

**LAPORAN PRAKTIKUM  
SOCIAL MEDIA MINING  
#pilpres2024**



**Oleh Kelompok 5 :**

- |                                     |                |
|-------------------------------------|----------------|
| 1. Ilham Ahmad Kamil                | (162012133019) |
| 2. Dheandra Azzahra Triyasnanda     | (162012133028) |
| 3. Melodi Aurellia Permata Siswanto | (162012133049) |
| 4. Ignatius Arvantya Hartono        | (162012133061) |

**MATA KULIAH  
DATA MINING II SD-A1  
PROGRAM STUDI TEKNOLOGI SAINS DATA  
FAKULTAS TEKNOLOGI MAJU DAN MULTIDISIPLIN  
UNIVERSITAS AIRLANGGA  
2022**

## **1. Latar Belakang**

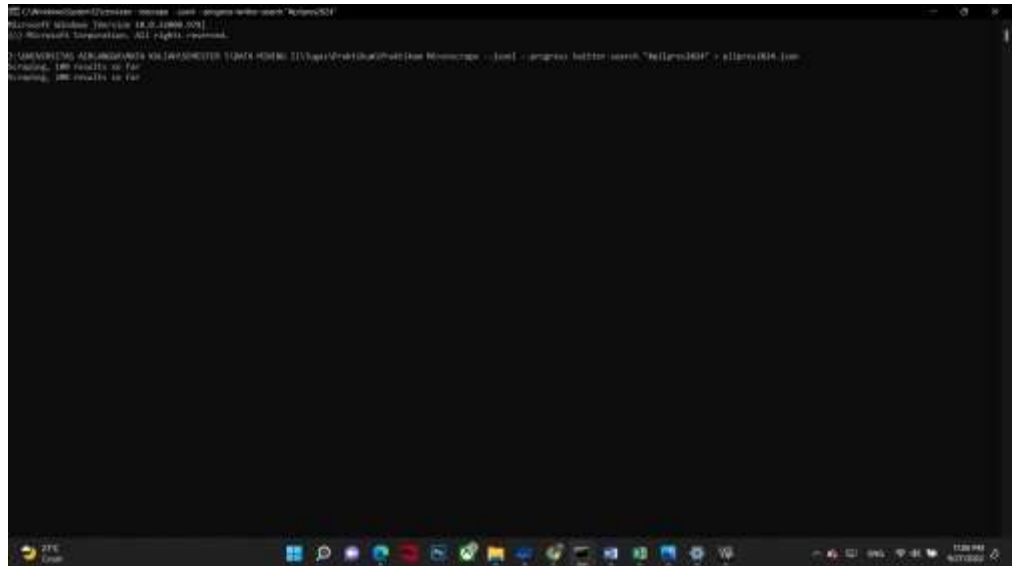
Indonesia merupakan negara yang menganut sistem demokrasi. Dalam hal ini didukung dengan adanya pemilu guna mencari seorang kepala negara. Di Indonesia, pemilu biasanya diselenggarakan secara serentak lima tahun sekali (Hidayatillah, 2019). Pada tahun 2024 ini dilakukan pemilihan presiden Indonesia. Sebagai seorang politikus yang ingin maju mencalonkan sebagai presiden Indonesia, tentu akan mempertimbangkan tingkat popularitas berdasarkan tweet yang ada dalam aplikasi Twitter. Salah satu media yang digunakan oleh masyarakat di Indonesia untuk mendapatkan informasi adalah media sosial, salah satunya adalah Twitter dengan jumlah pengguna harian Twitter secara global terus meningkat (Clinton, 2019). Pada awalnya Twitter hanya digunakan sebagai jejaring pertemanan saja, dengan seiring berjalannya waktu, Twitter telah mengalami pergeseran sebagai sarana interaksi untuk mempengaruhi orang lain atau berpolitik dan berguna memberikan informasi kegiatan para tokoh politik atau publik. Setiap pengguna Twitter bebas mengunggah posting-an atau komentar tanpa ada batasan. Dengan keberadaan interaksi ini akan memberi efek interaksi sosial dan membentuk semacam jaringan pertemanan.

Pada umumnya posting-an di Twitter digunakan untuk mengunggah perihal tentang diri pengguna dan berbagi informasi, serta penyampaian sebuah berita. Isi dari posting-an juga dapat mengungkapkan sebuah perasaan dari pengguna. Oleh karena itu dibutuhkan sebuah analisis yang tepat, guna mengolah data teks sehingga menyuguhkan informasi yang berharga mengenai opini seseorang terhadap tokoh politik atau publik. Kumpulan komentar berupa teks tersebut dikumpulkan dan dapat diolah menggunakan analisis sentimen, yaitu mempelajari opini atau pendapat yang menyatakan atau mengungkapkan persepsi yang mengandung sentimen positif atau negatif. Pendapat yang dibutuhkan untuk melakukan analisis bersumber dari komentar halaman Twitter yang memiliki tagar #pilpres2024.

Dengan adanya komentar-komentar yang membahas tentang pemilihan presiden 2024, kami akan melihat interaksi antara komen atau tweet yang dibuat oleh akun yang tersedia di Twitter. Maka kami menggunakan social data mining untuk melihat interaksi-interaksi yang ada dalam pilpres 2024 hastag di twitter.

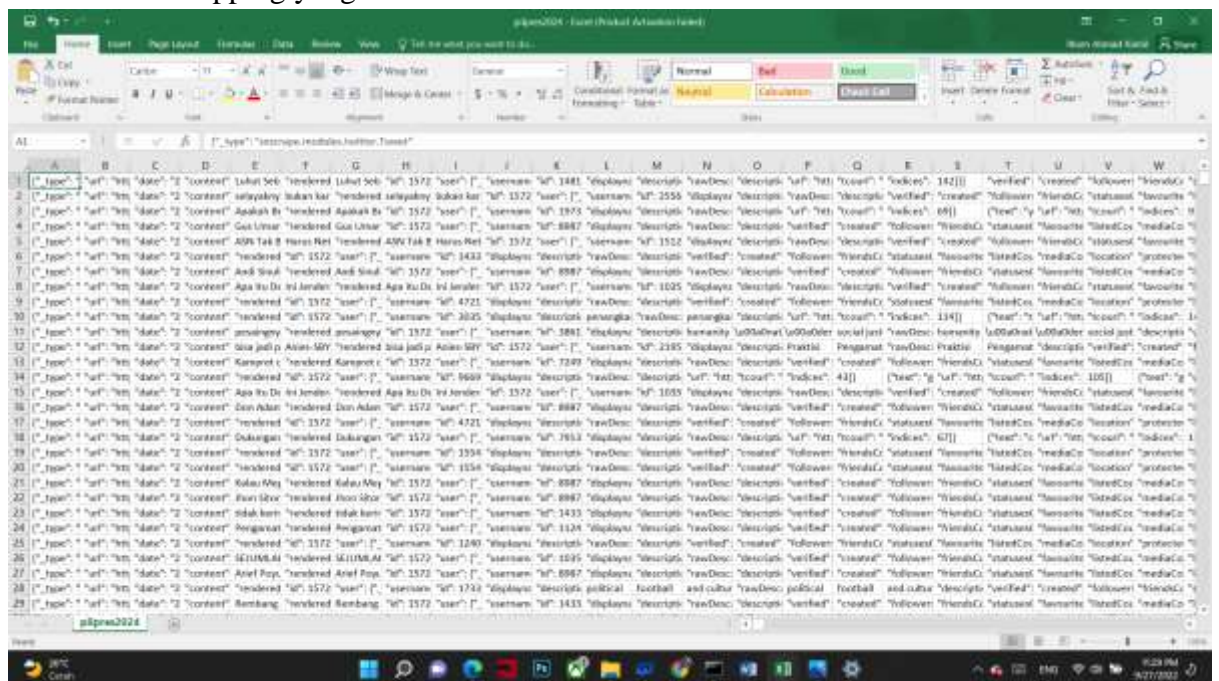
## **2. Hasil Praktikum**

### **a. Data**

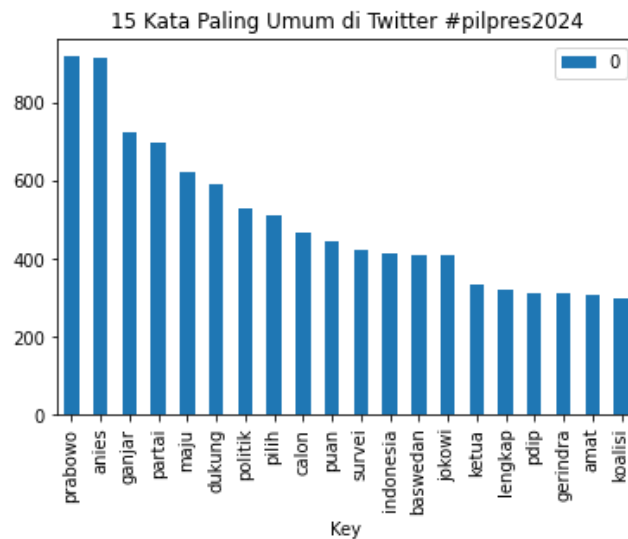


### Gambar 1. Scarpping Data

Melakukan scrapping data dengan hastag #pilpres2024 dengan menggunakan snsrape seperti diatas dan dihasilkan 5907 data yang dihasilkan dari scrapping yang dilakukan.



### b. Wordcloud



### Gambar 3. Visualisasi 15 Kata Umum

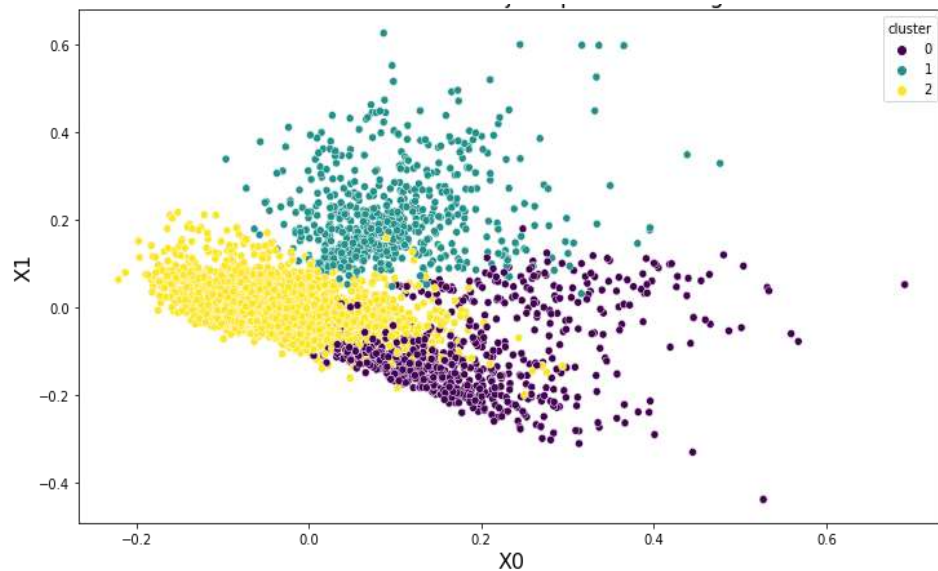
Hasil yang didapatkan untuk kata yang paling umum di twitter dalam hastag #pilpres2024 kata calon presiden yang umum dibicarakan antara lain : prabowo serta anies yang memiliki frekuensi kemunculan paling tinggi lalu disusul dengan ganjar dan partai.



#### Gambar 4. WordCloud

Hasil wordcloud terlihat bahwa kata-kata yang memiliki frekuensi besar diantaranya adalah : “Prabowo”, “Anies”, “Ganjar”. Sehingga dari hasil tersebut dapat diasumsikan bahwa di dalam topik tentang pemilihan presiden 2024, kata yang paling sering disebut di ranah twitter adalah Prabowo serta Anies.

### c. Clustering



Gambar 5. Visualisasi Clustering

Dari hasil clustering dengan TF-IDF dan Kmeans didapatkan hasil 3 cluster. Cluster 0 memiliki tingkat terendah, diikuti cluster 2 dan cluster 1 untuk yang tertinggi.

### d. Centrality Measures

```
degreecentral = nx.degree centrality(mentionnet)
```

```
[19] dict(sorted(degreecentral.items(), key=lambda item: item[1]))
```

```
{ '@maghnaul_ulum': 0.0008110300081103001,
```

```
 '@Iaro_Rajagukguk': 0.0008110300081103001,
```

```
 '@Triyogo22684337': 0.0008110300081103001,
```

```
 '@PadmavatiPrati1': 0.0008110300081103001,
```

```
 '@PartaiPerindo': 0.0008110300081103001,
```

```
 '@MediaCellPPP': 0.0008110300081103001,
```

```
 '@benediktus2302': 0.0008110300081103001,
```

```
 '@KitaSemuaJokowi': 0.0008110300081103001,
```

```
 '@rethiesri': 0.0008110300081103001,
```

```
 '@partainasdem': 0.0008110300081103001,
```

```
 '@BennyHarmanID': 0.0008110300081103001,
```

```
 '@BeEm_TOENTAK16': 0.0008110300081103001,
```

```
 '@Ganjar4RI': 0.0008110300081103001,
```

```
 '@RiuRizkiUtomo': 0.0008110300081103001,
```

```
 '@kangsobary': 0.0008110300081103001,
```

```
 '@_MbakSri_': 0.0008110300081103001,
```

```
 '@g_politik2024': 0.0008110300081103001,
```

```
 '@FerdinandHutah4': 0.0008110300081103001,
```

```
 '@Cucu4Anak4': 0.0008110300081103001,
```

```
 '@purwanto010170': 0.0008110300081103001,
```

Gambar 6. Centrality Measures

Hasil centrality measures, beberapa akun didapatkan memiliki nilai centrality terkecil, yaitu sebesar 0.000811. Diikuti nilai centrality sebesar 0.00162 dan seterusnya. nilai centrality yang kecil menandakan kecilnya signifikansi data terhadap graf.

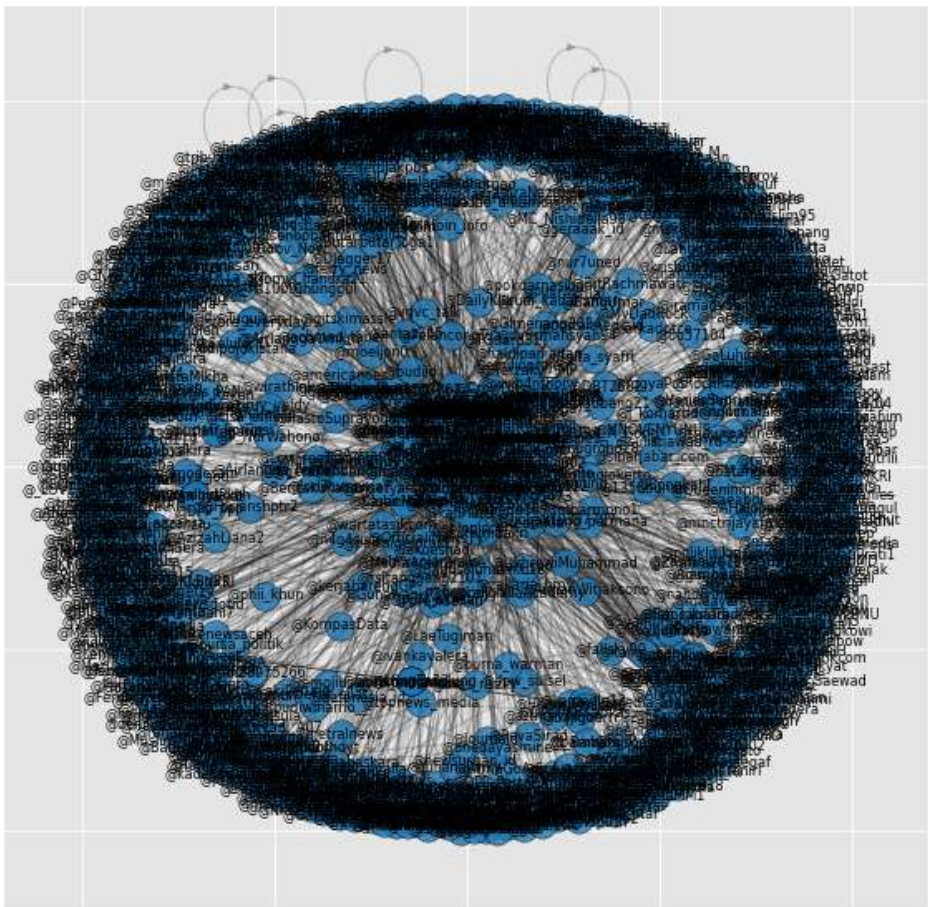


Untuk nilai degree sendiri didapatkan hasil seperti berikut dengan urutan terbesar sampai terkecil.

Id	Label	Interval	In-Degree	Out-Degree	Degree
@BeritaDekhoCo...	@BeritaDekhoCo...		0	103	103
@HulkifliMarbun	@HulkifliMarbun		0	101	101
@aniesbaswedan	@aniesbaswedan		94	0	94
@ganjarpranowo	@ganjarpranowo		81	0	81
@prabowo	@prabowo		68	0	68
@nyapres2024	@nyapres2024		0	67	67
@bob_bay	@bob_bay		0	64	64
@jokowi	@jokowi		49	0	49
@cakimiNOW	@cakimiNOW		48	0	48
@jpnncom	@jpnncom		17	27	44

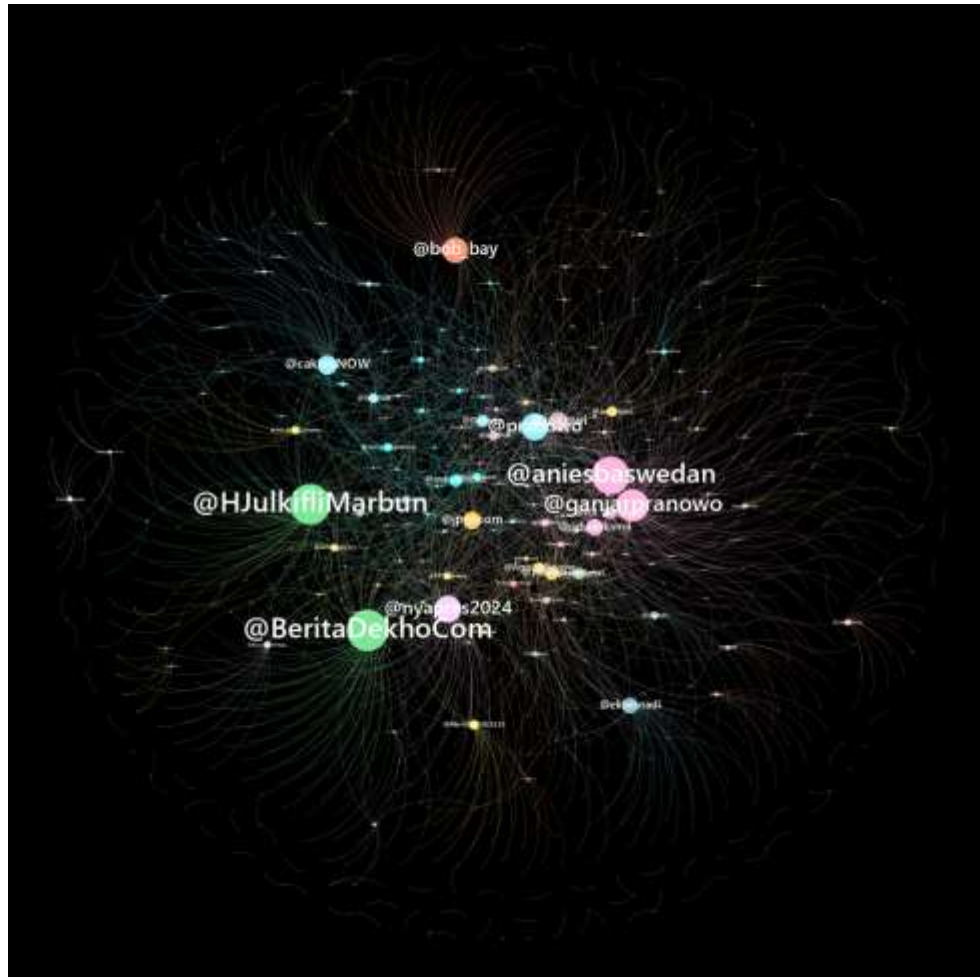
Gambar 7. Hasil Data Laboratory

e. Visualisasi



Gambar 8. Social Network

Hasil visualisasi social network tersebut masih kurang dapat terlihat jelas sehingga digunakan bantuan gephi untuk membuat visualisasi yang lebih jelas dan bagus.



Gambar 9. Social Network

Didapatkan hasil dari visualisasi Fruchterman Reingold layout bahwa dari hastag #pilpres2024 yang memiliki degree tertinggi secara keseluruhan dari analisis sosial media menggunakan data twitter yaitu ada @BeritaDekhoCom dan @HjulkifliMarbun yang menempati posisi pertama dan kedua. Terdapat hasil yang menarik juga bahwa akun twitter para kemungkinan calon presiden yang memiliki degree tertinggi yaitu @aniesbaswedan terbesar pertama, @ganjarpranowo terbesar kedua dan ketiga @ridwankamil dengan jarak node yang berdekatan. Selain itu, terdapat beberapa akun twitter yang cukup tinggi degree-nya antara lain @nyapres2024, @prabowo, @jokowi, @cakimiNOW, dan @bob\_bay.

Id	Label	Interval	In-Degree	Out-Degree	Degree
@BeritaDekhoCo...	@BeritaDekhoCo...		0	103	103
@HjulkifliMarbun	@HjulkifliMarbun		0	101	101
@aniesbaswedan	@aniesbaswedan		94	0	94
@ganjarpranowo	@ganjarpranowo		81	0	81
@prabowo	@prabowo		68	0	68
@nyapres2024	@nyapres2024		0	67	67
@bob_bay	@bob_bay		0	64	64
@jokowi	@jokowi		49	0	49
@cakimiNOW	@cakimiNOW		48	0	48
@jpnncom	@jpnncom		17	27	44

Gambar 10. Hasil Data Laboratory

Hasil visualisasi tersebut juga didukung dengan hasil data laboratory diatas dengan 10 data degree tertinggi yang dibuktikan bahwa @BeritaDekhoCom menempati posisi pertama dengan 103 degree , @HjulkifliMarbun dengan 101 degree, @aniesbaswedan dengan 94 degree, dst.

#### f. Kesimpulan

Kesimpulan yang didapatkan adalah pemantik diskusi yang paling besar dalam hastag #pilpres2024 yaitu @BeritaDekhoCom dan @HjulkifliMarbun yang mana nama atau kata yang sering disebut dalam #pilpres2024 antara lain prabowo, anies, dan ganjar dengan urutan paling besar sampai paling kecil. Dapat dilihat juga bahwa @anisbaswedan, @ganjarpranowo, dan @ridwan kamil masih dalam kubu yang sama untuk data ini sedangkan untuk @prabowo dan @cakimiNOW masuk dalam kubu berlawanan dari ketiga tokoh tersebut dan mereka satu kubu hal tersebut memungkinkan juga karena banyak berita bahwa mereka koalisi. Analisis yang kami lakukan menggunakan layout Fruchterman Reingold yang membentuk ring bulat.

#### Daftar Pustaka

Hidayatillah, R., Mirwan, M., Hakam, M., & Nugroho, A. (2019). Levels of Political Participation Based on Naive Bayes Classifier. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 13(1), 73-82.

B. Clinton, "Pengguna Aktif Harian Twitter Indonesia Di Klaim Terbanyak,"

Kompas.com, 2019. [Online]. URL:

<https://tekno.kompas.com/read/2019/10/30/16062477/pengguna-aktif-harian-twitter-indonesia-diklaim-terbanyak> (Diakses pada 26 September 2022).



## LAMPIRAN

### 1. Cleaning Data

```
import json
import pandas as pd
data = [json.loads(line) for line in open('pilpres2024.json', 'r')]
content, username = [], []
for i in data:
    content.append(i['content'])
    username.append(i['user']['username'])
```

```
df = pd.DataFrame({
    "content": content,
    "username": username
})
df
```

	content	username
0	Bismillah Allahumma Cak Imin Presiden 2024\n\n...	SelviEvhy4
1	#Pilpres2024 #LuhutBinsarPandjaitan\nPilpres 2...	PostGorontalo
2	@Mdy_Asmara1701 Dalam pilpres, selayaknya ada...	maqnaul_ulum
3	Hasto Kristiyanto: "Dewan Kolonel" Guyonan Po...	kompastv_jabar
4	Ade Armando Pertanyakan Kinerja Puan Maharani,...	fajaronline
...	...	...
5898	Nih, #puisi #Pilpres2024 dari saya di http://t...	Gol_A_Gong
5899	"@detikcom: Salam Dua Jari, Slank Ajak Warga B...	susansut13
5900	Cpt amt Dôgôj thn 2024 RT @doni_dee: Debat ca...	YulianaPinky
5901	Debat capres 2014 now on metrotv #pilpres2024 ...	doni_dee
5902	rumput tetangga memang hijau, maka dari itu ma...	anggitsatrio

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import re
import nltk
import string
```

```
def text_preproc(x):
    x = str(x).lower() # Huruf kecil teks
    x = x.encode('ascii', 'ignore').decode() # Hapus karakter unicode
    x = re.sub(r'https*\S+', ' ', x) # Hapus URL
    x = re.sub(r'@\S+', ' ', x) # Hapus sebutan
    x = re.sub(r'#\S+', ' ', x) # Hapus Tagar
    x = re.sub(r'pilpres', ' ', x) # Hapus pilpres
    x = re.sub(r'presiden', ' ', x) # Hapus presiden
    x = re.sub(r'capres', ' ', x) # Hapus capres
    x = re.sub(r'\'w+', ' ', x) # Hapus centang dan karakter berikutnya
    x = re.sub(r'[\'s]': '% re.escape(string.punctuation)', ' ', x) # Hapus tanda baca
    x = re.sub(r'\'w*\d+\'w+', ' ', x) # Hapus nomor
    x = re.sub(r'\'s[2,]', ' ', x) # Ganti spasi
    return x
text = df.content.apply(text_preproc)
```

```
for i in text:
    print (i)
```

Python

Output exceeds the [size limit](#). Open the full output data [in a text editor](#).

bismillah allahumma cak imin amin ya allah ya rabb alamin

luhut sebut non jawa mustahil jadi

dalam selayaknya adalah bagaimana memilih pemimpin indonesia yg membawa kesejahteraan dan keadilan bagi rakyatnya bukan karna like atau dislike baru indonesia akan bangkit dan maju ayo menjadi pemilih cerdas

hasto kristiyanto dewan kolonel guyonan politik apakah benar

ade armando pertanyakan kinerja puan maharani gus umar emang prestasi ganjar apa ade ingat asn tak boleh macam macam saat pemilu harus netral baca selengkapnya

bismillah allahumma cak imin amin ya allah ya rabb alamin

don adam sebut prabowo dijegal dan andes dikriminalisasi andi sinulingga strateginya mau calon tunggal

trending apa itu dewan kolonel terungkap kuda puan maharani untuk ini jenderal nya lewat

```
from nltk.tokenize import word_tokenize
def text_token(token):
    token = word_tokenize(str(token))
    return token
token = text.apply(text_token)
token
```

```
for i in token:
    print (i)
```

Python

Output exceeds the [size limit](#). Open the full output data [in a text editor](#).

['bismillah', 'allahumma', 'cak', 'imin', 'amin', 'ya', 'allah', 'ya', 'rabb', 'alamin']

['luhut', 'sebut', 'non', 'jawa', 'mustahil', 'jadi']

['dalam', 'selayaknya', 'adalah', 'bagaimana', 'memilih', 'pemimpin', 'indonesia', 'yg', 'membawa', 'kesejahteraan', 'dan', 'keadilan', 'bagi', 'rakyatnya', 'bukan', 'karna', 'like', 'atau', 'dislike', 'baru', 'indonesia', 'akan', 'bangkit', 'dan', 'maju', 'ayo', 'menjadi', 'pemilih', 'cerdas']

['hasto', 'kristiyanto', 'dewan', 'kolonel', 'guyonan', 'politik', 'apakah', 'benar']

```
from nltk.corpus import stopwords
#tokenize text
list_stopwords = set(stopwords.words('Indonesian'))# get Indonesian stopwords

def text_sw(sw):
    tokens_without_stopword = [word for word in sw if not word in list_stopwords]
    return tokens_without_stopword
tokens_without_stopword = token.apply(text_sw)
tokens_without_stopword
```

Python

0 [bismillah, allahumma, cak, imin, amin, ya, al...

1 [luhut, non, jawa, mustahil]

2 [selayaknya, memilih, pemimpin, indonesia, yg,...

3 [hasto, kristiyanto, dewan, kolonel, guyonan, ...

4 [ade, armando, kinerja, puan, maharani, gus, u...

...

5898 [nih]

5899 [salam, jari, slank, ajak, warga, bandung, nga...

5900 [cpt, amt, thn, rt, debat, now, on, metrotv, l...

5901 [debat, now, on, metrotv, live]

5902 [rumpun, tetangga, hijau, mari, hijaukan, rump...

Name: content, length: 5903, dtype: object

# Stemming

```
# Import Sastrawi Package
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

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

# token without stopwords
for i in range(len(tokens_without_stopword)):
    for j in range(len(tokens_without_stopword[i])):
        stemming[i][j] = stemmer.stem(tokens_without_stopword[i][j])
stemming
```

```
0      [bismillah, allahumma, cak, amin, amin, ya, al...
1      [luhut, non, jawa, mustahil]
2      [layak, pilih, pimpin, indonesia, yg, bawa, se...
3      [basto, kristiyanto, dewan, kolonel, guyon, po...
4      [ade, armando, kerja, puan, maharani, gus, uma...
      ...
5898      [nib]
5899      [salam, jari, slank, ajak, warga, bandung, nga...
5900      [cpt, ant, tin, rt, debat, now, on, metrotv, l...
5901      [debat, now, on, metrotv, live]
5902      [rumput, tetangga, hijau, mari, hijau, rumput,...
Name: content, Length: 5903, dtype: object
```

```
words = []
for i in stemming:
    words.extend(i)

from nltk.probability import FreqDist
fdist = FreqDist(words)
fdist1 = dict(fdist.most_common(20))
fdist1
```

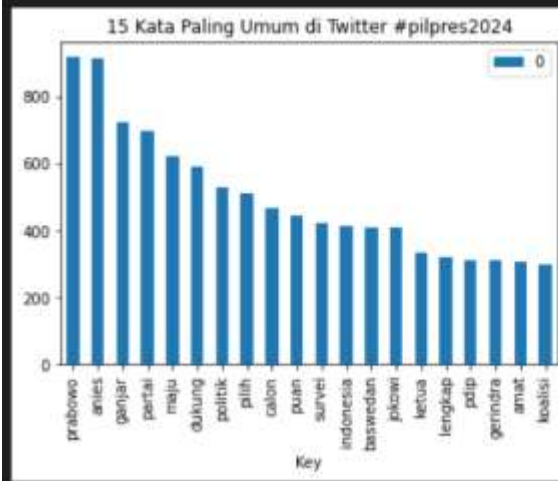
```
{'prabowo': 919,
 'anies': 914,
 'ganjar': 723,
 'partai': 699,
 'maju': 623,
 'dukung': 590,
 'politik': 529,
 'pilih': 511,
 'calon': 467,
 'puan': 443,
 'survei': 422,
 'indonesia': 412,
 'baswedan': 410,
 'jokowi': 407,
 'ketua': 331,
 'lengkap': 320,
 'pdip': 311,
 'gerindra': 309,
 'amat': 308,
 'koalisi': 296}
```

```
import matplotlib.pyplot as plt
df_freq_tokens = pd.DataFrame.from_dict(fdist1, orient='index')
df_freq_tokens.columns = ['Frequency']
df_freq_tokens.index.name = 'Key'
```

```
df_freq_tokens.plot(kind='bar')
plt.title("15 Kata Paling Umum di Twitter #pilpres2024")
plt.show()
```

<ipython-input-10-a6e4c263c42c>:3: UserWarning: Pandas doesn't allow created via a new attribute name - see <https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access>

```
df_freq_tokens.columns = ['Frequency']
```



## 2. Wordcloud

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

wordcloud = WordCloud(background_color="lavender").generate_from_frequencies(fdist)

plt.figure(figsize=(12,12))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Python



### 3. Clustering

```
def join(token):
    text = " ".join(token)
    return text
final_text = stemming.apply(join)
final_text

0      bismillah allahumma cak imin amin ya allah ya ...
1                                luhut non jawa mustahil
2      layak pilih pimpin indonesia yg bawa sejahtera...
3      hasto kristiyanto dewan kolonel guyon politik
4      ade armando kerja puan maharani gus umar emang...
...
5898                                nih
5899      salam jari slank ajak warga bandung ngabuburit...
5900                                cpt amt thn rt debat now on metrotv live
5901                                debat now on metrotv live
5902      rumput tetangga hijau mari hijau rumput gw
Name: content, Length: 5903, dtype: object
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

# initialize the vectorizer
vectorizer = TfidfVectorizer(sublinear_tf=True, min_df = 5, max_df = 0.95)
# fit_transform applies TF-IDF to clean texts - we save the array of vector in X
X = vectorizer.fit_transform(final_text)

from sklearn.cluster import KMeans

# initialize kmeans with 3 centroid
kmeans = KMeans(n_clusters=3, random_state=42)
# fit the model
kmeans.fit(X)
# store cluster label in a variable
cluster = kmeans.labels_

from sklearn.decomposition import PCA
import numpy as np

# initialize PCA with 2 components
pca = PCA(n_components= 2, random_state=42)
# pass our X to the PCA and store the reduced vector into pca_vecs
pca_vecs = pca.fit_transform(X.toarray())
# save our dimensions into x0 and x1
x0 = pca_vecs[:, 0]
x1 = pca_vecs[:, 1]

words = pd.DataFrame()

# assign clusters and pca vectors to our dataframe
words['cluster'] = cluster
words['x0'] = x0
words['x1'] = x1
```



words

	cluster	x0	x1
0	2	-0.091016	-0.009108
1	2	-0.055728	-0.038854
2	2	-0.091112	-0.007471
3	2	-0.078179	-0.022782
4	2	0.066156	-0.020948
...	...	...	...
5898	2	-0.067491	-0.031414
5899	2	-0.058564	-0.031180
5900	2	-0.067809	-0.031578
5901	2	-0.066252	-0.031241
5902	2	-0.060118	-0.030235

```
def get_top_keywords(n_terms):  
    """This function return the keywords for each centroid of the KMeans"""  
    df = pd.DataFrame(X.todense()).groupby(cluster).mean() # groups the TF-IDF  
  
    terms = vectorizer.get_feature_names() # acces tf-idf terms  
    for i,r in df.iterrows():  
        print('\ncluster {}'.format(i))  
        print(', '.join([terms[t] for t in np.argsort(r)[-n_terms:]])) #for e  
  
get_top_keywords(10)
```

Cluster 0

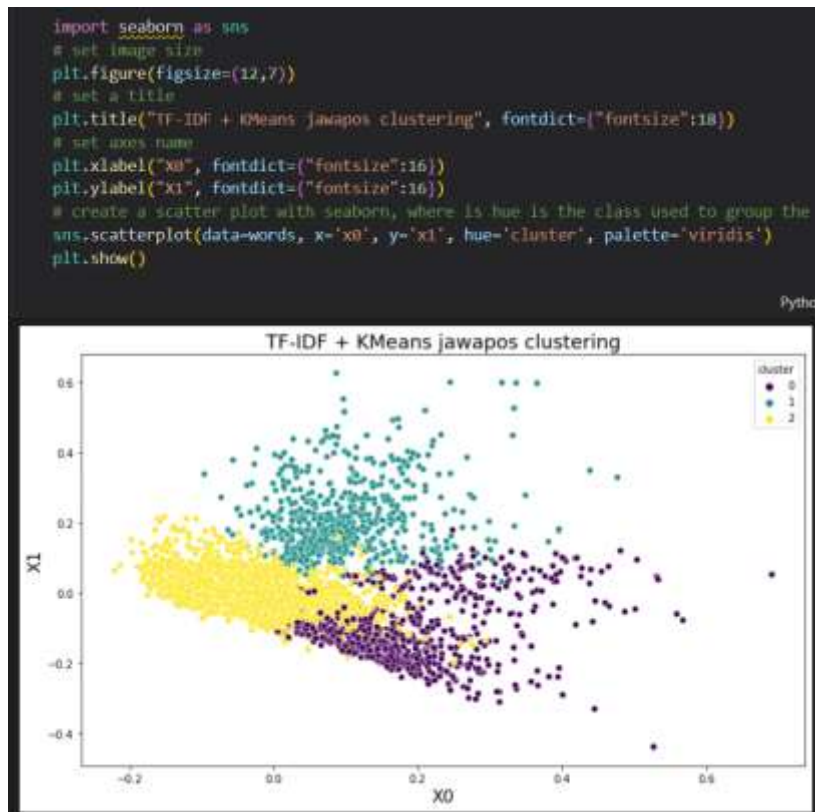
gubernur,puan,ahy,dki,dukung,duet,prabowo,ganjar,baswedan,anies

Cluster 1

survei,pasang,partai,calon,jokowi,puan,maju,gerindra,subianto,prabowo

Cluster 2

koalisi,jokowi,indonesia,calon,politik,pilih,maju,dukung,ganjar,partai



#### 4. Social Network

```
df = pd.read_json('pilpres2024.json', lines=True)
df.columns
```

Index(['\_type', 'url', 'date', 'content', 'renderedContent', 'id', 'user',  
'replyCount', 'retweetCount', 'likeCount', 'quoteCount',  
'conversationId', 'lang', 'source', 'sourceUrl', 'sourceLabel',  
'outlinks', 'tcooutlinks', 'media', 'retweetedTweet', 'quotedTweet',  
'inReplyToTweetId', 'inReplyToUser', 'mentionedUsers', 'coordinates',  
'place', 'hashtags', 'cashtags'],  
dtype='object')

```
scrap = df.drop(['_type', 'url', 'date', 'renderedContent', 'id',  
'replyCount', 'retweetCount', 'likeCount', 'quoteCount',  
'conversationId', 'lang', 'source', 'sourceUrl', 'sourceLabel',  
'outlinks', 'tcooutlinks', 'media', 'retweetedTweet', 'quotedTweet',  
'inReplyToTweetId', 'inReplyToUser', 'mentionedUsers', 'coordinates',  
'place', 'hashtags', 'cashtags'], axis=1)
scrap
```

Python

	content	user
0	Bismillah Allahumma Cak Imin Presiden 2024/n/n...	['_type': 'snsrape.modules.twitter.User', 'us...
1	#Pilpres2024 #LuhutBinsarPandaiar/n/nPilpres 2...	['_type': 'snsrape.modules.twitter.User', 'us...
2	@Mdy_Asmara1701 Dalam pilpres, selayaknya adal...	['_type': 'snsrape.modules.twitter.User', 'us...
3	Hasto Kristiyanto: "Dewan Kolonel" Guyonan Pol...	['_type': 'snsrape.modules.twitter.User', 'us...
4	Ade Armando Pertanyakan Kinerja Puan Maharani...	['_type': 'snsrape.modules.twitter.User', 'us...
...	...	...

```
listuser = []
for row in scrap['user']:
    listuser.append('@'+row['username'])
scrap['user'] = listuser
scrap
```

	content	user
0	Bismillah Allahumma Cak Imin Presiden 2024\n\n...	@SelviEvhy4
1	#Pilpres2024 #LuhutBinsarPandjaitan\nPilpres 2...	@PostGorontalo
2	@Mdy_Asmara1701 Dalam pilpres, selayaknya adal...	@magnaui_ukum
3	Hasto Kristiyanto: "Dewan Kolonel" Guyonan Pol...	@kompastv_jabar
4	Ade Armando Pertanyakan Kinerja Puan Maharani,...	@fajaronline
...	...	...
5898	Nih, #puisi #Pilpres2024 dari saya di http://t...	@Gol_A_Gong
5899	"@detikcom: Salam Dua Jari, Slank Ajak Warga B...	@susansut13
5900	Cpt amir D00j thn 2024 RT @doni_dee: Debat ca...	@YulianaPinky
5901	Debat capres 2014 now on metrotv #pilpres2024 ...	@doni_dee
5902	rumpun tetangga memang hijau, maka dari itu ma...	@anggitsatrio

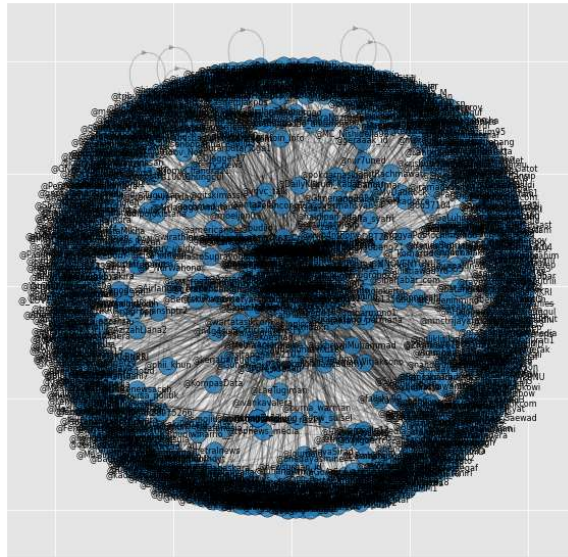
```
mentionnet = nx.DiGraph()

for row in scrap.iterrows():
    author = row[1]['user']
    text = row[1]['content']

    mentions = set(re.findall(r'@(\w+)', text))

    if len(mentions) > 0:
        for u in mentions:
            u = f'@{u}'
            mentionnet.add_edge(author, u)
```

```
# G = nx.from_pandas_edgelist(scrap, 'source', 'target')
pos = nx.spring_layout(mentionnet)
G=mentionnet
import matplotlib.pyplot as plt
f, ax = plt.subplots(figsize=(10,10))
plt.style.use('ggplot')
nodes = nx.draw_networkx_nodes(G, pos, alpha=0.8)
nodes.set_edgecolor('k')
nx.draw_networkx_labels(G, pos, font_size=8)
nx.draw_networkx_edges(G, pos, width=1.0, alpha=0.2)
```



```
dfexport = nx.to_pandas_edgelist(mentionnet)
dfexport.to_csv('mentions.csv')
```

## 5. Centrality Measure

```
degreecentral = nx.degree_centrality(mentionnet)
```

```
print(degreecentral)
```

Python

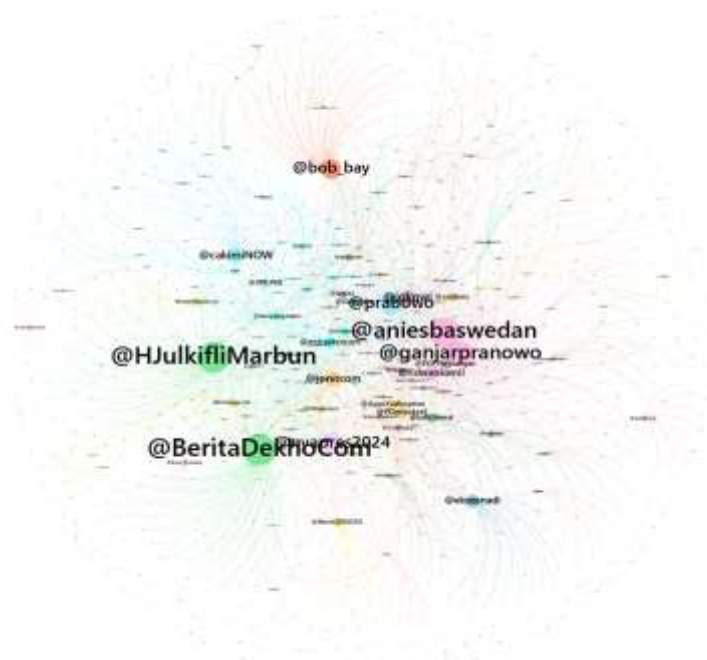
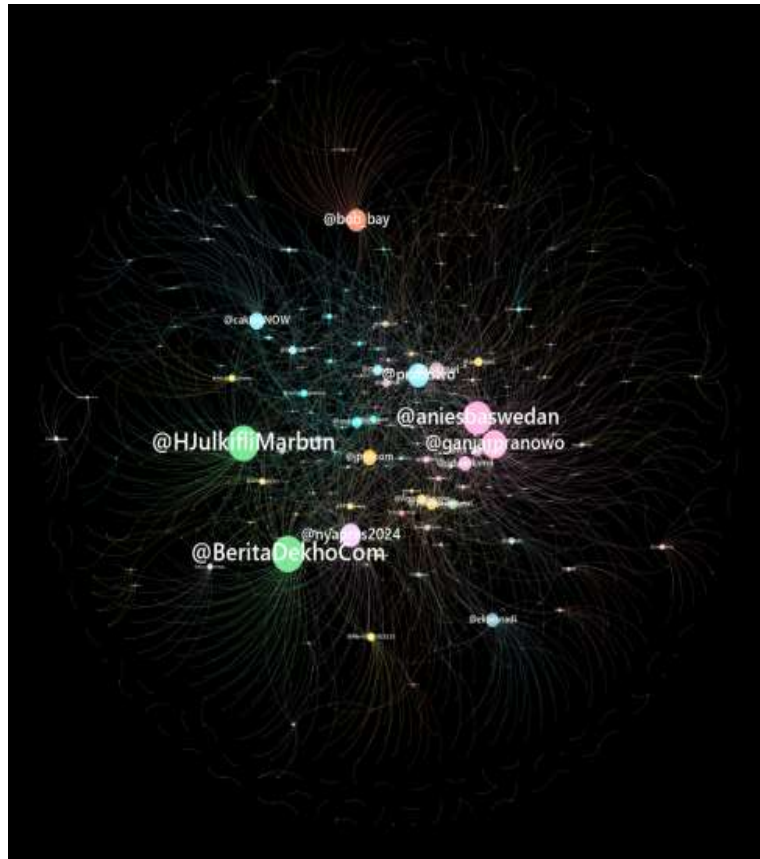
```
{ '@maqnaul_ulum': 0.0008110300081103001, '@Mdy_Asmara1701': 0.0024330900243309003,
  '@tribunkaltim': 0.0024330900243309003, '@at_Frederich18': 0.0016220600162206002,
  '@CNNIndonesia': 0.004866180048661801, '@elditweet': 0.004866180048661801,
  '@puanmaharani_ri': 0.012165450121654502, '@erickthohir': 0.015409570154095702,
  '@prabowo': 0.055150040551500405, '@ganjarpranowo': 0.06569343065693431,
  '@rsitanggang2101': 0.0016220600162206002, '@Haro_Rajagukguk': 0.0008110300081103001,
  '@Triyogo22684337': 0.0008110300081103001, '@rmp_officials': 0.0040551500405515,
  '@cakimiNOW': 0.038929440389294405, '@MartinS85825235': 0.019464720194647202,
  '@partainasdem': 0.0008110300081103001, '@rethiesri': 0.0008110300081103001,
```



```
dict(sorted(degreecentral.items(), key=lambda item: item[1]))
```

```
{ '@maqnaul_ulum': 0.0008110300081103001,
  '@Haro_Rajagukguk': 0.0008110300081103001,
  '@Triyogo22684337': 0.0008110300081103001,
  '@PadmavatiPrati1': 0.0008110300081103001,
  '@PartaiPerindo': 0.0008110300081103001,
  '@MediaCellPPP': 0.0008110300081103001,
  '@benediktus2302': 0.0008110300081103001,
  '@KitaSemuaJokowi': 0.0008110300081103001,
  '@rethiesri': 0.0008110300081103001,
  '@partainasdem': 0.0008110300081103001,
  '@BennyHarmanID': 0.0008110300081103001,
  '@BeEm_JOENTAK16': 0.0008110300081103001,
  '@Ganjar4RI': 0.0008110300081103001,
  '@RiuRizkiUtomo': 0.0008110300081103001,
  '@kangsobary': 0.0008110300081103001,
  '@_MbakSri_': 0.0008110300081103001,
  '@g_politik2024': 0.0008110300081103001,
```

## 6. Gephi





Hadi Data - Facebook Activation Rates																			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	M	Label	likeset	indigo	outdegree	Degree	facebook	disseminated	harmonious	betweenness	modularity	class							
2	@maspaul_uhm	@maspaul_uhm	0	1	1	1	1	1	1	1	0	1							
3	@MMy_Aurora1701	@MMy_Aurora1701	1	0	1	0	0	0	0	0	0	1							
4	@tribusakalm	@tribusakalm	1	2	3	1	1	1	1	1	0	0							
5	@Triken_Kalena	@Triken_Kalena	1	0	1	0	0	0	0	0	0	0							
6	@at_friedrich18	@at_friedrich18	0	2	2	1	1	1	1	1	0	0							
7	@CNIndonesia	@CNIndonesia	4	0	4	0	0	0	0	0	0	0							
8	@kompasTV	@kompasTV	4	0	4	0	0	0	0	0	0	0							
9	@tribusant	@tribusant	0	4	4	1	1	1	1	1	0	0							
10	@puanmaharati_1	@puanmaharati_1	11	0	11	0	0	0	0	0	0	1							
11	@sachiboke	@sachiboke	19	0	19	0	0	0	0	0	0	1							
12	@paulosm	@paulosm	64	0	64	0	0	0	0	0	0	1							
13	@pangiprenesno	@pangiprenesno	81	0	81	0	0	0	0	0	0	1							
14	@paulad	@paulad	2	0	2	0	0	0	0	0	0	0							
15	@trunggong1301	@trunggong1301	0	2	2	1	1	1	1	1	0	1							
16	@Hiro_Rajapagala	@Hiro_Rajapagala	1	0	1	0	0	0	0	0	0	1							
17	@Triyogo2284357	@Triyogo2284357	1	0	1	0	0	0	0	0	0	1							
18	@trp_official	@trp_official	3	2	5	1	1	1	1	1	0.000002	1							
19	@rubenNOW	@rubenNOW	44	0	44	0	0	0	0	0	0	1							
20	@FadyHadji	@FadyHadji	1	0	1	0	0	0	0	0	0	1							
21	@Martin58582335	@Martin58582335	0	24	24	1	1	1	1	1	0	1							
22	@partinawaden	@partinawaden	1	0	1	0	0	0	0	0	0	1							
23	@rethorn	@rethorn	1	0	1	0	0	0	0	0	0	1							
24	@yul_14	@yul_14	4	0	4	0	0	0	0	0	0	1							
25	@PESajihana	@PESajihana	20	0	20	0	0	0	0	0	0	1							
26	@FauziPKB	@FauziPKB	7	0	7	0	0	0	0	0	0	1							
27	@FantiGibbar	@FantiGibbar	7	0	7	0	0	0	0	0	0	1							
28	@FakhrulPrati3	@FakhrulPrati3	1	4	5	0	0	0	0	0	0	1							
29	@MedhaCulP79	@MedhaCulP79	1	0	1	0	0	0	0	0	0	1							