**file main.py**

di sini menampilkan menu menggunakan package simple-term-menu dan juga menggunakan looping untuk memilih menjalankan script scrapper comment instagram, Youtube, dan juga analisis.

**INSTAGRAM**

untuk code instagram comment scrapper menggunakan package instaloader, fungsi scrapper juga di bagi 3 yaitu :

1. List User(ig-lists-craper.py)

Maksudnya adalah kita membuat wordlist username instagram yang akan kita scraping semua post commentnya.

/\* code membutuhkan akses login instagram jika bisa gunakan akun smurf \*/

user\_login = input("Username Login : ")

user\_pass = getpass.getpass("Password Login : ")

L = instaloader.Instaloader(max\_connection\_attempts=0)

L.login(user\_login, user\_pass)

/\* code untuk membaca file wordlist \*/

data\_json\_file = ()

with open("userlist.json", 'r') as f:

data\_json\_file = json.loads(f.read())

/\* code untuk mengambil semua comment yang ada di post username di wordlist menggunakan bantuan looping \*/

for loop\_userlist in data\_json\_file:

count = 1

profile = instaloader.Profile.from\_username(L.context, loop\_userlist)

usernamelist = []

captionlist = []

hashtaglist = []

likeslist = []

commentlist = []

followerlist = []

json\_data = []

for post in profile.get\_posts():

print("Data dari " + loop\_userlist + ", postingan ke " + str(count) + " dari " + str(profile.mediacount) + ".")

caption = post.caption

if caption is None:

caption = ""

if caption is not None:

caption = caption.encode('ascii', 'ignore').decode('ascii')

hashtag = post.caption\_hashtags

likes = post.likes

comments = []

for comment in post.get\_comments() :

comments.append(comment.text.encode('ascii', 'ignore').decode('ascii'))

usernamelist.append(loop\_userlist)

captionlist.append(caption)

hashtaglist.append(hashtag)

likeslist.append(likes)

commentlist.append(comments)

count = count+1

/\* code akan membuat Path folder bernama ‘hasil’ untuk menyimpan output ke csv file \*/

pd.options.display.max\_colwidth = 100

data = pd.DataFrame({"Account":usernamelist, "Post":captionlist, "Tag":hashtaglist, "Likes":likeslist, "Comments":commentlist})

timestring = time.strftime("%Y%m%d\_%H%M%S")

path =r'hasil/'

nama\_file = os.path.join(path, "Dataset\_" + loop\_userlist + "\_" + timestring + ".csv")

data.to\_csv(nama\_file)

print("Dataset\_" + loop\_userlist + "\_" + timestring + ".csv")

print("DONE!!! Check csv file di Directori hasil")

1. Post User Scraper(ig-post-scraper.py)

Code ini menscrape comment satu link post yang di input oleh user, code disini sama dengan List User perbedaanya user di minta menginput username dan IDlink postingan yang akan di scraping melalui terminal.

/\* User input username dan IDlink postingan username \*/

user\_name = input("Username : ")

postingan = input("Id Postingan : ")

/\* syntax username \*/

username = user\_name

/\* Membaca post menggunakan package instaloader \*/

profile = instaloader.Profile.from\_username(L.context, username)

/\* Mengambil data yang ada pada IDlink post yang telah di inputkan \*/

usernamelist = [None]

captionlist = [None]

hashtaglist = [None]

likeslist = [None]

commentlist = [None]

followerlist = [None]

shortcode = [None]

for post in profile.get\_posts():

post = post.from\_shortcode(L.context, postingan)

print("Data dari " + username + "(" + postingan + ")")

caption = post.caption

if caption is None:

caption = ""

if caption is not None:

caption = caption.encode('ascii', 'ignore').decode('ascii')

hashtag = post.caption\_hashtags

likes = post.likes

comments = []

for comment in post.get\_comments() :

comments.append(comment.text.encode('ascii', 'ignore').decode('ascii'))

usernamelist.append(username)

shortcode.append(postingan)

captionlist.append(caption)

hashtaglist.append(hashtag)

likeslist.append(likes)

commentlist.append(comments)

break

1. User Scrapper(ig-user-scraper.py)  
   Code ini mengambil semua Post yang ada pada username yang di inputkan oleh user perbedaannya adalah

/\* Code hanya mengambil inputan username dari user lalu menscrappingnya \*/

user\_name = input("Username : ")

username = user\_name

**YOUTUBE(youtube.py)**

Di sini Code menggunakan Open Api yang telah disediakan oleh Google Api.

Data yang didapatkan selanjutnya dapat dianalisis untuk mengetahui performa dan aktivitas dari video/channel tertentu.

Untuk mendapaktan Api key bisa dilihat dalam file petunjuk.txt

/\* Membuat Folder Path Bernama Hasil Lalu menyimpan Output Ke dalam file txt \*/

stdoutOrigin=sys.stdout

filename = 'log\_youtube\_'+user\_input+'.txt'

sys.stdout = open(os.path.join('hasil/', filename), "w")

/\* User Diminta Memasukan Youtube Video Link ID \*/

user\_input = input("Input Video ID (https://www.youtube.com/watch?v=VIDEO\_ID): ")

/\* Code Membaca API KEY Dan Membaca Inputan user \*/

URL = 'https://www.googleapis.com/youtube/v3/'

#Masukan API KEY di Sini

API\_KEY = 'DISINI'

VIDEO\_ID = (user\_input)

youtube = build('youtube', 'v3', developerKey=API\_KEY)

/\* Code Print Output \*/

results = youtube.videos().list(id=VIDEO\_ID, part='snippet,statistics').execute()

for result in results.get('items', []):

videoID = print ('Video ID = ' + result['id'])

line = print ('---------------------------------------------------')

publish = print ('Publish = ' + result['snippet']['publishedAt'])

line = print ('---------------------------------------------------')

views = print ('Views = ' + result['statistics']['viewCount'])

line = print ('---------------------------------------------------')

likes = print ('Likes = ' + result['statistics']['likeCount'])

line = print ('---------------------------------------------------')

#print ('Disliked :' + result['statistics']['dislikeCount'])

judul = print ('Judul = ' + result['snippet']['title'])

line = print ('---------------------------------------------------')

deskripsi = print ('Deskripsi =\n'+ result['snippet']['description'])

line = print ('---------------------------------------------------')

comment = print ('COMENT = ' + result['statistics']['commentCount'])

line = print ('---------------------------------------------------')

/\* Code Mulai Membaca Dan Mengambil Comment,Reply,Dan Data Statistik Dari Video ID \*/

def print\_video\_comment(video\_id, next\_page\_token):

params = {

'key': API\_KEY,

'part': 'snippet',

'videoId': video\_id,

'order': 'relevance',

'textFormat': 'plaintext',

}

if next\_page\_token is not None:

params['pageToken'] = next\_page\_token

response = requests.get(URL + 'commentThreads', params=params)

resource = response.json()

for comment\_info in resource['items']:

user\_name = print("Username :",comment\_info['snippet']['topLevelComment']['snippet']['authorDisplayName'])

text = print("Comment :",comment\_info['snippet']['topLevelComment']['snippet']['textDisplay'])

like\_cnt = print("Comment Like :",comment\_info['snippet']['topLevelComment']['snippet']['likeCount'])

reply\_cnt = print("Total Reply :",comment\_info['snippet']['totalReplyCount'])

parentId = comment\_info['snippet']['topLevelComment']['id']

print\_video\_reply(video\_id, next\_page\_token, parentId)

print('===========================================================')

if 'nextPageToken' in resource:

print\_video\_comment(video\_id, resource["nextPageToken"])

def print\_video\_reply(video\_id, next\_page\_token, id):

params = {

'key': API\_KEY,

'part': 'snippet',

'videoId': video\_id,

'textFormat': 'plaintext',

'parentId': id,

}

if next\_page\_token is not None:

params['pageToken'] = next\_page\_token

response = requests.get(URL + 'comments', params=params)

resource = response.json()

for comment\_info in resource['items']:

print("==>")

user\_name = print("Username Reply :",comment\_info['snippet']['authorDisplayName'])

text = print("Reply :",comment\_info['snippet']['textDisplay'])

like\_cnt = print("Reply Like :",comment\_info['snippet']['likeCount'])

if 'nextPageToken' in resource:

print\_video\_reply(video\_id, resource["nextPageToken"], id)

**ANALISIS(analisis.py)**

Code Menggunakan Package NLTK untuk Stopwords dan Word Tokenize dan Tabel Diagram Menggunakan Package matplotlib

/\* user diminta menginputkan nama file yang akan di analisis \*/

filename = input('Enter a filename: ')

path = r'hasil/'

f = open(os.path.join(path, filename))

/\* Membaca File yang di inputkan user \*/

comment = f.read()

/\* Mode 1 Mengambil 20 kata yang sering keluar di comment Dan Membatasi Kata-Kata Ambigu dengan namafile mode1 ,jika ingin menambahkan stopwords kata-kata bisa di tambahkan di file tersebut \*/

word\_tokens = word\_tokenize(comment)

stop\_words = set(stopwords.words('mode1'))

word\_tokens\_no\_stopwords = [w for w in word\_tokens if not w in stop\_words]

freq\_kata\_2 = nltk.FreqDist(word\_tokens\_no\_stopwords)

print(freq\_kata\_2.most\_common(20))

freq\_kata\_2.plot(20)

plt.show()

/\* Mode 2 Mengambil Kata-Kata Yang hanya dibuat oleh user di dalam file analisis.py \*/

word\_tokens = word\_tokenize(comment)

stop\_words1 = ['bagus', 'sekali', 'enak', 'banget', 'tidak', 'suka', 'recomen'] #tambah di sini

word\_tokens\_no\_stopwords1 = dict([(match, len([w for w in word\_tokens if match in w])) for match in stop\_words1])

freq\_kata\_1 = nltk.FreqDist(word\_tokens\_no\_stopwords1)

print(freq\_kata\_1.most\_common())

freq\_kata\_1.plot()

plt.show()

**Sekian Laporan Youtube Dan Instagram Scrappernya**

**Happy Scrapping!!!**