langkah pertama Import Library

```
import tweepy
from textblob import TextBlob
import pandas as pd
import numpy as np
import re,string
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')

from nltk.corpus import stopwords
from wordcloud import WordCloud,STOPWORDS

import csv
```

Kemudian kita masukan API Keynya yang di dapatkan dari twitter

```
Consumer_Key = "ee71LNIhBorkkXStapMD8Xf8a"
Consumer_Secret_Key = "CtimpCO2188bPm8Dx6j5eK43AbWJPnX8BJlIwO6irNJt1w2lmB"
Access_Token = "711763537971044353-ydboIEZFp6rvi7tT8Iqzx0odvr15ygt"
Access_Token_Secret = "YT5tehyL7rNqge2H992GAMHNEfWCcVBjf3J8UjrHq7SrC"

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth, wait_on_rate_limit=True)
```

kita mengambil data dari Gojek indonesia

```
csvFile = open('Gojek Indonesia.csv','a', encoding='utf-8')
csvWriter = csv.writer(csvFile)
for tweet in tweepy.Cursor(api.search, q= "#gojek",count=2000,
                           lang='id').items(2000):
    print(tweet.text)
    csvWriter.writerow([tweet.text])
```

Membaca file data yang sudah kita ambil

```
df = pd.read_csv("Gojek Indonesia.csv", header=None)
df
```

KIta bersihkan data kata kita yang tidak penting

```
def praproses(teks):
    teks = re.sub(r'http\S+','',teks)
    teks = hapus_tanda(teks)
```

```
        teks = hapus_tanda(teks)
        teks = re.sub(r'#([^\s]+)', r'\1', teks) #hapus #tagger
        teks = re.sub('@[A-Za-z0-9]+', '',teks)  #hapus @
        teks = re.sub(r':([^\s]+)', r'\1', teks) #hapus #tagger
        teks = re.sub('RT[\s]+', '',teks)#hapus RT
        teks = re.sub('https?:\/\/\S+', '',teks)#hapus hyperlink
        teks = re.sub(r'\w*\d\w*', '',teks).strip()#hapus angka dan angka yang berada dalam st
        teks = hapus_katadouble(teks)#hapus repetisi karakter
        teks = teks.lower() #ubah jadi lower case
        return teks
    def hapus_emoji(teks):
        regrex_pattern = re.compile(pattern = "["
            u"\U0001F600-\U0001F64F"  # emoticons
            u"\U0001F300-\U0001F5FF"  # symbols & pictographs
            u"\U0001F680-\U0001F6FF"  # transport & map symbols
            u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                               "]+", flags = re.UNICODE)
        return regrex_pattern.sub(r'',teks)
    #     teks = to_kbbi(teks)


    def hapus_tanda(teks):
        tanda_baca = set(string.punctuation)
        tanda_baca.update(['…'])
        teks = ''.join(ch for ch in teks if ch not in tanda_baca)
        return teks

    def hapus_katadouble(s):
        #look for 2 or more repetitions of character and replace with the character itself
        pattern = re.compile(r"(.)\1{1,}", re.DOTALL)
        return pattern.sub(r"\1\1", s)

    def kbbi(kata): # penyeragaman kata berdasarkan kbbi
      #kbba = [kamus.strip('\n').strip('\r') for kamus in open('kamus\\kbba.txt')]
      kamus_kata = [kamus.strip('\n').strip('\r') for kamus in open('kbba.txt')]
      #ubah list menjadi dictionary
      dic = {}
      for i in kamus_kata:
        (key, val) = i.split('\t')
        dic[str(key)] = val
      #kbbi cocokan
      final_string = ' '.join(str(dic.get(word, word)) for word in kata).split()
      return final_string

    def to_kbbi(teks):
        tek = teks.split()
        tek = kbbi(tek)
        return tek

    #Removing the noisy text
    def cleanText(text):
        text = remove_stopwords(text)
        text = praproses(text)
        return text

    #Removing the stopwords from text
```

```
#Removing the stopwords from text
def remove_stopwords(text):
    final_text = []
    for i in text.split():
        if i.strip().lower() not in stop_w:
            final_text.append(i.strip())
    return " ".join(final_text)



# Load stopword Bahasa Indonesia
stopword_id = pd.read_csv('stopword_id.csv', sep='\t', header=None)
stopword_id.columns = ['word']
stop_w = stopword_id['word'].to_list() #diubah ke list
# print('ada' in stop_w) # test periksa kata di dalam list stop_w

# def cleanText(teks):
#     teks = re.sub('@[A-Za-z0-9]+', '',teks)  #hapus @
#     teks = re.sub(r'#([^\s]+)', r'\1', teks) #hapus #tagger
#     teks = re.sub('RT[\s]+', '',teks)#hapus RT
#     teks = re.sub('https?:\/\/\S+', '',teks)#hapus hyperlink
#     teks = teks.lower() #ubah jadi lower case
#     teks = re.sub(r"[-()\"#/@;:<>{}=~|.?,]", "", teks)
#     import string
#     killpunctuation = str.maketrans('', '', string.punctuation)
#     return teks

#Apply function on review column
df[0] = df[0].apply(cleanText)

df
```

Lakukan untuk melakukan sentiment popularity

```
def getSubjectivity(text):
    return TextBlob(text).sentiment.subjectivity

def getPolarity(text):
    return TextBlob(text).sentiment.polarity


df['Polarity'] = df[0].apply(getPolarity)
df['Subjectivity'] = df[0].apply(getSubjectivity)

df
```

Setkah itu kita menampilkan visual dengan dengan wordCLoud

```
allWords =' '.join([twts for twts in df[0]])
wc = WordCloud(width = 500 , height = 300 , random_state=10, max_font_size=110).generate(a
plt.imshow(wc , interpolation = 'bilinear')
plt.axis('off')
```

```
plt.show()
```

kita perthitungan sentiment

```
#untuk menambahkan sentimen positif, negatif / netral dari polarity yg sudah dihitung
def getAnalysisSentiment(score):
    if score < 0:
        return 'Negative'
    elif score == 0:
        return 'Neutral'
    else:
        return 'Positive'


df['Analysis'] = df['Polarity'].apply(getAnalysisSentiment)

df
```

Menampilkan hasil analisis sentiment

```
plt.figure(figsize=(8,6))
for i in range(0, df.shape[0]):
    plt.scatter(df["Polarity"][i], df["Subjectivity"][i], color="Blue")

plt.title('Sentiment Analysis')
plt.xlabel('Polarity')
plt.ylabel('Subjectivity')
plt.show()
```

hasil perhitungan analisis sentiment

```
df['Analysis'].value_counts()
```

Menampilakn hasil berdasarkan grafik

```
plt.title('Sentiment Analysis')
plt.xlabel('Sentiment')
plt.ylabel('Counts')
df['Analysis'].value_counts().plot(kind ='bar')
plt.show()
```