Analisis Sentimen Komen Instagram

"Kenapa Elektabilitas Anies Tak Kunjung Naik?"

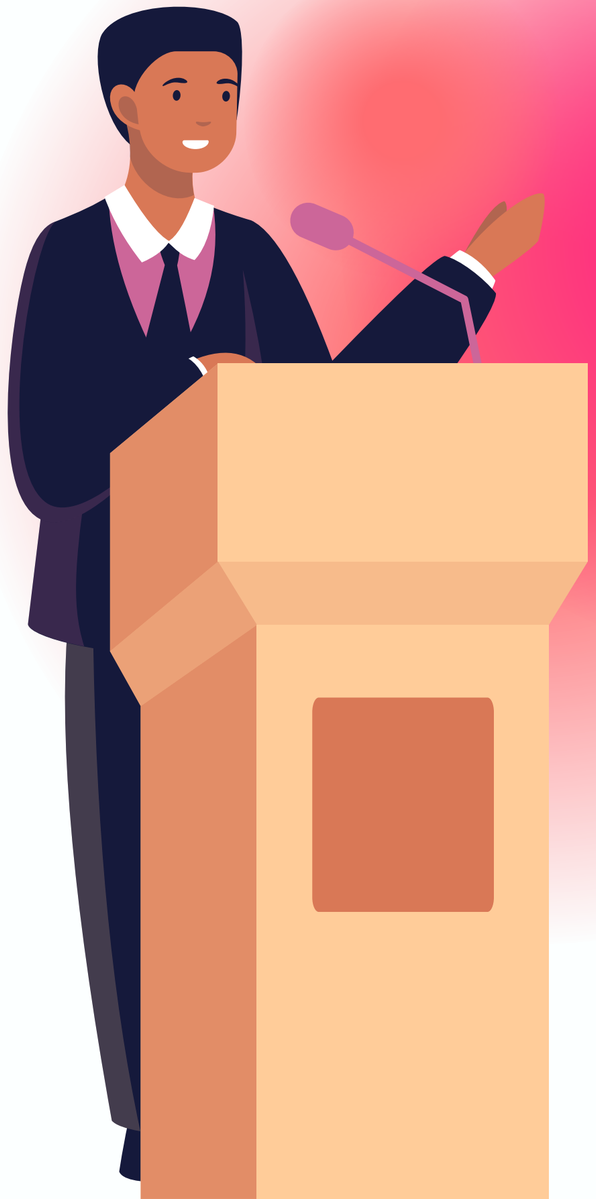
KELOMPOK 3

- ANDREW LAUWIRA - 2440035904
- BELVA FITHRIYAH - 2440084231
- FARRAH A MAHARANI - 2440109732
- SHAREN IVANA - 2440086155

IDE

P O L I T I K

DI DUNIA POLITIK BEGITU BANYAK PRO KONTRA AKAN SUATU HAL, TERMASUK SUATU PERNYATAAN. PENTINGNYA UNTUK KITA MENCOBA UNTUK MENGANALISA TENTANG PERNYATAAN-PERNYATAAN TERSEBUT UNTUK DAPAT MENGETAHUI SISI POSITIF DAN NEGATIFNYA. OLEH KARENA ITU, KITA MELAKUKAN ANALISIS SENTIMEN UNTUK DAPAT MEMBEDAKAN PERNYATAAN POSITIF DAN NEGATIF DARI KOMEN DI INSTAGRAM.



INSTAGRAM @ASUMSICO

POLITICS

SCRAPING KOMEN INSTAGRAM KE CSV

```
!pip install instaloader
import instaloader
import random
import time
import csv
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: instaloader in /usr/local/lib/python3.10/dist-packages (4.9.6)
Requirement already satisfied: requests>=2.4 in /usr/local/lib/python3.10/dist-packages (from instaloader) (2.27.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.4->instaloader) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.4->instaloader) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests>=2.4->instaloader) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.4->instaloader) (3.4)
```

IMPORTING NECESSARY LIBRARIES

- **Instaloader: instaloader adalah library Python yang digunakan untuk mendownload dan berinteraksi dengan data-data Instagram.**
- **Random: Library random adalah library bawaan Python yang menyediakan fungsi untuk mengenerate angka acak.**
- **Time: Modul time adalah modul bawaan Python yang menyediakan fungsi untuk bekerja dengan waktu.**
- **CSV: Modul csv adalah modul bawaan Python yang digunakan untuk membaca dan menulis file CSV (Comma-Separated Values).**

```python
def login_to_instagram(username, password):
    loader = instaloader.Instaloader()

    try:
        loader.login(username, password)
        print("Login successful!")
        return loader
    except instaloader.exceptions.BadCredentialsException:
        print("Invalid login credentials!")
        return None
```

DEFINING FUNCTION

Fungsi login_to_instagram menerima dua parameter, yaitu username dan password, dan bertugas untuk melakukan proses login ke Instagram menggunakan library instaloader.

```python
def scrape_instagram_post(loader, shortcode):
    if loader is None:
        loader.login(username, password)
        print("Login successful!")
        return loader

    with open('SEresult.csv', 'a+', newline='', encoding='utf-8') as file:
      writer = csv.writer(file)
      try:
          post = instaloader.Post.from_shortcode(loader.context, shortcode)

          # Access the post data
          print("Post ID:", post.mediaid)
          time.sleep(random.uniform(2, 3))
          postid = post.mediaid
          print("Caption:", post.caption)
          time.sleep(random.uniform(2, 3))
          caption = post.caption
          print("Likes:", post.likes)
          time.sleep(random.uniform(2, 3))
          likes = post.likes
          writer.writerow(["post.ID", "Caption", "Likes"])
          writer.writerow([postid, caption, likes])

          # Scrape and print the comments
          print("Comments:")
          writer.writerow(["Username", "Comment"])
          for comment in post.get_comments():
              print("Username:", comment.owner.username)
              time.sleep(random.uniform(2, 3))
              username=comment.owner.username
              print("Comment:", comment.text)
              time.sleep(random.uniform(2, 3))
              comment_user=comment.text
              print("---")
              writer.writerow([username, comment_user])
      except instaloader.exceptions.InstaloaderException as e:
          print("Error occurred while scraping the post:", str(e))
```

DEFINING FUNCTION

Fungsi scrape_instagram_post

menerima dua parameter, yaitu loader dan shortcode, dan bertanggung jawab untuk melakukan pengambilan data dari sebuah postingan Instagram menggunakan instaloader.

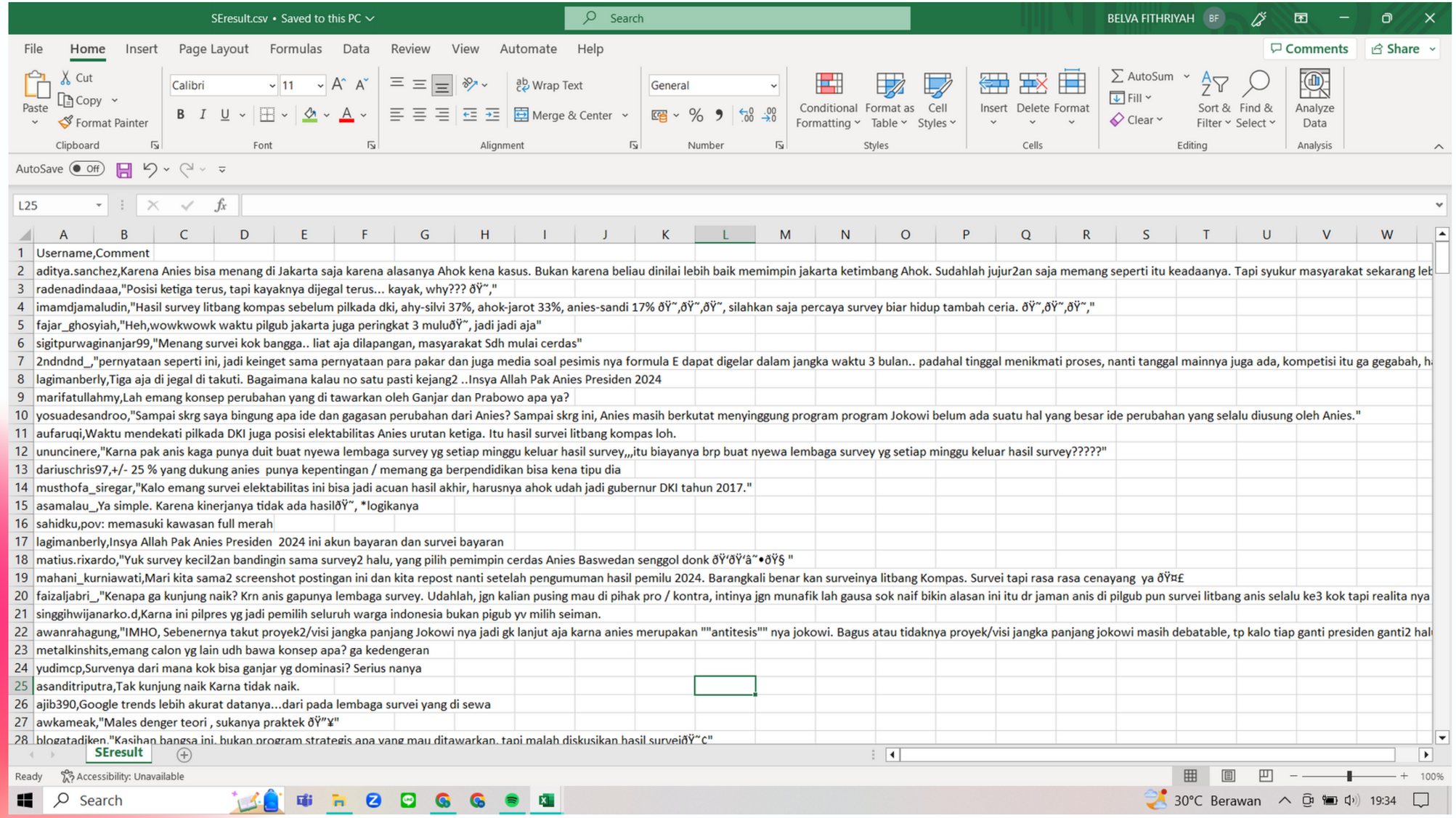
```
shortcode = 'CsqkhP3vXz6'

username = 'sharenbaubabi'
password = 'ranicantiksharenbabi'

loader = login_to_instagram(username, password)
time.sleep(random.uniform(2, 3))  # Sleep for a random duration between 2 and 3 seconds
scrape_instagram_post(loader, shortcode)
```

DEFINING FUNCTION

- **shortcode = 'CsqkhP3vXz6': Baris ini mengassign shortcode sebenarnya dari postingan Instagram ke dalam variabel shortcode.**
- **username & password for login credentials: Username dan pass akun ig**
- **time.sleep(random.uniform(2, 3)): Baris ini menggunakan fungsi time.sleep() dari modul time untuk memasukkan jeda acak antara 2 hingga 3 detik sebelum menjalankan instruksi berikutnya. Hal ini berguna untuk menambahkan penundaan acak agar program terlihat lebih realistis dan menghindari terlalu cepatnya permintaan ke server Instagram.**

HASIL SCRAPING:

SEresult.csv



Analisis Sentimen

POSITIF / NEGATIF

DATA PREPROCESSING

1. Import Library

```
1 import nltk
2 import pandas as pd
3 import numpy as np
4 import string
5 import re
```

2. Case Folding

```
[5]  1 df = df.iloc[5:]
     2 df = df.drop(['post.ID','Likes'], axis =1)
     3 df.head(10)
```

```
1 print('Case Folding Result : \n')
2 print(df['Caption'].head(5))
3 print('\n\n\n')
```

```
Case Folding Result :

5    Karena Anies bisa menang di Jakarta saja karen...
6    Posisi ketiga terus, tapi kayaknya dijegal ter...
7    Hasil survey litbang kompas sebelum pilkada dk...
8    Heh,wowkwowk waktu pilgub jakarta juga peringk...
9    Menang survei kok bangga.. liat aja dilapangan...
Name: Caption, dtype: object
```

3. Tokenizing

```
# import word_tokenize & FreqDist from NLTK
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist

# ------ Tokenizing ---------

def remove_special(text):
    text = str(text)
    # remove tab, new line, ans back slice
    text = text.replace('\\t'," ").replace('\\n'," ").replace('\\u'," ").replace('\\',"")
    # remove non ASCII (emoticon, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    text = text.lower()
    # remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", text).split())
    # remove incomplete URL
    return text.replace("http://", " ").replace("https://", " ")

df['Caption'] = df['Caption'].apply(remove_special)
```

NLTK Tokenizing

```
# NLTK word rokenize
def word_tokenize_wrapper(text):
    return word_tokenize(text)

df['tokenized'] = df['Caption'].apply(word_tokenize_wrapper)

print('Tokenizing Result : \n')
print(df['tokenized'].head())
print('\n\n\n')
```

```
kenizing Result :

  [karena, anies, bisa, menang, di, jakarta, saj...
  [posisi, ketiga, terus, tapi, kayaknya, dijega...
  [hasil, survey, litbang, kompas, sebelum, pilk...
  [hehwowkwowk, waktu, pilgub, jakarta, juga, pe...
  [menang, survei, kok, bangga, liat, aja, dilap...
me: tokenized, dtype: object
```

4. Filtering (removing stopwords)

```
from nltk.corpus import stopwords

#stopword indonesia
list_stopwords = stopwords.words('indonesian')


# menggabungkan additional stopword
list_stopwords.extend(["yg", "dg", "rt", "dgn", "ny", "d", 'klo',
                       'kalo', 'amp', 'biar', 'bikin', 'bilang',
                       'gak', 'ga', 'krn', 'nya', 'nih', 'sih',
                       'si', 'tau', 'tdk', 'tuh', 'utk', 'ya',
                       'jd', 'jgn', 'sdh', 'aja', 'n', 't',
                       'nyg', 'hehe', 'pen', 'u', 'nan', 'loh', 'rt',
                       '&amp', 'yah','awokawok','b aja','yang','jg'])

```

```python
# read txt stopword using pandas
txt_stopword = pd.read_csv("stopwords.txt", names= ["stopwords"], header = None)

# convert stopword string to list & append additional stopword
list_stopwords.extend(txt_stopword["stopwords"][0].split(' '))


# memasukan list stopwords ke dalam corpus
list_stopwords = set(list_stopwords)


#remove stopword pada list token
def stopwords_removal(words):
    return [word for word in words if word not in list_stopwords]

df['tokens_WSW'] = df['tokenized'].apply(stopwords_removal)


print(df['tokens_WSW'].head())
```

```
5    [anies, menang, jakarta, alasanya, ahok, kena,...
6      [posisi, ketiga, kayaknya, dijegal, kayak, why]
7    [hasil, survey, litbang, kompas, pilkada, dki,...
8      [hehwowkwowk, pilgub, jakarta, peringkat, mulu]
9    [menang, survei, bangga, liat, dilapangan, mas...
Name: tokens_WSW, dtype: object
```

5. Normalization

```
normalizad_word = pd.read_excel("normalisasi.xlsx")

normalizad_word_dict = {}

for index, row in normalizad_word.iterrows():
    if row[0] not in normalizad_word_dict:
        normalizad_word_dict[row[0]] = row[1]

def normalized_term(document):
    return [normalizad_word_dict[term] if term in normalizad_word_dict else term for term in document]

df['normalized'] = df['tokens_WSW'].apply(normalized_term)

df['normalized'].head(10)
```

6. Stemming

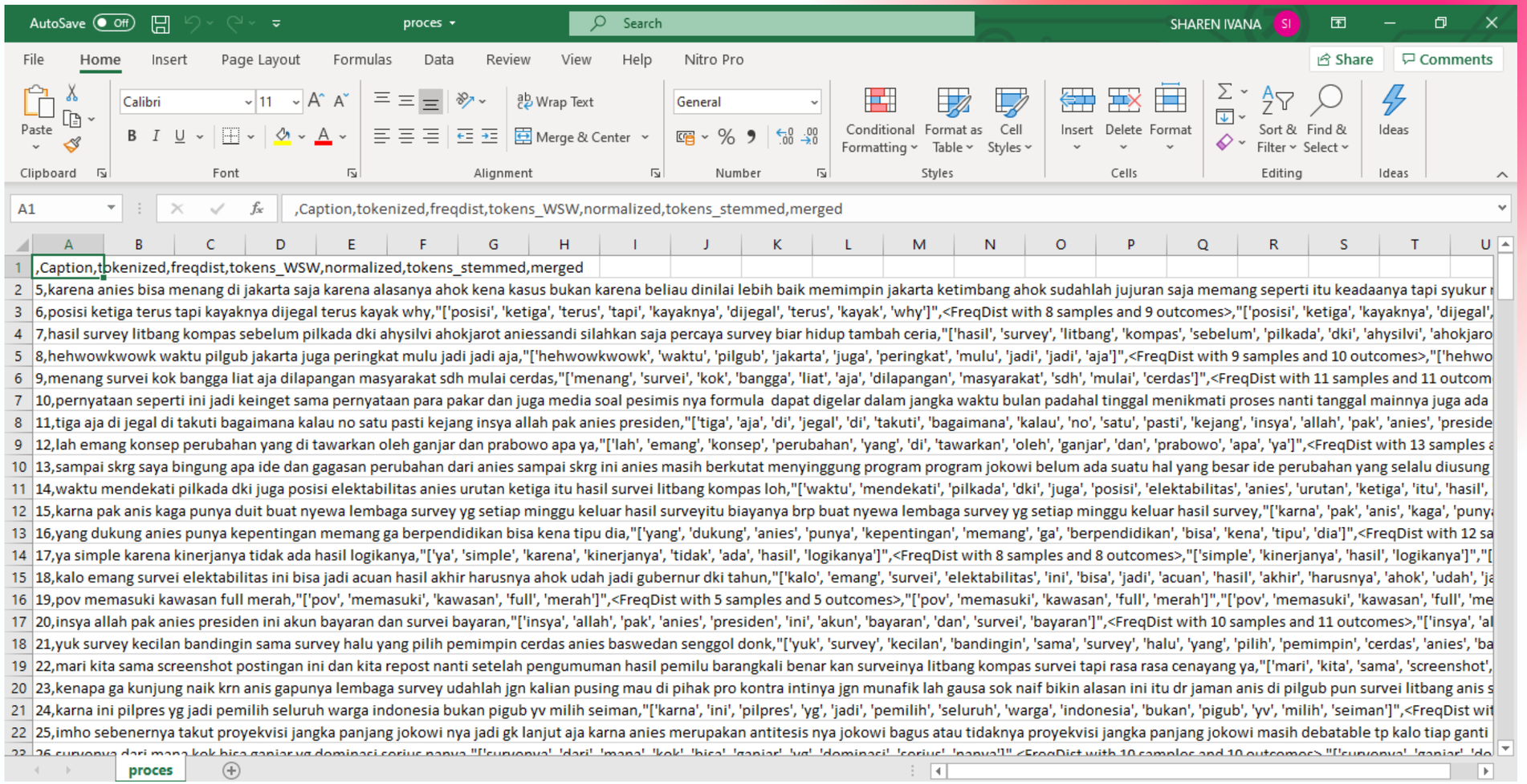
```python
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import swifter


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

# stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in df['normalized']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '
```

```python
print(len(term_dict))

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":" ,term_dict[term])

print(term_dict)
print("===")


# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

df['tokens_stemmed'] = df['normalized'].swifter.apply(get_stemmed_term)
print(df['tokens_stemmed'])
```

```
program : program
terealisasikan : realisasi
datanya : data
posisi : posisi
ketiga : tiga
kayaknya : kayak
dijegal : jegal
```

HASIL SETELAH DILAKUKAN PREPROCESSING

proces



MEMBUAT MODEL BUAT TRAIN TEST

https://github.com/devildances/twittersentimentanalysis_final_project/blob/master/data/indonesia_tweet/clean_tweets.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Tweet,label																
2	cowok usaha lacak perhati gue lantas remeh perhati gue kasih khusus basic cowok bego,negative																
3	rt telat tau edan sarap gue gaul cigax jifla cal licew,negative																
4	kadang pikir percaya tuhan jatuh kali kali kadang tuhan tinggal orang tua rencana pisah kakak pilih kristen anak ter,positive																
5	tau mata sipit lihat,positive																
6	kaum cebong kafir lihat dongok dungu haha,negative																
7	ya bani taplak kawan kawan,negative																
8	deklarasi pilih kepala daerah aman anti hoaks warga dukuh sari jabon,positive																
9	gue selesai re watch aldnoah zero kampret karakter utama cowok kena friendzone bro xd,negative																
10	admin belanja port nak makan ais kepal milo ais kepal horlicks cendol toping kau kau doket gerai rozak mertua taipan kembar baby amp romantika bank islam senawang,positive																
11	enak ngewe,negative																
12	gue jari gue ukur nyali bacot,negative																
13	banci kaleng malu nyungsep koe,negative																
14	ajar ekonomi mesti jago privatisasi hati orang aduh ironi,positive																
15	aktor huru hara prabowo si lengser perintah jokowi nyata,negative																
16	bu guru enak jablay guru sekolah dasar sih kayak nikmat pecun guru,negative																
17	lawan bicara gue intelek kayak otak kencing onta gue hadis nabi sahih kafir kayak tolak ahlun nar,negative																
18	pikir ampas banget ya,positive																
19	ari bek rapi monyet,negative																
20	cowok gantle gantle nama banci,negative																

METODE BERT

Bidirectional Encoder Representations from Transformers

BERT adalah model bahasa yang dikembangkan oleh Google pada tahun 2018. Model ini menggunakan arsitektur Transformer yang merupakan jaringan saraf rekurensi (RNN) yang kuat dalam pemrosesan data berurutan.

BERT secara khusus dirancang untuk tugas pemahaman bahasa alami (natural language understanding, NLU) seperti pemrosesan teks, analisis sentimen, pemisahan kalimat, dan tugas-tugas lainnya. Keunikan utama BERT terletak pada kemampuannya untuk memahami konteks kata dalam kalimat dengan mempertimbangkan kata-kata yang ada sebelum dan sesudahnya, yang dikenal sebagai pemrosesan arah ganda (bidirectional processing).

SENTIMENT ANALYSIS MENGGUNAKAN METODE BERT

a. import library

Import necessary library

```
[1]  !pip install transformers
     !pip install Sastrawi

     import pandas as pd
     import numpy as np
     import torch
     from transformers import BertTokenizer, BertForSequenceClassification, AdamW, get_linear_schedule_with_warmup
     from sklearn.model_selection import train_test_split
     from torch.utils.data import TensorDataset, DataLoader, RandomSampler, SequentialSampler
     from sklearn.metrics import f1_score
     from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
```

31s

b. Tokenize dengan BERT

Tokenize data using BERT tokenizer

[3] 9s

```
tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased', do_lower_case=True)

# Encode train and test data using BERT tokenizer
train_encodings = tokenizer(train_text.tolist(), truncation=True, padding=True)
test_encodings = tokenizer(test_text.tolist(), truncation=True, padding=True)

# Convert encoded inputs to PyTorch tensors
train_inputs = torch.tensor(train_encodings['input_ids'])
train_masks = torch.tensor(train_encodings['attention_mask'])
train_labels = torch.tensor(train_labels)
test_inputs = torch.tensor(test_encodings['input_ids'])
test_masks = torch.tensor(test_encodings['attention_mask'])
test_labels = torch.tensor(test_labels)

# Create data loaders for efficient batching
batch_size = 16
train_data = TensorDataset(train_inputs, train_masks, train_labels)
train_sampler = RandomSampler(train_data)
train_dataloader = DataLoader(train_data, sampler=train_sampler, batch_size=batch_size)
test_data = TensorDataset(test_inputs, test_masks, test_labels)
test_sampler = SequentialSampler(test_data)
test_dataloader = DataLoader(test_data, sampler=test_sampler, batch_size=batch_size)
```

```
Downloading (…)solve/main/vocab.txt: 100% 996k/996k [00:00<00:00, 8.46MB/s]
Downloading (…)okenizer_config.json: 100% 29.0/29.0 [00:00<00:00, 1.72kB/s]
Downloading (…)lve/main/config.json: 100% 625/625 [00:00<00:00, 15.1kB/s]
```

c. Train the model

```
[4]  model = BertForSequenceClassification.from_pretrained('bert-base-multilingual-cased', num_labels=2, output_attentions=False, output_hidden_states=False)

     # Set device to GPU if available
     device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
     model.to(device)

     # Set up optimizer and scheduler
     epochs = 5
     optimizer = AdamW(model.parameters(), lr=2e-5, eps=1e-8)
     total_steps = len(train_dataloader) * epochs
     scheduler = get_linear_schedule_with_warmup(optimizer, num_warmup_steps=0, num_training_steps=total_steps)

     from tqdm import tqdm

     # Train the model
     for epoch in range(epochs):
         model.train()
         total_loss = 0
         progress_bar = tqdm(train_dataloader, desc=f'Epoch {epoch+1}', leave=False)
         for step, batch in enumerate(progress_bar):
             batch_inputs = batch[0].to(device)
             batch_masks = batch[1].to(device)
             batch_labels = batch[2].to(device)
             optimizer.zero_grad()
             outputs = model(batch_inputs, attention_mask=batch_masks, labels=batch_labels)
             loss = outputs[0]
             total_loss += loss.item()
             loss.backward()
             torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
             optimizer.step()
             scheduler.step()
             avg_train_loss = total_loss / (step+1)
             progress_bar.set_postfix({'training_loss': avg_train_loss})
         print(f'Epoch: {epoch + 1}, Training Loss: {avg_train_loss}')
```

Save the trained model

```
[5]  model_path = 'bert_sa_sentiment_model.pt'
     torch.save(model.state_dict(), model_path)
```

d. Evaluasi model

Evaluate the model

```
[6]  # Load the saved model
     model = BertForSequenceClassification.from_pretrained('bert-base-multilingual-cased', num_labels=2, output_attentions=False, output_hidden_states=False)
     model.load_state_dict(torch.load(model_path))
     model.to(device)

     # Evaluate the model on test data
     model.eval()
     total_preds = []
     total_labels = []
     with torch.no_grad():
         for batch in test_dataloader:
             batch_inputs = batch[0].to(device)
             batch_masks = batch[1].to(device)
             batch_labels = batch[2].to('cpu')
             outputs = model(batch_inputs, attention_mask=batch_masks)
             logits = outputs[0]
             preds = torch.argmax(logits, axis=1)
             total_preds.extend(preds)
             total_labels.extend(batch_labels)

     total_labels = torch.tensor(total_labels)
     total_preds = torch.tensor(total_preds)

     # Calculate F1 score
     f1 = f1_score(total_labels.detach().cpu().numpy(), total_preds.detach().cpu().numpy(), average='weighted')
     print(f'Weighted F1 score: {f1:.4f}')
```

```
Some weights of BertForSequenc
You should probably TRAIN this
Weighted F1 score: 0.8491
```

e. Memasukkan text nya untuk memprediksi hasil sentiment

```
[7]  def predict_sentiment(text):
         # Remove stop words from input text

         text = stop_words.remove(text)
         # Tokenize input text using BERT tokenizer
         encoding = tokenizer.encode_plus(text, max_length=64, truncation=True, padding='max_length', add_special_tokens=True, return_attention_mask=True, return_tensors='pt')
         inputs = encoding['input_ids'].to(device)
         masks = encoding['attention_mask'].to(device)
         # Pass input through the trained model and return predicted sentiment
         with torch.no_grad():
             outputs = model(inputs, attention_mask=masks)
             logits = outputs[0]
             probs = torch.softmax(logits, dim=1)
             _, pred_label = torch.max(probs, dim=1)
         return 'positive' if pred_label == 1 else 'negative'
```

Result:

```
[8]  text = "Bodoh kamu"
     sentiment = predict_sentiment(text)
     print(sentiment)
```

```
negative
```

```
[9]  text = "Ditengah masyarakat yg gak begitu suka baca dan lebih tertarik dengan hal2 visual, Ganjar dan Prabowo lebih ahlinya, sedikit membicarakan data, jaranga mengkritiki, tapi lebih sering tampil di medi
     sentiment = predict_sentiment(text)
     print(sentiment)
```

```
positive
```

```
[10] text = "Insya Allah Pak Anies Presiden  2024 ini akun bayaran dan survei bayaran"
     sentiment = predict_sentiment(text)
     print(sentiment)
```

```
positive
```

```
[11] text = "Sampai skrg saya bingung apa ide dan gagasan perubahan dari Anies? Sampai skrg ini, Anies masih berkutat menyinggung program program Jokowi belum ada suatu hal yang besar ide perubahan yang selalu
     sentiment = predict_sentiment(text)
     print(sentiment)
```

```
positive
```

f. Memprediksi proces.csv (hasil data preprocessing)

```
[13] predict = pd.read_csv('proces.csv')
     predict.head()
```

	Unnamed: 0	Caption	tokenized	freqdist	tokens_WSW	normalized	tokens_stemmed	merged
0	5	karena anies bisa menang di jakarta saja karen...	['karena', 'anies', 'bisa', 'menang', 'di', 'j...	<FreqDist with 45 samples and 54 outcomes>	['anies', 'menang', 'jakarta', 'alasanya', 'ah...	['anies', 'menang', 'jakarta', 'alasanya', 'ah...	['anies', 'menang', 'jakarta', 'alasanya', 'ah...	anies menang jakarta alasanya ahok kena beliau...
1	6	posisi ketiga terus tapi kayaknya dijegal teru...	['posisi', 'ketiga', 'terus', 'tapi', 'kayakny...	<FreqDist with 8 samples and 9 outcomes>	['posisi', 'ketiga', 'kayaknya', 'dijegal', 'k...	['posisi', 'ketiga', 'kayaknya', 'dijegal', 'k...	['posisi', 'tiga', 'kayak', 'jegal', 'kayak', ...	posisi ketiga kayaknya dijegal kayak why
2	7	hasil survey litbang kompas sebelum pilkada dk...	['hasil', 'survey', 'litbang', 'kompas', 'sebe...	<FreqDist with 17 samples and 18 outcomes>	['hasil', 'survey', 'litbang', 'kompas', 'pilk...	['hasil', 'survey', 'litbang', 'kompas', 'pilk...	['hasil', 'survey', 'litbang', 'kompas', 'pilk...	hasil survey litbang kompas pilkada dki ahysil...
3	8	hehwowkwowk waktu pilgub jakarta juga peringka...	['hehwowkwowk', 'waktu', 'pilgub', 'jakarta', ...	<FreqDist with 9 samples and 10 outcomes>	['hehwowkwowk', 'pilgub', 'jakarta', 'peringka...	['hehwowkwowk', 'pilgub', 'jakarta', 'peringka...	['hehwowkwowk', 'pilgub', 'jakarta', 'peringka...	hehwowkwowk pilgub jakarta peringkat mulu
4	9	menang survei kok bangga liat aja dilapangan m...	['menang', 'survei', 'kok', 'bangga', 'liat', ...	<FreqDist with 11 samples and 11 outcomes>	['menang', 'survei', 'bangga', 'liat', 'dilapa...	['menang', 'survei', 'bangga', 'lihat', 'dilap...	['menang', 'survei', 'bangga', 'lihat', 'lapan...	menang survei bangga lihat dilapangan masyarak...

g. Menghapus data yang tidak diperlukan

```
[14] predict.drop(['Unnamed: 0', 'tokenized', 'freqdist', 'tokens_WSW', 'tokens_stemmed', 'normalized'], inplace=True, axis=1)
```

h. Memasukkan kolom hasil sentimen analisis ke dalam dataframe

```
[17] # Add the predicted_sentiments list as a new column to the DataFrame
     predict['predicted_sentiments'] = predicted_sentiments

     # Write the updated DataFrame to the CSV file
     predict.to_csv('predict2.csv', index=False)
```

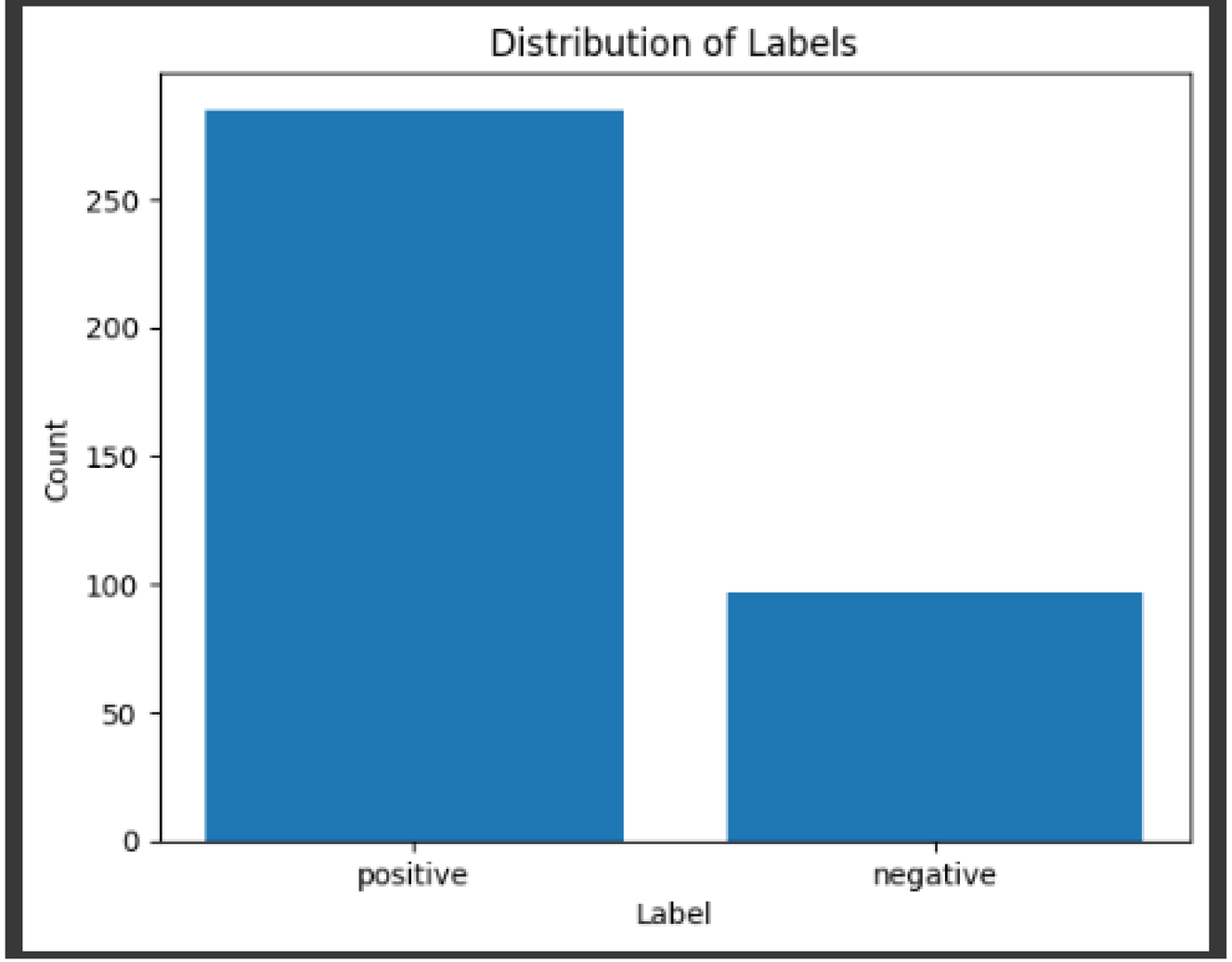
print hasil:

```
[16] # Assuming you have already imported the necessary libraries and have the DataFrame `data` with the desired column

     # Create an empty list to store the predicted sentiments
     predicted_sentiments = []

     # Iterate through the column values using a loop
     for text in predict['merged']:
         # Check if the value is a string
         if isinstance(text, str):
             # Apply the predict_sentiment() function to each text value
             sentiment = predict_sentiment(text)
         else:
             # Handle non-string values (e.g., NaN or float)
             sentiment = 'Unknown'
         # Append the predicted sentiment to the list
         predicted_sentiments.append(sentiment)

     # Print or manipulate the list of predicted sentiments
     for sentiment in predicted_sentiments:
         print(sentiment)
```

```
positive
negative
negative
positive
negative
positive
Unknown
positive
positive
negative
positive
positive
positive
positive
negative
positive
```

i. Histogram hasil perbandingan analisis sentimen

```
10
11 # Show the plot
12 plt.show()
13
```



KESIMPULAN

HASIL :

predict2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Caption,merged,predicted_sentiments																			
2	karena anies bisa menang di jakarta saja karena alasanya ahok kena kasus bukan karena beliau dinilai lebih baik memimpin jakarta ketimbang ahok sudahlah jujuran saja memang seperti itu keadaanya tapi sy																			
3	posisi ketiga terus tapi kayaknya dijegal terus kayak why,posisi ketiga kayaknya dijegal kayak why,positive																			
4	hasil survey litbang kompas sebelum pilkada dki ahysilvi ahokjarot aniessandi silahkan saja percaya survey biar hidup tambah ceria,hasil survey litbang kompas pilkada dki ahysilvi ahokjarot aniessandi silahka																			
5	hehwowkwowk waktu pilgub jakarta juga peringkat mulu jadi jadi aja,hehwowkwowk pilgub jakarta peringkat mulu,positive																			
6	menang survei kok bangga liat aja dilapangan masyarakat sdh mulai cerdas,menang survei bangga lihat dilapangan masyarakat cerdas,positive																			
7	pernyataan seperti ini jadi keinget sama pernyataan para pakar dan juga media soal pesimis nya formula dapat digelar dalam jangka waktu bulan padahal tinggal menikmati proses nanti tanggal mainnya juga																			
8	tiga aja di jegal di takuti bagaimana kalau no satu pasti kejang insya allah pak anies presiden,jegal takuti no kejang insya allah anies presiden,positive																			
9	lah emang konsep perubahan yang di tawarkan oleh ganjar dan prabowo apa ya,emang konsep perubahan tawarkan ganjar prabowo,negative																			
10	sampai skrg saya bingung apa ide dan gagasan perubahan dari anies sampai skrg ini anies masih berkutat menyinggung program program jokowi belum ada suatu hal yang besar ide perubahan yang selalu diu																			
11	waktu mendekati pilkada dki juga posisi elektabilitas anies urutan ketiga itu hasil survei litbang kompas loh,mendekati pilkada dki posisi elektabilitas anies urutan ketiga hasil survei litbang kompas,positive																			
12	karna pak anis kaga punya duit buat nyewa lembaga survey yg setiap minggu keluar hasil surveyitu biayanya brp buat nyewa lembaga survey yg setiap minggu keluar hasil survey,karna anies tidak duit nyewa l																			
13	yang dukung anies punya kepentingan memang ga berpendidikan bisa kena tipu dia,dukung anies kepentingan berpendidikan kena tipu,positive																			
14	ya simple karena kinerjanya tidak ada hasil logikanya,simple kinerjanya hasil logikanya,positive																			
15	kalo emang survei elektabilitas ini bisa jadi acuan hasil akhir harusnya ahok udah jadi gubernur dki tahun,emang survei elektabilitas acuan hasil ahok udah gubernur dki,positive																			
16	pov memasuki kawasan full merah,pov memasuki kawasan full merah,positive																			
17	insya allah pak anies presiden ini akun bayaran dan survei bayaran,insya allah anies presiden akun bayaran survei bayaran,positive																			
18	yuk survey kecilan bandingin sama survey halu yang pilih pemimpin cerdas anies baswedan senggol donk,yuk survey kecilan bandingin survey halu pilih pemimpin cerdas anies baswedan senggol donk,positiv																			
19	mari kita sama screenshot postingan ini dan kita repost nanti setelah pengumuman hasil pemilu barangkali benar kan surveinya litbang kompas survei tapi rasa rasa cenayang ya,mari screenshot postingan rep																			
20	kenapa ga kunjung naik krn anis gapunya lembaga survey udahlah jgn kalian pusing mau di pihak pro kontra intinya jgn munafik lah gausa sok naif bikin alasan ini itu dr jaman anis di pilgub pun survei litbang a																			
21	karna ini pilpres yg jadi pemilih seluruh warga indonesia bukan pigub yv milih seiman,karna pilpres pemilih warga indonesia pigub yv milih seiman,positive																			
22	imho sebenernya takut proyekvisi jangka panjang jokowi nya jadi gk lanjut aja karna anies merupakan antitesis nya jokowi bagus atau tidaknya proyekvisi jangka panjang jokowi masih debatable tp kalo tiap g																			

KESIMPULAN

Hasil analisis sentimen untuk topik "Kenapa Elektabilitas Anies Tak Kunjung Naik?" membuahkan dua bagian sentimen, positif dan negatif. Namun, dari hasil analisis, terlihat bahwa lebih banyak terdapatnya sentimen positif dari komen netizen di instagram.

Sentimen positif:

- bismillah anis menang pilpres tak peduli apa kata bazer
- saya pilih anis

Sentimen negatif:

- anies jarang mandi kyny

TERIMA KASIH