 plt.xlabel('Predicted Label')
10 plt.ylabel('True Label')
11 plt.show()
```



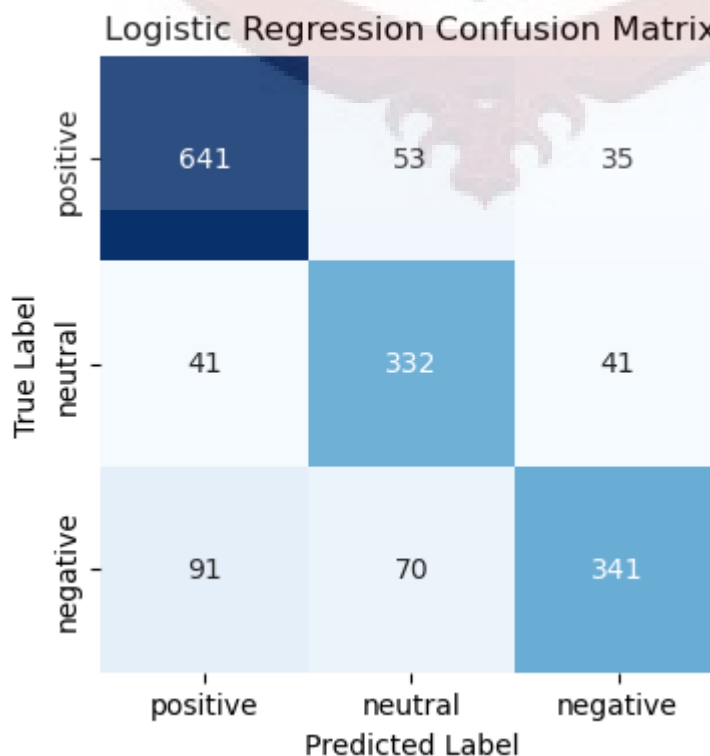
```
In [12]: 1 classes = y.unique() # atau Anda bisa secara manual menyebutkan kelas
          2
          3 print("Confusion Matrix for Naive Bayes:")
          4 plot_confusion_matrix(y_test, y_pred_nb, classes, title='Naive Bayes Co
```

Confusion Matrix for Naive Bayes:



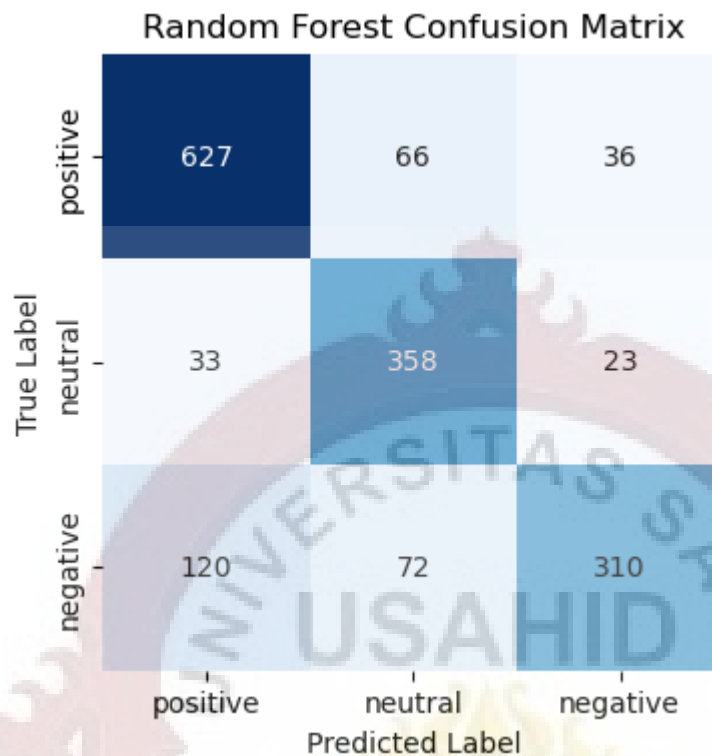
```
In [13]: 1 print("Confusion Matrix for Logistic Regression:")
          2 plot_confusion_matrix(y_test, y_pred_lr, classes, title='Logistic Regre
```

Confusion Matrix for Logistic Regression:



```
In [14]: 1 print("Confusion Matrix for Random Forest:")
2 plot_confusion_matrix(y_test, y_pred_rf, classes, title='Random Forest')
```

Confusion Matrix for Random Forest:



```
In [15]: 1 results = pd.DataFrame({
2     'Model': ['Naive Bayes', 'Logistic Regression', 'Random Forest'],
3     'Accuracy': [accuracy_score(y_test, y_pred_nb),
4     accuracy_score(y_test, y_pred_lr),
5     accuracy_score(y_test, y_pred_rf)]
6 })
7
8 print("Model Comparison:")
9 print(results)
```

Model Comparison:

	Model	Accuracy
0	Naive Bayes	0.589058
1	Logistic Regression	0.798784
2	Random Forest	0.787234

```

In [16]: 1 from sklearn.metrics import accuracy_score, f1_score, recall_score
2
3 # Fungsi untuk menghitung metrik
4 def evaluate_model(y_true, y_pred):
5     accuracy = accuracy_score(y_true, y_pred)
6     f1 = f1_score(y_true, y_pred, average='weighted') # average='weigh
7     recall = recall_score(y_true, y_pred, average='weighted') # averag
8     return accuracy, f1, recall
9
10 # Evaluasi Naive Bayes
11 accuracy_nb, f1_nb, recall_nb = evaluate_model(y_test, y_pred_nb)
12
13 # Evaluasi Logistic Regression
14 accuracy_lr, f1_lr, recall_lr = evaluate_model(y_test, y_pred_lr)
15
16 # Evaluasi Random Forest
17 accuracy_rf, f1_rf, recall_rf = evaluate_model(y_test, y_pred_rf)

```

```

In [17]: 1 results_df = pd.DataFrame({
2     'Model': ['Naive Bayes', 'Logistic Regression', 'Random Forest'],
3     'Accuracy': [accuracy_nb, accuracy_lr, accuracy_rf],
4     'F1 Score': [f1_nb, f1_lr, f1_rf],
5     'Recall': [recall_nb, recall_lr, recall_rf]
6 })
7
8 print("Model Comparison:")
9 print(results_df)

```

Model Comparison:

	Model	Accuracy	F1 Score	Recall
0	Naive Bayes	0.589058	0.537599	0.589058
1	Logistic Regression	0.798784	0.797020	0.798784
2	Random Forest	0.787234	0.783517	0.787234

```
In [18]: 1 best_model_accuracy = results_df.loc[results_df['Accuracy'].idxmax()]
2 best_model_f1 = results_df.loc[results_df['F1 Score'].idxmax()]
3 best_model_recall = results_df.loc[results_df['Recall'].idxmax()]
4
5 print("\nBest Model based on Accuracy:")
6 print(best_model_accuracy)
7
8 print("\nBest Model based on F1 Score:")
9 print(best_model_f1)
10
11 print("\nBest Model based on Recall:")
12 print(best_model_recall)
```

Best Model based on Accuracy:
Model Logistic Regression
Accuracy 0.798784
F1 Score 0.79702
Recall 0.798784
Name: 1, dtype: object

Best Model based on F1 Score:
Model Logistic Regression
Accuracy 0.798784
F1 Score 0.79702
Recall 0.798784
Name: 1, dtype: object

Best Model based on Recall:
Model Logistic Regression
Accuracy 0.798784
F1 Score 0.79702
Recall 0.798784
Name: 1, dtype: object

In []:

1