






# UTS BIG DATA

KELOMPOK 3





# ANGGOTA KELOMPOK



- IMAMULKHAIR
  - AKMAL BAHARUDDIN SYAM
  - AOURA ANSAR
  - SELVI SAHRENA
  - AKBAR
- 
- 



# LATAR BELAKANG



Konflik antara Gaza dan Israel sering menjadi sorotan dunia dan banyak dibahas di media sosial serta portal berita. Keyword "Gaza" dan "Israel" dipilih karena sering muncul dalam percakapan publik, mencerminkan opini dan emosi masyarakat.

Dengan melakukan scraping data teks dari internet, lalu membersihkannya, analisis dapat dilakukan untuk melihat tren, sentimen, dan pola pembahasan. Ini membantu memahami bagaimana isu tersebut dipersepsikan secara luas di dunia digital.







# RUMUSAN MASALAH

1. Bagaimana proses pengambilan data teks (scraping) dengan keyword "Gaza" dan "Israel" dilakukan?.
  2. Bagaimana tahapan pembersihan (cleaning) data teks yang diperoleh?
  3. Apa saja hasil analisis yang dapat diperoleh dari data teks terkait "Gaza" dan "Israel"?
  4. Bagaimana persepsi publik terhadap konflik Gaza-Israel berdasarkan data yang dianalisis?
- 
- 



# PROSES PENGAMBILAN DATA TEXT

- Proses instalasi
  - Proses pengambilan data
  - Proses Pandas membaca data
- 
- 





```
# Import required Python package  
!pip install pandas
```

menginstal pandas digunakan untuk analisis dan manipulasi data ( !pip install pandas )

```
# Install Node.js (because tweet-harvest built using Node.js)  
!sudo apt-get update  
!sudo apt-get install -y ca-certificates curl gnupg  
!sudo mkdir -p /etc/apt/keyrings  
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg
```



menginstal node.js digunakan untuk scroll halaman tweet secara otomatis dan mengambil data dalam kurun waktu tertentu



```
# Crawl Data

filename = 'Gaza.csv'
search_keyword = 'Gaza lang:id'
limit = 200

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}
```



```
# Crawl Data



filename = 'Gaza.csv'
search_keyword = 'Gaza lang:id'
limit = 200

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}
```



kode ini mengambil 200 tweet terbaru berbahasa Indonesia yang membahas tentang Gaza dan Israel, lalu menyimpannya dalam file csv untuk dianalisis lebih lanjut.





# ALUR DARI PROSES CLEANING DATA


- Proses pandas membaca data
  - Proses load
  - Tahapan cleaning data
  - Membuat variable baru dan perbandingan sebelum dan sesudah cleaning
- 
- 





Selanjutnya tahap pengolahan data setelah pengumpulan tweet, di mana data dari file CSV dimasukkan ke dalam struktur data pandas untuk memudahkan analisis lebih lanjut.



```
 import pandas as pd

# Specify the path to your CSV file
file_path = f"tweets-data/{filename}"

# Read the CSV file into a pandas DataFrame
df = pd.read_csv(file_path, delimiter=",")


# Display the DataFrame
display(df)
```

```
import pandas as pd  
  
Gaza = pd.read_csv('Gaza.csv')  
print(Gaza.head())
```

Menggunakan library pandas untuk membaca data dari file CSV bernama "Gaza.csv" dan menyimpannya ke dalam variabel Gaza dalam bentuk DataFrame. Kemudian, kode mencetak lima baris pertama dari data tersebut menggunakan Gaza.head() agar pengguna bisa melihat cuplikan awal isi dan struktur tabel secara ringkas.

```
Gaza.info()
```

Menampilkan informasi ringkas tentang DataFrame Gaza, seperti jumlah baris, jumlah kolom, nama kolom, tipe data setiap kolom, jumlah data non-null (tidak kosong), dan penggunaan memori. Ini berguna untuk memahami struktur data dan mengecek apakah ada data yang hilang (missing values).



```
Gaza["full_text"]
```

Menampilkan isi kolom full\_text yang berisi  
full text mentah yang dari tweet

```
#membuat function cleaning

import re
def clean(s):
    s = s.replace(r'<lb>', "\n")
    s = s.replace(r'<tab>', "\t")
    s = re.sub(r'<br */*>', "\n", s)
    s = s.replace("&lt;", "<").replace("&gt;", ">").replace("&#", "&")
    s = s.replace("&#", "&")
    # markdown urls
    s = re.sub(r'\(https*://[^\)]*\)', "", s)
    # normal urls
    s = re.sub(r'https*://[^\s]*', "", s)
    s = re.sub(r'_+', ' ', s)
    s = re.sub(r'"'+', '""', s)
    return str(s)
```

Membuat function cleaning untuk membersihkan  
dari symbol dan link



```
Gaza["text_clean"] = ''
```

membuat variable baru untuk menyimpan hasilnya



```
Gaza[["full_text", "text_clean"]].head()
```

	full_text	text_clean
0	@Sujudzz @mohzen51 @KompasTV iki opo cuk? wkwk...	@Sujudzz @mohzen51 @KompasTV iki opo cuk? wkwk...
1	IYALAH kan mereka perang demi mempertahankan t...	IYALAH kan mereka perang demi mempertahankan t...
2	Perang Gaza bukan baru mula 7 oktober dah berl...	Perang Gaza bukan baru mula 7 oktober dah berl...
3	@penaikhwan Rachel Corrie adalah seorang aktiv...	@penaikhwan Rachel Corrie adalah seorang aktiv...
4	Pas udah pulih terus pas balik ke Gaza lagi ga...	Pas udah pulih terus pas balik ke Gaza lagi ga...

Membandingkan data sesudah dan sebelum di clean yang  
Dimana teks sebelum berisi tag dan link



# ALUR DARI ANALISIS DATA TEKS

- Load data dan tokenisasi
  - Stopword word indonesia
  - Stemming
- 
- 

```
# Tokenisasi kata pada kolom 'text_clean'
Gaza['tokenized_text'] = Gaza['text_clean'].apply(lambda x: word_tokenize(x))

# Menampilkan beberapa baris pertama data dengan kolom tokenized_text
print(Gaza[['text_clean', 'tokenized_text']].head())
```

	text_clean \	tokenized_text
0	@Sujudzz @mohzen51 @KompasTV iki opo cuk? wkwk...	[@, Sujudzz, @, mohzen51, @, KompasTV, iki, op...
1	IYALAH kan mereka perang demi mempertahankan t...	[IYALAH, kan, mereka, perang, demi, mempertahankan...
2	Perang Gaza bukan baru mula 7 oktober dah berl...	[Perang, Gaza, bukan, baru, mula, 7, oktober, ...
3	@penaikhwan Rachel Corrie adalah seorang aktiv...	[@, penaikhwan, Rachel, Corrie, adalah, seoran...
4	Pas udah pulih terus pas balik ke Gaza lagi ga...	[Pas, udah, pulih, terus, pas, balik, ke, Gaza...

Melakukan tokenisasi (memecah teks menjadi kata-kata) pada kolom 'text\_clean' di DataFrame Gaza. Proses ini dilakukan dengan fungsi `word_tokenize`, yang diterapkan ke setiap baris menggunakan `.apply()`.

Hasil tokenisasi disimpan di kolom baru bernama 'tokenized\_text'. Kemudian, kode menampilkan beberapa baris pertama dari kolom 'text\_clean' dan 'tokenized\_text' untuk melihat hasil pemecahan teks menjadi kata-kata.



```
stop_words_indonesia = stopwords.words('indonesian')

def remove_stopwords(token_list):
    return [word for word in token_list if word.lower() not in stop_words_indonesia]

Gaza['tokenized_stopwords'] = Gaza['tokenized_text'].apply(remove_stopwords)

Gaza[['tokenized_text', 'tokenized_stopwords']].head()
```

	tokenized_text	tokenized_stopwords
0	[@, Sujudzz, @, mohzen51, @, KompasTV, iki, op...	[@, Sujudzz, @, mohzen51, @, KompasTV, iki, op...
1	[YALAH, kan, mereka, perang, demi, mempertahankan...	[YALAH, perang, mempertahankan, tanah, airnya...
2	[Perang, Gaza, bukan, baru, mula, 7, oktober, ...	[Perang, Gaza, 7, oktober, dah, berlaku, tu, s...
3	[@, penaikhwan, Rachel, Corrie, adalah, seoran...	[@, penaikhwan, Rachel, Corrie, aktivis, pro-P...
4	[Pas, udah, pulih, terus, pas, balik, ke, Gaza...	[Pas, udah, pulih, pas, Gaza, gaza, nya, Uda, ...

Mengambil stopwords Indonesia, lalu menerapkan fungsi "remove" stopwords ke data yang telah di tokenisasi sebelumnya untuk menampilkan data hasilnya.

```
!pip install Sastrawi
```

Menginstal paket yang dibutuhkan

```

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

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

def stemming_sastrawi(token_list):
    return [stemmer.stem(word) for word in token_list]

Gaza['stemmed_words'] = Gaza['tokenized_stopwords'].apply(stemming_sastrawi)

Gaza[['tokenized_stopwords', 'stemmed_words']]

```



	tokenized_stopwords	stemmed_words
0	[@, Sujudzz, @, mohzen51, @, KompasTV, iki, op...	[, sujudzz, , mohzen51, , kompastv, iki, opo, ...
1	[IYALAH, perang, mempertahankan, tanah, airnya...	[iya, perang, tahan, tanah, air, wilayah, gaza...
2	[Perang, Gaza, 7, oktober, dah, berlaku, tu, s...	[perang, gaza, 7, oktober, dah, laku, tu, sinc...
3	[@, penaikhwan, Rachel, Corrie, aktivis, pro-P...	[, penaikhwan, rachel, corrie, aktivis, pro-pa...
4	[Pas, udah, pulih, pas, Gaza, gaza, nya, Uda, ...	[pas, udah, pulih, pas, gaza, gaza, nya, uda, ...
...	...	...
208	[@, tempodotco, Mala, memudahkan, zionist, cla...	[, tempodotco, mala, mudah, zionist, claim, gaza]
209	[@, nu, online, @, ulil, Senada, kebijakan, Ch...	[, nu, online, , ulil, nada, bijak, china, gaz...
210	[Aktivis, Amerika, Chicago, memegang, anak, ya...	[aktivis, amerika, chicago, pegang, anak, yati...
211	[@, TirtolD, Jualan, gaza, aja, ..., -, el, st...	[, tirtoid, jual, gaza, aja, , -, el, stafsus,...
212	[PROSES, PEMAKAMAN, EYANG, TITIEK, PUSPA, TPU,...	[proses, makam, eyang, titiek, puspa, tpu, tan...

Mengimport paket yang telah diinstal dan membuat objek stemmer untuk membentuk fungsi steeming yang kemudian nantinya akan diterapkan pada data yang telah di stopwords sebelumnya lalu menampilkan hasilnya.





# HASIL ANALISIS DATA TEKS

- Sentiment score
  - Membuat grafik plot
  - Membuat word cloud
- 
- 

```
# Menampilkan beberapa baris pertama data dengan kolom sentiment_score
print(Gaza[['stemmed_words', 'sentiment_score']].head())
```

Kode diatas menampilkan beberapa baris pertama dari kolom `stemmed_words` dan `sentiment_score`. Tujuannya untuk melihat hasil tokenisasi kata dan skor sentimen yang sudah dihitung. Fungsinya sebagai cuplikan hasil analisis sentimen di data.

```
# Menampilkan beberapa baris pertama data dengan kolom sentiment_score
print(Gaza[['stemmed_words', 'sentiment_score']].head())
```

	stemmed_words	sentiment_score
0	[, sujudzz, , mohzen51, , kompastv, iki, opo, ...	-0.2500
1	[iya, perang, tahan, tanah, air, wilayah, gaza...	0.0000
2	[perang, gaza, 7, oktober, dah, laku, tu, sinc...	0.6369
3	[, penaikhwan, rachel, corrie, aktivis, pro-pa...	0.0000
4	[pas, udah, pulih, pas, gaza, gaza, nya, uda, ...	0.0000

Menampilkan 5 baris pertama dari dua kolom di DataFrame Gaza, yaitu `stemmed_words` (hasil tokenisasi dan stemming kata) dan `sentiment_score` (nilai sentimen dari teks tersebut). Tujuannya agar pengguna bisa melihat hasil sementara dari proses tokenisasi dan analisis sentimen yang sudah dilakukan.

```
# Membuat label setiap komentar

# Fungsi untuk menentukan label sentimen berdasarkan sentiment_score
def label_sentiment(score):
    if score < 0:
        return 'negatif'
    elif score == 0:
        return 'netral'
    else:
        return 'positif'

# Buat kolom baru 'sentiment_label' berdasarkan kolom 'sentiment_score'
Gaza['sentiment_label'] = Gaza['sentiment_score'].apply(label_sentiment)

# Menampilkan beberapa baris pertama data dengan kolom sentiment_label
print(Gaza[['stemmed_words', 'sentiment_score', 'sentiment_label']].head())
```

	stemmed_words	sentiment_score \
0	[, sujudzz, , mohzen51, , kompastv, iki, opo, ...	-0.2500
1	[iya, perang, tahan, tanah, air, wilayah, gaza...	0.0000
2	[perang, gaza, 7, oktober, dah, laku, tu, sinc...	0.6369
3	[, penaikhwan, rachel, corrie, aktivis, pro-pa...	0.0000
4	[pas, udah, pulih, pas, gaza, gaza, nya, uda, ...	0.0000

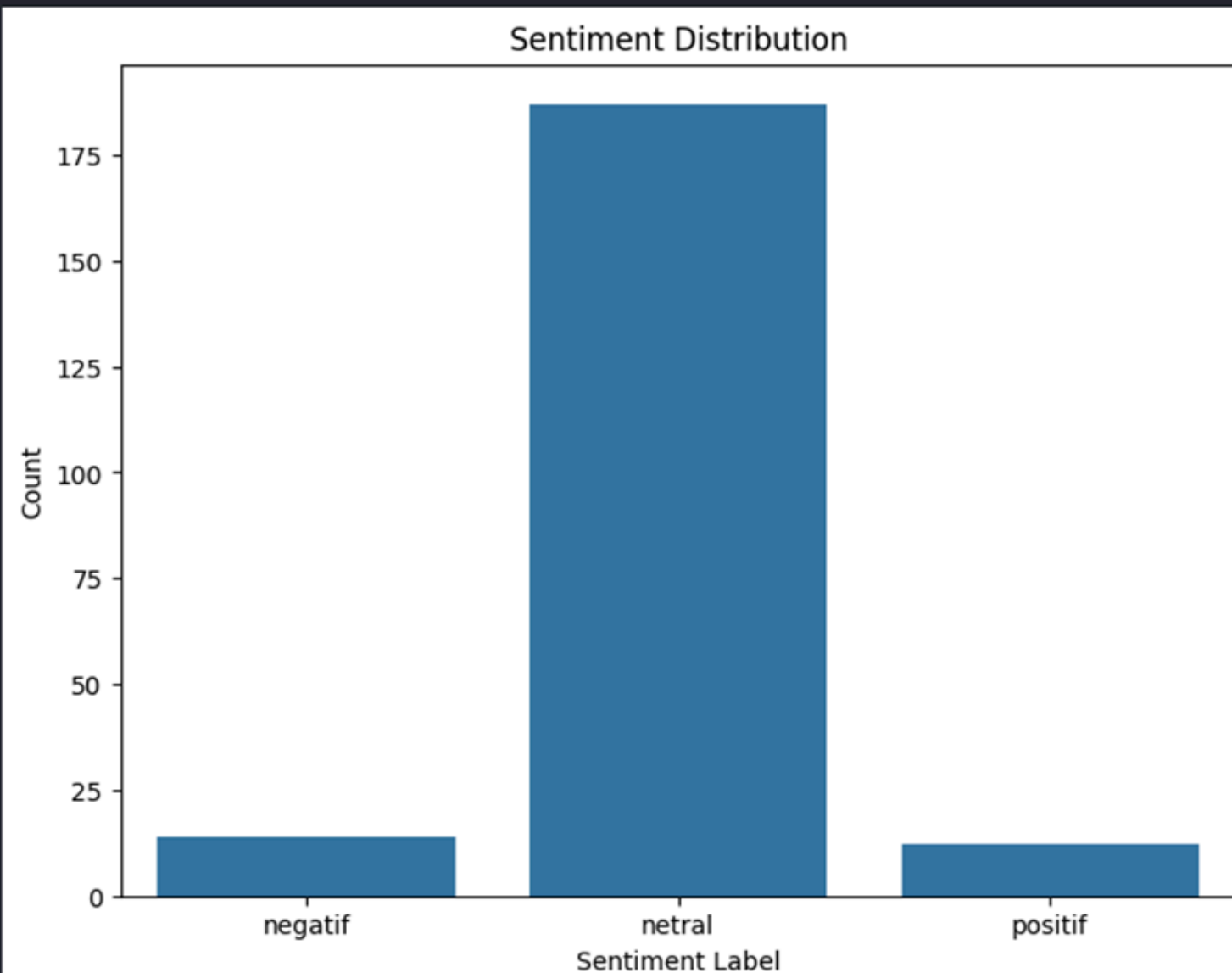
  

	sentiment_label
0	negatif
1	netral
2	positif
3	netral
4	netral

Membuat label sentimen berdasarkan nilai sentiment\_score. Fungsi label\_sentiment menentukan kategori 'negatif', 'netral', atau 'positif' sesuai skornya. Hasil label disimpan di kolom sentiment\_label, lalu ditampilkan bersama beberapa data awal.

```
#Membuat grafik plot
import matplotlib.pyplot as plt
import seaborn as sns

# Create the bar chart
plt.figure(figsize=(8, 6))
sns.countplot(x='sentiment_label', data=Gaza)
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment Label')
plt.ylabel('Count')
plt.show()
```



Membuat grafik batang untuk menampilkan jumlah data pada tiap kategori sentimen.

Menggunakan matplotlib dan seaborn, grafik menunjukkan distribusi label 'positif', 'netral', dan 'negatif'. Tujuannya untuk memvisualisasikan sebaran sentimen dalam data Gaza.



```
# Group data by sentiment label
sentiment_groups = Gaza.groupby('sentiment_label')

# Create word clouds for each sentiment category
for sentiment, group in sentiment_groups:
    # Combine all stemmed words in the group
    all_words = ' '.join([' '.join(words) for words in group['stemmed_words']])

    # Generate the word cloud
    wordcloud = WordCloud(width=800, height=400, background_color='white').generate(all_words)

    # Display the generated image:
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.title(f'Word Cloud for {sentiment} Sentiment')
    plt.show()
```

Data dikelompokkan berdasarkan `sentiment\_label`, lalu semua kata dalam tiap grup digabung. Hasilnya divisualisasikan jadi Word Cloud per kategori sentimen menggunakan `WordCloud` dan ditampilkan dengan `matplotlib`.



**LINK .....**

**TERIMA KASIH**

