

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Pengumpulan Data

Proses pengumpulan data dilakukan dengan menggunakan Bahasa Python dan *library python scrapper* yang dijalankan di Google Collaboratory. Berikut adalah tahapan-tahapan pengambilan data ulasan:

1. *Install Library Google Play Scrapper* kemudian lakukan impor *reviews* ke dalam *Notebook Google Collaboratory*.

```
[ ] !pip install google-play-scraper  
  
Collecting google-play-scraper  
  Downloading google_play_scraper-1.2.7-py3-none-any.whl.metadata (50 kB)  
      50.2/50.2 kB 2.1 MB/s eta 0:00:00  
  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
```

```
from google_play_scraper import app, reviews  
import pandas as pd  
import datetime
```

google_play_scraper: Library ini digunakan untuk mengambil informasi aplikasi dan ulasan dari Google Play Store. pandas: Digunakan untuk bekerja dengan data dalam format DataFrame. datetime: Digunakan untuk menentukan rentang waktu (start_date dan end_date).

Gambar 4. 1 Install Library Google Play Scrapper

Sumber: Data yang diolah (2025)

UNIVERSITAS
NUSA MANDIRI

```

from google_play_scraper import reviews, Sort

app_id = 'com.dev.jakone.mbanking'

def get_reviews(app_id, lang='id', count=5000, sort=Sort.NEWEST, filter_score_with=None, filter_device_with=None, continuation_token=None):
    try:
        result, continuation_token = reviews(
            app_id=app_id,
            lang=lang,
            country='id',
            sort=sort,
            count=count,
            filter_score_with=filter_score_with,
            filter_device_with=filter_device_with,
            continuation_token=continuation_token
        )

        return result, continuation_token
    except Exception as e:
        print("Error:", e)
        return None, None

reviews, continuation_token = get_reviews(app_id)

if reviews is not None:
    print("Jumlah ulasan:", len(reviews))
    if len(reviews) > 0:
        print("Contoh ulasan:")
        print(reviews[0])
    else:
        print("Tidak dapat mengambil ulasan.")

```

Jumlah ulasan: 5000

Gambar 4. 2 Notebook Google Collaboratory

Sumber: Data yang diolah (2025)

2. Kemudian *Scrapping* sesuai dengan app id *Google Play Store* dari JakOne Mobile.

Contoh ulasan:

```

{'reviewId': 'e211fd67-65ac-4c2d-8f86-43538a24bb81', 'userName':
'Pengguna Google', 'userImage': 'https://play-
lh.googleusercontent.com/EGemoI2NTXmTsBVtJqk8jxF9rh8ApRWfsIMQSt2u
E4OcpQqbFu7f7NbTK051x80nuSijCz7sc3a277R67g', 'content': 'Aplikasi
jakone mobile sangat keren bagus karena fitur fiturnya sangat
lengkap dan mudah difahami pokonya mantap 👍👍👍', 'score': 5,
'thumbsUpCount': 0, 'reviewCreatedVersion': None, 'at':
datetime.datetime(2025, 1, 13, 7, 19, 41), 'replyContent': 'Terima
kasih atas ulasannya. Semoga fitur-fitur JakOne Mobile memudahkan
transaksi perbankan Bapak / Ibu.\n', 'repliedAt':
datetime.datetime(2025, 1, 13, 9, 55, 31), 'appVersion': None}

```

3. Kemudian data di impor kedalam format CSV.

```
[ ] import csv

def export_to_csv(reviews, file_name='hasil_scraper_ulasan_app_jakOne.csv'):
    if reviews:
        fieldnames = ['Review ID', 'Username', 'Rating', 'Review Text', 'Date']

        with open(file_name, mode='w', newline='', encoding='utf-8') as csv_file:
            writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
            writer.writeheader()

            for review in reviews:
                writer.writerow({
                    'Review ID': review['reviewId'],
                    'Username': review['userName'],
                    'Rating': review['score'],
                    'Review Text': review['content'],
                    'Date': review['at']
                })

        print(f"Data berhasil diekspor ke '{file_name}'")
    else:
        print("Tidak ada data ulasan untuk diekspor.")

export_to_csv(reviews)
```

➤ Data berhasil diekspor ke 'hasil_scraper_ulasan_app_jakOne.csv'

Gambar 4. 3 Format CSV

Sumber: Data yang diolah (2025)

4. Menampilkan data hasil scrapper ulasan JakOne Mobile.

```
[ ] import pandas as pd

data = pd.read_csv("hasil_scraper_ulasan_app_jakOne.csv")
data.info()
```

➤ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 5 columns):
Column Non-null Count Dtype

0 Review ID 5000 non-null object
1 Username 5000 non-null object
2 Rating 5000 non-null int64
3 Review Text 5000 non-null object
4 Date 5000 non-null object
dtypes: int64(1), object(4)
memory usage: 195.4+ KB

```
[ ] data.head(10)
```

➤

	Review ID	Username	Rating	Review Text	Date
0	e211fd67-65ac-4c2d-8f86-43538a24bb81	Pengguna Google	5	Aplikasi jakone mobile sangat keren bagus kare...	2025-01-13 07:19:41
1	f5152b64-fb0e-46ef-b972-763afab4dc61	Pengguna Google	5	memudahkan transaksi kemanapun, tanpa ngelag l...	2025-01-13 06:24:28
2	e137e6ff-1ef4-4fb9-8a1e-fcb4dc9c942a	Pengguna Google	5	Aplikasi ini membantu banget dengan tampilan y...	2025-01-13 02:30:56
3	b4f89480-63eb-4c2d-b03f-6b7be0ad929d	Pengguna Google	5	Aplikasinya bagus banget fiturnya memudahkan u...	2025-01-13 02:27:36
4	937bda7a-fda6-4fdc-b37a-1f966d346591	Pengguna Google	5	Aplikasinya bagus, fiturnya lengkap dan mudah ...	2025-01-13 00:25:38
5	8b84ad6d-7ca8-415c-9ce0-5f81f4235234	Pengguna Google	5	aplikasi ini rekomendasi bgt buat topup game m...	2025-01-13 00:23:31
6	d257678a-42ab-4dd4-8143-272f916776d3	Pengguna Google	5	Transaksi disini lebih gampang , fiturnya juga...	2025-01-12 06:01:24
7	db56debb-a209-489d-92a8-3cccf96ae3df	Pengguna Google	5	Aplikasinya praktis dan mudah digunakan juga b...	2025-01-12 05:58:56

Gambar 4. 4 Data hasil scrapper ulasan JakOne Mobile

Sumber: Data yang diolah (2025)

4.2 Preprocessing Data

Berikut adalah proses *Preprocessing* Data.

1. Proses hapus data duplikat.

Dari 5000 data yang di ambil dari ulasan Google Play Store akan di hapus data yang sama dan didapatkan hasil sebanyak 4611 data ulasan dan data yang diambil adalah data yang paling pertama.

```
[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 4 columns):
#   column      Non-Null Count  Dtype
---  -
0   Date        5000 non-null        object
1   Username    5000 non-null        object
2   Rating      5000 non-null        int64
3   Review Text  5000 non-null        object
dtypes: int64(1), object(3)
memory usage: 156.4+ KB

[ ] df.drop_duplicates(subset = "Review Text", keep = 'first', inplace = True)

[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 4611 entries, 0 to 4999
Data columns (total 4 columns):
#   column      Non-Null Count  Dtype
---  -
0   Date        4611 non-null        object
1   Username    4611 non-null        object
2   Rating      4611 non-null        int64
3   Review Text  4611 non-null        object
dtypes: int64(1), object(3)
memory usage: 180.1+ KB
```

Gambar 4. 5 Preprocessing Data

Sumber: Data yang diolah (2025)

UNIVERSITAS
NUSA MANDIRI

2. Wordcloud Sebelum *Preproccesing*

Proses ini menampilkan kata yang sering muncul sebelum data di bersihkan.

```
[ ] import pandas as pd
import numpy as np
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt

# Mengisi nilai NaN dengan string kosong ('')
df['Review Text'] = df['Review Text'].fillna('')

# Menggabungkan teks dari kolom 'content'
text = ' '.join(df['Review Text'].astype(str).tolist())

stopwords = set(STOPWORDS)
stopwords.update(['https', 'co', 'RT', '...', 'amp'])

wc = WordCloud(stopwords=stopwords, background_color="black", max_words=500, width=800, height=400)

wc.generate(text)

plt.figure(figsize=(10, 5))
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Gambar 4. 6 Wordcloud Sebelum Preproccesing

Sumber: Data yang diolah (2025)



Gambar 4. 7 Kata yang sering muncul sebelum data di bersihkan

Sumber: Data yang diolah (2025)

```
[ ] import matplotlib.pyplot as plt
    from collections import Counter

    text = " ".join(df["Review Text"])

    tokens = text.split()
    word_counts = Counter(tokens)

    top_words = word_counts.most_common(10)

    word, count = zip(*top_words)

    # Definisikan palet warna
    colors = plt.cm.tab10(range(len(word)))

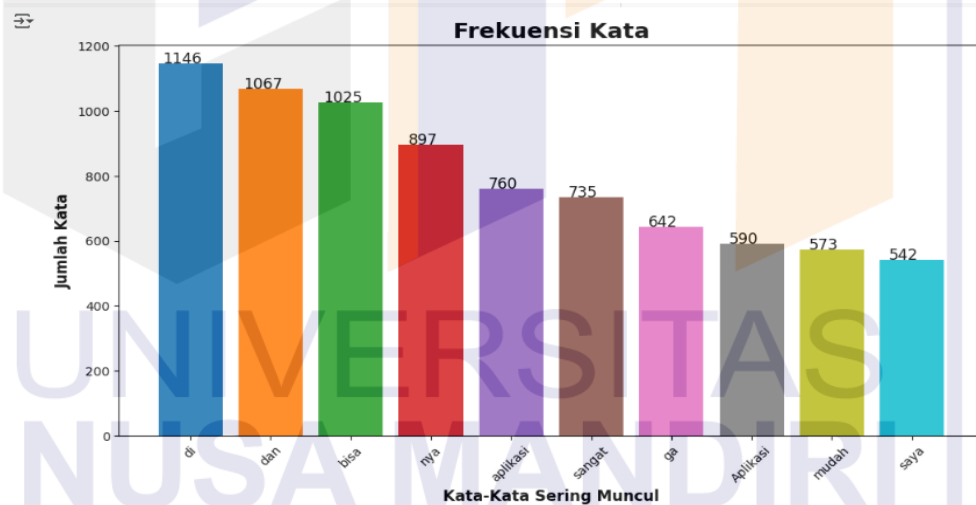
    plt.figure(figsize=(12, 6))
    bars = plt.bar(word, count, color=colors)
    plt.xlabel("Kata-Kata Sering Muncul", fontsize=12, fontweight='bold')
    plt.ylabel("Jumlah Kata", fontsize=12, fontweight='bold')
    plt.title("Frekuensi Kata", fontsize=18, fontweight='bold')
    plt.xticks(rotation=45)

    # Menambahkan angka rata tengah di atas setiap bar
    for bar, num in zip(bars, count):
        plt.text(bar.get_x() + bar.get_width() / 2 - 0.1, num + 1, str(num), fontsize=12, color='black', ha='center')

    plt.show()
```

Gambar 4. 8 Worcloud Sebelum Preprocessing

Sumber: Data yang diolah (2025)



Gambar 4. 9 Frekuensi Kata Sebelum Preprocessing

Sumber: Data yang diolah (2025)

Dari grafik diatas dapat dilihat bahwa kata-kata yang tidak mempunyai arti masih banyak dan ini perlu dilanjutkan ke proses *cleaning* data.

3. Proses *Cleaning* Data

Cleaning bertujuan untuk membersihkan data *reviews* yang sudah diperoleh. Komponen-komponen yang dibersihkan merupakan komponen yang tidak berarti atau tidak relevan untuk proses pengklasifikasian data. Komponen tersebut di antaranya karakter spesial pada ulasan di antaranya *hashtag*, URL, link, tanda baca, angka, dan simbol emoji.

```
[ ] import re
import string
import nltk

# Fungsi untuk menghapus URL
def remove_URL(tweet):
    if tweet is not None and isinstance(tweet, str):
        url = re.compile(r'https?://\S+|www\.\S+')
        return url.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus HTML
def remove_html(tweet):
    if tweet is not None and isinstance(tweet, str):
        html = re.compile(r'<.*?>')
        return html.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus emoji
def remove_emoji(tweet):
    if tweet is not None and isinstance(tweet, str):
        emoji_pattern = re.compile("["
            u"\U0001F600-\U0001F64F" # emoticons
            u"\U0001F300-\U0001F5FF" # symbols & pictographs
            u"\U0001F680-\U0001F6FF" # transport & map symbols
            u"\U0001F700-\U0001F77F" # alchemical symbols
            u"\U0001F780-\U0001F7FF" # Geometric Shapes Extended
            u"\U0001F800-\U0001F8FF" # Supplemental Arrows-C
            u"\U0001F900-\U0001F9FF" # Supplemental Symbols and Pictographs
```

Gambar 4. 10 Cleaning

Sumber: Data yang diolah (2025)

```

        u"\U0001FA00-\U0001FA6F" # Chess Symbols
        u"\U0001FA70-\U0001FAFF" # Symbols and Pictographs Extended-A
        u"\U0001F004-\U0001F0CF" # Additional emoticons
        u"\U0001F1E0-\U0001F1FF" # flags
        "]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', tweet)
else:
    return tweet

# Fungsi untuk menghapus simbol
def remove_symbols(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'[^\w-zA-Z0-9\s]', '', tweet)
    return tweet

# Fungsi untuk menghapus angka
def remove_numbers(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'\d', '', tweet)
    return tweet

df['cleaning'] = df['Review Text'].apply(lambda x: remove_URL(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_html(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_emoji(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_symbols(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_numbers(x))

df.head(10)

```

Gambar 4. 11 Reviews

Sumber: Data yang diolah (2025)

	Date	Username	Rating	Review Text	cleaning
0	2025-01-13 07:19:41	Pengguna Google	5	Aplikasi jakone mobile sangat keren bagus kare...	Aplikasi jakone mobile sangat keren bagus kare...
1	2025-01-13 06:24:28	Pengguna Google	5	memudahkan transaksi kemanapun, tanpa ngelag l...	memudahkan transaksi kemanapun tanpa ngelag la...
2	2025-01-13 02:30:56	Pengguna Google	5	Aplikasi ini membantu banget dengan tampilan y...	Aplikasi ini membantu banget dengan tampilan y...
3	2025-01-13 02:27:36	Pengguna Google	5	Aplikasinya bagus banget fiturnya memudahkan u...	Aplikasinya bagus banget fiturnya memudahkan u...
4	2025-01-13 00:25:38	Pengguna Google	5	Aplikasinya bagus, fiturnya lengkap dan mudah ...	Aplikasinya bagus fiturnya lengkap dan mudah d...
5	2025-01-13 00:23:31	Pengguna Google	5	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...
6	2025-01-12 06:01:24	Pengguna Google	5	Transaksi disini lebih gampang , fiturnya juga...	Transaksi disini lebih gampang fiturnya juga ...
7	2025-01-12 05:58:56	Pengguna Google	5	Aplikasinya praktis dan mudah digunakan juga b...	Aplikasinya praktis dan mudah digunakan juga b...
8	2025-01-12 05:23:18	Pengguna Google	5	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...
9	2025-01-12 01:59:56	Pengguna Google	5	Aplikasi nya jelas dan sangat recommended buat...	Aplikasi nya jelas dan sangat recommended buat...

Gambar 4. 12 Hashtag

Sumber: Data yang diolah (2025)

4. Proses Case Folding

Pada proses ini huruf kapital pada data ulasan diganti menjadi huruf kecil (*lowercase*).

Data ulasan sudah diganti menjadi *lowercase*.

```
[ ] def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower()
        return lowercase_text
    else:
        return text

df['case_folding'] = df['cleaning'].apply(case_folding)
df.head(10)
```

	Date	Username	Rating	Review Text	cleaning	case_folding
0	2025-01-13 07:19:41	Pengguna Google	5	Aplikasi jakone mobile sangat keren bagus kare...	Aplikasi jakone mobile sangat keren bagus kare...	aplikasi jakone mobile sangat keren bagus kare...
1	2025-01-13 06:24:28	Pengguna Google	5	memudahkan transaksi kemanapun, tanpa ngelag l...	memudahkan transaksi kemanapun tanpa ngelag la...	memudahkan transaksi kemanapun tanpa ngelag la...
2	2025-01-13 02:30:56	Pengguna Google	5	Aplikasi ini membantu banget dengan tampilan y...	Aplikasi ini membantu banget dengan tampilan y...	aplikasi ini membantu banget dengan tampilan y...
3	2025-01-13 02:27:36	Pengguna Google	5	Aplikasinya bagus banget fiturnya memudahkan u...	Aplikasinya bagus banget fiturnya memudahkan u...	aplikasinya bagus banget fiturnya memudahkan u...
4	2025-01-13 00:25:38	Pengguna Google	5	Aplikasinya bagus, fiturnya lengkap dan mudah ...	Aplikasinya bagus fiturnya lengkap dan mudah d...	aplikasinya bagus fiturnya lengkap dan mudah d...
5	2025-01-13 00:23:31	Pengguna Google	5	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...
6	2025-01-12 06:01:24	Pengguna Google	5	Transaksi disini lebih gampang , fiturnya juga...	Transaksi disini lebih gampang fiturnya juga ...	transaksi disini lebih gampang fiturnya juga ...
7	2025-01-12 05:58:56	Pengguna Google	5	Aplikasinya praktis dan mudah digunakan juga b...	Aplikasinya praktis dan mudah digunakan juga b...	aplikasinya praktis dan mudah digunakan juga b...
8	2025-01-12 05:23:18	Pengguna Google	5	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...

Gambar 4. 13 Proses Case Folding

Sumber: Data yang diolah (2025)

5. Normalisasi kata

Normalisasi kata adalah mengganti kata tidak baku pada ulasan menjadi kata yang baku sesuai kamus besar Bahasa Indonesia, pada tahap ini peneliti mengambil *dataset* dari kaggle.com untuk *dataset* kamus Bahasa Indonesia.

```
[ ] import pandas as pd

# Fungsi penggantian kata tidak baku
def replace_taboo_words(text, kamus_tidak_baku):
    if isinstance(text, str):
        words = text.split()
        replaced_words = []
        kalimat_baku = []
        kata_diganti = []
        kata_tidak_baku_hash = []

        for word in words:
            if word in kamus_tidak_baku:
                baku_word = kamus_tidak_baku[word]
                if isinstance(baku_word, str) and all(char.isalpha() for char in baku_word):
                    replaced_words.append(baku_word)
                    kalimat_baku.append(baku_word)
                    kata_diganti.append(word)
                    kata_tidak_baku_hash.append(hash(word))
            # else:
            #     replaced_words.append('< karakter eksplisit >')
            #     kalimat_baku.append(word)
        else:
            replaced_words.append(word)
            replaced_text = ' '.join(replaced_words)
        else:
            replaced_text = ''
            kalimat_baku = []
            kata_diganti = []
            kata_tidak_baku_hash = []
```

Gambar 4. 14 Normalisasi kata

Sumber: Data yang diolah (2025)

```
[ ] return replaced_text, kalimat_baku, kata_diganti, kata_tidak_baku_hash

[ ] !pip install kaggle
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
!kaggle datasets download -d fornigulo/kamus-slag
!unzip kamus-slag.zip

Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.6.17)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle) (1.17.0)
Requirement already satisfied: certifi>=2023.7.22 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2024.12.14)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from kaggle) (4.67.1)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.3.0)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle) (6.2.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.10)
cp: cannot stat 'kaggle.json': No such file or directory
chmod: cannot access '/root/.kaggle/kaggle.json': No such file or directory
Dataset URL: https://www.kaggle.com/datasets/fornigulo/kamus-slag
License(s): RAIL
kamus-slag.zip: Skipping, found more recently modified local copy (use --force to force download)
Archive: kamus-slag.zip
replace kamuskatabaku.xlsx? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

Gambar 4. 15 Mengambil dataset dari kaggle.com

Sumber: Data yang diolah (2025)

```
[ ] # Baca dataset
data = pd.DataFrame(df[['Date', 'Username', 'Rating', 'Review Text', 'cleaning', 'case_folding']])
data.head(10)
```

	Date	Username	Rating	Review Text	cleaning	case_folding
0	2025-01-13 07:19:41	Pengguna Google	5	Aplikasi jakone mobile sangat keren bagus kare...	Aplikasi jakone mobile sangat keren bagus kare...	aplikasi jakone mobile sangat keren bagus kare...
1	2025-01-13 06:24:28	Pengguna Google	5	memudahkan transaksi kemanapun, tanpa ngelag l...	memudahkan transaksi kemanapun tanpa ngelag la...	memudahkan transaksi kemanapun tanpa ngelag la...
2	2025-01-13 02:30:56	Pengguna Google	5	Aplikasi ini membantu banget dengan tampilan y...	Aplikasi ini membantu banget dengan tampilan y...	aplikasi ini membantu banget dengan tampilan y...
3	2025-01-13 02:27:36	Pengguna Google	5	Aplikasinya bagus banget fiturnya memudahkan u...	Aplikasinya bagus banget fiturnya memudahkan u...	aplikasinya bagus banget fiturnya memudahkan u...
4	2025-01-13 00:25:38	Pengguna Google	5	Aplikasinya bagus, fiturnya lengkap dan mudah ...	Aplikasinya bagus fiturnya lengkap dan mudah d...	aplikasinya bagus fiturnya lengkap dan mudah d...
5	2025-01-13 00:23:31	Pengguna Google	5	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...
6	2025-01-12 06:01:24	Pengguna Google	5	Transaksi disini lebih gampang , fiturnya juga...	Transaksi disini lebih gampang fiturnya juga ...	transaksi disini lebih gampang fiturnya juga ...
7	2025-01-12 05:58:56	Pengguna Google	5	Aplikasinya praktis dan mudah digunakan juga b...	Aplikasinya praktis dan mudah digunakan juga b...	aplikasinya praktis dan mudah digunakan juga b...
8	2025-01-12 05:23:18	Pengguna Google	5	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...
9	2025-01-12 01:59:56	Pengguna Google	5	Aplikasi nya jelas dan sangat recommended buat...	Aplikasi nya jelas dan sangat recommended buat...	aplikasi nya jelas dan sangat recommended buat...

Gambar 4. 16 Kata-kata yang tidak baku sudah berubah

Sumber: Data yang diolah (2025)

```
[ ] # Baca kamus kata tidak baku
kamus_data = pd.read_excel("kamuskatabaku.xlsx")
kamus_tidak_baku = dict(zip(kamus_data['tidak_baku'], kamus_data['kata_baku']))

[ ] # Terapkan fungsi penggantian kata tidak baku
data['normalisasi'] = data['kata_baku'].apply(lambda x: replace_taboo_words(x, kamus_tidak_baku))
# data.head(100)

df = pd.DataFrame(data[['Date', 'Username', 'Rating', 'Review Text', 'cleaning', 'case_folding', 'normalisasi']])

df.head(10)
```

	Date	Username	Rating	Review Text	cleaning	case_folding	normalisasi	tokenize
0	2025-01-13 07:19:41	Pengguna Google	5	Aplikasi jakone mobile sangat keren bagus kare...	Aplikasi jakone mobile sangat keren bagus kare...	aplikasi jakone mobile sangat keren bagus kare...	aplikasi jakone mobile sangat keren bagus kare...	[aplikasi, jakone, mobile, sangat, keren, bagus]
1	2025-01-13 06:24:28	Pengguna Google	5	memudahkan transaksi kemanapun tanpa ngelag l...	memudahkan transaksi kemanapun tanpa ngelag l...	memudahkan transaksi kemanapun tanpa ngelag l...	memudahkan transaksi kemanapun tanpa ngelag l...	[memudahkan, transaksi, kemanapun, tanpa, ngelag, l...]
2	2025-01-13 02:30:56	Pengguna Google	5	Aplikasi ini membantu banget dengan tampilan y...	Aplikasi ini membantu banget dengan tampilan y...	aplikasi ini membantu banget dengan tampilan y...	aplikasi ini membantu banget dengan tampilan y...	[aplikasi, ini, membantu, banget, dengan, tamp...
3	2025-01-13 02:27:36	Pengguna Google	5	Aplikasinya bagus banget flumnya memudahkan u...	Aplikasinya bagus banget flumnya memudahkan u...	aplikasinya bagus banget flumnya memudahkan u...	aplikasinya bagus banget flumnya memudahkan u...	[aplikasinya, bagus, banget, flumnya, mudahah...
4	2025-01-13 00:25:38	Pengguna Google	5	Aplikasinya bagus, flumnya lengkap dan mudah d...	Aplikasinya bagus, flumnya lengkap dan mudah d...	aplikasinya bagus, flumnya lengkap dan mudah d...	aplikasinya bagus, flumnya lengkap dan mudah d...	[aplikasinya, bagus, flumnya, lengkap, dan, m...
5	2025-01-13 00:23:31	Pengguna Google	5	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	[aplikasi, ini, rekomendasi, banget, buat, top...
6	2025-01-12 06:01:24	Pengguna Google	5	Transaksi disini lebih gampang flumnya juga b...	Transaksi disini lebih gampang flumnya juga b...	transaksi disini lebih gampang flumnya juga b...	transaksi disini lebih gampang flumnya juga b...	[transaksi, disini, lebih, gampang, flumnya, ...]
7	2025-01-12 05:58:56	Pengguna Google	5	Aplikasinya praktis dan mudah digunakan juga b...	Aplikasinya praktis dan mudah digunakan juga b...	aplikasinya praktis dan mudah digunakan juga b...	aplikasinya praktis dan mudah digunakan juga b...	[aplikasinya, praktis, dan, mudah, digunakan, ...]
8	2025-01-12 05:23:18	Pengguna Google	5	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	[aplikasi, yang, bagus, buat, transaksi, dan, ...]
9	2025-01-12 01:59:56	Pengguna Google	5	Aplikasi nya jelas dan sangat recommended buat...	Aplikasi nya jelas dan sangat recommended buat...	aplikasi nya jelas dan sangat recommended buat...	aplikasi nya jelas dan sangat recommended buat...	[aplikasi, ya, jelas, dan, sangat, recommended...

Kata-kata yang tidak baku sudah berubah.

6. Tokenisasi

Tokenisasi bertujuan untuk memecah teks pada ulasan menjadi potongan-potongan kata, dan potongan kata tersebut disebut token ke dalam bentuk token.

```
[ ] def tokenize(text):
    tokens = text.split()
    return tokens

df['tokenize'] = df['normalisasi'].apply(tokenize)

df.head(10)
```

Gambar 4. 17 Tokenisasi

Sumber: Data yang diolah (2025)

7. Stopword Removal

Stopword removal berfungsi menghilangkan kata yang tidak memiliki makna dalam ulasan.

```
[ ] from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('indonesian')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

def remove_stopwords(text):
    return [word for word in text if word not in stop_words]

df['stopword removal'] = df['tokenize'].apply(lambda x: remove_stopwords(x))
df.head(10)
```

	Date	Username	Rating	Review Text	cleaning	case_folding	normalisasi	tokenize	stopword removal
0	2025-01-13 07:19:41	Pengguna Google	5	Aplikasi jakone mobile sangat keren bagus kare...	Aplikasi jakone mobile sangat keren bagus kare...	aplikasi jakone mobile sangat keren bagus kare...	aplikasi jakone, mobile, sangat, keren, bagus, kare...	[aplikasi, jakone, mobile, keren, bagus, flur...	[aplikasi, jakone, mobile, keren, bagus, flur...
1	2025-01-13 06:24:28	Pengguna Google	5	memudahkan transaksi kemanapun, tanpa ngelag l...	memudahkan transaksi kemanapun tanpa ngelag la...	memudahkan transaksi kemanapun tanpa ngelag la...	memudahkan, transaksi, kemanapun, tanpa, ngelag, l...	[memudahkan, transaksi, kemanapun, tanpa, ngelag, l...	[memudahkan, transaksi, kemanapun, ngelag, lag...
2	2025-01-13 02:30:56	Pengguna Google	5	Aplikasi ini membantu banget dengan tampilan y...	Aplikasi ini membantu banget dengan tampilan y...	aplikasi ini membantu banget dengan tampilan y...	aplikasi, ini, membantu, banget, dengan, tampil, y...	[aplikasi, ini, membantu, banget, dengan, tamp...	[aplikasi, membantu, banget, tampilan, sederhana...
3	2025-01-13 02:27:36	Pengguna Google	5	Aplikasinya bagus banget flurnya memudahkan u...	Aplikasinya bagus banget flurnya memudahkan u...	aplikasinya bagus banget flurnya memudahkan u...	aplikasinya, bagus, banget, flurnya, memudahkan, u...	[aplikasinya, bagus, banget, flurnya, memudahkan...	[aplikasinya, bagus, banget, flurnya, memudahkan...
4	2025-01-13 00:25:38	Pengguna Google	5	Aplikasinya bagus, flurnya lengkap dan mudah...	Aplikasinya bagus flurnya lengkap dan mudah d...	aplikasinya bagus flurnya lengkap dan mudah d...	aplikasinya, bagus, flurnya, lengkap, dan, mudah, d...	[aplikasinya, bagus, flurnya, lengkap, dan, m...	[aplikasinya, bagus, flurnya, lengkap, mudah...
5	2025-01-13 00:23:31	Pengguna Google	5	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi, ini, rekomendasi, banget, buat, topup, gam...	[aplikasi, ini, rekomendasi, banget, buat, top...	[aplikasi, rekomendasi, banget, topup, game, m...
6	2025-01-12 06:01:24	Pengguna Google	5	Transaksi disini lebih gampang flurnya juga...	Transaksi disini lebih gampang flurnya juga...	transaksi disini lebih gampang flurnya juga...	transaksi, disini, lebih, gampang, flurnya, juga...	[transaksi, disini, lebih, gampang, flurnya, ...	[transaksi, gampang, flurnya, lengkap, bel...
7	2025-01-12 05:56:56	Pengguna Google	5	Aplikasinya praktis dan mudah digunakan juga b...	Aplikasinya praktis dan mudah digunakan juga b...	aplikasinya praktis dan mudah digunakan juga b...	aplikasinya, praktis, dan, mudah, digunakan, juga, b...	[aplikasinya, praktis, dan, mudah, digunakan, ...	[aplikasinya, praktis, mudah, bayar, bayar, qis]
8	2025-01-12 05:23:16	Pengguna Google	5	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	aplikasi, yang, bagus, buat, transaksi, dan, gampang...	[aplikasi, yang, bagus, buat, transaksi, dan, ...	[aplikasi, bagus, transaksi, gampang, bikin, t...
9	2025-01-12 01:59:56	Pengguna Google	5	Aplikasi nya jelas dan sangat recommended buat...	Aplikasi nya jelas dan sangat recommended buat...	aplikasi nya jelas dan sangat recommended buat...	aplikasi, ya, jelas, dan, sangat, recommended...	[aplikasi, ya, jelas, dan, sangat, recommended...	[aplikasi, ya, recommended, topup, topup, flu...

Gambar 4. 18 Stopword Removal

Sumber: Data yang diolah (2025)

8. Stemming Data

Stemming data bertujuan mengganti kata dalam ulasan ke dalam kata dasar sesuai kaidah Bahasa Indonesia. Pada tahap ini semua kata dalam ulasan telah ternormalisasi dan data yang dibutuhkan sudah bersih, sehingga proses selanjutnya sudah bisa dilanjutkan.

```
[ ] !pip install Sastrawi

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer

Requirement already satisfied: Sastrawi in /usr/local/lib/python3.10/dist-packages (1.0.1)

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

def stem_text(text):
    return [stemmer.stem(word) for word in text]

df['stemming_data'] = df['stopword removal'].apply(lambda x: ' '.join(stem_text(x)))
df.head(10)
```

Gambar 4. 19 Stemming data

Sumber: Data yang diolah (2025)

	Date	Username	Rating	Review Text	Cleaning	case_folding	normalisasi	tokenize	stopword removal	stemming_data
0	2025-01-13 07:19:41	Pengguna Google	5	Aplikasi jakone mobile sangat keren bagus kare...	Aplikasi jakone mobile sangat keren bagus kare...	aplikasi jakone mobile sangat keren bagus kare...	aplikasi jakone mobile sangat keren bagus kare...	[aplikasi, jakone, mobile, sangat, keren, bagu...	[aplikasi, jakone, mobile, keren, bagus, flur...	aplikasi jakone mobile keren bagus flur flur...
1	2025-01-13 06:24:28	Pengguna Google	5	memudahkan transaksi kemanapun, tanpa ngelag l...	memudahkan transaksi kemanapun tanpa ngelag la...	memudahkan transaksi kemanapun tanpa ngelag la...	memudahkan transaksi kemanapun tanpa ngelag la...	(memudahkan, transaksi, kemanapun, tanpa, ngel...	(memudahkan, transaksi, kemanapun, ngelag, lag...	mudah transaksi mana ngelag lag an proses cepa...
2	2025-01-13 02:30:56	Pengguna Google	5	Aplikasi ini membantu banget dengan tampilan y...	Aplikasi ini membantu banget dengan tampilan y...	aplikasi ini membantu banget dengan tampilan y...	aplikasi ini membantu banget dengan tampilan y...	[aplikasi, ini, membantu, banget, dengan, tamp...	[aplikasi, membantu, banget, tampilan, sederhana...	aplikasi bantu banget tampil sederhana mudah p...
3	2025-01-13 02:27:36	Pengguna Google	5	Aplikasinya bagus banget fiturnya memudahkan u...	Aplikasinya bagus banget fiturnya memudahkan u...	aplikasinya bagus banget fiturnya memudahkan u...	aplikasinya bagus banget fiturnya memudahkan u...	[aplikasinya, bagus, banget, fiturnya, memudah...	[aplikasinya, bagus, banget, fiturnya, memudah...	aplikasi bagus banget fiturnya mudah transaksi...
4	2025-01-13 00:25:38	Pengguna Google	5	Aplikasinya bagus, fiturnya lengkap dan mudah ...	Aplikasinya bagus fiturnya lengkap dan mudah d...	aplikasinya bagus fiturnya lengkap dan mudah d...	aplikasinya bagus fiturnya lengkap dan mudah d...	[aplikasinya, bagus, fiturnya, lengkap, dan, m...	[aplikasinya, bagus, fiturnya, lengkap, mudah...	aplikasi bagus fiturnya lengkap mudah erti rib...
5	2025-01-13 00:23:31	Pengguna Google	5	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	aplikasi ini rekomendasi bgt buat topup game m...	[aplikasi, ini, rekomendasi, banget, buat, top...	[aplikasi, rekomendasi, banget, topup, game, m...	aplikasi rekomendasi banget topup game mi akumul...
6	2025-01-12 06:01:24	Pengguna Google	5	Transaksi disini lebih gampang, fiturnya juga ...	Transaksi disini lebih gampang fiturnya juga ...	transaksi disini lebih gampang fiturnya juga ...	transaksi disini lebih gampang fiturnya juga l...	[transaksi, disini, lebih, gampang, fiturnya, ...	[transaksi, gampang, fiturnya, lengkap, beli, ...	transaksi gampang fiturnya lengkap beli pulsa ...
7	2025-01-12 05:58:56	Pengguna Google	5	Aplikasinya praktis dan mudah digunakan juga b...	Aplikasinya praktis dan mudah digunakan juga b...	aplikasinya praktis dan mudah digunakan juga b...	aplikasinya praktis dan mudah digunakan juga b...	[aplikasinya, praktis, dan, mudah, digunakan, ...	[aplikasinya, praktis, mudah, bayar, bayar, qris]	aplikasi praktis mudah bayar bayar qris
8	2025-01-12 05:23:18	Pengguna Google	5	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	aplikasi yang bagus buat transaksi dan gampang...	[aplikasi, yang, bagus, buat, transaksi, dan, ...	[aplikasi, bagus, transaksi, gampang, bikin, r...	aplikasi bagus transaksi gampang bikin rekena...

9. Word Cloud Setelah Preprocessing

```
[ ] import pandas as pd
import numpy as np
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt

# Menggabungkan teks dari kolom 'stemming_data'
text = ' '.join(df['stemming_data'].astype(str).tolist())

stopwords = set(STOPWORDS)
stopwords.update(['https', 'co', 'RT', '...', 'amp', 'lu', 'deh', 'fyp', 'ya', 'gue', 'sih'])

wc = WordCloud(stopwords=stopwords, background_color="white", max_words=500, width=800, height=400)

wc.generate(text)

plt.figure(figsize=(10, 5))
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Gambar 4. 20 Cloud Setelah Preprocessing

Sumber: Data yang diolah (2025)



Gambar 4. 21 Wordcloud Setelah Preprocessing

Sumber: Data yang diolah (2025)

10. Diagram Kata Yang Sering Muncul

```
[ ] import matplotlib.pyplot as plt
from collections import Counter
from wordcloud import STOPWORDS

text = ".join(df["stemming_data"])

stopwords = set(STOPWORDS)
stopwords.update(['https', 'co', 'RT', '...', 'amp', 'lu', 'deh', 'fyp', 'ya', 'gue', 'sinh'])

tokens = [word for word in text.split() if word not in stopwords]
word_counts = Counter(tokens)

top_words = word_counts.most_common(10)
word, count = zip(*top_words)
# Menggunakan palet warna lebih soft
colors = plt.cm.Pastell(range(len(word)))

# Membuat plot
plt.figure(figsize=(12, 5))
bars = plt.bar(word, count, color=colors)
plt.xlabel("Kata-Kata Sering Muncul", fontsize=12, fontweight='bold')
plt.ylabel("Jumlah Kata", fontsize=12, fontweight='bold')
plt.title("Frekuensi Kata", fontsize=18, fontweight='bold')
plt.xticks(rotation=45)

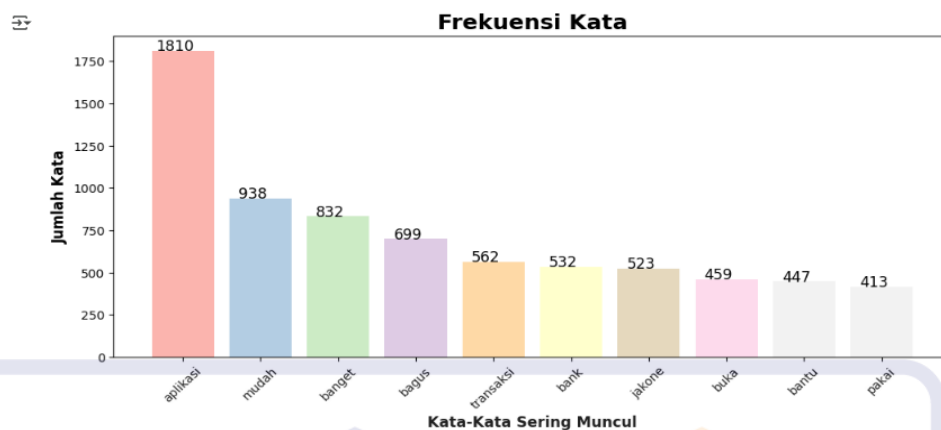
# Menambahkan angka rata tengah di atas setiap bar
for bar, num in zip(bars, count):
    plt.text(bar.get_x() + bar.get_width() / 2 - 0.1, num + 1, str(num), fontsize=12, color='black', ha='center')

# Menampilkan plot
plt.show()
```

Gambar 4. 22 Kata Yang Sering Muncul

Sumber: Data yang diolah (2025)

Menampilkan kata yang sering muncul dalam bentuk diagram batang.



Gambar 4. 23 Frekuensi Kata Setelah Preprocessing

Sumber: Data yang diolah (2025)

4.3 Labelling Data Metode Lexicon Based

Klasifikasi dilakukan dengan cara melabeli data dengan mengelompokkan ulasan berdasarkan kamus kata positif atau negatif, jika kata tersebut tidak termasuk dalam kata positif dan negatif, kata tersebut akan dianggap sebagai kata netral. Kemudian jumlah kata akan dihitung dan dapat dilihat bahwa ulasan tergolong menjadi sentimen positif atau negatif. Keuntungan dari metode ini adalah lebih akurat daripada metode rating, karena memeriksa ulasan terhadap kata-kata yang dikandungnya. Namun kelemahan dari metode ini adalah dalam sangat mengandalkan kebutuhan kamus *lexicon* Bahasa Indonesia yang mungkin terdapat kata-kata yang tidak sesuai dengan keadaan yang sebenarnya.

Mempertimbangkan penjabaran sebelumnya peneliti menggunakan Metode *lexicon based* karena klasifikasi ini dapat mengklasifikasikan data *reviews* apakah data tersebut merupakan sentimen positif, negatif dan netral dari sentimen per kata dari setiap ulasan dan cocok dengan JakOne Mobile yang memiliki ribuan ulasan dari pengguna.

Klasifikasi dengan *lexicon* sangat mengandalkan kebutuhan kamus *lexicon* bahasa Indonesia, karena kata pada data dapat diberikan *score* berdasarkan kamus apakah kata positif atau negatif. Untuk penilaian sentimen pada ulasan sendiri berbeda-beda, Sentimen dibagi menjadi tiga kelas sentimen yaitu kelas sentimen positif, kelas sentimen negatif dan kelas sentiment netral . Nilai yang dihasilkan dari penjumlahan total nilai kata positif pada keseluruhan kalimat ulasan yang dikurangi total nilai kata negatif pada kalimat ulasan disebut nilai sentimen. Nilai sentimen diatas 0 (sentimen ≥ 0) dinyatakan sebagai kelas positif, sedangkan nilai sentimen di bawah 0 (sentimen < 0) dinyatakan sebagai kelas negatif.

```
[ ] import pandas as pd

data = pd.read_csv("Hasil_Preprocessing_Data.csv")
data.info()
```

<class 'pandas.core.frame.DataFrame'>
 RangeIndex: 4611 entries, 0 to 4610
 Data columns (total 10 columns):

#	Column	Non-Null Count	Dtype
0	Date	4611 non-null	object
1	Username	4611 non-null	object
2	Rating	4611 non-null	int64
3	Review Text	4611 non-null	object
4	cleaning	4602 non-null	object
5	case_folding	4602 non-null	object
6	normalisasi	4600 non-null	object
7	tokenize	4611 non-null	object
8	stopword removal	4611 non-null	object
9	stemming_data	4580 non-null	object

dtypes: int64(1), object(9)
 memory usage: 360.4+ KB

```
[ ] import pandas as pd
import requests

# Unduh kamus leksikon positif dan negatif dari GitHub
positive_url = "https://raw.githubusercontent.com/fajri91/InSet/master/positive.tsv"
negative_url = "https://raw.githubusercontent.com/fajri91/InSet/master/negative.tsv"

positive_lexicon = set(pd.read_csv(positive_url, sep="\t", header=None)[0])
negative_lexicon = set(pd.read_csv(negative_url, sep="\t", header=None)[0])

# Fungsi untuk menentukan sentimen dan menghitung skornya
def determine_sentiment(text):
    if isinstance(text, str):
        positive_count = sum(1 for word in text.split() if word in positive_lexicon)
        negative_count = sum(1 for word in text.split() if word in negative_lexicon)
        sentiment_score = positive_count - negative_count
        if sentiment_score > 0:
            sentiment = "Positif"
        elif sentiment_score < 0:
            sentiment = "Negatif"
        else:
            sentiment = "Netral"
        return sentiment_score, sentiment
    return 0, "Netral"

# Tentukan sentimen dan skor untuk setiap ulasan
data[['Score', 'Sentiment']] = data['stemming_data'].apply(lambda x: pd.Series(determine_sentiment(x)))

# Tampilkan hasilnya
data.head(10)
```

	Date	Username	Rating	stemming_data	Score	Sentiment
0	2025-01-13 07:19:41	Pengguna Google	5	aplikasi jakone mobile keren bagus fitur fitur...	0	Netral
1	2025-01-13 06:24:28	Pengguna Google	5	mudah transaksi mana ngelag lag an proses cepa...	2	Positif
2	2025-01-13 02:30:56	Pengguna Google	5	aplikasi bantu banget tampil sederhana mudah p...	2	Positif
3	2025-01-13 02:27:36	Pengguna Google	5	aplikasi bagus banget fiturnya mudah transaksi...	0	Netral
4	2025-01-13 00:25:38	Pengguna Google	5	aplikasi bagus fiturnya lengkap mudah erti rib...	0	Netral
5	2025-01-13 00:23:31	Pengguna Google	5	aplikasi rekomendasi banget topup game ml akum...	1	Positif
6	2025-01-12 06:01:24	Pengguna Google	5	transaksi gampang fiturnya lengkap beli pulsa ...	-1	Negatif
7	2025-01-12 05:58:56	Pengguna Google	5	aplikasi praktis mudah bayar bayar qris	0	Netral
8	2025-01-12 05:23:18	Pengguna Google	5	aplikasi bagus transaksi gampang bikin rekenin...	2	Positif
9	2025-01-12 01:59:56	Pengguna Google	5	aplikasi ya recommended topup topup fitur fitu...	0	Netral

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

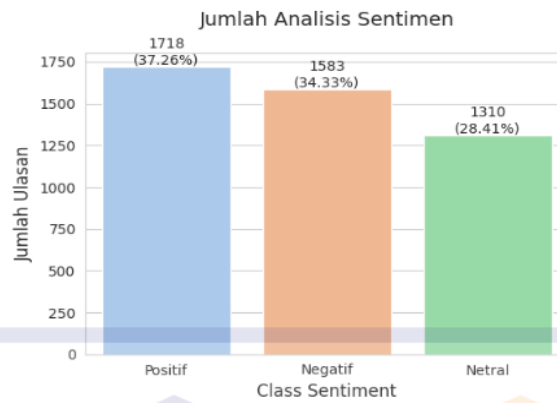
sentiment_count = data['Sentiment'].value_counts()
sns.set_style('whitegrid')

fig, ax = plt.subplots(figsize=(6, 4))
ax = sns.barplot(x=sentiment_count.index, y=sentiment_count.values, palette='pastel')
plt.title('Jumlah Analisis Sentimen', fontsize=14, pad=20)
plt.xlabel('Class Sentiment', fontsize=12)
plt.ylabel('Jumlah Ulasan', fontsize=12)

total = len(data['Sentiment'])

for i, count in enumerate(sentiment_count.values):
    percentage = f'{100 * count / total:.2f}%'
    ax.text(i, count + 0.10, f'{count}\n({percentage})', ha='center', va='bottom')

plt.show()
```



```
] data.to_csv('Hasil_Labelling_Data.csv', encoding='utf8', index=False)
```

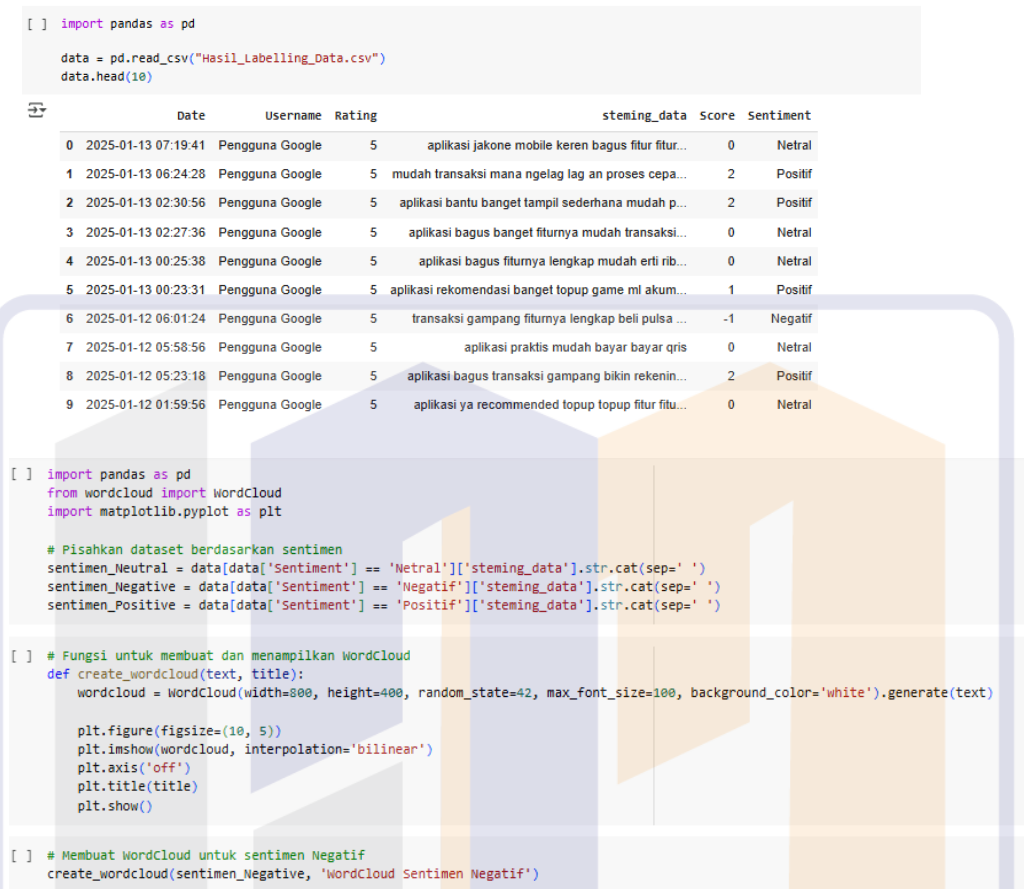
Gambar 4. 24 Jumlah Analisis Sentimen

Sumber: Data yang diolah (2025)

Dari hasil pelabelan data diatas didapatkan hasil 1.718 untuk hasil positif, 1.583 untuk hasil negatif dan 1.310 untuk hasil netral.

UNIVERSITAS
NUSA MANDIRI

Berikut ditampilkan *wordcloud* hasil labeling positif, negatif dan netral



Gambar 4. 25 Wordcloud hasil labeling positif, negatif dan netral

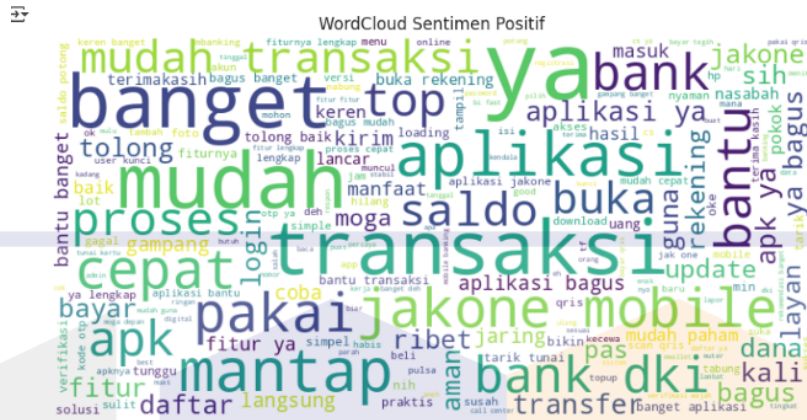
Wordcloud sentimen negatif



Gambar 4. 26 Wordcloud Sentimen Negatif

Wordcloud sentiment positif

```
[ ] # Membuat WordCloud untuk sentimen Positif
    create_wordcloud(sentimen_Positive, 'WordCloud Sentimen Positif')
```



Gambar 4. 27 Worcloud Sentimen Positif

Wordcloud setimen netral

```
[ ] # Membuat WordCloud untuk sentimen Netral
create_wordcloud(sentimen_Neutral, 'WordCloud Sentimen Netral')
```



Gambar 4. 28 Worcloud Sentimen Netral

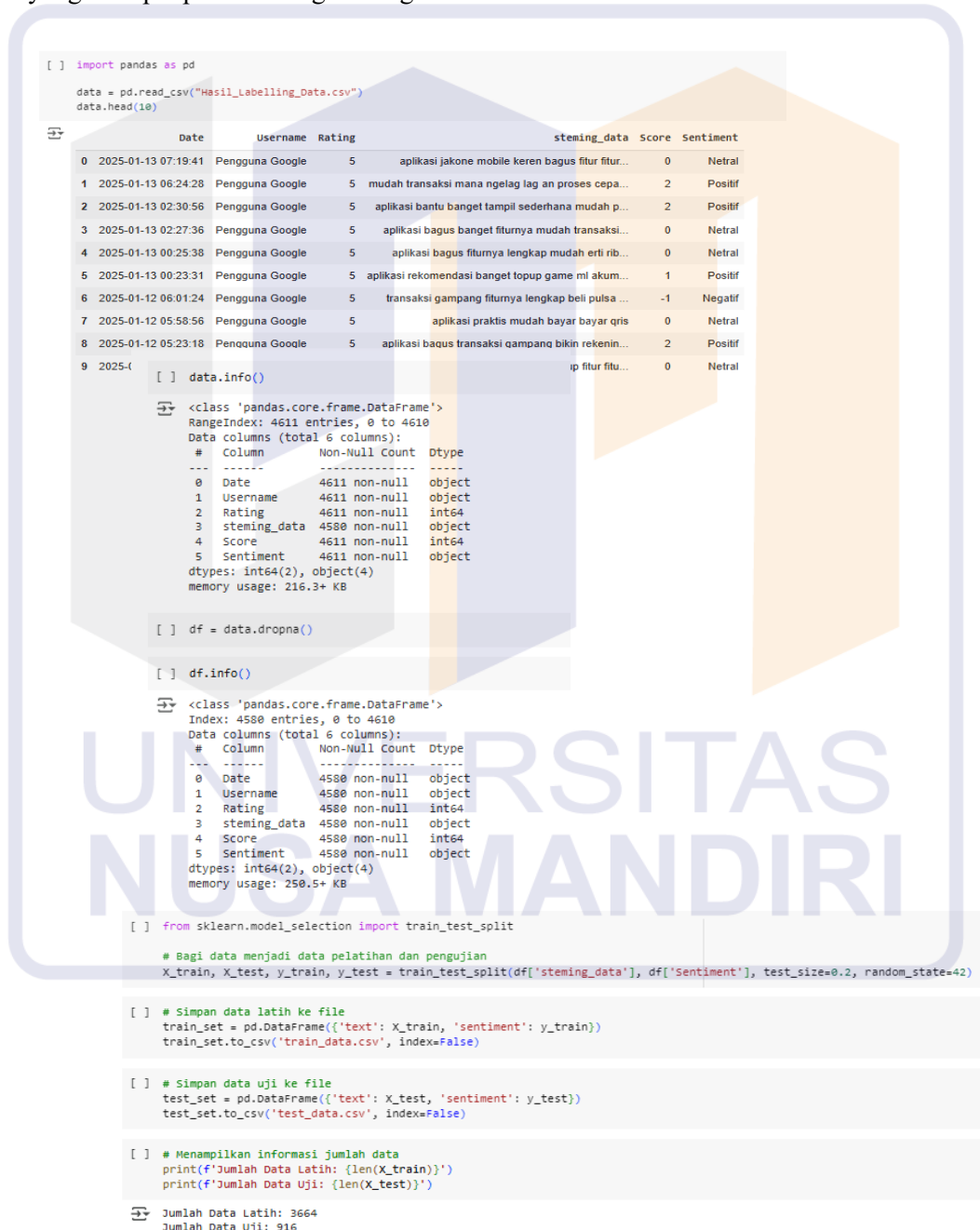
4.4 Pembagian Data Latih Dan Data Uji

Data review harus dipisahkan terlebih dahulu menjadi data pelatihan dan pengujian sebelum diklasifikasikan menggunakan *Support Vector Machine*. Untuk memperkirakan sentimen kelas baru, model klasifikasi yang terdiri dari pengetahuan yang dilatih menggunakan data pelatihan. Algoritma akan memahami data dengan lebih baik jika model semakin mendalaminya.

Persentase algoritma klasifikasi yang mengkategorikan dengan benar ditentukan dengan

menggunakan data uji. Data pelatihan dan pengujian dipisahkan dalam analisis ini berdasarkan perbandingan 80:20.

Support Vector Machine melakukan prediksi kelas positif atau negatif dengan menguji data serta mempelajari pengetahuan yang terkandung dalam data latih. Pada data latih terdapat kelas sentimen positif dan negatif, SVM akan mempelajari karakteristik kata-kata yang terdapat pada masing-masing kelas.



```
[ ] import pandas as pd

data = pd.read_csv("Hasil_Labelling_Data.csv")
data.head(10)
```

	Date	Username	Rating	steming_data	Score	Sentiment
0	2025-01-13 07:19:41	Pengguna Google	5	aplikasi jakone mobile keren bagus fitur futur...	0	Netral
1	2025-01-13 06:24:28	Pengguna Google	5	mudah transaksi mana ngelag lag an proses cepa...	2	Positif
2	2025-01-13 02:30:56	Pengguna Google	5	aplikasi bantu banget tampil sederhana mudah p...	2	Positif
3	2025-01-13 02:27:36	Pengguna Google	5	aplikasi bagus banget fiturnya mudah transaksi...	0	Netral
4	2025-01-13 00:25:38	Pengguna Google	5	aplikasi bagus fiturnya lengkap mudah erti rib...	0	Netral
5	2025-01-13 00:23:31	Pengguna Google	5	aplikasi rekomendasi banget topup game ml akum...	1	Positif
6	2025-01-12 06:01:24	Pengguna Google	5	transaksi gampang fiturnya lengkap beli pulsa ...	-1	Negatif
7	2025-01-12 05:58:56	Pengguna Google	5	aplikasi praktis mudah bayar bayar qris	0	Netral
8	2025-01-12 05:23:18	Pengguna Google	5	aplikasi bagus transaksi gampang bikin rekenin...	2	Positif
9	2025-01-12 05:23:18	Pengguna Google	5	aplikasi bagus transaksi gampang bikin rekenin...	0	Netral

```
[ ] data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4611 entries, 0 to 4610
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date         4611 non-null   object
1   Username     4611 non-null   object
2   Rating       4611 non-null   int64
3   steming_data 4580 non-null   object
4   Score        4611 non-null   int64
5   Sentiment    4611 non-null   object
dtypes: int64(2), object(4)
memory usage: 216.3+ KB

[ ] df = data.dropna()

[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 4580 entries, 0 to 4610
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date         4580 non-null   object
1   Username     4580 non-null   object
2   Rating       4580 non-null   int64
3   steming_data 4580 non-null   object
4   Score        4580 non-null   int64
5   Sentiment    4580 non-null   object
dtypes: int64(2), object(4)
memory usage: 250.5+ KB

[ ] from sklearn.model_selection import train_test_split

# Bagi data menjadi data pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(df['steming_data'], df['Sentiment'], test_size=0.2, random_state=42)

[ ] # Simpan data latih ke file
train_set = pd.DataFrame({'text': X_train, 'sentiment': y_train})
train_set.to_csv('train_data.csv', index=False)

[ ] # Simpan data uji ke file
test_set = pd.DataFrame({'text': X_test, 'sentiment': y_test})
test_set.to_csv('test_data.csv', index=False)

[ ] # Menampilkan informasi jumlah data
print(f'Jumlah Data Latih: {len(X_train)}')
print(f'Jumlah Data Uji: {len(X_test)}')
```

Jumlah Data Latih: 3664
Jumlah Data Uji: 916

Gambar 4. 29 Support Vector Machine

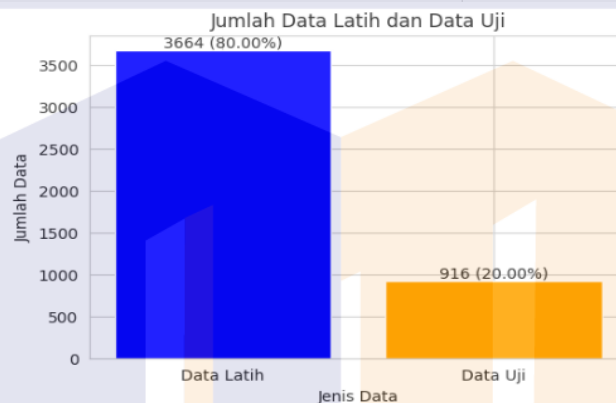
```
[ ] import matplotlib.pyplot as plt

# Jumlah data latih dan data uji
train_size = len(X_train)
test_size = len(X_test)

# Membuat plot
plt.figure(figsize=(6, 4))
bars = plt.bar(['Data Latih', 'Data Uji'], [train_size, test_size], color=['blue', 'orange'])

# Menambahkan label untuk setiap bar (dalam kurung persentase)
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 0.7, f'{height} ({height / (train_size + test_size) * 100:.2f}%)',
             ha='center', va='bottom')

plt.title('Jumlah Data Latih dan Data Uji')
plt.xlabel('Jenis Data')
plt.ylabel('Jumlah Data')
plt.show()
```



Gambar 4. 30 Hasil data Uji dan Data Latih

Sumber: Data yang diolah (2025)

Dari 4.580 data didapat 80% untuk data latih yaitu sebanyak 3.664 data dan 20% data sebanyak 916 data untuk data uji.

4.5 Pemodelan Support Vector Machine

Dalam konteks *machine learning*, khususnya *Support Vector Machine*, vektorisasi adalah proses mengubah data teks menjadi representasi numerik yang dapat dipahami oleh model. *Support Vector Machine* bekerja dengan data numerik, sehingga teks perlu diubah menjadi vektor fitur. Setiap fitur dalam vektor ini mewakili karakteristik tertentu dari teks.

Berikut adalah hasil *Support Vector Machine* dan *Confusion Matrix* tersebut.

```
[ ] from sklearn.svm import SVC
    from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
    import seaborn as sns
    import matplotlib.pyplot as plt
    from sklearn.feature_extraction.text import CountVectorizer
    from sklearn.feature_extraction.text import TfidfVectorizer

[ ] # Mendefinisikan vectorizer
    vectorizer = TfidfVectorizer()
    X_train_vectorized = vectorizer.fit_transform(X_train)
    X_test_vectorized = vectorizer.transform(X_test)

[ ] # Menampilkan hasil vektorisasi
    print("Matriks Vektorisasi untuk Data Latih:")
    print(X_train_vectorized.toarray())

    # menampilkan sebagian kecil matriks
    print("\nsebagian kecil Matriks Vektorisasi untuk Data Latih:")
    print(X_train_vectorized[5, :].toarray())

⇒ Matriks Vektorisasi untuk Data Latih:
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]

[ ] # Menampilkan sebagian kecil matriks vektorisasi untuk Data Latih:
⇒
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]

[ ] # Membuat dan melatih model SVM
    svm = SVC(kernel='linear')
    svm.fit(X_train_vectorized, y_train)

⇒ SVC
SVC(kernel='linear')

[ ] # Evaluasi model
    y_pred_svm = svm.predict(X_test_vectorized)

[ ] # Menampilkan confusion matrix dalam angka
    cm_svm = confusion_matrix(y_test, y_pred_svm)
    print("SVM Confusion Matrix:")
    print(cm_svm)

⇒ SVM Confusion Matrix:
[[288  29   3]
 [ 30 215  25]
 [   2  27 297]]

[ ] # Fungsi untuk plot confusion matrix
def plot_confusion_matrix(model_name, y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)

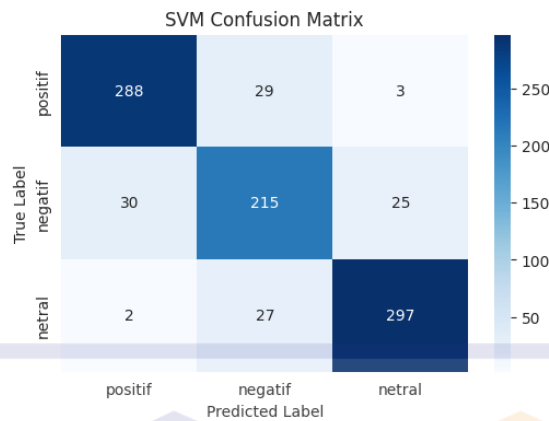
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
                xticklabels=['positif', 'negatif', 'netral'],
                yticklabels=['positif', 'negatif', 'netral'])

    plt.title(f"{model_name} Confusion Matrix")
    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.show()

plot_confusion_matrix("SVM", y_test, y_pred_svm)
```

Gambar 4. 31 Support Vector Machine dan Confusion Matrix

Sumber: Data yang diolah (2025)



Gambar 4. 32 Hasil Confusion Matrix

Sumber: Data yang diolah (2025)

Dari hasil *confusion matrix* dapat dijelaskan bahwa terdapat 288 data positif yang diprediksi termasuk ke dalam sentimen positif, 30 data positif yang diprediksi termasuk ke dalam sentimen negatif, dan 2 data positif yang termasuk ke dalam sentimen netral. Selain itu terdapat 215 data negatif yang diprediksi termasuk ke dalam sentimen negatif, 29 data negatif yang diprediksi termasuk kedalam sentiment positif, dan 27 data negatif yang termasuk kedalam sentimen netral. Kemudian terdapat 297 data netral yang termasuk ke dalam sentimen netral, 25 data netral yang termasuk ke dalam sentiment negatif, dan 3 data netral yang termasuk kedalam sentimen positif.

```
[ ] # Menampilkan accuracy untuk SVM
accuracy_svm = accuracy_score(y_test, y_pred_svm)
print("SVM Accuracy:", accuracy_svm)
print("")

accuracy_svm_percentage = accuracy_svm * 100
print("SVM Accuracy:", "{:.2f}%".format(accuracy_svm_percentage))
print("")

# Menampilkan classification report untuk SVM
print("SVM Classification Report:")
print(classification_report(y_test, y_pred_svm))
```

```
SVM Accuracy: 0.8733624454148472
SVM Accuracy: 87.34%

SVM Classification Report:
              precision    recall  f1-score   support

   Negatif         0.90      0.90      0.90       320
    Netral         0.79      0.80      0.79       270
    Positif         0.91      0.91      0.91       326

 accuracy         0.87
 macro avg        0.87
 weighted avg     0.87
```

Gambar 4. 33 Nilai akurasi Support Vector Machine

Sumber: Data yang diolah (2025)

Nilai akurasi *Support Vector Machine* menunjukkan angka 0.8733624454148472, angka ini berarti model berhasil memprediksi sentiment dengan benar sebanyak 87,34% dari total data.

Laporan klasifikasi ini memberikan informasi lebih detail tentang kinerja model untuk setiap kategori sentimen. Berikut penjelasannya:

1. *Precision* (Presisi): Presisi mengukur seberapa sering model benar dalam memprediksi suatu kategori sentimen dari semua data yang diprediksi memiliki sentimen tersebut.
 - a. Negatif: 0.90 (90% dari data yang diprediksi negatif benar-benar negatif)
 - b. Netral: 0.79 (79% dari data yang diprediksi netral benar-benar netral)
 - c. Positif: 0.91 (91% dari data yang diprediksi positif benar-benar positif)
2. *Recall*: *Recall* mengukur seberapa sering model benar dalam memprediksi suatu kategori sentimen dari semua data yang sebenarnya memiliki sentimen tersebut.
 - a. Negatif: 0.90 (90% dari data negatif yang sebenarnya berhasil diprediksi sebagai negatif)
 - b. Netral: 0.80 (80% dari data netral yang sebenarnya berhasil diprediksi sebagai netral)
 - c. Positif: 0.91 (91% dari data positif yang sebenarnya berhasil diprediksi sebagai positif)

3. *F1-Score*: *F1-score* adalah rata-rata harmonik antara presisi dan *recall*. Ini memberikan ukuran keseimbangan antara presisi dan *recall*.

- a. Negatif: 0.90
- b. Netral: 0.79
- c. Positif: 0.91

4. *Support* (Jumlah Data): Jumlah data aktual untuk setiap kategori sentimen.

- a. Negatif: 320
- b. Netral: 270
- c. Positif: 326

4.6 Rating

```
[ ] import pandas as pd

data = pd.read_csv("Hasil_Labeling_Data.csv")
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4611 entries, 0 to 4610
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        4611 non-null    object
1   Username    4611 non-null    object
2   Rating      4611 non-null    int64
3   stemming_data 4580 non-null    object
4   Score       4611 non-null    int64
5   Sentiment   4611 non-null    object
dtypes: int64(2), object(4)
memory usage: 216.3+ KB
```

Gambar 4. 34 Hasil labeling data

Sumber: Data yang diolah (2025)

```
import matplotlib.pyplot as plt

rating_counts = data['Rating'].value_counts()
rating_counts = rating_counts.sort_index()

colors = ['skyblue', 'lightcoral', 'lightgreen', 'lightsalmon', 'lightblue']

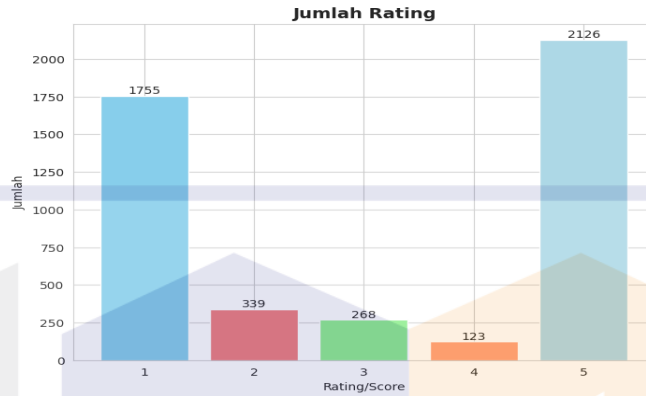
plt.figure(figsize=(8, 6))
bars = plt.bar(rating_counts.index, rating_counts.values, color=colors)
plt.title('Jumlah Rating', fontsize=14, fontweight='bold')
plt.xlabel('Rating/Score')
plt.ylabel('Jumlah')
plt.xticks(rating_counts.index)

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, height, str(int(height)), ha='center', va='bottom')

plt.show()
```

Gambar 4. 35 Import matplotlib.pyplot

Sumber: Data yang diolah (2025)



Gambar 4. 36 Hasil Rating Pengguna

Sumber: Data yang diolah (2025)

Berdasarkan gambar diatas menunjukan rating 5 yang memiliki data terbanyak yaitu 2.126 data, dan *rating* 1 menunjukan angka 1.755 data yang artinya berdasarkan *rating* dari ulasan pengguna aplikasi ini juga masih banyak belum memuaskan pengguna.

UNIVERSITAS
NUSA MANDIRI