MINI PROJECT UTS PEMROSESAN BAHASA ALAMI

Analisis Sentiment Bersumber dari Video Youtube dengan Hashtag #indonesiagelap Menggunakan Library TextBlob dan Pre-train Model BERT

disusun untuk memenuhi
Ujian Tengah Semester matakuliah
Pemrosesan Bahasa Alami

oleh :

NAUFAL AQIL
(2208107010043)

DEPARTEMEN INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SYIAH KUALA

TAHUN 2025

DAFTAR ISI

1. Pendahuluan

1.1 Latar Belakang

Perkembangan media sosial telah membuka peluang bagi masyarakat untuk mengekspresikan opini mereka terhadap berbagai isu sosial, politik, dan ekonomi. Salah satu platform yang sering digunakan untuk berdiskusi adalah YouTube, di mana komentar dalam video dapat memberikan wawasan terhadap persepsi publik.

Hashtag #indonesiagelap menjadi salah satu topik yang menarik perhatian karena berisi diskusi mengenai berbagai permasalahan yang dihadapi Indonesia. Analisis sentimen terhadap komentar yang muncul pada video dengan hashtag ini dapat membantu memahami kecenderungan opini masyarakat, baik dalam kategori positif, negatif, maupun netral.

Dalam penelitian ini, dilakukan perbandingan dua pendekatan analisis sentimen, yaitu TextBlob, yang berbasis leksikon, dan model BERT (nlptown/bert-base-multilingual-uncased-sentiment), yang berbasis pembelajaran mesin. Dengan membandingkan kedua metode ini, diharapkan diperoleh pemahaman mengenai keakuratan dan efektivitas masing-masing model dalam menganalisis sentimen dalam bahasa Indonesia setelah diterjemahkan ke bahasa Inggris.

1.2 Tujuan

Penelitian ini bertujuan untuk:

1. Menganalisis sentimen komentar pada YouTube terkait hashtag #indonesiagelap.
2. Membandingkan performa metode berbasis leksikon (TextBlob) dengan metode berbasis deep learning (BERT) dalam klasifikasi sentimen.
3. Mengevaluasi efektivitas model dalam mengklasifikasikan sentimen positif, negatif, dan netral.

1.3 Rumusan Masalah

1. Bagaimana kecenderungan sentimen masyarakat terhadap hashtag #indonesiagelap di YouTube?
2. Bagaimana perbandingan performa metode TextBlob dan BERT dalam analisis sentimen?
3. Apa kelebihan dan kekurangan masing-masing metode dalam menangani komentar berbahasa Indonesia?

2. Kajian Pustaka

2.1 Analisis Sentimen

Analisis sentimen merupakan cabang dari pemrosesan bahasa alami (Natural Language Processing/NLP) yang bertujuan untuk menentukan sikap atau perasaan seseorang terhadap suatu subjek berdasarkan teks yang ditulisnya. Analisis ini sering digunakan dalam berbagai bidang seperti pemasaran, politik, dan layanan pelanggan untuk memahami opini publik. Dalam konteks media sosial, analisis sentimen sangat berguna untuk mengolah dan memahami persepsi masyarakat terhadap isu tertentu.

Dalam penelitian yang dilakukan oleh F. Illia et al. (2021) tentang analisis sentimen terhadap aplikasi PeduliLindungi menggunakan TextBlob dan VADER, ditemukan bahwa metode berbasis leksikon dapat memberikan wawasan yang cukup baik mengenai opini masyarakat. Studi tersebut menyoroti bahwa penggunaan analisis sentimen berbasis leksikon, seperti TextBlob dan VADER, mampu memberikan pemetaan umum terhadap sentimen yang berkembang di media sosial. Namun, terdapat beberapa keterbatasan, terutama dalam menangani konteks yang lebih kompleks dan nuansa bahasa yang lebih dalam. Hal ini mengindikasikan bahwa metode berbasis pembelajaran mesin dapat memberikan hasil yang lebih akurat dalam beberapa kasus.

2.2 Model Berbasis Pembelajaran Mesin untuk Analisis Sentimen

Selain metode berbasis leksikon, pendekatan berbasis pembelajaran mesin seperti BERT telah terbukti memberikan hasil yang lebih akurat dalam analisis sentimen. Model BERT (Bidirectional Encoder Representations from Transformers) adalah salah satu model deep learning yang banyak digunakan dalam tugas NLP karena kemampuannya dalam memahami konteks kalimat secara lebih baik dibandingkan metode sebelumnya. Dalam studi oleh H. Koto et al. (2022), BERT menunjukkan kinerja yang superior dalam analisis sentimen terhadap ulasan film. Dengan menggunakan model IndoBERT, penelitian tersebut mampu mengklasifikasikan sentimen dalam bahasa Indonesia dengan lebih akurat dibandingkan dengan pendekatan berbasis leksikon.

Penelitian ini menunjukkan bahwa model pembelajaran mesin, terutama yang berbasis transformer seperti BERT, memiliki kemampuan lebih baik dalam menangani nuansa bahasa yang lebih kompleks dan memahami konteks dengan lebih baik dibandingkan metode berbasis leksikon. Oleh karena itu, dalam proyek ini, digunakan dua pendekatan, yaitu metode berbasis leksikon (TextBlob) dan metode berbasis pembelajaran mesin (IndoBERT), untuk membandingkan efektivitas keduanya dalam menganalisis sentimen komentar YouTube terkait hashtag #indonesiagelap.

2.3 Pentingnya Preprocessing dalam Analisi Sentimen

Proses preprocessing data teks sangat krusial dalam analisis sentimen karena dapat meningkatkan akurasi model yang digunakan. Beberapa teknik preprocessing yang umum dilakukan meliputi:

1. **Cleansing**: Menghapus karakter yang tidak diperlukan seperti tanda baca, angka, dan tautan URL.
2. **Tokenisasi**: Memecah teks menjadi kata-kata atau frasa untuk diproses lebih lanjut.
3. **Stemming dan Lemmatization**: Mengubah kata menjadi bentuk dasar agar lebih mudah dianalisis.
4. **Stopword Removal**: Menghapus kata-kata umum yang tidak memiliki makna signifikan dalam analisis sentimen.

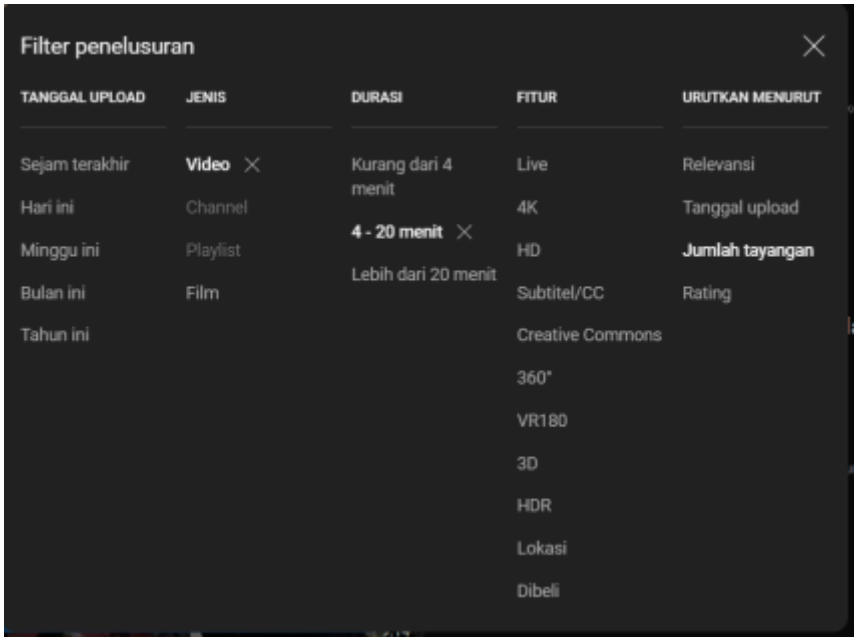
Dalam penelitian ini, preprocessing dilakukan dengan membersihkan teks dari elemen-elemen yang tidak relevan, melakukan normalisasi, menghapus stopwords, serta menerapkan stemming menggunakan library NLP dalam Python. Selain itu, data komentar yang awalnya dalam bahasa Indonesia diterjemahkan ke bahasa Inggris agar dapat diproses dengan lebih optimal oleh model TextBlob dan BERT.

Dari berbagai studi yang telah ditinjau, dapat disimpulkan bahwa pendekatan berbasis leksikon dan berbasis pembelajaran mesin masing-masing memiliki kelebihan dan kekurangan dalam analisis sentimen. Pendekatan berbasis leksikon seperti TextBlob lebih mudah diterapkan dan memberikan hasil yang cepat, namun kurang akurat dalam menangani konteks yang kompleks. Sementara itu, pendekatan berbasis pembelajaran mesin seperti IndoBERT lebih unggul dalam menangkap nuansa bahasa dan konteks dalam teks, meskipun memerlukan sumber daya komputasi yang lebih besar. Oleh karena itu, penelitian ini mengadopsi kedua pendekatan tersebut untuk memperoleh gambaran yang lebih komprehensif mengenai sentimen publik terhadap isu yang diangkat dalam hashtag #indonesiagelap.

3. Metodologi

3.1 Data

Data dikumpulkan dengan metode web scraping dari komentar pada video YouTube yang memiliki hashtag #indonesiagelap. Pengambilan data dilakukan pada 10 video teratas yang muncul dalam pencarian menggunakan hashtag tersebut. Scraping dilakukan menggunakan API YouTube untuk mengakses dan mengekstrak komentar. berikut adalah filter yang digunakan dalam melakukan scrapping 10 video teratas.



3.2 Algoritma dan Teknik yang Digunakan

Preprocessing Data

1. **Data Cleaning**: Menghapus karakter khusus, emoji, angka, dan tanda baca yang tidak diperlukan.
2. **Normalisasi**: Mengonversi teks menjadi bentuk standar, seperti mengubah kata-kata tidak baku menjadi bentuk baku.
3. **Stopword Removal**: Menghapus kata-kata umum yang tidak memberikan makna signifikan.
4. **Tokenisasi**: Memecah teks menjadi kata-kata atau frasa kecil.
5. **Stemming**: Mengubah kata menjadi bentuk dasarnya.
6. **Penerjemahan ke Bahasa Inggris**: Menggunakan IndoBERT untuk menerjemahkan teks hasil preprocessing agar sesuai dengan model analisis sentimen berbasis bahasa Inggris.

Analisis Sentimen

1. **TextBlob**: Menggunakan pendekatan berbasis leksikon untuk menentukan polaritas komentar (positif, negatif, atau netral).
2. **BERT (nlptown/bert-base-multilingual-uncased-sentiment)**: Menggunakan model deep learning yang telah dilatih untuk analisis sentimen multibahasa.

Pelabelan data dilakukan menggunakan kedua metode tersebut dan hasilnya dibandingkan.

Visualisasi Hasil

Hasil analisis sentimen dari kedua metode akan divisualisasikan dalam bentuk grafik menggunakan seaborn dan matplotlib untuk mempermudah perbandingan tren sentimen.

4. Hasil

4.1 Scraping Data dari Youtube

Proses pengumpulan data dilakukan melalui YouTube Data API v3 untuk mendapatkan komentar dari video dengan hashtag **#indonesiagelap**. API key yang diperoleh dari Google digunakan untuk mengakses data tersebut. Dari hasil scraping, diperoleh **12.484 baris data** yang berisi atribut utama seperti **Video_ID, Author, Published At, dan Comment**. Namun, untuk keperluan analisis sentimen, hanya kolom **Comment** yang digunakan. Berikut adalah cuplikan kode yang saya gunakan untuk melakukan scraping data dari youtube.

```python
import csv
import googleapiclient.discovery
import time

# API Key dari Google Cloud Console
API_KEY = "AIzaSyBCQeaksnlpZlc0bTVkp6yIN2bWkZuJRKI"

# Daftar Video ID dari link yang diberikan
VIDEO_IDS = [
    "yDfOFd8V8os", "4yO8mKKUTr8", "oYOYvoJqs4Q", "_zuaYeyy52g", "g2f-Nm0CSTk",
    "63XR0GGJciQ", "2dDtKmDgK18", "uN2RQyPEl1Y", "CzAwaR-dXqY"
]

# Fungsi untuk mengambil komentar dari video YouTube
def get_youtube_comments(video_id, max_comments=100000):
    youtube = googleapiclient.discovery.build("youtube", "v3", developerKey=API_KEY)

    comments = []
    next_page_token = None
    count = 0

    while True:
        try:
            request = youtube.commentThreads().list(
                part="snippet",
                videoId=video_id,
                maxResults=100,  # Maksimum per halaman
                pageToken=next_page_token
            )
            response = request.execute()

            for item in response.get("items", []):
                comment = item["snippet"]["topLevelComment"]["snippet"]
                comments.append([
                    video_id,
                    comment["authorDisplayName"],
                    comment["publishedAt"],
                    comment["textDisplay"]
                ])
                count += 1

                if count >= max_comments:
                    break

            next_page_token = response.get("nextPageToken")
            if not next_page_token or count >= max_comments:
                break

            # Tunggu sebentar untuk menghindari rate limit
            time.sleep(0.5)

        except Exception as e:
            print(f"Error pada video {video_id}: {e}")
            break

    return comments

# Simpan hasil ke CSV
def save_comments_to_csv(comments, filename="youtube_comments.csv"):
    with open(filename, "w", newline="", encoding="utf-8") as file:
        writer = csv.writer(file)
        writer.writerow(["Video_ID", "Author", "Published At", "Comment"])
        writer.writerows(comments)

# Scraping semua video
all_comments = []
for video_id in VIDEO_IDS:
    print(f"Scraping komentar dari video: {video_id}")
    comments = get_youtube_comments(video_id)
    all_comments.extend(comments)

# Simpan ke file CSV
save_comments_to_csv(all_comments)
print(f"Berhasil menyimpan {len(all_comments)} komentar ke 'youtube_comments.csv'")

```

4.2 Preprocessing Data

Sebelum dilakukan analisis sentimen, data komentar harus melalui beberapa tahap preprocessing:

1. **Penghapusan Data yang Hilang dan Duplikat**
    - Baris dengan nilai kosong dan duplikat dihapus untuk memastikan keakuratan data.
2. **Pembersihan Data (Cleaning)**
    - Menghapus karakter khusus seperti emoji, angka, tanda baca, serta format teks yang tidak diperlukan.
3. **Normalisasi Teks**
    - Mengonversi kata-kata tidak baku menjadi bentuk baku.
4. **Penghapusan Stopwords**
    - Menghilangkan kata-kata yang tidak memberikan informasi penting dalam analisis sentimen.
5. **Tokenisasi dan Stemming**
    - Memecah kalimat menjadi kata-kata serta mengubahnya ke bentuk dasar menggunakan metode stemming.

Berikut kode yang saya gunakan :

```python
# Fungsi untuk membersihkan teks
def clean_text(text):
    # Menghapus karakter khusus seperti @mentions, #hashtags, dan URL
    text = re.sub(r'@[A-Za-z0-9_]+', '', text)
    text = re.sub(r'#\w+', '', text)
    text = re.sub(r'RT[\s]+', '', text)
    text = re.sub(r'https?://\S+', '', text)

    text = re.sub(r'[^A-Za-z0-9 ]', '', text)
    text = re.sub(r'\s+', ' ', text).strip()

    return text
```

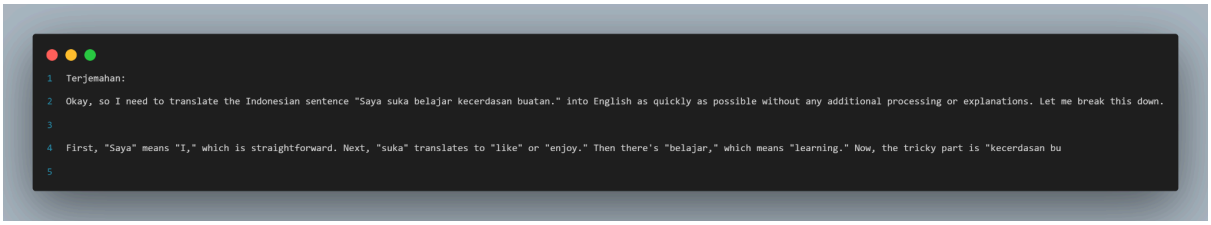
4.3 Kendala dalam Translasi Komentar

Pada tahap awal, direncanakan translasi komentar dari bahasa Indonesia ke bahasa Inggris untuk memanfaatkan model BERT berbasis bahasa Inggris. Namun, beberapa kendala ditemukan:

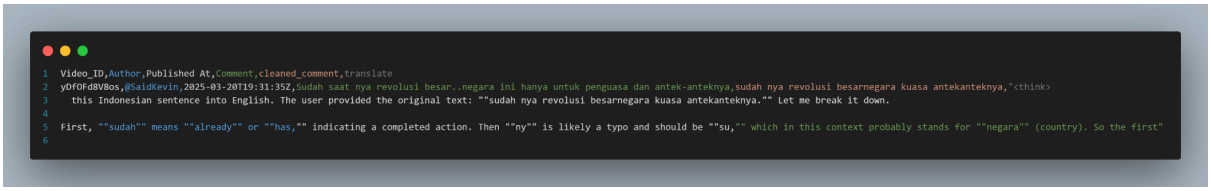
- **Batasan API DeepSeek**: Website DeepSeek memiliki limit pada jumlah request yang dapat dilakukan, sehingga saya mencoba menggunakan alternatif lain agar tetap menggunakan model DeepSeek sebagai sumber utama dalam translasi.
- **Solusi Alternatifnya Yaitu Menjalankan Model Secara Lokal**: Model **DeepSeek-R1-Distill-Qwen-7B-GGUF** dijalankan menggunakan **LM Studio**, tetapi hasilnya tidak langsung berupa terjemahan. Model cenderung memberikan respons tambahan seperti proses berpikir. Berikut cuplikan kode saat saya melakukan pengetesan pada [tes_translate_deepseek.csv]() :

```python
import requests

API_URL = "http://localhost:1234/v1/chat/completions"

payload = {
    "model": "DeepSeek-R1-Distill-Qwen-7B",
    "messages": [
        {"role": "system", "content": "Terjemahkan teks berikut dari Bahasa Indonesia ke Bahasa Inggris secepat mungkin tanpa pemrosesan tambahan atau penjelasan."},
        {"role": "user", "content": "Saya ingin menerjemahkan kalimat ini: 'Saya suka belajar kecerdasan buatan.'"}
    ],
    "max_tokens": 100,  # Batasi agar tidak berpikir terlalu lama
    "temperature": 0.1,  # Turunkan suhu agar hasil lebih deterministik
    "top_p": 0.8  # Batasi cakupan kemungkinan jawaban
}

response = requests.post(API_URL, json=payload)

if response.status_code == 200:
    print("Terjemahan:", response.json()["choices"][0]["message"]["content"])
else:
    print("Error:", response.text)
```

dan menghasilkan keluaran sebagai berikut



dapat dilihat, hasil yang dikeluarkan tidak langsung mengeluarkan terjemahan dari kalimat yang dimasukkan. Dia juga mengeluarkan proses think dari melakukan traslate tersebut. sehingga tidak bisa dimasukkan ke dalam dataframe untuk melanjutkan ke tahap selanjutnya. Hasil testing dari kode ini bisa dilihat di [translate.csv] berikut cuplikan hasilnya:



- **Alternatif Model Translasi juga Telah Dicoba**: Yaitu Model **Wikidepia/IndoT5-small**, tetapi hasilnya masih kurang akurat dan beberapa komentar tetap dalam bahasa Indonesia. Hasil translatenya dapat dilihat di [translated_comments_indot5.csv].
- **Kendala Skala Data**: Saya juga sudah mencoba untuk melakukan translate dengan menggunakan library **translate** namun hasil yang didapat juga tidak bagus dan juga dikarenakan data yang terlalu besar yang menyebabkan terkena limit untuk melakukan translasi. Hasil dari translate dengan library translate ini dapat dilihat di [final.csv].

Karena kendala ini, proses translasi **dilompati**, dan analisis sentimen dilakukan langsung pada teks dalam bahasa Indonesia.

4.4 Analisis Sentiment

Dua metode analisis sentimen diterapkan untuk membandingkan hasil klasifikasi komentar:

4.4.1 TextBlob (Pendekatan Leksikon)

TextBlob menghitung **polaritas** komentar berdasarkan leksikon.
Jika nilai polaritas **> 0**, komentar dikategorikan **Positif**.
Jika nilai polaritas **= 0**, komentar dikategorikan **Netral**.
Jika nilai polaritas **< 0**, komentar dikategorikan **Negatif**.

Hasil analisis ditampilkan dalam file [sentiment_labeled_comments.csv]. untuk proses labeling dengan textblob saya menggunakan kode berikut:

```python
import pandas as pd
from textblob import TextBlob
import nltk

nltk.download('punkt')

# **Membaca data hasil preprocessing**
file_path = "preprocessed_comments_stemming.csv"  # Sesuaikan dengan nama file hasil preprocessing
df = pd.read_csv(file_path)

df = df.dropna()

# **Pastikan kolom yang digunakan ada dalam data**
if "cleaned_comment" not in df.columns:
    raise ValueError("Kolom 'cleaned_comment' tidak ditemukan dalam file CSV. Pastikan preprocessing benar.")

# **Variabel untuk menyimpan hasil klasifikasi sentimen**
total_positif = total_negatif = total_netral = total = 0
status = []

# **Proses Sentiment Analysis**
for tweet in df["cleaned_comment"]:
    analysis = TextBlob(tweet)
    polarity = analysis.sentiment.polarity  # Skor polaritas dari TextBlob

    if polarity > 0.0:
        total_positif += 1
        status.append('Positif')
    elif polarity == 0.0:
        total_netral += 1
        status.append('Netral')
    else:
        total_negatif += 1
        status.append('Negatif')

    total += 1

# **Menambahkan hasil klasifikasi ke dalam DataFrame**
df["sentiment_label"] = status

# **Menampilkan hasil analisis**
print(f"Hasil Analisis Data:\nPositif = {total_positif}\nNetral = {total_netral}\nNegatif = {total_negatif}")
print(f"\nTotal Data: {total}")

# **Menyimpan hasil ke file baru**
output_file = "sentiment_labeled_comments.csv"
df.to_csv(output_file, index=False)

print(f"Data dengan label sentimen telah disimpan ke '{output_file}'.")

```

Kode ini melakukan analisis sentimen pada komentar yang telah diproses sebelumnya menggunakan TextBlob. Pertama, kode mengimpor pustaka yang diperlukan, seperti pandas untuk manipulasi data, TextBlob untuk analisis sentimen, dan nltk untuk tokenisasi teks, lalu mengunduh paket ‘punkt’ yang diperlukan oleh TextBlob. Selanjutnya, kode membaca data dari file CSV (**preprocessed_comments_stemming.csv**) yang berisi komentar yang telah dibersihkan melalui preprocessing. Untuk memastikan tidak ada nilai kosong dalam dataset, kode menghapus baris yang mengandung nilai NaN dan memverifikasi keberadaan kolom cleaned_comment. Jika kolom tersebut tidak ditemukan, program akan menghentikan eksekusi dengan menampilkan pesan error.

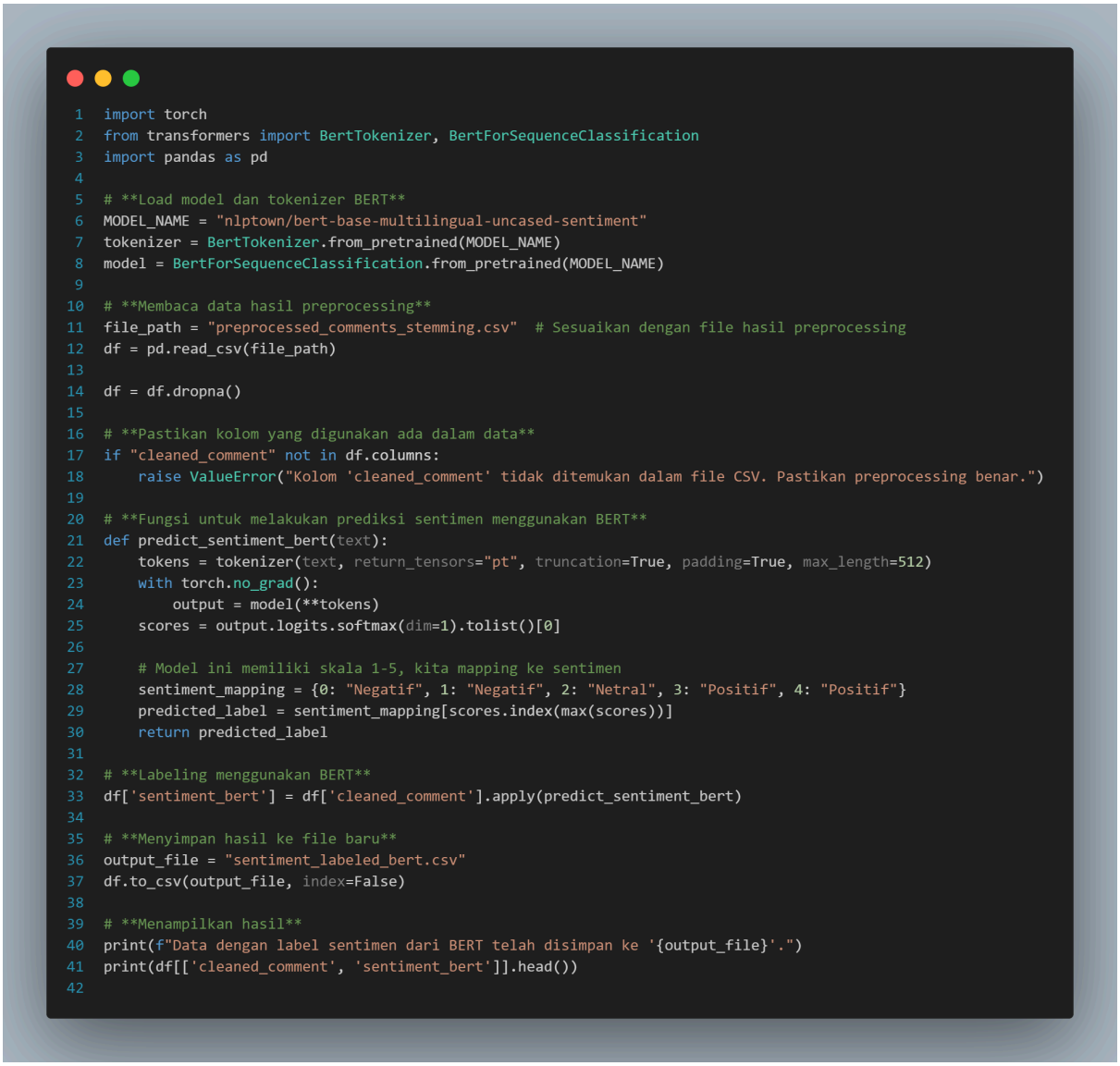
Setelah data siap, kode menginisialisasi beberapa variabel untuk menyimpan jumlah komentar positif, netral, dan negatif, serta daftar status sentimen. Kemudian, setiap komentar

dianalisis menggunakan TextBlob untuk menghitung polarity score, yaitu nilai antara -1 (negatif) hingga +1 (positif). Jika nilai polaritas lebih besar dari 0, komentar dikategorikan sebagai positif; jika sama dengan 0, dikategorikan netral; dan jika kurang dari 0, dikategorikan negatif. Klasifikasi ini disimpan dalam daftar status yang kemudian ditambahkan sebagai kolom baru (sentiment_label) dalam DataFrame.

Setelah semua data dianalisis, program mencetak hasil analisis berupa jumlah komentar dalam setiap kategori sentimen dan total komentar yang diproses. Akhirnya, DataFrame yang telah diberi label sentimen disimpan ke dalam file CSV baru (**sentiment_labeled_comments.csv**), dan pesan konfirmasi ditampilkan untuk memberi tahu pengguna bahwa hasil analisis telah berhasil disimpan.

4.4.2 BERT (nlptown/bert-base-multilingual-uncased-sentiment)

Model BERT digunakan untuk klasifikasi sentimen berbasis deep learning. Output model dalam skala 1-5 dikonversi ke **Negatif (0-1), Netral (2), dan Positif (3-4)**. Hasil klasifikasi disimpan dalam **sentiment_labeled_bert.csv**. Untuk proses labeling dengan menggunakan bert saya menggunakan kode berikut



```python
import torch
from transformers import BertTokenizer, BertForSequenceClassification
import pandas as pd

# **Load model dan tokenizer BERT**
MODEL_NAME = "nlptown/bert-base-multilingual-uncased-sentiment"
tokenizer = BertTokenizer.from_pretrained(MODEL_NAME)
model = BertForSequenceClassification.from_pretrained(MODEL_NAME)

# **Membaca data hasil preprocessing**
file_path = "preprocessed_comments_stemming.csv"  # Sesuaikan dengan file hasil preprocessing
df = pd.read_csv(file_path)

df = df.dropna()

# **Pastikan kolom yang digunakan ada dalam data**
if "cleaned_comment" not in df.columns:
    raise ValueError("Kolom 'cleaned_comment' tidak ditemukan dalam file CSV. Pastikan preprocessing benar.")

# **Fungsi untuk melakukan prediksi sentimen menggunakan BERT**
def predict_sentiment_bert(text):
    tokens = tokenizer(text, return_tensors="pt", truncation=True, padding=True, max_length=512)
    with torch.no_grad():
        output = model(**tokens)
    scores = output.logits.softmax(dim=1).tolist()[0]

    # Model ini memiliki skala 1-5, kita mapping ke sentimen
    sentiment_mapping = {0: "Negatif", 1: "Negatif", 2: "Netral", 3: "Positif", 4: "Positif"}
    predicted_label = sentiment_mapping[scores.index(max(scores))]
    return predicted_label

# **Labeling menggunakan BERT**
df['sentiment_bert'] = df['cleaned_comment'].apply(predict_sentiment_bert)

# **Menyimpan hasil ke file baru**
output_file = "sentiment_labeled_bert.csv"
df.to_csv(output_file, index=False)

# **Menampilkan hasil**
print(f"Data dengan label sentimen dari BERT telah disimpan ke '{output_file}'.")
print(df[['cleaned_comment', 'sentiment_bert']].head())

```

Kode ini menggunakan model BERT multilingual dari nlptown untuk melakukan analisis sentimen terhadap komentar yang telah diproses sebelumnya. Pertama, kode mengimpor pustaka yang diperlukan, seperti torch untuk komputasi tensor, transformers dari Hugging Face untuk memuat model BERT, serta pandas untuk manipulasi data. Model yang digunakan adalah "nlptown/bert-base-multilingual-uncased-sentiment", yang dirancang untuk klasifikasi sentimen dalam berbagai bahasa. Setelah itu, tokenizer dan model BERT dimuat ke dalam memori.

Selanjutnya, kode membaca dataset dari file CSV (**preprocessed_comments_stemming.csv**) yang berisi komentar yang telah dibersihkan. Data yang mengandung nilai kosong dihapus untuk menghindari error saat pemrosesan. Kode juga memastikan bahwa kolom cleaned_comment tersedia dalam dataset, karena kolom ini yang akan dianalisis. Jika kolom tidak ditemukan, eksekusi dihentikan dengan pesan error.

Proses analisis sentimen dilakukan melalui fungsi predict_sentiment_bert(), yang menerima teks komentar sebagai input, lalu mengonversinya menjadi token menggunakan tokenizer BERT. Token ini diberikan ke model BERT untuk mendapatkan logits, yang merepresentasikan skor probabilitas dari setiap kelas sentimen. Hasil logits kemudian dinormalisasi menggunakan softmax untuk mendapatkan probabilitas dalam skala 1 hingga 5. Model ini secara asli menggunakan 5 label sentimen (0-4), sehingga dilakukan pemetaan ke dalam kategori Negatif (0-1), Netral (2), dan Positif (3-4) berdasarkan indeks skor tertinggi.

Setelah setiap komentar diklasifikasikan, hasil sentimen ditambahkan sebagai kolom baru (sentiment_bert) dalam dataset. Kemudian, dataset yang telah diberi label sentimen disimpan dalam file CSV baru (**sentiment_labeled_bert.csv**). Kode juga mencetak beberapa baris pertama dari hasil klasifikasi untuk memberikan gambaran kepada pengguna mengenai output yang dihasilkan.

4.5 Visualisasi Hasil

Untuk memahami distribusi sentimen dari komentar yang telah dianalisis, digunakan dua jenis visualisasi:

4.5.1 WordCloud

WordCloud digunakan untuk melihat kata-kata yang paling sering muncul dalam komentar. **Hasil WordCloud dari TextBlob dan BERT menunjukkan kemunculan kata-kata utama seperti "rakyat", "mahasiswa", dan "demo"** sebagai kata yang dominan dalam komentar. Berikut gambar WordCloud yang dihasilkan:

WordCloud TEXTBLOB

WordCloud BERT

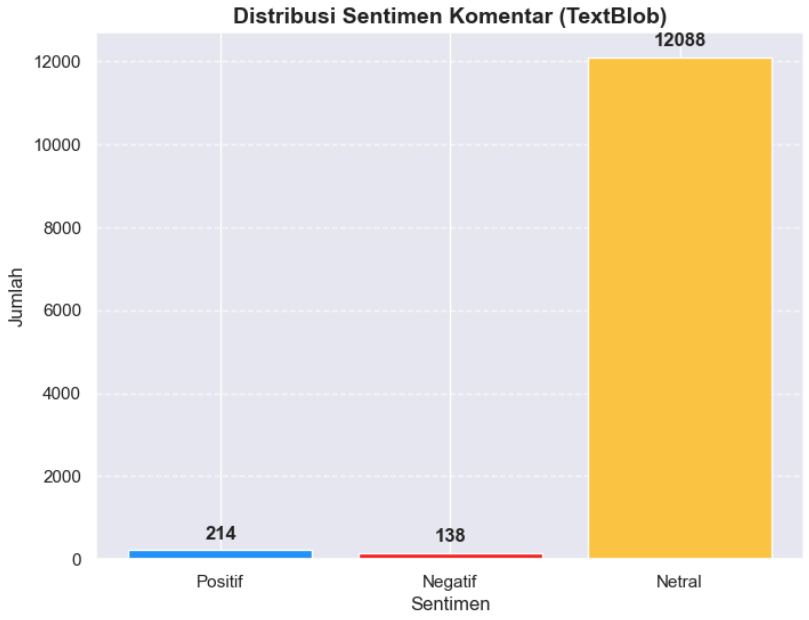
Dikarenakan tidak melalui proses translate maka katanya masih **berbahasa indonesia**.

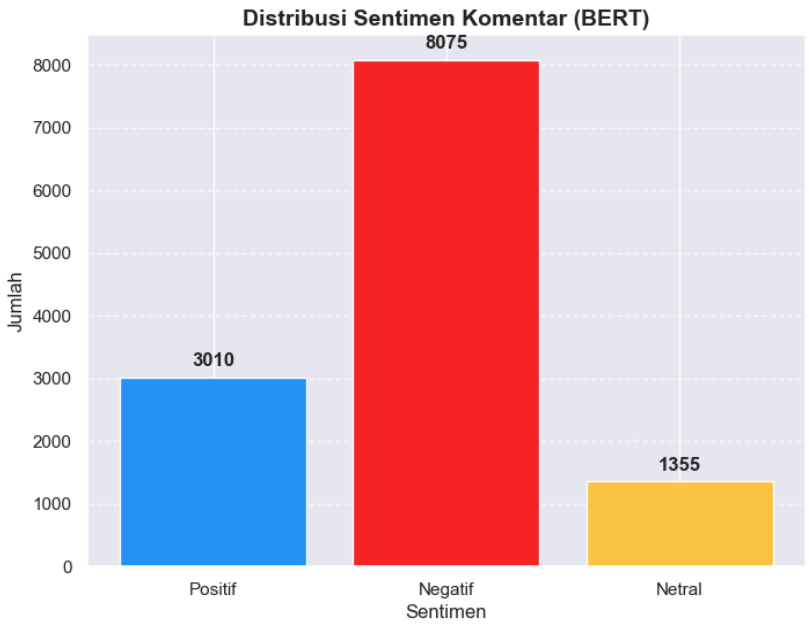
4.5.2 Distribusi Sentimen (Bar Chart)

- **TextBlob** menunjukkan distribusi yang sangat **jomplang**, dengan mayoritas komentar dikategorikan sebagai **Netral (12088 baris)**.

- **BERT** menunjukkan distribusi yang lebih **seimbang**, dengan dominasi sentimen **Negatif (8075)**, diikuti oleh Netral (1355) dan Positif (3010).

Berikut adalah gambar visualisasi yang dihasilkan dari proses labeling:





4.6 Source Code dan Dataset

Source Code :
https://drive.google.com/file/d/1IX3NX6ESDbSaUz6Lr5EVCqyR7_aI6d8T/view?usp=drive_link

Dataset :
https://drive.google.com/drive/folders/1rb6-VtHbsEVvDMfoJuExzbPQacj1oYXe?usp=drive_link

Github : NaufalAqil18/UTS-NLP-2025

5. Kesimpulan

Berdasarkan hasil analisis sentimen, kesimpulan yang dapat diambil diantaranya:

- **Keterbatasan waktu** dan **sumber daya** menyebabkan proses translasi **tidak bisa berjalan sesuai dengan semestinya,** dikarenakan adanya limit untuk traslate dan trial and error yang harus dijalani untuk mencapai hasil yang maksimal.
- **TextBlob** memiliki kecenderungan untuk mengklasifikasikan sebagian besar komentar sebagai **Netral**, yang menunjukkan keterbatasan metode berbasis leksikon dalam menangkap konteks bahasa yang lebih kompleks.
- **BERT** menunjukkan distribusi sentimen yang lebih variatif, dengan **sentimen Negatif sebagai kategori dominan**. Ini menunjukkan bahwa metode berbasis deep learning lebih efektif dalam memahami nuansa sentimen dalam komentar yang lebih panjang dan kompleks.
- **Keputusan untuk tidak menerjemahkan teks ke bahasa Inggris tidak menghambat analisis**, karena BERT masih mampu menangani bahasa Indonesia dengan cukup baik.

6. Daftar Pustaka

1. Illia, F., Eugenia, M. P., & Rutba, S. A. (2021). Sentiment Analysis on PeduliLindungi Application Using TextBlob and VADER Library. Politeknik Statistika STIS.
2. Liu, Z., Yang, S., & Ma, J. (2021). Movie Reviews Sentiment Analysis Using BERT. Proceedings of the International Conference on Artificial Intelligence and Data Science.
3. hugginaface.co
4. nlptown/bert-base-multilingual-uncased-sentiment
5. TextBlob: Simplified Text Processing — TextBlob 0.19.0 documentation