

KLASIFIKASI KEPERIBADIAN BERBASIS SENTIMENT DI SOSIAL MEDIA TWITTER MENGUNAKAN METODE PBSC

User Manual

Warih Maharani | Joshua Panjaitan

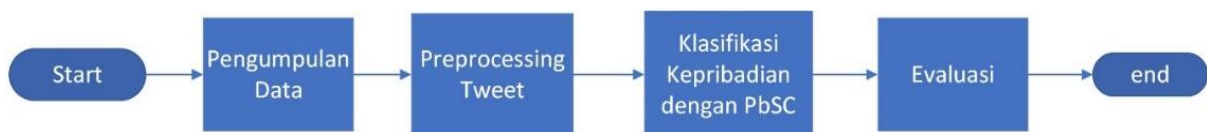


About

Disclaimer

Program dibuat dengan bahasa pemrograman python versi 3.7. Terdapat beberapa *Library/Dependency* yang digunakan untuk mendukung eksekusi program klasifikasi kepribadian ini. Harap membaca detail *library/dependency* yang digunakan dalam github : <https://github.com/joshuapanjaitan/Personality-Prediction-Using-PbSC> untuk proses instalasi.

Alur Sistem



Gambar 1 Alur Sistem

Program dibangun dengan alur tahapan diatas dimana terdapat 3 tahapan utama yang dilakukan. Tahapn yang pertama adalah pengumpulan data, dimana tahapan ini bertujuan untuk mengumpulkan *tweet* dari tiap tiap user dengan menggunakan *twitter* API. Tahapan kedua adalah *pre-processing tweet*, dimana tahapan ini bertujuan untuk membersihkan *tweet* dari *string*, *special character* atau url yang menempel dalam setiap *tweet* dari user. Tahapan ketiga adalah klasifikasi dengan metode utama yaitu PbSC. Tahapan ke-3 ini juga bertujuan menghitung akurasi dari metode yang digunakan.

User Guide

1. Pengumpulan Data

Tahapan pertama yang dalam penelitian ini adalah tahap pengumpulan data. Download semua file yang ada di URL Github diatas dan buka folder Crawl lalu jalankan dengan menggunakan jupyter notebook dengan mengetik 'jupyter notebook' via *command prompt* pada *directory* file.

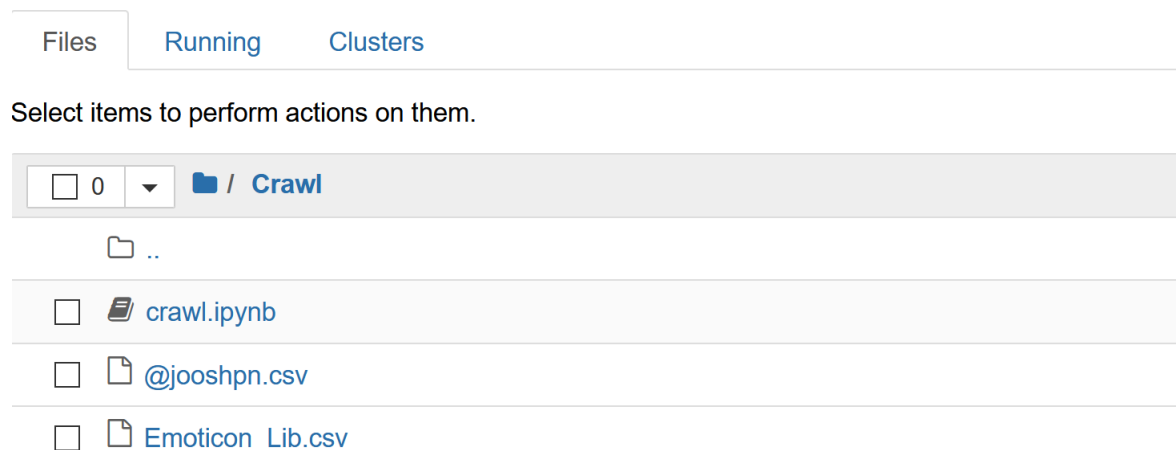
```
C:\Windows\System32\cmd.exe - jupyter notebook
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Nuclear Codes\SEMESTER 8\Tugas Akhir 2\AFTER SIDANG\Apps>jupyter notebook
[I 13:08:14.011 NotebookApp] Serving notebooks from local directory: D:\Nuclear Codes\SEMESTER 8\Tugas Akhir 2\AFTER SIDANG\Apps
[I 13:08:14.012 NotebookApp] The Jupyter Notebook is running at:
[I 13:08:14.014 NotebookApp] http://localhost:8888/?token=195890fadd373dccc0ca40663b06d3d11b1c508b52bbf995
[I 13:08:14.014 NotebookApp] or http://127.0.0.1:8888/?token=195890fadd373dccc0ca40663b06d3d11b1c508b52bbf995
[I 13:08:14.014 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 13:08:14.153 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/Joshua/AppData/Roaming/jupyter/runtime/nbserver-4024-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=195890fadd373dccc0ca40663b06d3d11b1c508b52bbf995
or http://127.0.0.1:8888/?token=195890fadd373dccc0ca40663b06d3d11b1c508b52bbf995
```

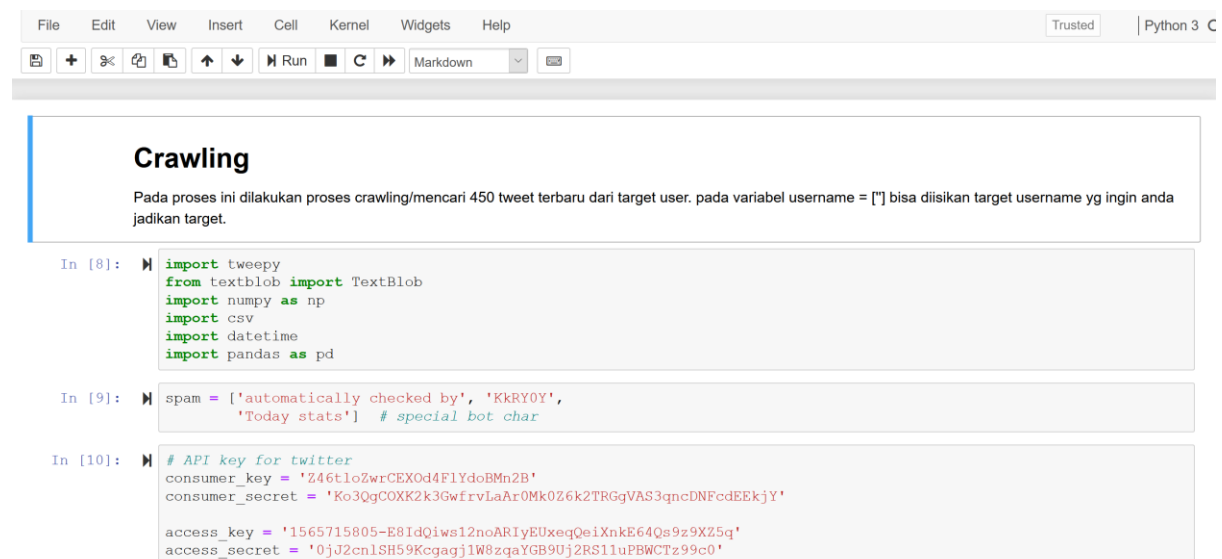
Gambar 2 Jalankan dengan jupyter notebook

Setelah membuka dengan jupyter maka, anda akan di *redirect* kedalam *browser default* anda. Jika berjalan dengan lancar folder yang terbuka adalah seperti dibawah,



Gambar 3 Isi folder Crawl

Setelah itu buka file `crawl.ipynb`. Anda akan masuk kedalam halaman jupyter notebook dan dapat mengeksekusi file yang ada. Pada blok 1-3 Berisi konfigurasi untuk melakukan koneksi dengan *twitter*.



Gambar 4 Blok API

Tekan tombol **Run** untuk setiap blok untuk mengeksekusi program agar bisa terkoneksi dengan *dependency* dan *API twitter*. Pada Blok Selanjutnya berisi fungsi untuk mengambil 450 *tweet* terbaru dari *username* yang masukkan. Perhatikan pada gambar 5 blok 2, terdapat *username* @jooshpn, itu berarti target usernya adalah @jooshpn, dan program akan melakukan *crawling* terhadap 450 tweet user yang terbaru dan menyimpannya dalam sebuah file csv bernama @jooshpn.csv. Setelah proses crawling selesai maka file sudah bisa diakses dan pindahkan file hasil ke folder berikutnya Preprocessing untuk tahapan berikutnya.

```
def get_tweets(username):
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_key, access_secret)
    api = tweepy.API(auth)

    # tweets = api.user_timeline(
    #     screen_name=username, count=200, page=0)
    pages = [] # store indo tweets
    for t in range(10):
        store = api.user_timeline(
            screen_name=username, count=200, page=t, tweet_mode='extended')
        for v in store:
            tweet = v.full_text
            lang = v.lang
            if lang == 'in': # get only indo tweet
                if len(tweet) > 1:
                    rtCek = tweet[0]+tweet[1]
                    if rtCek != 'RT':
                        if spam[0] not in tweet and spam[1] not in tweet and spam[2] not in tweet:
                            pages.append(v.full_text)

    #tulis tweet yang sudah di crawl ke csv
    with open(username+".csv", 'w', newline='') as f:
        tulis = csv.writer(f)
        for j in range(len(pages)):
            v = str(pages[j])
            g = str(v.encode('unicode-escape')) # emoji ubah ke unicode
            tulis.writerow([g])
            if j == 450: # jumlah row
                break

if __name__ == '__main__':
    username = ['@jooshpn'] # masukkan username target
    for uname in username:
        get_tweets(uname)
```

Gambar 5. Crawling

2. Preprocessing

Preprocessing merupakan tahapan untuk membersihkan *tweet* yang baru di *crawl* dari tahapan sebelumnya. Hal-hal yang dibersihkan meliputi *special character*, *url*, *emoticon* dan kata-kata yang berlebihan. Hasil dari tahapan ini adalah *tweet* dari setiap user yang sudah bersih dan siap untuk dipakai dalam tahapan berikutnya. Untuk mengeksekusi program masuk ke folder Preprocessing lewat jupyter notebook seperti yang sebelumnya dilakukan.



Gambar 6 Isi File Preprocessing

File *Pre-processing* terbagi menjadi 2 file, yaitu *clean1.ipynb* dan *clean2.ipynb*. Dimana kedua file memiliki fungsi masing-masing. Buka file *Clean1.ipynb* terlebih dahulu untuk melakukan tahap awal *pre-processing*. Tujuan dari file *clean1* ini berfungsi untuk membersihkan tweet

dari special character dan melakukan konversi *emoticon* dari *unicode* menjadi bahasa yang bisa dikenali oleh manusia.

Proses Preprocessing

Pada proses ini dilakukan pembersihan tweet dari special char dan convert emoji

```
In [13]: import tweepy
from textblob import TextBlob
import numpy as np
import csv
import datetime
import nltk
from nltk.tokenize import TweetTokenizer
from collections import defaultdict
import re
import pandas as pd
```

```
In [14]: # deteksi key dalam dictionary
def checkKey(dict, key):
    hasil = ''
    if key in dict:
        hasil = 'yes'
    else:
        hasil = 'No'
    return hasil
```

```
In [15]: def replace_emoji(key):
    kunci = 'U'+key
    tes = checkKey(kamus, kunci)
    hasil = ''
    if tes == 'yes':
        hasil = '#' + kamus[kunci] + '#'
    elif tes == 'No':
        hasil = ''
    return hasil
```

Gambar 7 clean1.ipynb

Run semua blok sampai kebawah untuk menyimpan hasil *preprocessing* kedalam file .csv setelah file clean1.ipynb selesai di run sampai selesai maka anda bisa melanjutkan untuk merun file clean2.ipynb. Clean2 bertujuan untuk membersihkan *tweet* dari url.

```
In [5]: # remove yg URL, dll.
import tweepy
from textblob import TextBlob
import numpy as np
import csv
import datetime
import nltk
from nltk.tokenize import TweetTokenizer
from collections import defaultdict
import re
import pandas as pd
```

```
In [6]: uname = ['@jooshpn'] #ganti dengan username twitter target
newSentence = []
userTw = []
```

```
In [7]: for nama in uname:
    with open(nama+".csv", 'r') as csv_file:
        csv_reader = csv.reader(csv_file)
        for line in csv_reader:
            userTw.append(line)

    for i in range(len(userTw)):
        sent = ''.join(userTw[i].copy())
        token = sent.split()
        newToken = sent.split()
        for x in range(len(token)):
            # tambahkan fitur yg mau dihapus
            if 'http' in token[x]:
                newToken.remove(token[x])
            elif 'ue6f4' in token[x]:
                newToken.remove(token[x])
        concate = ''.join(newToken).lower()
        newSentence.append(concate)

    with open(nama+".csv", 'w', newline='') as f:
        tulis = csv.writer(f)
```

Gambar 8 clean2.ipynb

```

In [8]: data = pd.read_csv(uname[0]+' .csv', usecols=[0], names=['tweet'])
        pd.set_option('display.max_colwidth', -1)
        data

Out[8]:

```

	tweet
0	@yolanda_n28 iya kan, emg ngeselin
1	apa cm aku yg benci bet liat slide yg isinya kata2 semua?
2	yaa anak jaman sekarang emg jago2 pakai teknologi2 onlen
3	pelajaran hari ini : bahkan bosnya g-suite belajar pakai g-suite dari anaknya..
4	@coachjustinl klopp keknya ngelawak coc, jadi malas deh nonton pool,
...	...
446	@coachjustinl gatau coc
447	baru sadar kalau besok ada acara,, dan udh terlanjur begadang buat nonton nihh bola ufe0f ufe0f
448	@coachjustinl udah,, blok aja coc
449	semangats jo, kamu punya tuhan yg tidak pernah menyerah buat mu..
450	astaga,, ini aku terkadang suka banget ambil keputusan yang salah, padahal udh di timbang2 tetap aja rasa ragu itu ada. apakah cuma aku ? #disappointed face#

```

451 rows x 1 columns

In [ ]:

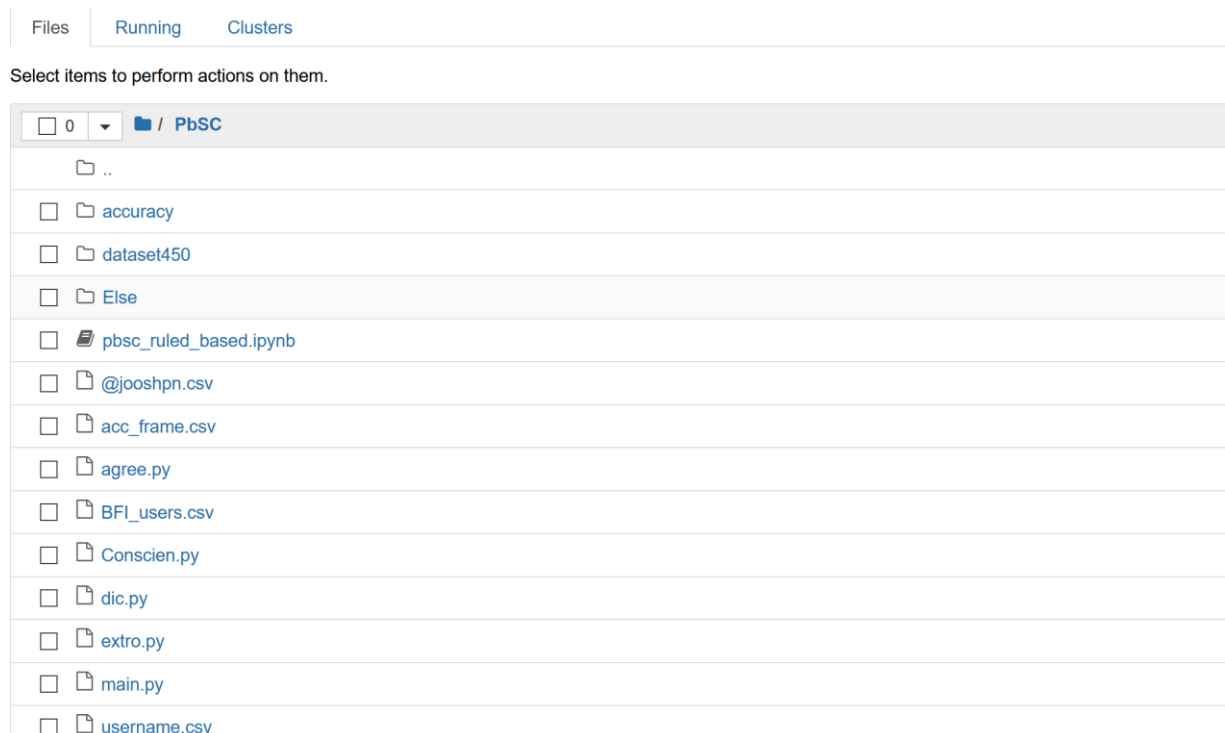
```

Gambar 9. Hasil Preprocessing

Setelah 2 file tersebut berhasil di eksekusi sampai selesai, berikut *preview tweet* yang sudah sibersihkan tersebut. Jika previewnya sudah muncul itu berarti file *tweet* sudah berhasil di *save* dan siap untuk dipakai untuk tahapan berikutnya. Pindahkan file result yaitu @username.csv kedalam folder berikutnya yaitu PbSC untuk proses klasifikasi kepribadian.

3. Klasifikasi PbSC

Tahapan terakhir dalam sistem ini adalah tahap klasifikasi. Untuk menjalankan file klasifikasi pastikan file hasil @username.csv dari tahap *preprocessing* sudah ada dalam folder PbSC. Buka Folder PbSC dengan menggunakan *command prompt* dan jalankan jupyter notebook seperti sebelumnya.



Gambar 10 Folder Klasifikasi PbSC

Pada folder tersebut terdapat 2 file yang harus anda perhatikan yaitu **pbsc_ruled_based.ipynb** dan **@jooshpn.csv** dimana file **pbsc_ruled_based.ipynb** merupakan file utama klasifikasi dan file **@jooshpn.csv** merupakan file yang berisi 450 tweet terbaru dari user yang sudah dilakukan proses *preporcessing*. Buka file **pbsc_ruled_based.ipynb**



Gambar 11 Blok 1-3 PbSC

Pada blok 1-3 berisi fungsi konfigurasi dengan library,parameter dan file **@jooshpn.csv**. Jalankan semua blok dengan menekan tombol **run**.



Gambar 12 Algoritma PbSC

Blok berikutnya berisi algoritma berbasis aturan dari PbSC. Dimana bisa dilihat penggunaan *if then else* sesuai dengan paper acuan dengan menggunakan parameter yang sudah dijelaskan dalam jurnal. Algoritma ini bertujuan untuk memprediksi kepribadian user berdasarkan 450 *tweet* yang terkumpul. Hasilnya bisa dilihat dibawah bahwa **@jooshpn** memiliki kepribadian

low,low,netral yang dimana secara berurutan merepresentasikan dimensi *extroversion*, *agreeableness* dan *constciouness*.

4. Evaluasi

Untuk hasil dari 122 user yang sudah dikumpulkan dengan kuesioner dan 288 user dengan kelas BFI anda bisa terus mengeksekusi file **pbsc_ruled_based.ipynb** sampai kebawah. Sedikit kebawah anda bisa melihat hasil kalkulasi untuk 122 user responden.

```
In [6]: ► extroValue = extro.driver(uname[0])
conValue = cons.driver(uname[0])
agreValue = agree.driver(uname[0])
print(extroValue)
print(agreValue)
print(conValue)

[79, 0.17517, 105, 0.23282]
[56, 0.12417, 105, 0.23282]
[28, 0.06208, 16, 0.03548]
```

```
In [7]: ► frame = pd.read_csv('acc_frame.csv')
pd.set_option('display.max_colwidth', -1)
frame
```

Out[7]:

	Username	Extro	Agree	Cons	LE	LA	LC
0	@lidyajustn	#	1	#	1	1	0
1	@naufal_fadh	1	#	#	0	0	0
2	@erbinaselvia4	0	#	#	0	0	0
3	@bearbee33	#	#	#	0	1	0
4	@enkit2	#	#	#	0	0	0
...
117	@irnaa_sp	1	0	#	1	0	0
118	@callme_Ruth	0	0	#	1	0	0
119	@mayasabrinaas	#	0	#	1	1	1
120	@marthamgdln	1	1	#	1	0	0
121	@natasha_forsa	1	0	#	0	0	0

122 rows × 7 columns

Gambar 13 Hasil Perhitungan 122 User Responden

Dimana Extro, Agree dan Cons merupakan hasil prediksi dan LE, LA, LC adalah label yang ditentukan oleh peneliti secara berurutan. Tanda # merupakan hasil prediksi Netral. Untuk melihat akurasi yang didapatkan anda bisa mengeksekusi blok selanjutnya

Akurasi Keseluruhan Data

```
In [8]: ► bawah = len(frame['Extro']) #sebagai pembanding
ext = 0
agree = 0
cons = 0
for i, x in zip(frame['Extro'], frame['LE']):
    pred = str(i)
    label = str(x)
    if pred == label:
        ext += 1
for i, x in zip(frame['Agree'], frame['LA']):
    pred = str(i)
    label = str(x)
    if pred == label:
        agree += 1
for i, x in zip(frame['Cons'], frame['LC']):
    pred = str(i)
    label = str(x)
    if pred == label:
        cons += 1

print('Akurasi Extrovert : ', ext/bawah)
print('Akurasi Agreeableness : ', agree/bawah)
print('Akurasi Consciouness : ', cons/bawah)
```

```
Akurasi Extrovert : 0.319672131147541
Akurasi Agreeableness : 0.3360655737704918
Akurasi Consciouness : 0.00819672131147541
```

Gambar 14 Akurasi

Akurasi Perbandingan dengan BFI users

```
In [9]: ► frame2 = pd.read_csv('BFI_users.csv')
pd.set_option('display.max_colwidth', -1)
frame2
```

```
Out[9]:
```

	username	Extroversion	Agreeableness	Consciouness	LE	LA	LC
0	h3llatrash	1	0	#	0	0	1
1	azharizkita	1	0	#	0	0	1
2	jooshpn	0	0	#	0	1	1
3	yaelahir	1	#	#	0	1	0
4	azwardfauzan	#	#	#	0	1	1
...
283	rahmanizar_	1	#	#	0	0	1
284	prialitaf	#	0	#	0	1	0
285	innocentpep	#	0	#	1	1	0
286	evaxevi	#	#	#	1	1	1
287	ekkayuliana	1	1	#	1	1	1

288 rows × 7 columns

Gambar 15. Hasil Klasifikasi BFI User

```
In [12]: ▮ bawah = len(frame2['Extroversion']) #sebagai pembanding
ext = 0
agree = 0
cons = 0
for i, x in zip(frame2['Extroversion'], frame2['LE']):
    pred = str(i)
    label = str(x)
    if pred == label:
        ext += 1
for i, x in zip(frame2['Agreeableness'], frame2['LA']):
    pred = str(i)
    label = str(x)
    if pred == label:
        agree += 1
for i, x in zip(frame2['Consciouness'], frame2['LC']):
    pred = str(i)
    label = str(x)
    if pred == label:
        cons += 1

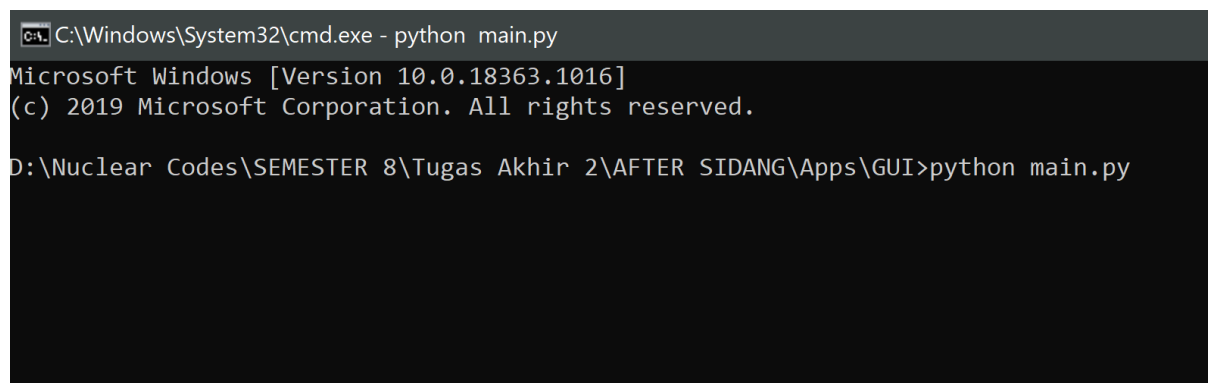
print('Akurasi Extrovert : ',ext/bawah)
print('Akurasi Agreeableness : ',agree/bawah)
print('Akurasi Consciouness : ',cons/bawah)

Akurasi Extrovert :  0.23263888888888889
Akurasi Agreeableness :  0.10763888888888889
Akurasi Consciouness :  0.03472222222222224
```

Gambar 16 Akurasi BFI Users

5. GUI

Pada penelitian ini juga menyediakan tampilan dengan GUI yang interaktif yang memudahkan pengguna untuk berinteraksi dengan sistem klasifikasi kepribadian berbasis sentimen twitter dengan algoritma PbSC sebagai metode klasifikasi utama yang dilakukan. Untuk menjalankan GUI ini buka folder GUI dan arahkan directory command prompt kedalam folder ini dan jalan kan file main.py dengan mengetik 'python main.py'



```
C:\Windows\System32\cmd.exe - python main.py
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Nuclear Codes\SEMESTER 8\Tugas Akhir 2\AFTER SIDANG\Apps\GUI>python main.py
```

Gambar 17 Openning GUI

Setelah anda berhasil membukanya maka akan membuka aplikasi GUI. Jika terjadi error maka pastikan semua directory sudah terinstall.



Gambar 18 GUI PBSC

Diatas merupakan tampilan dari aplikasinya. Anda dapat memasukkan *username twitter* dari *target user* untuk melihat tingkat kepribadiannya. Masukkan *username*nya dalam kasus ini saya akan memasukkan username akun twitter saya yaitu @joosnpn pada text input *Username Twitter*, Setelah dimasukkan maka tekan tombol predict. Pastikan anda terkoneksi ke internet dan usernamenya sudah benar, aplikasi akan melakukan semua proses diatas tadi seperti pengumpulan 450 *tweet* terbaru secara *real-time*, *preprocessing* dan klasifikasi. Jika tidak ada yang error maka hasil yang didapatkan bisa dilihat di gambar 19. Dapat dilihat kolom dimensi *extroversion*, *agreeableness* dan *conscientiousness* sudah terisi dengan kelas klasifikasi yang diprediksi dengan metode PbSC.

Hello Python

Prediksi Kepribadian dengan Algoritma PbSC

Username Twitter

Berikut hasil kepribadian mu @jooshpn

Extroversion

Agreeableness

Conscientiousness

Gambar 19 Hasil Klasifikasi Aplikasi PbSC

Jika *username* salah maka akan ditampilkan seperti gambar 20. Salah dalam artian username tersebut tidak pernah terdaftar di twitter.

Hello Python

Prediksi Kepribadian dengan Algoritma PbSC

Username Twitter

@jooshpnX tidak berhasil di crawl
pastikan username sudah benar dan akun tidak di private
atau periksa koneksi internet anda

Extroversion

Agreeableness

Conscientiousness

Gambar 20 Error