FINAL PROJECT

INFRASTRUKTUR DAN PLATFORM SAINS DATA

ANALISIS SENTIMEN ULASAN GOOGLE PLAY STORE PADA APLIKASI DIGITAL KORLANTAS POLRI

Disusun oleh:

Dian Sasya Fitriani	**(L200220021)**
Nasyawa Adesty Riyanni	**(L200220023)**
Adinda Aulia Hapsari	**(L200220037)**

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

TAHUN 2024

A. PENDAHULUAN

Di era digital saat ini, ulasan pengguna aplikasi di platform Google Play Store memiliki peran penting dalam mengevaluasi kualitas dan kinerja suatu aplikasi. Ulasan tersebut mencerminkan pengalaman, kepuasan, dan keluhan pengguna yang dapat digunakan untuk analisis lebih lanjut seperti pengukuran kepuasan pelanggan, analisis sentimen, dan pengambilan keputusan bisnis. Proses pengumpulan data ulasan secara manual dapat memakan waktu dan tidak efektif. Oleh karena itu, teknik web scraping digunakan untuk mengotomatisasi pengambilan data dari platform tersebut. Dalam proyek ini, kami melakukan scraping data ulasan aplikasi Digital Korlantas POLRI dari Google Play Store menggunakan pustaka google-play-scraper. Data yang diperoleh meliputi rating bintang, isi ulasan, nama pengguna, dan tanggal posting ulasan. Informasi ini diolah menjadi dataset terstruktur yang dapat digunakan untuk berbagai analisis seperti pemantauan performa aplikasi, identifikasi masalah yang sering dikeluhkan, dan pengambilan keputusan untuk pengembangan aplikasi yang lebih baik.

Analisis Sentimen adalah proses untuk mengidentifikasi dan mengkategorikan opini atau perasaan yang terkandung dalam suatu teks, seperti ulasan atau komentar. Dalam konteks ini, analisis sentimen digunakan untuk menilai apakah ulasan yang diberikan oleh pengguna terhadap aplikasi Digital Korlantas POLRI bersifat positif, negatif, atau netral. Dalam pendekatan leksikon (*Lexicon-based Approach*), analisis dilakukan dengan membandingkan kata-kata dalam ulasan dengan kamus kata-kata yang telah diberi nilai sentimen tertentu (misalnya, kata-kata positif seperti "baik" atau negatif seperti "buruk"). Setiap kata yang ditemukan dalam teks akan memberikan kontribusi skor yang dihitung untuk menghasilkan nilai keseluruhan. Berdasarkan skor ini, ulasan dapat dikategorikan sebagai positif, negatif, atau netral.

Melalui proyek ini, kami berharap dapat memanfaatkan data ulasan pengguna secara maksimal untuk mendukung pengembangan aplikasi yang lebih responsif terhadap kebutuhan pengguna.

B. PROSES SCRAPING DATA

Scraping data adalah proses mengumpulkan informasi dari suatu situs web secara otomatis menggunakan program komputer. Dalam konteks ini, scraping data digunakan untuk mengumpulkan ulasan pengguna aplikasi dari Google Play Store dengan cepat dan efisien. Berikut langkah-langkah scraping data ulasan aplikasi Digital Korlantas POLRI.

1. Install library google-play-scraper untuk mengolah data. Tunggu proses import library selesai.

```
[ ] !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 2.0 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
```

2. Import library google-play-scraper. Selanjutnya panggil fungsi reviews_all untuk mengambil seluruh review/ulasan dari aplikasi korlantas. Proses ini sudah masuk kedalam tahap ekstrasi data karena data ulasan diekstrasi langsung dari Google Play Store menggunakan pustaka google-play-scraper.
   - 'id.qoin.korlantas.user' merupakan ID aplikasi korlantas.
   - sleep_milliseconds merupakan jeda waktu antar-pengambilan data, 0 berarti tidak ada jeda (default).
   - lang='id' merupakan bahasa ulasan yang ingin diambil.
   - country='id' merupakan negara tempat ulasan diambil.
   - sort=Sort.NEWEST merupakan kode untuk menentukan agar ulasan diambil berdasarkan yang paling baru.

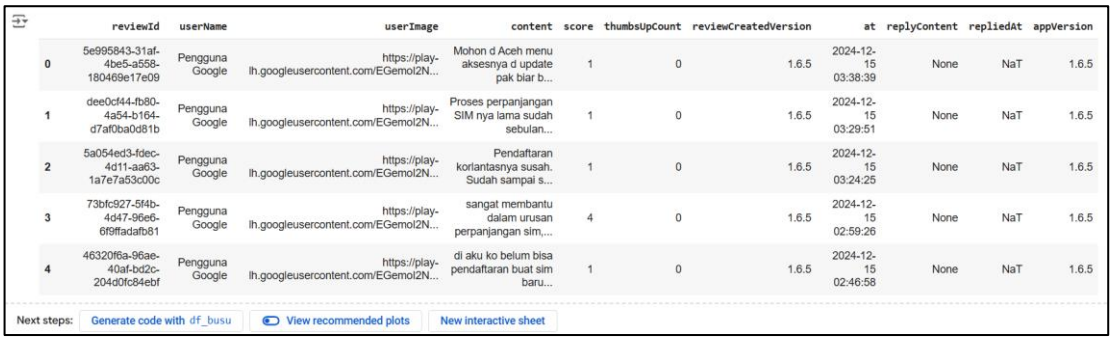
```
[ ] from google_play_scraper import Sort, reviews_all

    result = reviews_all(
        'id.qoin.korlantas.user',
        sleep_milliseconds=0, # defaults to 0
        lang='id', # defaults to 'en'
        country='id', # defaults to 'us'
        sort=Sort.NEWEST, # defaults to Sort.MOST_RELEVANT ,
    )
```

3. Membuat dataframe awal dari hasil ulasan yang telah diambil sebelumnya, dan mengekstrak informasi menjadi kolom terpisah. Proses ini sudah masuk kedalam tahap transformasi karena mencakup pengolahan data agar lebih terstruktur.

```
[ ] df_busu = pd.DataFrame(np.array(result),columns=['review'])

    df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))

    df_busu.head()
```

- Baris pertama digunakan untuk mengonversi daftar ulasan result menjadi array NumPy, menetapkan nama kolom sebagai ‘review’.
- Baris kedua digunakan untuk memisahkan kolom ulasan. Kolom review akan dihapus dalam DataFrame dan akan dikembalikan lagi.
- Baris ketiga digunakan untuk menampilkan DataFrame.

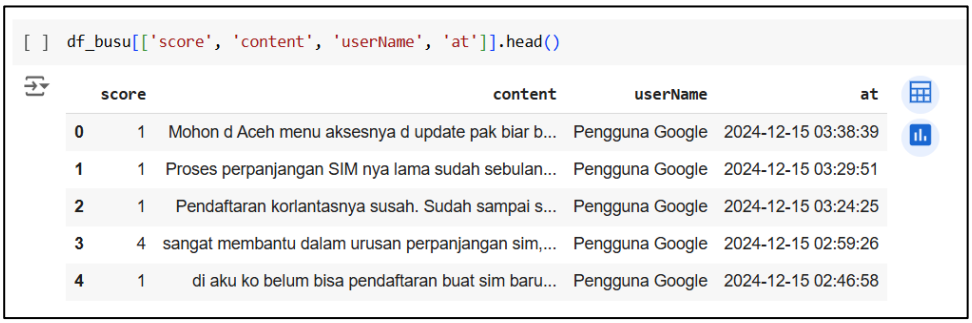
Hasil data:



	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	appVersion
0	5e995843-31af-4be5-a558-180469e17e09	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Mohon d Aceh menu aksesnya d update pak biar b...	1	0	1.6.5	2024-12-15 03:38:39	None	NaT	1.6.5
1	dee0cf44-fb80-4a54-b164-d7af0ba0d81b	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Proses perpanjangan SIM nya lama sudah sebulan...	1	0	1.6.5	2024-12-15 03:29:51	None	NaT	1.6.5
2	5a054ed3-fdec-4d11-aa63-1a7e7a53c00c	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	Pendaftaran korlantasnya susah. Sudah sampai s...	1	0	1.6.5	2024-12-15 03:24:25	None	NaT	1.6.5
3	73bfc927-5f4b-4d47-96e6-6f9ffadafb81	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	sangat membantu dalam urusan perpanjangan sim,...	4	0	1.6.5	2024-12-15 02:59:26	None	NaT	1.6.5
4	46320f6a-96ae-40af-bd2c-204d0fc84ebf	Pengguna Google	https://play-lh.googleusercontent.com/EGemol2N...	di aku ko belum bisa pendaftaran buat sim baru...	1	0	1.6.5	2024-12-15 02:46:58	None	NaT	1.6.5

Next steps: Generate code with df_busu | View recommended plots | New interactive sheet

4. Menampilkan 5 baris pertama kolom score, content, username, dan at. Proses ini sudah masuk kedalam tahap transformasi karena mencakup pengolahan data agar lebih terstruktur.



```
[ ] df_busu[['score', 'content', 'userName', 'at']].head()
```

	score	content	userName	at
0	1	Mohon d Aceh menu aksesnya d update pak biar b...	Pengguna Google	2024-12-15 03:38:39
1	1	Proses perpanjangan SIM nya lama sudah sebulan...	Pengguna Google	2024-12-15 03:29:51
2	1	Pendaftaran korlantasnya susah. Sudah sampai s...	Pengguna Google	2024-12-15 03:24:25
3	4	sangat membantu dalam urusan perpanjangan sim,...	Pengguna Google	2024-12-15 02:59:26
4	1	di aku ko belum bisa pendaftaran buat sim baru...	Pengguna Google	2024-12-15 02:46:58

5. Membuat DaraFrame baru yang hanya berisi kolom-kolom penting yang dibutuhkan, lalu menyimpannya kedalam file CSV Bernama ‘korlantas.csv’. index=False ini berfungsi untuk menghilangkan kolom indeks agar tidak ikut disimpan.

   Yang terakhir adalah mengunduh file CSV yang telah disimpan ke perangkat local pengguna. Proses ini sudah termasuk kedalam proses Load/penyimpanan.

```
[ ] new_df = df_busu[['score', 'content','userName', 'at']]

[ ] new_df.to_csv("korlantas.csv", index=False) #Save the file as CSV , to download:

    #download
    from google.colab import files
    files.download('korlantas.csv')
```

6. Menghubungkan Google Colab dengan akun Google Drive agar file dapat disimpan langsung ke Google Drive.

```
[ ]  from google.colab import drive
     drive.mount('/content/drive')
```

C. PROSES PENGOLAHAN DATA MENGGUNAKAN JUPYTER NOTEBOOK

1. Mengimport library umum yang digunakan untuk memproses data.
   - Pandas dan Numpy digunakan untuk memproses dan memanipulasi data.
   - matplotlib.pyplot dan seaborn digunakan untuk visualisasi data. Seperti membuat grafik histogram, diagram batang, dan heatmap untuk analisis data.
   - Nltk digunakan untuk membersihkan dan memproses data teks untuk analisis lebih lanjut.
   - Sastrawi digunakan untuk Mendukung analisis teks dalam bahasa Indonesia seperti stemming dan menghapus kata-kata tidak penting.
   - Tensorflow digunakan untuk membuat model pembelajaran mesin berbasis teks seperti analisis sentimen, klasifikasi teks, dan pemrosesan bahasa alami.

```
[1]: import pandas as pd  # Pandas untuk manipulasi dan analisis data
     pd.options.mode.chained_assignment = None  # Menonaktifkan peringatan chaining
     import numpy as np  # NumPy untuk komputasi numerik
     seed = 0
     np.random.seed(seed)  # Mengatur seed untuk reproduktibilitas
     import matplotlib.pyplot as plt  # Matplotlib untuk visualisasi data
     import seaborn as sns  # Seaborn untuk visualisasi data statistik, mengatur gaya visualisasi

     !pip install nltk
     import datetime as dt  # Manipulasi data waktu dan tanggal
     import re  # Modul untuk bekerja dengan ekspresi reguler
     import string  # Berisi konstanta string, seperti tanda baca
     from nltk.tokenize import word_tokenize  # Tokenisasi teks
     from nltk.corpus import stopwords  # Daftar kata-kata berhenti dalam teks

     !pip install sastrawi
     from Sastrawi.Stemmer.StemmerFactory import StemmerFactory  # Stemming (penghilangan imbuhan kata) dalam
     from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory  # Menghapus kata-kata

     !pip install tensorflow
     import tensorflow as tf
     from tensorflow.keras.preprocessing.text import Tokenizer
     from tensorflow.keras.preprocessing.sequence import pad_sequences
     from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Embedding, LSTM, Dense, SpatialDropout1D
```

2. Mengunduh dataset yang diperlukan oleh NLTK (Natural Language Toolkit). Dataset punkt digunakan untuk tokenisasi, yaitu proses memecah teks menjadi unit-unit yang lebih kecil seperti kata, kalimat, atau frasa. Dataset stopwords berisi daftar kata-kata

berhenti dalam berbagai Bahasa, sehingga memudahkan untuk menghapus kata-kata yang kurang berguna.

```
[2]: import nltk  # Import pustaka NLTK (Natural Language Toolkit).
     nltk.download('punkt')  # Mengunduh dataset yang diperlukan untuk tokenisasi teks.
     nltk.download('stopwords')  # Mengunduh dataset yang berisi daftar kata-kata berhenti

     [nltk_data] Downloading package punkt to
     [nltk_data]     C:\Users\Acer\AppData\Roaming\nltk_data...
     [nltk_data]   Package punkt is already up-to-date!
     [nltk_data] Downloading package stopwords to
     [nltk_data]     C:\Users\Acer\AppData\Roaming\nltk_data...
     [nltk_data]   Package stopwords is already up-to-date!
[2]: True
```

3. Membaca file korlantas.csv dan menampilkan 5 baris pertama.

```
[3]: df = pd.read_csv('korlantas.csv')
     df.head()
```

[3]:

	score	content	userName	at
0	1	Mohon d Aceh menu aksesnya d update pak biar b...	Pengguna Google	2024-12-15 03:38:39
1	1	Proses perpanjangan SIM nya lama sudah sebulan...	Pengguna Google	2024-12-15 03:29:51
2	1	Pendaftaran korlantasnya susah. Sudah sampai s...	Pengguna Google	2024-12-15 03:24:25
3	4	sangat membantu dalam urusan perpanjangan sim,...	Pengguna Google	2024-12-15 02:59:26
4	1	di aku ko belum bisa pendaftaran buat sim baru...	Pengguna Google	2024-12-15 02:46:58

4. Menampilkan struktur tabel korlantas.csv

```
[4]: df.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 68582 entries, 0 to 68581
     Data columns (total 4 columns):
      #   Column    Non-Null Count  Dtype
     ---  ------    --------------  -----
      0   score     68582 non-null  int64
      1   content   68582 non-null  object
      2   userName  68582 non-null  object
      3   at        68582 non-null  object
     dtypes: int64(1), object(3)
     memory usage: 2.1+ MB
```

5. Menganalisis dan memvisualisasikan distribusi data dalam kolom score.
   - `print(df['score'].value_counts())` digunakan untuk menampilkan jumlah frekuensi kemunculan rating.
   - `sns.countplot(x='score', data=df)` digunakan untuk membuat plot batang (bar plot) yang menunjukkan distribusi jumlah kemunculan setiap nilai unik pada kolom score.

- plt.title('Distribusi Skor') digunakan untuk memberi judul pada plot.

```
[5]: import matplotlib.pyplot as plt
     # Menampilkan jumlah nilai unik pada kolom 'score'
     print(df['score'].value_counts())

     # Membuat visualisasi jumlah nilai unik pada kolom 'score'
     sns.countplot(x='score', data=df)
     plt.title('Distribusi Skor')
     plt.xlabel('Skor')
     plt.ylabel('Jumlah')
     plt.show()
```

```
score
5    31382
1    21042
3     5890
2     5316
4     4952
Name: count, dtype: int64
```

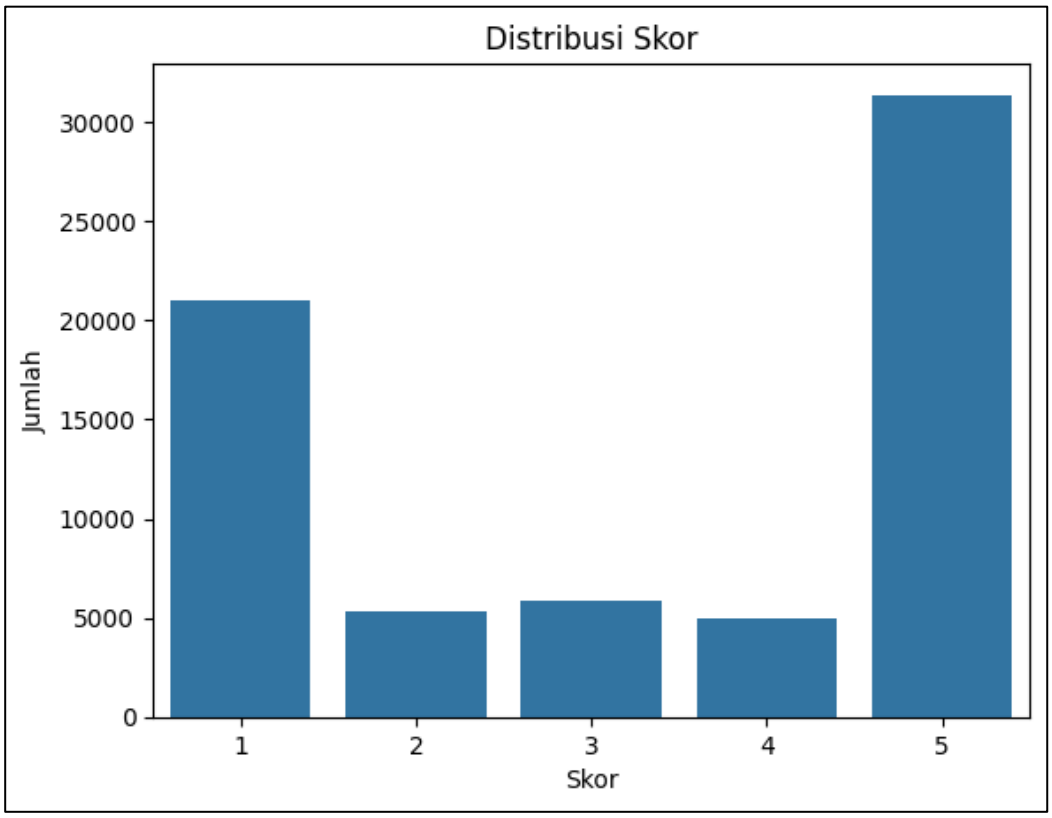


Diagram diatas menunjukkan bahwa Sebagian besar pengguna memberikan skor 5, namun tidak sedikit pula yang memberikan skor 1.

6. Membersihkan data dengan menghapus nilai yang hilang dan memeriksa apakah ada nilai yang hilang setelah pembersihan.

- clean_df = df.dropna() digunakan untuk menghapus baris kosong dan menyimpannya dalam DataFrame yang baru Bernama ‘clean_df’.
- clean_df.isnull().sum() digunakan untuk memeriksa apakah setelah penghapusan nilai yang hilang, masih ada kolom yang memiliki nilai NaN. Jika hasilnya semua nol (0), berarti tidak ada lagi nilai hilang pada DataFrame clean_df.

```
[6]: # Membuat DataFrame baru (clean_df) dengan
     # menghapus baris yang memiliki nilai yang hilang (NaN) dari app_reviews_df
     clean_df = df.dropna()

[7]: clean_df.isnull().sum()

[7]: score       0
     content     0
     userName    0
     at          0
     dtype: int64
```

7. Menghapus duplikat dalam DataFrame dan menghitung dimensi (jumlah baris dan kolom) setelah penghapusan duplikat.
   - clean_df = clean_df.drop_duplicates() digunakan untuk menghapus baris-baris yang berulang.
   - jumlah_ulasan_setelah_hapus_duplikat, jumlah_kolom_setelah_hapus_duplikat = clean_df.shape digunakan untuk menghitung jumlah baris dan kolom setelah penghapusan duplikat.
   - print(jumlah_ulasan_setelah_hapus_duplikat,jumlah_kolom_setelah_hapus_duplikat) digunakan untuk menampilkan hasil dari variabel yang menyimpan jumlah baris (ulasan) dan jumlah kolom pada DataFrame clean_df setelah proses penghapusan duplikat

```
[9]: # Menghapus baris duplikat dari DataFrame clean_df
     clean_df = clean_df.drop_duplicates()

     # Menghitung jumlah baris dan kolom dalam DataFrame clean_df setelah menghapus duplikat
     jumlah_ulasan_setelah_hapus_duplikat, jumlah_kolom_setelah_hapus_duplikat = clean_df.shape

[10]: print(jumlah_ulasan_setelah_hapus_duplikat, jumlah_kolom_setelah_hapus_duplikat)

      68582 4
```

8. Mengambil sampel acak sebanyak 15.000 baris dari DataFrame clean_df dan menyimpannya dalam DataFrame baru new_df.

```
[11]: # Mengambil 15 ribu baris data secara acak dan menyimpannya ke dalam
new_df = clean_df.sample(n=15000, random_state=seed)

# Menampilkan jumlah baris dan kolom dalam DataFrame new_df
jumlah_baris_new_df, jumlah_kolom_new_df = new_df.shape
print(jumlah_baris_new_df, jumlah_kolom_new_df)

15000 4
```

9. Menampilkan 5 baris pertama data dew_df

```
[12]: new_df.head()
```

[12]:

	score	content	userName	at
68031	1	Abis download buka aplikasi, masukkan nomer HP...	dwi merak	2021-04-14 02:58:55
45149	5	Terimakasih...SIM nya sudah tadang... Mudah da...	Pengguna Google	2022-03-10 10:53:35
53324	5	Alhamdulillah sangat terbantu	Pengguna Google	2021-10-12 09:06:51
44641	5	Adakan pembuatan sim secara online jg min	Pengguna Google	2022-03-16 18:10:54
43733	3	sayang nya untuk pembuatan SIM online di daera...	Pengguna Google	2022-03-29 12:11:26

10. Membersihkan data.
    - `cleaningText(text)` digunakan untuk membersihkan data dari berbagai elemen yang tidak diinginkan, seperti mention : `re.sub(r'@[A-Za-z0-9]+', '', text)`, hastag : `re.sub(r'#[A-Za-z0-9]+', '', text)`, retweet : `re.sub(r'RT[\s]', '', text)`, link atau URL `re.sub(r"http\S+", '', text)`, tanda baca dan symbol : `re.sub(r'[^\w\s]', '', text)`, format tanggal dan waktu : `re.sub(r'\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}', '', text)`, emoji : `re.sub(r'[^\x00-\x7F]+', '', text)`, spasi di awal dan akhir teks : `text.strip(' ')`, dan mengganti newline dengan spasi : `text.replace('\n', ' ')`.
    - `casefoldingText(text)` digunakan untuk mengubah semua karakter dalam teks menjadi huruf kecil.
    - `tokenizingText(text)` digunakan memecah teks menjadi unit yang lebih kecil, seperti kata atau frasa, agar dapat dianalisis lebih lanjut.
    - `filteringText(text)` digunakan untuk menghapus stopwords (kata-kata umum yang tidak memberikan banyak informasi, seperti "dan", "atau", "the") dari teks, sehingga fokus pada kata-kata yang lebih bermakna.
    - `stemmingText(text)` digunakan untuk menyatukan berbagai bentuk kata ke dalam satu bentuk dasar, yang memudahkan analisis. Misalnya, "berlari" dan "berlarian" dapat dianggap sebagai kata yang sama.

- toSentence(list_words) figunakan untuk mengubah daftar kata-kata (list of words) menjadi kalimat yang utuh.

```
[13]: #Preprocessing - Transform
def cleaningText(text):
    text = re.sub(r'@[A-Za-z0-9]+', '', text) # remove mentions
    text = re.sub(r'#[A-Za-z0-9]+', '', text) # remove hashtag
    text = re.sub(r'RT[\s]', '', text) # remove RT
    text = re.sub(r"http\S+", '', text) # remove link
    text = re.sub(r'[^\w\s]', '', text) # remove punctuation
    text = re.sub(r'\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}', '', text) # remove date-time in format 2021-04-14 02:58:55
    text = re.sub(r'[^\x00-\x7F]+', '', text)  # remove non-ASCII characters (including emojis)

    text = text.replace('\n', ' ') # replace new line into space
    text = text.translate(str.maketrans('', '', string.punctuation)) # remove all punctuations
    text = text.strip(' ') # remove characters space from both left and right text
    return text

def casefoldingText(text): # Converting all the characters in a text into lower case
    text = text.lower()
    return text

def tokenizingText(text): # Tokenizing or splitting a string, text into a list of tokens
    text = word_tokenize(text)
    return text
```

```
def filteringText(text): # Remove stopwors in a text
    listStopwords = set(stopwords.words('indonesian'))
    listStopwords1 = set(stopwords.words('english'))
    listStopwords.update(listStopwords1)
    listStopwords.update(['iya','yaa','gak','nya','na','sih','ku',"di","ga","ya","gaa","loh","kah","woi","woii","woy"])
    filtered = []
    for txt in text:
        if txt not in listStopwords:
            filtered.append(txt)
    text = filtered
    return text

def stemmingText(text): # Reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words
    # Membuat objek stemmer
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()

    # Memecah teks menjadi daftar kata
    words = text.split()

    # Menerapkan stemming pada setiap kata dalam daftar
    stemmed_words = [stemmer.stem(word) for word in words]

    # Menggabungkan kata-kata yang telah distem
    stemmed_text = ' '.join(stemmed_words)

    return stemmed_text

    return stemmed_text
def toSentence(list_words): # Convert list of words into sentence
    sentence = ' '.join(word for word in list_words)
    return sentence
```

Fungsi-fungsi ini adalah bagian dari *pipeline preprocessing* untuk mempersiapkan data teks dalam analisis atau pemodelan bahasa alami. Dengan membersihkan, menormalisasi, dan mengurangi teks, data akan lebih siap untuk langkah-langkah analisis selanjutnya, yaitu analisis sentimen.

11. Mengganti kata slang (Bahasa gaul/singkatan) menjadi teks yang lebih formal dan sesuai konteks. Teks akan dipecah menjadi daftar kata-kata dengan memisahkannya

berdasarkan spasi, kemudian kita akan membuat daftar kosong untuk menyimpan kata-kata yang telah diperbaiki. Setelah itu, teks akan kembali digabungkan.

```
[14]: slangwords = {"@": "di", "abis": "habis", "wtb": "beli", "masi": "masih", "wts": "jual",
def fix_slangwords(text):
    words = text.split()
    fixed_words = []

    for word in words:
        if word.lower() in slangwords:
            fixed_words.append(slangwords[word.lower()])
        else:
            fixed_words.append(word)

    fixed_text = ' '.join(fixed_words)
    return fixed_text
```

12. preprocessing teks secara bertahap dan menyimpan hasilnya dalam kolom baru di DataFrame new_df

```
[17]: # Membersihkan teks dan menyimpannya di kolom 'text_clean'
new_df['text_clean'] = new_df['content'].apply(cleaningText)

# Mengubah huruf dalam teks menjadi huruf kecil dan menyimpannya di 'text_casefoldingText'
new_df['text_casefoldingText'] = new_df['text_clean'].apply(casefoldingText)

# Mengganti kata-kata slang dengan kata-kata standar dan menyimpannya di 'text_slangwords'
new_df['text_slangwords'] = new_df['text_casefoldingText'].apply(fix_slangwords)

# Memecah teks menjadi token (kata-kata) dan menyimpannya di 'text_tokenizingText'
new_df['text_tokenizingText'] = new_df['text_slangwords'].apply(tokenizingText)

# Menghapus kata-kata stop (kata-kata umum) dan menyimpannya di 'text_stopword'
new_df['text_stopword'] = new_df['text_tokenizingText'].apply(filteringText)

# Menggabungkan token-token menjadi kalimat dan menyimpannya di 'text_akhir'
new_df['text_akhir'] = new_df['text_stopword'].apply(toSentence)
```

13. Menampilkan beberapa baris data new_df

```
[18]: new_df
```

[18]:

	score	content	userName	at	text_clean	text_casefoldingText	text_slangwords	text_tokenizingText	text_stopword	text_akhir
68031	1	Abis download buka aplikasi, masukkan nomer HP...	dwi merak	2021-04-14 02:58:55	Abis download buka aplikasi masukkan nomer HP ...	abis download buka aplikasi masukkan nomer hp ...	habis download buka aplikasi masukkan nomer hp...	[habis, download, buka, aplikasi, masukkan, no...	[habis, download, buka, aplikasi, masukkan, no...	habis download buka aplikasi masukkan nomer hp...
45149	5	Terimakasih...SIM nya sudah tadang... Mudah da...	Pengguna Google	2022-03-10 10:53:35	TerimakasihSIM nya sudah tadang Mudah dan aman	terimakasihsim nya sudah tadang mudah dan aman	terimakasihsim nya sudah datang mudah dan aman	[terimakasihsim, nya, sudah, datang, mudah, da...	[terimakasihsim, mudah, aman]	terimakasihsim mudah aman
53324	5	Alhamdulillah sangat terbantu	Pengguna Google	2021-10-12 09:06:51	Alhamdulillah sangat terbantu	alhamdulillah sangat terbantu	alhamdulillah sangat terbantu	[alhamdulillah, sangat, terbantu]	[alhamdulillah, terbantu]	alhamdulillah terbantu
44641	5	Adakan pembuatan sim secara online jg min	Pengguna Google	2022-03-16 18:10:54	Adakan pembuatan sim secara online jg min	adakan pembuatan sim secara online jg min	adakan pembuatan sim secara online juga min	[adakan, pembuatan, sim, secara, online, juga,...	[adakan, pembuatan, sim, online, min]	adakan pembuatan sim online min
43733	3	sayang nya untuk pembuatan SIM online di daera...	Pengguna Google	2022-03-29 12:11:26	sayang nya untuk pembuatan SIM online di daera...	sayang nya untuk pembuatan sim online di daera...	sayang nya untuk pembuatan sim online di daera...	[sayang, nya, untuk, pembuatan, sim, online, d...	[sayang, pembuatan, sim, online, daerah, manda...	sayang pembuatan sim online daerah mandailing ...
...	...	...	...	...	...	...	...	...	...	...
22820	1	Upload data eror trs...terjadi kesalahan,tungg...	Pengguna Google	2023-05-10 02:28:20	Upload data eror trsterjadi kesalahantunggu be...	upload data eror trsterjadi kesalahantunggu be...	upload data eror trsterjadi kesalahantunggu be...	[upload, data, eror, trsterjadi, kesalahantung...	[upload, data, eror, trsterjadi, kesalahantung...	upload data eror trsterjadi kesalahantunggu sa...
19341	5	Mantab sukses selalu untuk polri	Pengguna Google	2023-07-14 01:26:40	Mantab sukses selalu untuk polri	mantab sukses selalu untuk polri	mantab sukses selalu untuk polri	[mantab, sukses, selalu, untuk, polri]	[mantab, sukses, polri]	mantab sukses polri

14. Mengirim permintaan untuk membaca kamus kata positif dan negative dari file CSV GitHub untuk digunakan dalam analisis sentiment.
    - lexicon_positive = dict() digunakan untuk membuat kamus kosong untuk menyimpan kata-kata positif beserta skornya.
    - StringIO(response.text) digunakan untuk mengubah teks yang diunduh menjadi dile like object agar bisa dibaca sebagai file CSV.
    - lexicon_positive[row[0]] = int(row[1]) digunakan untuk menambahkan kata positif dan skornya ke kamus lexion_positive.
    - Begitu pula dengan kamus kata negatifnya.

    Tujuan utama membaca kamus kata positif dan negatif dari GitHub ini adalah untuk untuk digunakan dalam analisis sentimen berbasis leksikon (Lexicon-based Sentiment Analysis). Skor yang terkait dengan kata dapat digunakan untuk menghitung nilai sentimen teks tertentu.

[19]:
```python
#pelabelan
import csv
import requests
from io import StringIO

# Loads positive lexicon data from GitHub
# Membaca data kamus kata-kata positif dari GitHub
lexicon_positive = dict()

response = requests.get('https://raw.githubusercontent.com/angelmetanosaa/dataset/main/lexicon_positive.csv')
# Mengirim permintaan HTTP untuk mendapatkan file CSV dari GitHub

if response.status_code == 200:
    # Jika permintaan berhasil
    reader = csv.reader(StringIO(response.text), delimiter=',')
    # Membaca teks respons sebagai file CSV menggunakan pembaca CSV dengan pemisah koma

    for row in reader:
        # Mengulangi setiap baris dalam file CSV
        lexicon_positive[row[0]] = int(row[1])
        # Menambahkan kata-kata positif dan skornya ke dalam kamus lexicon_positive
else:
    print("Failed to fetch positive lexicon data")

# Loads negative lexicon data from GitHub
# Membaca data kamus kata-kata negatif dari GitHub
lexicon_negative = dict()

response = requests.get('https://raw.githubusercontent.com/angelmetanosaa/dataset/main/lexicon_negative.csv')
# Mengirim permintaan HTTP untuk mendapatkan file CSV dari GitHub
```

```python
if response.status_code == 200:
    # Jika permintaan berhasil
    reader = csv.reader(StringIO(response.text), delimiter=',')
    # Membaca teks respons sebagai file CSV menggunakan pembaca CSV dengan pemisah koma

    for row in reader:
        # Mengulangi setiap baris dalam file CSV
        lexicon_negative[row[0]] = int(row[1])
        # Menambahkan kata-kata negatif dan skornya ke dalam kamus lexicon_negative
else:
    print("Failed to fetch negative lexicon data")
```

15. Menganalisis sentiment sebuah teks menggunakan pendekatan leksikon dengan kamus kata positif dan negative.
    - Score digunakan untuk menyimpan total skor sentiment teks yang dimulai dari 0.
    - Polarity digunakan untuk menyimpan label sentiment akhir.
    - Setiap kata dalam teks akan dibandingkan dengan kamus lexicon_positive, kemudian jika kata ditemukan maka skornya ditambahkan ke score. Begitu pula dengan kata negative.
    - Sentiment positif berskor >=1, sentiment negative berskor <=1, dan sentiment netral berskor antara -1 dan 1.

```
[20]: # Function to determine sentiment polarity of tweets
      # Fungsi untuk menentukan polaritas sentimen dari tweet

      def sentiment_analysis_lexicon_indonesia(text):
          #for word in text:

          score = 0
          # Inisialisasi skor sentimen ke 0

          for word in text:
              # Mengulangi setiap kata dalam teks

              if (word in lexicon_positive):
                  score = score + lexicon_positive[word]
                  # Jika kata ada dalam kamus positif, tambahkan skornya ke skor sentimen

          for word in text:
              # Mengulangi setiap kata dalam teks (sekali lagi)

              if (word in lexicon_negative):
                  score = score + lexicon_negative[word]
                  # Jika kata ada dalam kamus negatif, kurangkan skornya dari skor sentimen

          polarity=''
          # Inisialisasi variabel polaritas

          if (score >= 1):
              polarity = 'positive'
              # Jika skor sentimen lebih besar atau sama dengan 0, maka polaritas adalah positif
          elif (score < -1):
              polarity = 'negative'
              # Jika skor sentimen kurang dari 0, maka polaritas adalah negatif
          else:
              polarity = 'neutral'
          # Ini adalah bagian yang bisa digunakan untuk menentukan polaritas netral jika diperlukan

          return score, polarity
          # Mengembalikan skor sentimen dan polaritas teks
```

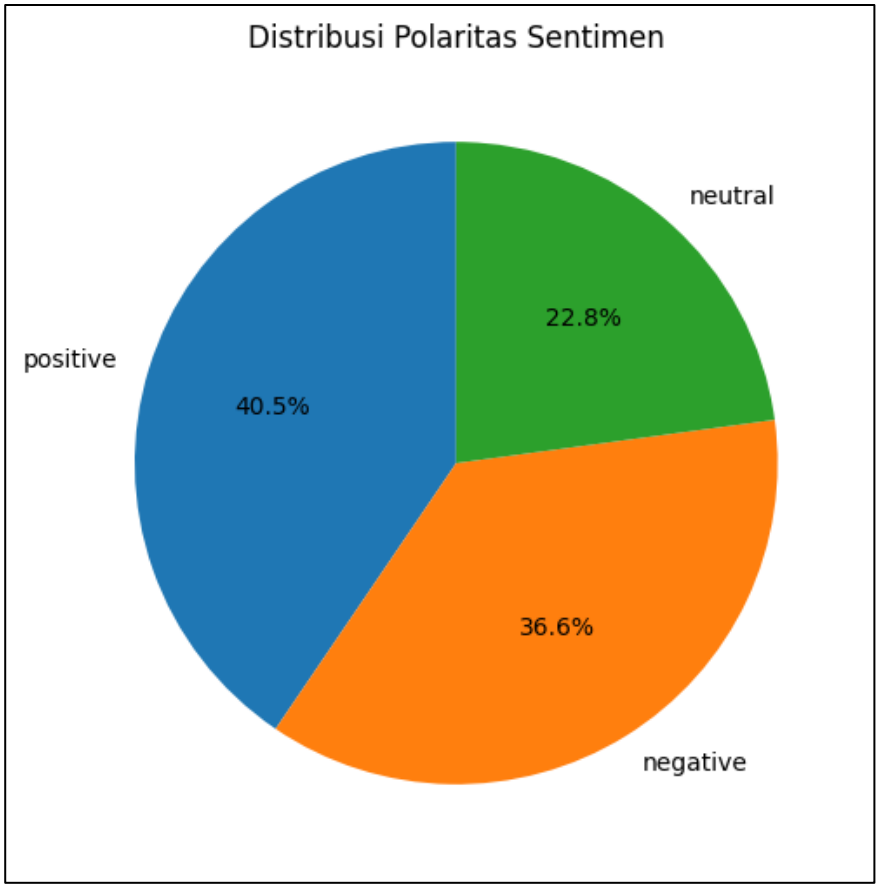
16. Penerapan analisis sentiment dan peyampaian hasil.
    - Pada baris pertama, kode digunakan untuk menerapkan fungsi analisis sentiment kepada setiap teks pada kolom file new_df.
    - Baris kedua digunakan untuk memisahkan skor dan polaritas dari daftar result.
    - Baris ketiga digunakan ntuk menyimpan skor sentiment.

- Baris keempat digunakan untuk menyimpan label polaritas (positive, negative, neutral).
- Baris kelima digunakan untuk menghitung jumlah kemunculan setiap label sentiment dan menampilkannya.

```
[21]: results = new_df['text_stopword'].apply(sentiment_analysis_lexicon_indonesia)
      results = list(zip(*results))
      new_df['polarity_score'] = results[0]
      new_df['polarity'] = results[1]
      print(new_df['polarity'].value_counts())

      polarity
      positive    6076
      negative    5497
      neutral     3427
      Name: count, dtype: int64
```

17. Membuat visualisasi distribusi polaritas sentiment dari dataset yang telah dianalisis, menggunakan diagram lingkaran.
    - polarity_counts = new_df['polarity'].value_counts() digunakan untuk menghitung jumlah kemunculan setiap kategori sentimen dalam kolom polarity.
    - polarilty_counts yaitu data untuk nilai yang akan digambarkan pada pie chart (jumlah setiap polaritas).
    - labels=polarity_counts.index digunakan untuk menambahkan label berdasarkan nilai yang ada pada indeks polarity_counts (seperti positive, negative, neutral).
    - autopct='%1.1f%%' digunakan untuk menampilkan persentase untuk masing-masing bagian diagram dalam format angka dengan satu desimal.
    - startangle=90 digunakan untuk memutar diagram sehingga bagian pertama dimulai dari sudut 90 derajat (searah jarum jam).

```
[22]: import matplotlib.pyplot as plt
      # Menghitung jumlah setiap nilai dalam kolom 'polarity'
      polarity_counts = new_df['polarity'].value_counts()

      # Membuat plot pie
      plt.figure(figsize=(8, 6))  # Mengatur ukuran plot
      plt.pie(polarity_counts, labels=polarity_counts.index, autopct='%1.1f%%', startangle=90)
      plt.title('Distribusi Polaritas Sentimen')  # Menambahkan judul plot
      plt.show()  # Menampilkan plot
```



18. Mengolah dataframe positif, negative, dan netral.

a. Dataframe positif

```
[23]: # Mengatur opsi tampilan Pandas agar kolom dapat menampilkan teks hingga 3000 karakter.
pd.set_option('display.max_colwidth', 3000)

# Membuat DataFrame baru 'positive_tweets' yang hanya berisi tweet dengan polaritas positif.
positive_tweets = new_df[new_df['polarity'] == 'positive']

# Memilih hanya kolom-kolom tertentu dari DataFrame 'positive_tweets'.
positive_tweets = positive_tweets[['text_akhir', 'polarity_score', 'polarity','text_stopword']]

# Mengurutkan DataFrame 'positive_tweets' berdasarkan 'polarity_score' secara menurun.
positive_tweets = positive_tweets.sort_values(by='polarity_score', ascending=False)

# Mengatur ulang indeks DataFrame agar dimulai dari 0.
positive_tweets = positive_tweets.reset_index(drop=True)

# Menambahkan 1 ke semua indeks DataFrame.
positive_tweets.index += 1
```

b. Dataframe negative

```
[24]: # Mengatur opsi tampilan Pandas agar kolom dapat menampilkan teks hingga 3000 karakter.
pd.set_option('display.max_colwidth', 3000)

# Membuat DataFrame baru 'negative_tweets' yang hanya berisi tweet dengan polaritas negatif.
negative_tweets = new_df[new_df['polarity'] == 'negative']

# Memilih hanya kolom-kolom tertentu dari DataFrame 'negative_tweets'.
negative_tweets = negative_tweets[['text_akhir', 'polarity_score', 'polarity','text_stopword']]

# Mengurutkan DataFrame 'negative_tweets' berdasarkan 'polarity_score' secara menaik (ascending).
negative_tweets = negative_tweets.sort_values(by='polarity_score', ascending=True)

# Memilih 10 baris pertama dari DataFrame yang sudah diurutkan.
negative_tweets = negative_tweets[0:10]

# Mengatur ulang indeks DataFrame agar dimulai dari 0.
negative_tweets = negative_tweets.reset_index(drop=True)

# Menambahkan 1 ke semua indeks DataFrame.
negative_tweets.index += 1
```

c. Dataframe netral

```
[25]: # Mengatur opsi tampilan Pandas agar kolom dapat menampilkan teks hingga 3000 karakter.
pd.set_option('display.max_colwidth', 3000)

# Membuat DataFrame baru 'negative_tweets' yang hanya berisi tweet dengan polaritas negatif.
neutral_tweets = new_df[new_df['polarity'] == 'neutral']

# Memilih hanya kolom-kolom tertentu dari DataFrame 'negative_tweets'.
neutral_tweets = negative_tweets[['text_akhir', 'polarity_score', 'polarity','text_stopword']]

# Mengurutkan DataFrame 'negative_tweets' berdasarkan 'polarity_score' secara menaik (ascending).
neutral_tweets = negative_tweets.sort_values(by='polarity_score', ascending=True)

# Memilih 10 baris pertama dari DataFrame yang sudah diurutkan.
neutral_tweets = negative_tweets[0:10]

# Mengatur ulang indeks DataFrame agar dimulai dari 0.
neutral_tweets = negative_tweets.reset_index(drop=True)

# Menambahkan 1 ke semua indeks DataFrame.
neutral_tweets.index += 1
```

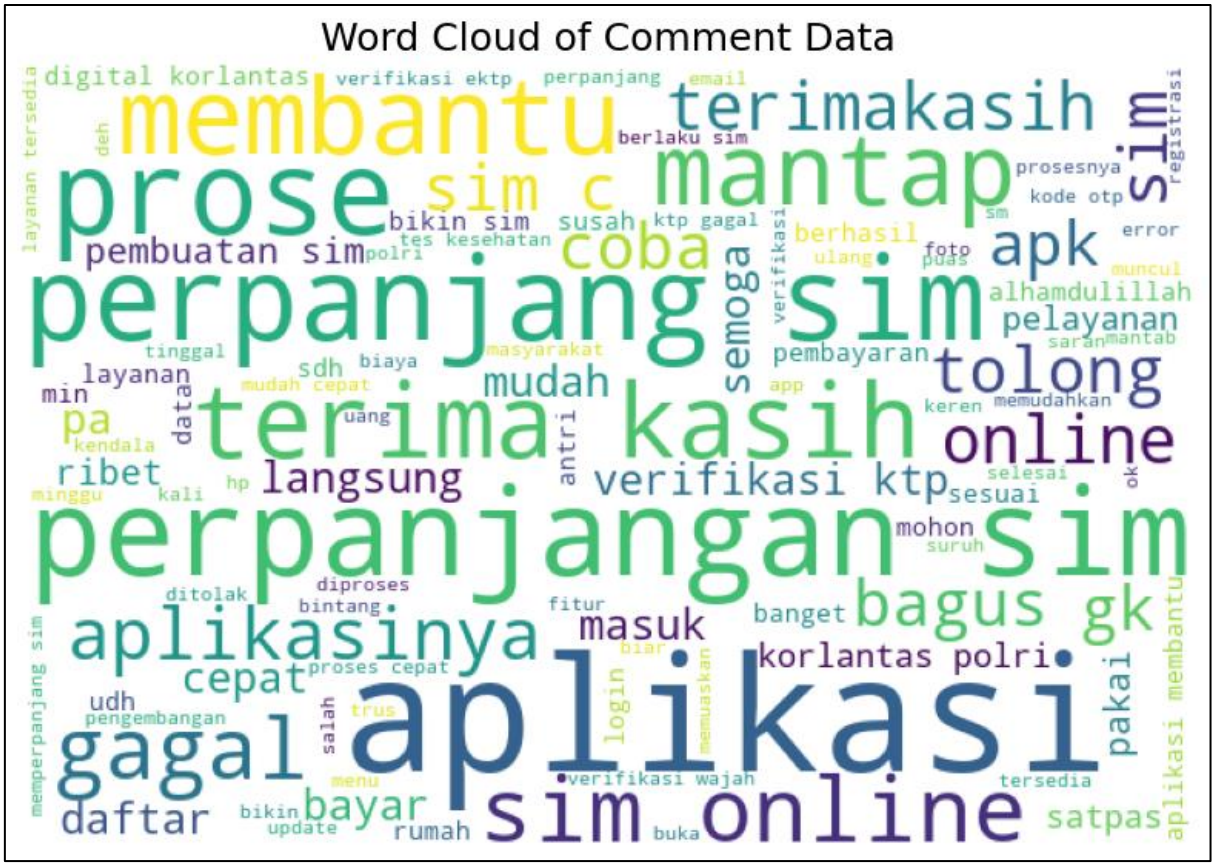
D. VISUALISASI

1. Menginstall WordCloud untuk visualisasi data teks. Wordcloud bekerja dengan menghitung frekuensi kemunculan kata dalam teks, kemudian menampilkan kata-kata tersebut dalam bentuk visual dengan ukuran font yang proporsional terhadap frekuensinya. Misalnya, kata yang muncul 100 kali akan lebih besar dari kata yang hanya muncul 5 kali.

```
[26]: pip install wordcloud

      Requirement already satisfied: wordcloud in c:\users\acer\appdata\local\programs\pytho
      Requirement already satisfied: numpy>=1.6.1 in c:\users\acer\appdata\local\programs\py
      Requirement already satisfied: pillow in c:\users\acer\appdata\local\programs\python\p
      Requirement already satisfied: matplotlib in c:\users\acer\appdata\local\programs\pyth
      Requirement already satisfied: contourpy>=1.0.1 in c:\users\acer\appdata\local\program
      Requirement already satisfied: cycler>=0.10 in c:\users\acer\appdata\local\programs\py
      Requirement already satisfied: fonttools>=4.22.0 in c:\users\acer\appdata\local\progra
      Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\acer\appdata\local\progra
      Requirement already satisfied: packaging>=20.0 in c:\users\acer\appdata\local\programs
      Requirement already satisfied: pyparsing>=2.3.1 in c:\users\acer\appdata\local\program
      Requirement already satisfied: python-dateutil>=2.7 in c:\users\acer\appdata\local\pro
      Requirement already satisfied: six>=1.5 in c:\users\acer\appdata\local\programs\python
      Note: you may need to restart the kernel to use updated packages.

[27]: from wordcloud import WordCloud
```

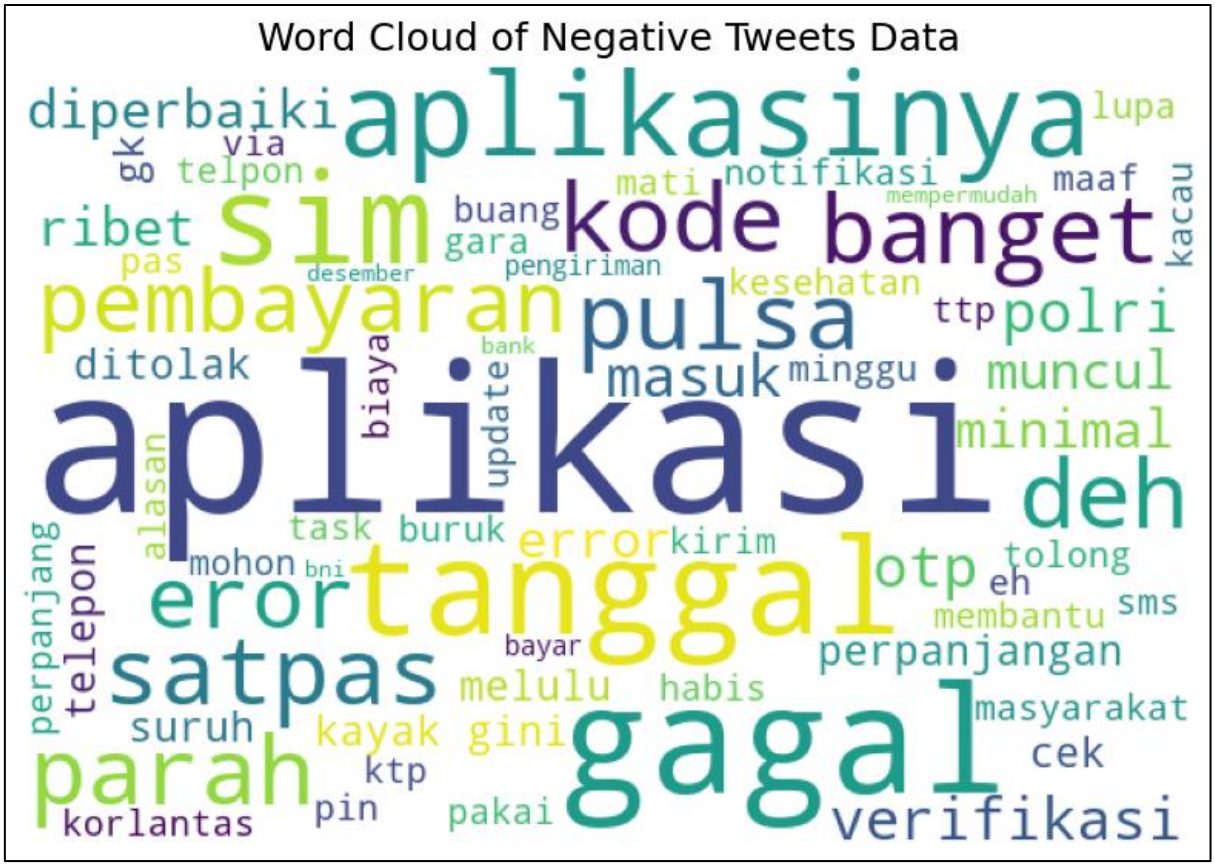
2. Menampilkan word cloud.
   - list_words adalah sebuah string kosong yang nantinya akan digunakan untuk mengumpulkan semua kata yang sudah dibersihkan dari kolom 'text_stopword' di DataFrame new_df
   - wordcloud = WordCloud(width=600, height=400, background_color='white', min_font_size=10).generate(list_words) digunakan untuk membuat objek WordCloud.
   - .generate(list_words) digunakan untuk menghasilkan word cloud berdasarkan list_words yang berisi semua kata yang telah dibersihkan.

```
[28]: # Membuat string kosong 'list_words' yang akan digunakan untuk mengumpulkan semua kata dari teks yang sudah dibersihkan.
      list_words = ''

      # Iterasi melalui setiap tweet dalam kolom 'text_stopword' dari DataFrame 'clean_df'.
      for tweet in new_df['text_stopword']:
          # Iterasi melalui setiap kata dalam tweet.
          for word in tweet:
              # Menambahkan kata ke dalam 'list_words'.
              list_words += ' ' + (word)

      # Membuat objek WordCloud dengan parameter tertentu.
      wordcloud = WordCloud(width=600, height=400, background_color='white', min_font_size=10).generate(list_words)

      # Membuat gambar dan sumbu untuk menampilkan word cloud.
      fig, ax = plt.subplots(figsize=(8, 6))

      # Menetapkan judul untuk word cloud.
      ax.set_title('Word Cloud of Comment Data', fontsize=18)

      # Menonaktifkan grid pada sumbu.
      ax.grid(False)

      # Menampilkan word cloud dalam gambar.
      ax.imshow((wordcloud))

      # Mengatur layout gambar.
      fig.tight_layout(pad=0)

      # Menyembunyikan sumbu.
      ax.axis('off')

      # Menampilkan word cloud.
      plt.show()
```



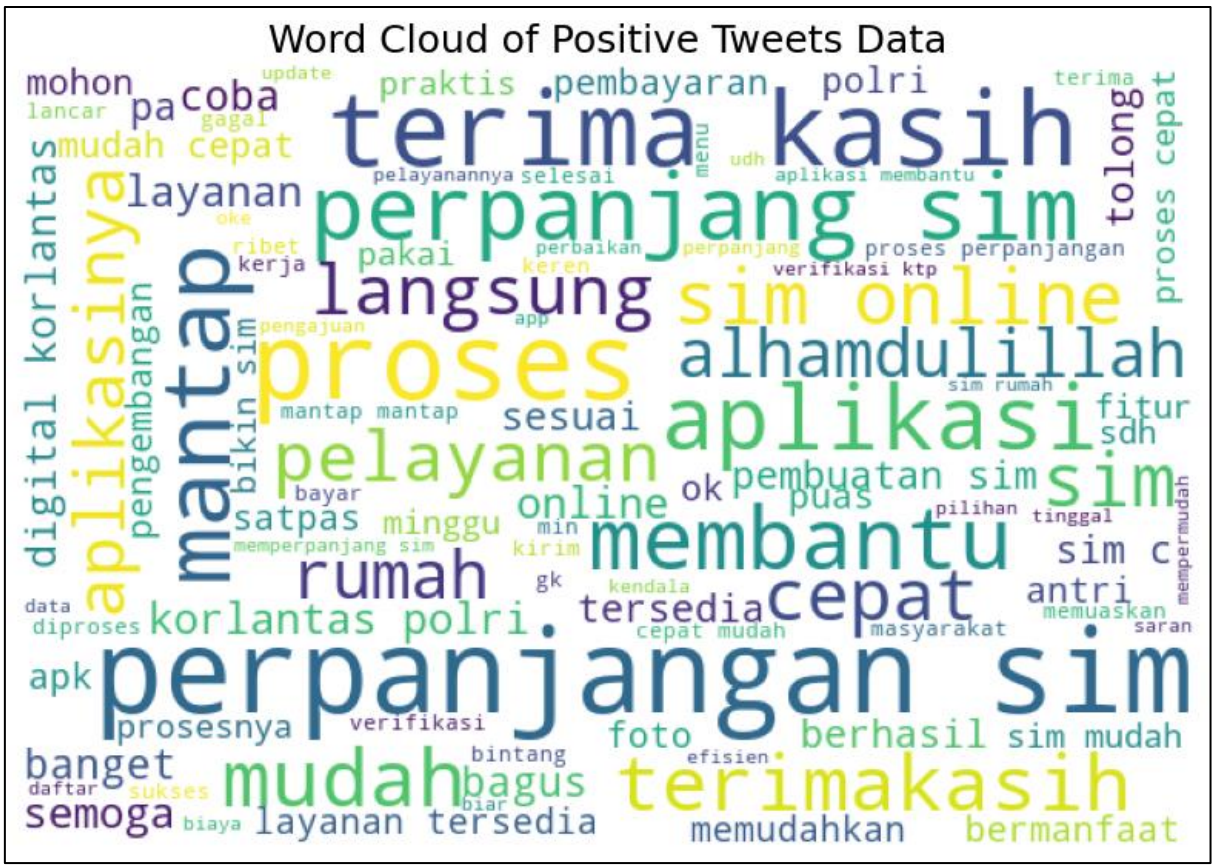
3. Menampilkan word cloud kata-kata negative.

```
[29]: # Membuat string kosong 'list_words' yang akan digunakan untuk mengumpulkan semua kata dari teks yang sudah dibersihkan dalam tweet negatif.
list_words = ''

# Iterasi melalui setiap tweet dalam kolom 'text_stopword' dari DataFrame 'negative_tweets'.
for tweet in negative_tweets['text_stopword']:
    # Iterasi melalui setiap kata dalam tweet.
    for word in tweet:
        # Menambahkan kata ke dalam 'list_words'.
        list_words += ' ' + (word)

# Membuat objek WordCloud dengan parameter tertentu.
wordcloud = WordCloud(width=600, height=400, background_color='white', min_font_size=10).generate(list_words)

# Membuat gambar dan sumbu untuk menampilkan word cloud.
fig, ax = plt.subplots(figsize=(8, 6))

# Menetapkan judul untuk word cloud.
ax.set_title('Word Cloud of Negative Tweets Data', fontsize=18)

# Menonaktifkan grid pada sumbu.
ax.grid(False)

# Menampilkan word cloud dalam gambar.
ax.imshow((wordcloud))

# Mengatur layout gambar.
fig.tight_layout(pad=0)

# Menyembunyikan sumbu.
ax.axis('off')

# Menampilkan word cloud.
plt.show()
```



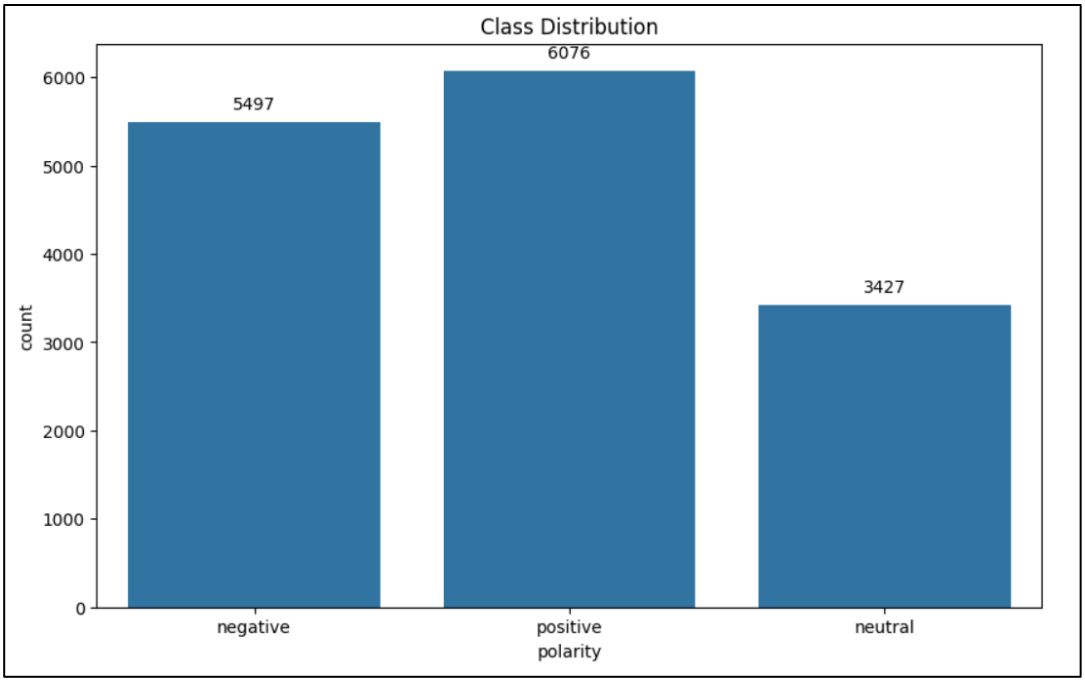
4. Menampilkan word cloud kata-kata positif.

```
[30]: # Membuat string kosong 'list_words' yang akan digunakan untuk mengumpulkan semua kata dari teks yang sudah dibersihkan dalam tweet positif.
list_words = ''

# Iterasi melalui setiap tweet dalam kolom 'text_stopword' dari DataFrame 'positive_tweets'.
for tweet in positive_tweets['text_stopword']:
    # Iterasi melalui setiap kata dalam tweet.
    for word in tweet:
        # Menambahkan kata ke dalam 'list_words'.
        list_words += ' ' + (word)

# Membuat objek WordCloud dengan parameter tertentu.
wordcloud = WordCloud(width=600, height=400, background_color='white', min_font_size=10).generate(list_words)

# Membuat gambar dan sumbu untuk menampilkan word cloud.
fig, ax = plt.subplots(figsize=(8, 6))

# Menetapkan judul untuk word cloud.
ax.set_title('Word Cloud of Positive Tweets Data', fontsize=18)

# Menonaktifkan grid pada sumbu.
ax.grid(False)

# Menampilkan word cloud dalam gambar.
ax.imshow((wordcloud))

# Mengatur layout gambar.
fig.tight_layout(pad=0)

# Menyembunyikan sumbu.
ax.axis('off')

# Menampilkan word cloud.
plt.show()
```



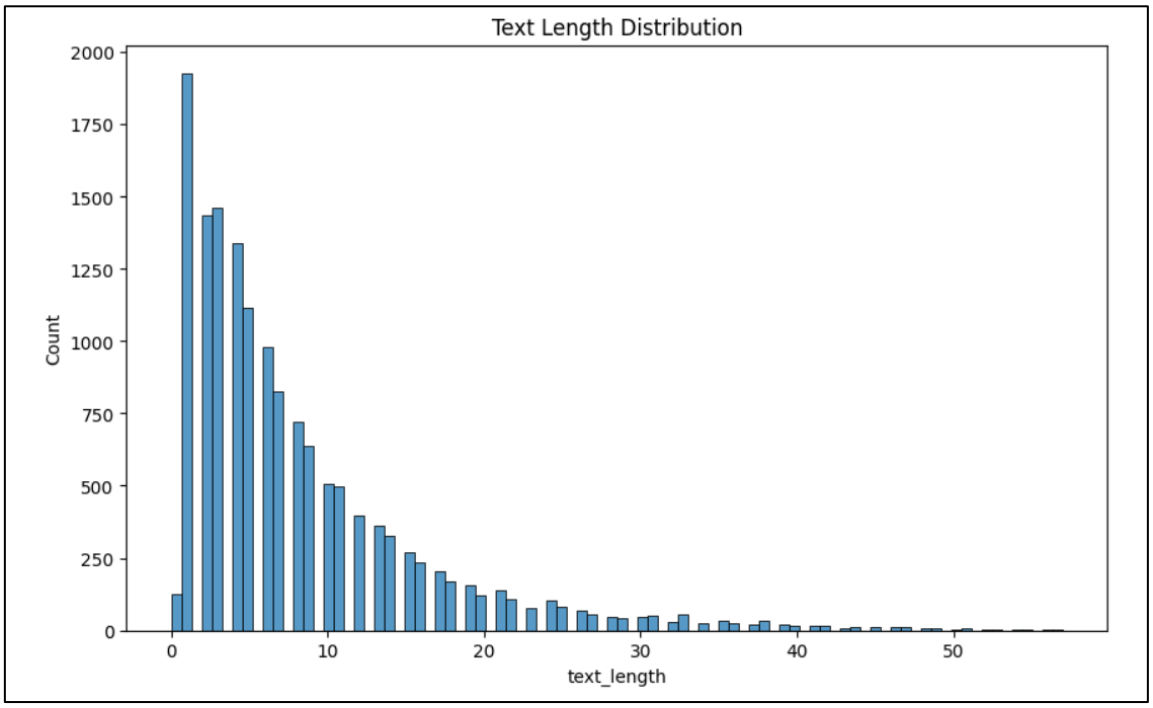
5. Visualisasi distribusi kelas sentiment,

```
[31]: import seaborn as sns
      import matplotlib.pyplot as plt
      from sklearn.feature_extraction.text import TfidfVectorizer

      # Set the figure size
      plt.figure(figsize=(10, 6))

      # Check class distribution
      class_dist_plot = sns.countplot(x='polarity', data=new_df)
      plt.title('Class Distribution')

      # Add counts above the bars on the plot
      for p in class_dist_plot.patches:
          class_dist_plot.annotate(format(p.get_height(), '.0f'),
                                   (p.get_x() + p.get_width() / 2., p.get_height()),
                                   ha='center', va='center',
                                   xytext=(0, 10),
                                   textcoords='offset points')

      plt.show()
```



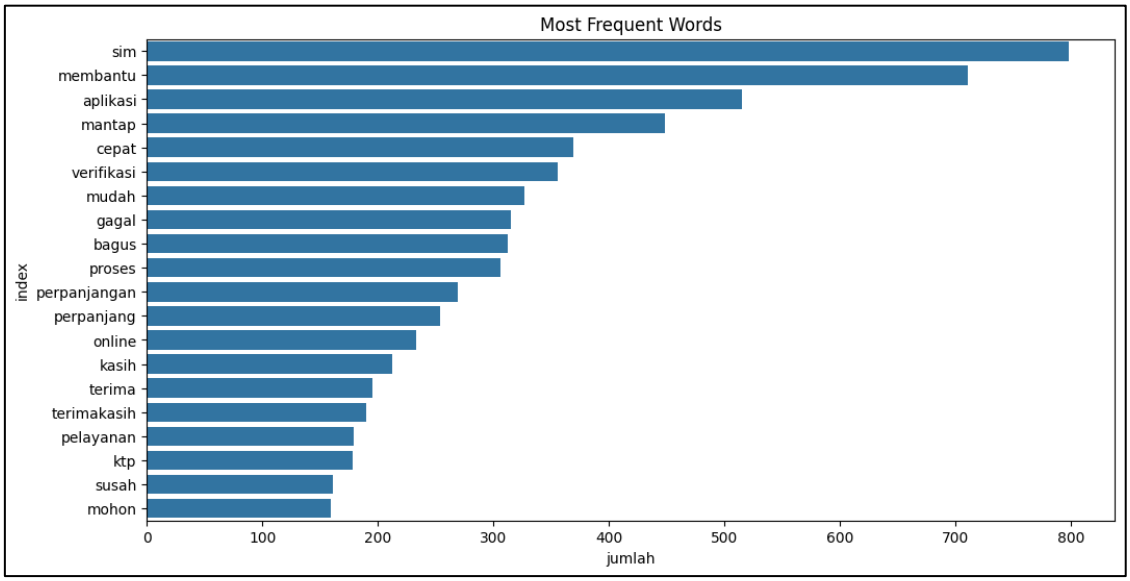
6. Visualisasi distribusi panjang teks.

```
# Set the figure size
plt.figure(figsize=(10, 6))

# Visualize text length distribution
new_df['text_length'] = new_df['text_akhir'].apply(lambda x: len(x.split()))
sns.histplot(new_df['text_length'])
plt.title('Text Length Distribution')
plt.show()
```



7. Visualisasi 20 kata paling sering muncul.

```
# Set the figure size
plt.figure(figsize=(12, 6))

# Visualize most frequent words
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(new_df['text_akhir'])
tfidf_df = pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names_out())
tfidf_df = tfidf_df.sum().reset_index(name='jumlah')
tfidf_df = tfidf_df.sort_values('jumlah', ascending=False).head(20)
sns.barplot(x='jumlah', y='index', data=tfidf_df)
plt.title('Most Frequent Words')
plt.show()
```



E. KESIMPULAN

Dalam laporan ini, kami telah menyelesaikan proses analisis sentimen terhadap ulasan aplikasi Digital Korlantas POLRI dengan pendekatan berbasis leksikon. Proses dimulai dari pengumpulan data menggunakan teknik web scraping, dilanjutkan dengan pembersihan teks, analisis sentimen, dan visualisasi hasil. Berdasarkan hasil analisis, dapat disimpulkan bahwa ulasan pengguna aplikasi Digital Korlantas POLRI di Google Play Store sebagian besar berisi ulasan bersifat positif, yang menunjukkan kepuasan pengguna terhadap fitur dan fungsi aplikasi. Namun, ulasan negatif juga memberikan masukan terkait masalah teknis dan kekurangan fitur yang perlu segera diperbaiki. Pendekatan berbasis leksikon yang digunakan dalam analisis sentimen ini terbukti efektif untuk memahami pola sentimen secara umum. Hasil analisis ini diharapkan dapat menjadi dasar bagi pengembang untuk meningkatkan performa aplikasi dan merancang pembaruan sistem yang lebih sesuai dengan kebutuhan pengguna.