

```
In [1]: import json
import pandas as pd
import nltk
from nltk.tokenize import TweetTokenizer as tw_tokenizer
from collections import Counter
from unicode import unicode

tweets_data = [] # Membuat list kosong untuk menyimpan data json perbaris
tweets_file = open('data/dataset.txt', "r") # membuka file
for line in tweets_file:
    try:
        tweet = json.loads(line) # Membaca data dalam format json dari file perbaris
        tweets_data.append(tweet) # Menambahkan data dari file ke dalam list
    except:
        continue
tweets_file.close()

tweets = pd.DataFrame()
# Informasi tweet
tweets['created_at'] = list(map(lambda tweet: tweet['created_at'], tweets_data))
tweets['id_str'] = list(map(lambda tweet: tweet['id_str'], tweets_data))
tweets['text'] = list(map(lambda tweet: tweet['text'], tweets_data))
tweets['source'] = list(map(lambda tweet: tweet['source'], tweets_data))
tweets['lang'] = list(map(lambda tweet: tweet['lang'], tweets_data))

# Informasi user
tweets['user_id'] = list(map(lambda tweet: tweet['user']['id_str'], tweets_data))
tweets['user_name'] = list(map(lambda tweet: tweet['user']['name'], tweets_data))
tweets['user_screen_name'] = list(map(lambda tweet: tweet['user']['screen_name'], tweets_data))
tweets['user_location'] = list(map(lambda tweet: tweet['user']['location'], tweets_data))
tweets['user_url'] = list(map(lambda tweet: tweet['user']['url'], tweets_data))
tweets['user_description'] = list(map(lambda tweet: tweet['user']['description'], tweets_data))
tweets['user_followers_count'] = list(map(lambda tweet: tweet['user']['followers_count'], tweets_data))
tweets['user_friends_count'] = list(map(lambda tweet: tweet['user']['friends_count'], tweets_data))
tweets['user_favourites_count'] = list(map(lambda tweet: tweet['user']['favourites_count'], tweets_data))
tweets['user_statuses_count'] = list(map(lambda tweet: tweet['user']['statuses_count'], tweets_data))
tweets['user_created_at'] = list(map(lambda tweet: tweet['user']['created_at'], tweets_data))

# Informasi Tempat
tweets['address'] = list(map(lambda tweet: tweet['place']['full_name'] if tweet['place'] != None else None, tweets_data))
tweets['country'] = list(map(lambda tweet: tweet['place']['country'] if tweet['place'] != None else None, tweets_data))

# Entities
hashtags = list(map(lambda tweet: tweet['entities']['hashtags'] if tweet['entities'] != None else None, tweets_data))
tweets['hashtags'] = list(map(lambda tweet: ', '.join(list(map(lambda tw: tw['text'], tweet))), hashtags))

urls = list(map(lambda tweet: tweet['entities']['urls'] if tweet['entities'] != None else None, tweets_data))
tweets['url'] = list(map(lambda tweet: ', '.join(list(map(lambda tw: tw['url'], tweet))), urls))

mentions = list(map(lambda tweet: tweet['entities']['user_mentions'] if tweet['entities'] != None else None, tweets_data))
tweets['mentions'] = list(map(lambda tweet: ', '.join(list(map(lambda tw: tw['screen_name'], tweet))), mentions))

mentions = list(map(lambda tweet: tweet['entities']['user_mentions'] if tweet['entities'] != None else None, tweets_data))
tweets['mentions'] = list(map(lambda tweet: ', '.join(list(map(lambda tw: tw['screen_name'], tweet))), mentions))

tweetsIn = tweets[tweets.lang == 'in']
```

```
In [2]: tweetsIn['created_at']=pd.to_datetime(tweetsIn['created_at'], utc=True)

# merubah index tabel berdasarkan pada kolom 'created_at'

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#i
ndexing-view-versus-copy
""""Entry point for launching an IPython kernel.
```

```
In [3]: tweetsIn.index = tweetsIn.created_at

# melihat hasil perubahan index
```

```
In [4]: tweetsIn.head()

# merubah zona waktu pada index 'created_at' yang merupakan standar UTC +0000 menjadi zona waktu 'Asia/Jakarta' +0700
```

Out[4]:

	created_at	id_str	text	source	lang
created_at					
2019-04-02 17:20:35+00:00	2019-04-02 17:20:35+00:00	1113129109989744640	@danangtrip whahaha iya sih bener	<a href="http://twitter.com/download/android" ...	in
2019-04-02 17:20:37+00:00	2019-04-02 17:20:37+00:00	1113129120429457409	#Repost antv_official\n••••\n•\nLedakan Me...	...	in
2019-04-02 17:20:38+00:00	2019-04-02 17:20:38+00:00	1113129123579224066	[00:16] #JAKARTA #MACET Kalideres arah Batu C...	twee...	in
2019-04-02 17:20:38+00:00	2019-04-02 17:20:38+00:00	1113129124913012737	[00:10] #JAKARTA Tol Cawang - TMI - Cibur ...	twee...	in
2019-04-02 17:20:39+00:00	2019-04-02 17:20:39+00:00	1113129127303716866	Well, makin kesini ku tahu yang mana yang mint...	<a href="http://twitter.com/download/iphone" f...	in

5 rows × 21 columns

```
In [5]: import pytz
from datetime import datetime
from pytz import timezone

JKT = pytz.timezone('Asia/Jakarta')
tweetsIn.index = tweetsIn.index.tz_convert(JKT)
# melihat hasil perubahan zona waktu di kolom index
```

```
In [6]: tweetsIn.head()
#tweetsIn.to_excel('export_dataframe.xlsx')

all_tweetsIn = [tweetsIn.loc['20190403'], tweetsIn.loc['20190405'], tweetsIn.loc['20190407']]

#print ("jumlah data tweetsIn04 :", len(tweetsIn04))

df = pd.DataFrame({'tanggal' : ['03 April 2019', '05 April 2019', '07 April 2019'],
                    'jumlah tweets' : [len(all_tweetsIn[0]), len(all_tweetsIn[1]), len(all_tweetsIn[2])])})
# melihat grafik bar total tweets per hari
```

```
In [7]: import matplotlib.pyplot as plt

fig, ax = plt.subplots()
x_pos = list(range(len(df)))
width = 0.8
plt.bar(x_pos, df['jumlah tweets'], width, alpha=1, color='r')
ax.set_ylabel('Jumlah Tweets', fontsize=12)
ax.set_title('Total Tweets per Hari', fontsize=12)
ax.set_xticks([p + 0.5 * width for p in x_pos])
ax.set_xticklabels(df['tanggal'])
plt.show()

<Figure size 640x480 with 1 Axes>
```

```
In [8]: import matplotlib.pyplot as plt
import seaborn as sns
from pylab import *
get_ipython().magic('matplotlib inline')
plt.rcParams['figure.figsize'] = (15, 5)

from pandas import Series
def f(x):
    return Series(dict(Number_of_tweets = x['text'].count()))

def grafik(tweet, judul = '', jenis = 'line', color = '', xlabel = '', ylabel = '', ket = []):
    count = tweet.groupby(tweet.index.hour).apply(f)

    if color == '':
        hourly_plot = count['Number_of_tweets'].plot(kind=jenis)
    else:
        hourly_plot = count['Number_of_tweets'].plot(kind=jenis, color=color)

    hours = list(range(0,24))
    xticks(np.arange(24), hours, rotation = 0, fontsize = 9)

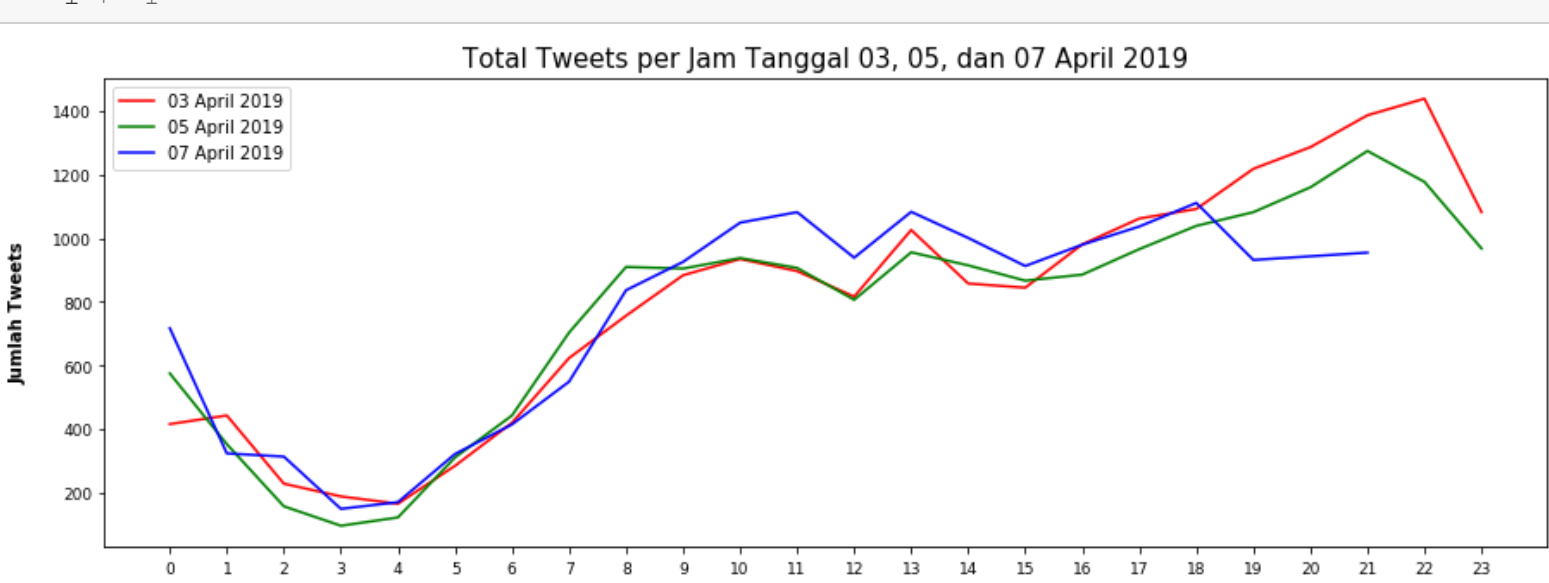
    hourly_plot.set_title(judul, fontsize=15)
    hourly_plot.set_xlabel(xlabel, weight='bold', labelpad=15)
    hourly_plot.set_ylabel(ylabel, weight='bold', labelpad=15)
    hourly_plot.legend(ket)

    xticks(fontsize = 9, rotation = 0, ha= "center")
    yticks(fontsize = 9)

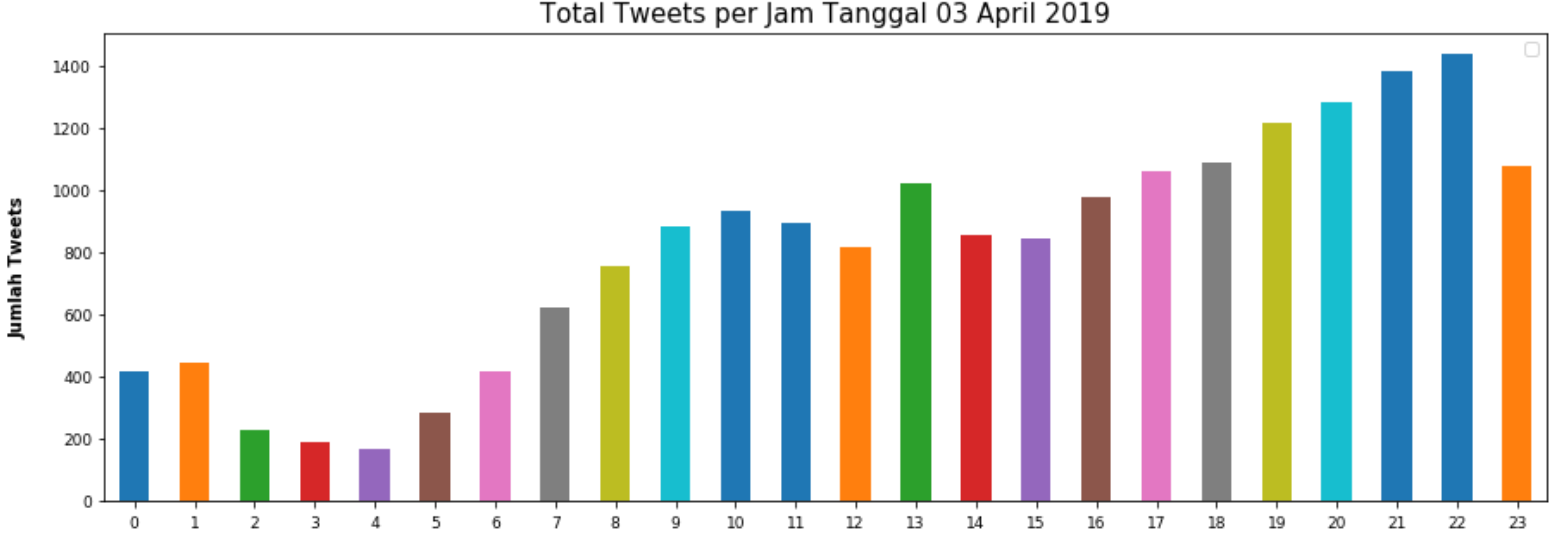
    hourly_plot.tick_params(axis='x', pad=5)
```

```
In [9]: colors = ['red', 'green', 'blue']
i = 0
judul = 'Total Tweets per Jam Tanggal 03, 05, dan 07 April 2019'
xlab = 'Jam'
ylab = 'Jumlah Tweets'
keterangan = list(df['tanggal'])

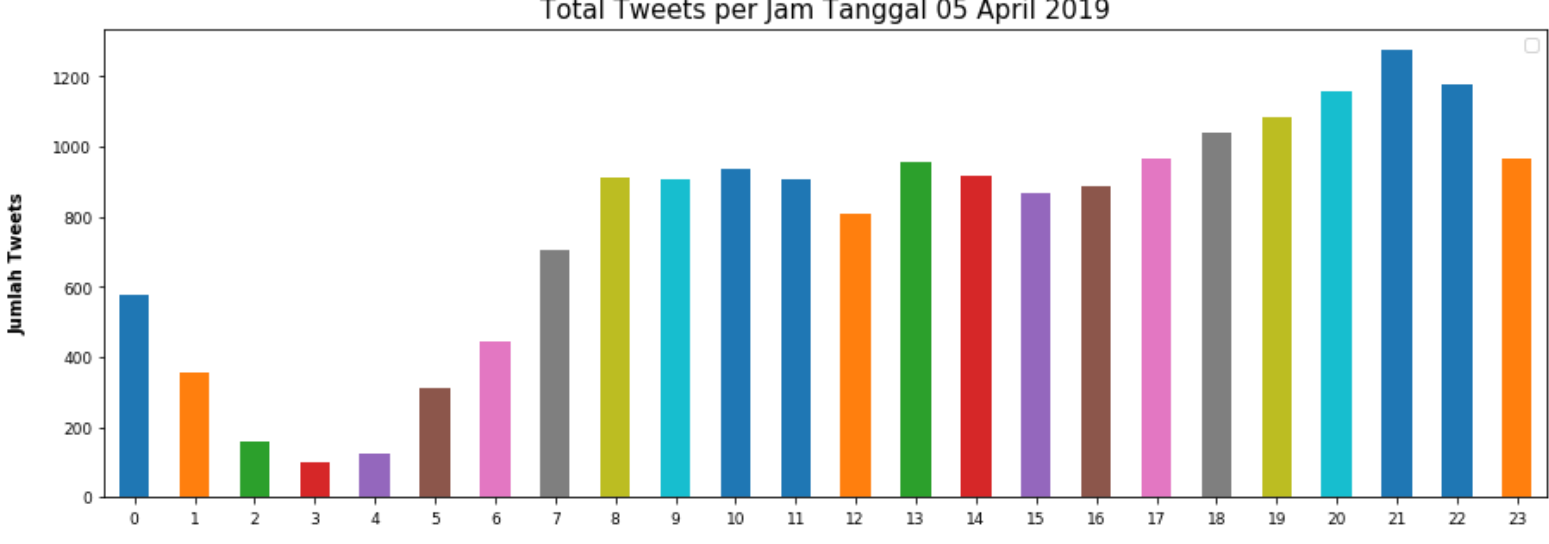
for tw in all_tweetsIn:
    grafik(tweet = tw, judul = judul, color = colors[i], xlabel = xlab, ylabel = ylab, ket = keterangan)
    i += 1
```



```
In [10]: judul = 'Total Tweets per Jam Tanggal 03 April 2019'
grafik(tweet = all_tweetsIn[0], jenis = 'bar', judul = judul, xlabel = xlab, ylabel = ylab)
```



```
In [11]: judul = 'Total Tweets per Jam Tanggal 05 April 2019'
grafik(tweet = all_tweetsIn[1], jenis = 'bar', judul = judul, xlabel = xlab, ylabel = ylab)
```



```
In [12]: judul = 'Total Tweets per Jam Tanggal 07 April 2019'
grafik(tweet = all_tweetsIn[2], jenis = 'bar', judul = judul, xlabel = xlab, ylabel = ylab)
```