
- ```
In [6]: ▶ #@title Twitter Auth Token
twitter_auth_token = 'XX'
```
- ```
In [10]: ▶ ! node -v  
! npm install curl
```

```
filename = 'jkt48.csv'
search_keyword = 'jkt48 senbatsu sousenkyo lang:id'
limit = 500

!npx -y tweet-harvest@2.6.1 -o "{filename}" -s "{search_keyword}" --tab "LATEST" -l {limit} --token {twitter_auth_token}
```

```
Tweet Harvest [v2.6.1]

Research by Helmi Satria
Use it for Educational Purposes only!

This script uses Chromium Browser to crawl data from Twitter with your Twitter auth token.
Please enter your Twitter auth token when prompted.

Note: Keep your access token secret! Don't share it with anyone else.
Note: This script only runs on your local device.

Opening twitter search page...

Found existing file ./tweets-data/jkt48.csv, renaming to ./tweets-data/jkt48.old.csv

Filling in keywords: jkt48 senbatsu sousenkyo lang:id

-- Scrolling... (1) (2) (3) (4)

Your tweets saved to: C:\Users\kevir\Coding\Semester 5\Datmin M7 Twitter\tweets-data\jkt48.csv
Total tweets saved: 16

-- Scrolling... (1) (2) (3)

Your tweets saved to: C:\Users\kevir\Coding\Semester 5\Datmin M7 Twitter\tweets-data\jkt48.csv
Total tweets saved: 35
```

PRAKTIKUM DATA MINING II

BAB : SOCIAL NETWORK ANALYSIS

MINGGU KE- : TUJUH (7)

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

TGL PRAKTIKUM : 27 SEPTEMBER 2024

Untuk melihat data yang telah disimpan dapat dijalankan kode di bawah ini.

```
import pandas as pd

# Specify the path to your CSV file
file_path = f"tweets-data/{filename}"

# Read the CSV file into a pandas DataFrame
df = pd.read_csv(file_path, delimiter=",")

# Display the DataFrame
display(df)
```

	conversation_id_str	created_at	favorite_count	full_text	id_str	image_url	in_reply_to_sc
0	1832049924902670383	Fri Sep 06 13:35:14 +0000 2024	0	[QUIZ] Tebak Lagu Senbatsu Sousenkyo JKT48 Han...	1832049924902670383	NaN	
1	1831989864629493837	Fri Sep 06 09:36:35 +0000 2024	0	Daftar Lagu Senbatsu Sousenkyo JKT48 Sepanjang...	1831989864629493837	NaN	
2	1831906511754555760	Fri Sep 06 04:05:33 +0000 2024	16	Maka kami mengajak teman-teman untuk mewujudkan...	1831906557677990179	https://pbs.twimg.com/media/GWw9TeVWcAATW2Z.jpg	
3	1831906511754555760	Fri Sep 06 04:05:22 +0000 2024	73	Together We Are Ready to Fight for Danellal Jo...	1831906511754555760	https://pbs.twimg.com/media/GWw9Q20bIAAYn3K.jpg	
4	1831623831779414204	Thu Sep 05 09:22:06 +0000 2024	0	Inilah Reward untuk Member yang Masuk Senbatsu...	1831623831779414204	NaN	
...
		Tue Sep 17		Terakhir ianqan lupa			

```
# Cek jumlah data yang didapatkan

num_tweets = len(df)
print(f"Jumlah tweet dalam dataframe adalah {num_tweets}.")

Jumlah tweet dalam dataframe adalah 292.
```

Hasil dari crawling twitter sebanyak 292 tweet yang terambil.

- 2. Buatlah wordcloud dari topik tersebut, lakukan clustering dengan TF-IDF (atau feature extraction lainnya), interpretasikan hasilnya

PRAKTIKUM DATA MINING II

BAB : SOCIAL NETWORK ANALYSIS

MINGGU KE- : TUJUH (7)

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

TGL PRAKTIKUM : 27 SEPTEMBER 2024

Sebelum membuat wordcloud dan clustering akan dijalankan pre-processing terlebih dahulu. Di sini hasil crawling akan dilakukan tokenize, penghapusan simbol, penghapusan stopwords, dan stemming dengan menggunakan library NLTK dan Sastrawi.

```
import nltk
import os
import re
from nltk import sent_tokenize
from nltk import word_tokenize
from nltk.corpus import stopwords

def preprocess(doc):
    sents = word_tokenize(doc)
    sents_tok = list() #tokenisasi kalimat
    for s in sents:
        s = s.strip().lower() # case folding dan menghilangkan new line
        s = s.replace("\n", " ") # menggantikan \n dengan spasi
        s = re.sub(r"http\S+|t.co\S+", "", s) # menghapus URL
        s = re.sub(r'[^a-zA-Z0-9 ]', ' ', s) # menghapus simbol
        s = re.sub(' +', ' ', s) #menghapus repetitive space
        sents_tok.append(s)
    return " ".join(sents_tok)

docs_clear = list()
for i in df["full_text"]:
    docs_clear.append(preprocess(i))
```

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from nltk.tokenize import word_tokenize
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()

kalimat = []
for i in docs_clear:
    stop = stopword.remove(i)
    kalimat.append(stop)
```

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
stemmer = StemmerFactory().create_stemmer()

hasil_stem = []
for i in kalimat:
    stemmers = stemmer.stem(i)
    hasil_stem.append(stemmers)
```

Setelah dilakukan pre-processing kemudian dilanjutkan dengan pembuatan WordCloud.

BAB	: SOCIAL NETWORK ANALYSIS
MINGGU KE-	: TUJUH (7)
NAMA	: AMALIKA ARI ANINDYA
NIM	: 164221029
TGL PRAKTIKUM	: 27 SEPTEMBER 2024

[illegible]

```
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist

text = " ".join(hasil_stem)
tokens = word_tokenize(text)

filtered_words = [word for word in tokens if len(word) > 1]
fdist = FreqDist(filtered_words)
print(fdist.most_common(10))

[('jkt48', 382), ('senbatsu', 335), ('sousenkyo', 313), ('single', 111), ('member', 75), ('tahun', 59), ('masuk', 53), ('pil
ih', 51), ('jadi', 46), ('original', 44)]
```

PRAKTIKUM DATA MINING II

BAB : SOCIAL NETWORK ANALYSIS

MINGGU KE- : TUJUH (7)

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

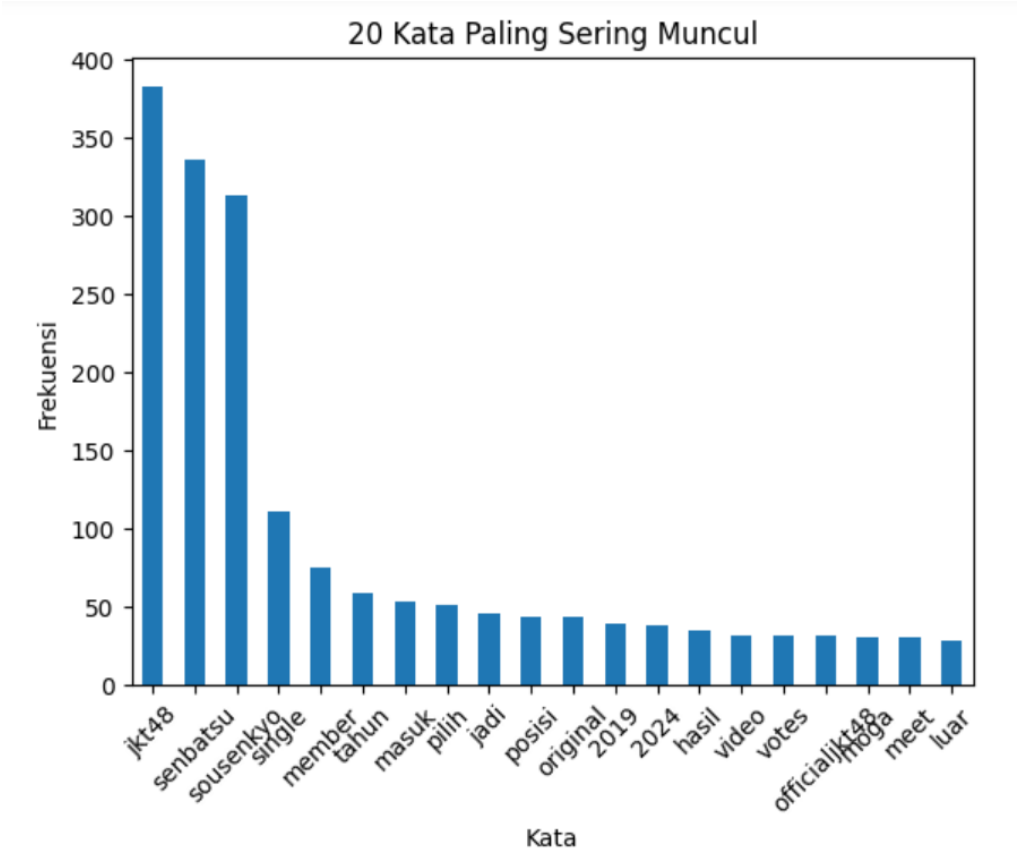
TGL PRAKTIKUM : 27 SEPTEMBER 2024

```
import pandas as pd

df_freq_tokens = pd.DataFrame.from_dict(fdist, orient='index')
df_freq_tokens.columns = ['Frequency']
df_freq_tokens.index.name = 'Key'

df_top_20 = df_freq_tokens.sort_values(by='Frequency', ascending=False).head(20)

df_top_20.plot(kind='bar', legend=False)
plt.title('20 Kata Paling Sering Muncul')
plt.ylabel('Frekuensi')
plt.xlabel('Kata')
plt.xticks(rotation=45)
plt.show()
```



Kata-kata yang paling banyak muncul pada tweet yang dicari tentu saja merupakan keywordnya yaitu ‘jkt48’, ‘senbatsu’, dan ‘sousenkyo’. Kata-kata selain tiga keyword tersebut yang paling banyak muncul lainnya adalah ‘single’, ‘member’, ‘tahun’, ‘masuk’, ‘pilih’, ‘jadi’, ‘posisi’, ‘original’, ‘2019’, ‘2024’, ‘hasil’, ‘video’, ‘votes’, ‘officialjkt48’, ‘moga’, ‘meet’, dan ‘luar’.

PRAKTIKUM DATA MINING II

BAB : SOCIAL NETWORK ANALYSIS
MINGGU KE- : TUJUH (7)
NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
TGL PRAKTIKUM : 27 SEPTEMBER 2024

Setelah pembuatan wordcloud dan barplot, dilanjutkan dengan ekstraksi fitur dengan menggunakan Word2Vec. Word2Vec digunakan karena lebih cocok dengan pengaplikasian text mining modern.

```
from gensim.models import Word2Vec
from sklearn.cluster import KMeans
import numpy as np

tokenized_docs = [doc.split() for doc in hasil_stem]

model = Word2Vec(sentences=tokenized_docs, vector_size=100, window=5, min_count=2, sg=0)

def document_vector(doc):
    word_vectors = [model.wv[word] for word in doc if word in model.wv]
    if len(word_vectors) > 0:
        return np.mean(word_vectors, axis=0)
    else:
        return np.zeros(model.vector_size)

doc_vectors = np.array([document_vector(doc) for doc in tokenized_docs])
doc_vectors = np.nan_to_num(doc_vectors)
doc_vectors
```

```
array([[ -0.01902281,  0.05546702, -0.00134967, ..., -0.0377186 ,
        -0.00378097, -0.01705713],
       [ -0.02823357,  0.07854185,  0.00020284, ..., -0.05422087,
        -0.00644087, -0.02688125],
       [ -0.01270096,  0.03911598, -0.00044179, ..., -0.0282682 ,
        -0.00404192, -0.01006434],
       ...,
       [ -0.02216624,  0.06015635, -0.00246282, ..., -0.04209365,
        -0.00536421, -0.02263166],
       [ -0.02242284,  0.06218068,  0.0002844 , ..., -0.04397226,
        -0.00562089, -0.0207651 ]],
```

Hasil ekstraksi fitur (doc_vectors) kemudian dimasukkan dalam algoritma clustering.

BAB : SOCIAL NETWORK ANALYSIS
MINGGU KE- : TUJUH (7)
NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
TGL PRAKTIKUM : 27 SEPTEMBER 2024

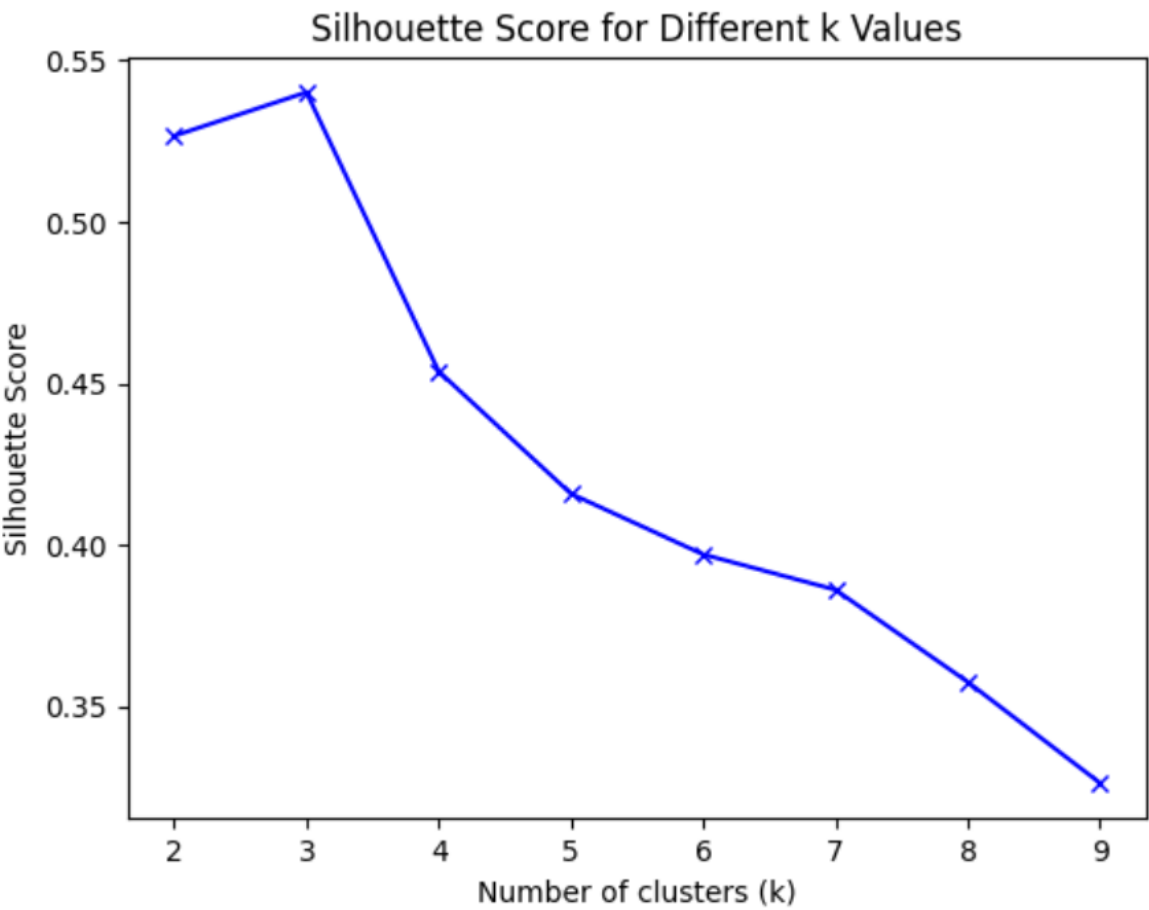
KMeans

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

silhouette_scores = []

for k in range(2, 10):
    km = KMeans(n_clusters=k, init='random', n_init=10, random_state=48)
    km.fit(doc_vectors)
    labels = km.labels_
    silhouette_avg = silhouette_score(doc_vectors, labels)
    silhouette_scores.append(silhouette_avg)

plt.plot(range(2, 10), silhouette_scores, 'bx-')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score for Different k Values')
plt.show()
```



PRAKTIKUM DATA MINING II

BAB : SOCIAL NETWORK ANALYSIS
MINGGU KE- : TUJUH (7)
NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
TGL PRAKTIKUM : 27 SEPTEMBER 2024

Dilihat dari hasil pengukuran di atas, jumlah cluster yang paling baik adalah sebanyak 3 cluster.

```
kmeans = KMeans(n_clusters=3, init='random', n_init=10, random_state=48)
kmeans.fit(doc_vectors)

labels = kmeans.labels_
print(labels)

[2 1 2 2 0 1 2 2 1 1 2 1 1 1 2 1 1 1 1 1 1 2 2 2 2 2 2 1 1 2 1 1
 1 2 2 2 1 1 1 1 2 0 2 2 2 2 1 2 0 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 1 1
 1 1 1 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 0 0 0 1 2 1 2 1 2 1 1 1 1 1 2 1 1 2 1 0 1 1 1 1 2 2 2 2 2 1 2 1 2 1 1 0 2
 2 2 1 2 2 1 2 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 1 1 1 1 1 2 2 2 2 2 1 2 1 2 2
 2 2 1 2 1 1 2 2 2 2 1 2 2 2 2 2 1 2 1 2 1 1 2 2 2 2 1 1 1 2 1 1 1 2 0 2 1
 1 1 2 2 1 1 1 2 1 2 2 2 2 1 1 1 1 1 1 2 2 2 2 2 1 1 1 1 1 2 1 1 1 2 1 1 1
 2 2 2 2 2 2 1 1 2 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2]
```

Visualisasi dengan PCA

```
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Perform PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(doc_vectors)

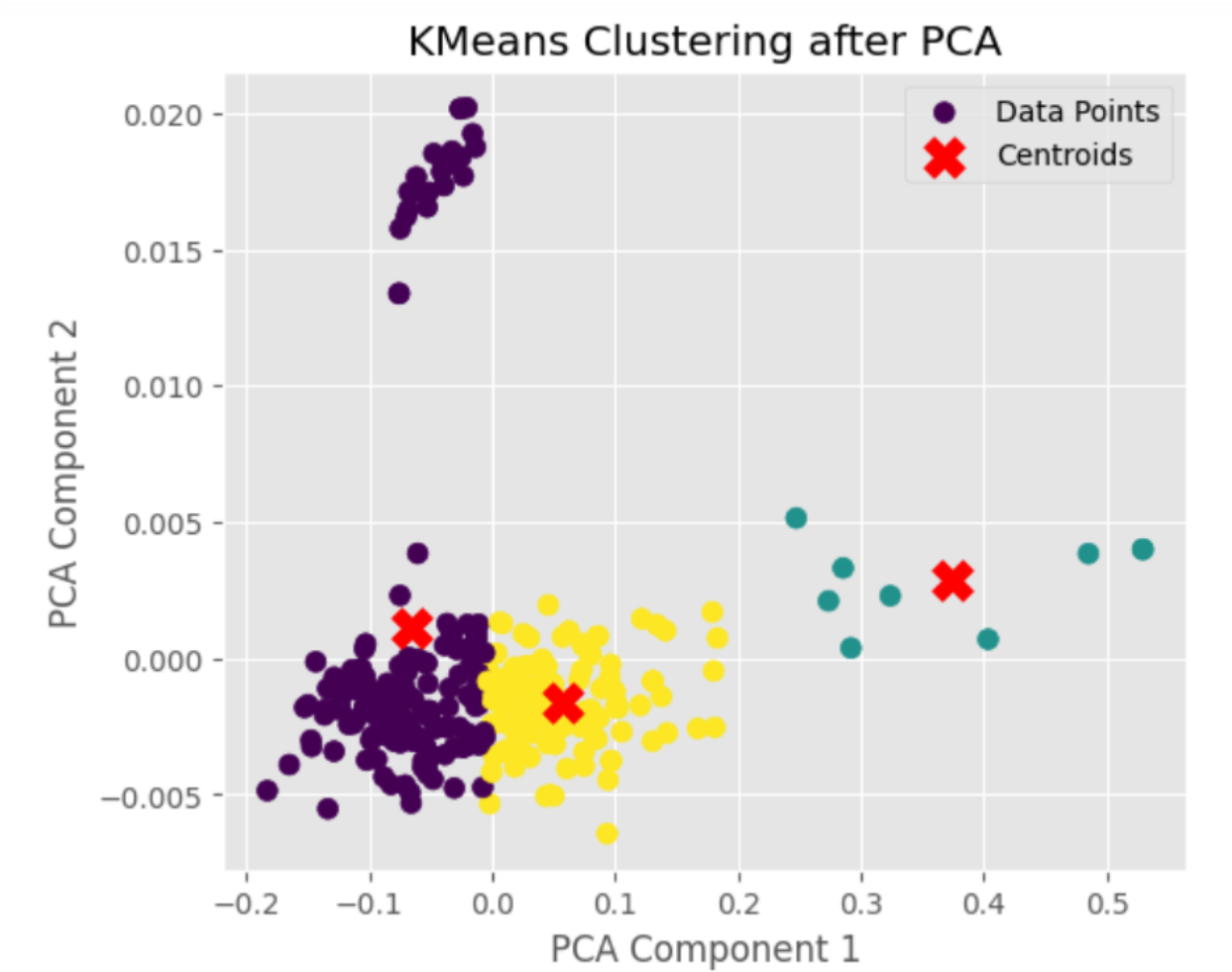
# Fit KMeans to the original data (doc_vectors) and get Labels
kmeans = KMeans(n_clusters=3, n_init=10, random_state=48)
labels = kmeans.fit_predict(doc_vectors)

# Transform the centroids to the PCA-reduced space
centroids_pca = pca.transform(kmeans.cluster_centers_)

# Plot the data points and centroids in PCA space
plt.figure(figsize=(6,5))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=labels, s=50, cmap='viridis', label='Data Points')
plt.scatter(centroids_pca[:, 0], centroids_pca[:, 1], c='red', s=200, marker='X', label='Centroids')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.title('KMeans Clustering after PCA')
plt.legend()
plt.show()
```


PRAKTIKUM DATA MINING II

BAB : SOCIAL NETWORK ANALYSIS
MINGGU KE- : TUJUH (7)
NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
TGL PRAKTIKUM : 27 SEPTEMBER 2024



```
from sklearn.metrics import silhouette_score  
  
silhouette_km = silhouette_score(doc_vectors, labels)  
print(f'Silhouette Score: {silhouette_km:.3f}')
```

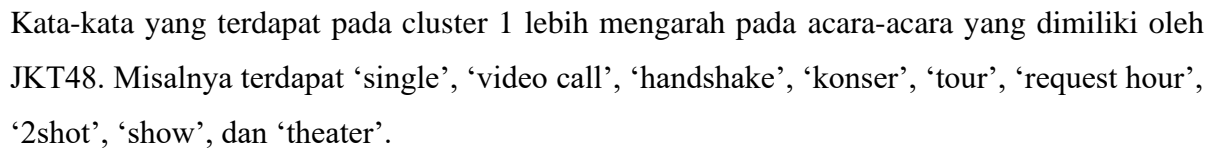
Silhouette Score: 0.540

Hasil Silhouette Score menunjukkan nilai 0.54 yang berarti hasil cluster sudah baik dalam menjelaskan pembagian tiap cluster dan tidak terlalu menumpuk satu sama lain.

BAB	: SOCIAL NETWORK ANALYSIS
MINGGU KE-	: TUJUH (7)
NAMA	: AMALIKA ARI ANINDYA
NIM	: 164221029
TGL PRAKTIKUM	: 27 SEPTEMBER 2024

```
# Menampilkan judul dokumen per cluster
clusters = {i: [] for i in range(num_clusters)}
for idx, label in enumerate(labels):
    clusters[label].append(hasil_stem[idx])
```

Cluster 1



BAB : SOCIAL NETWORK ANALYSIS

MINGGU KE- : TUJUH (7)

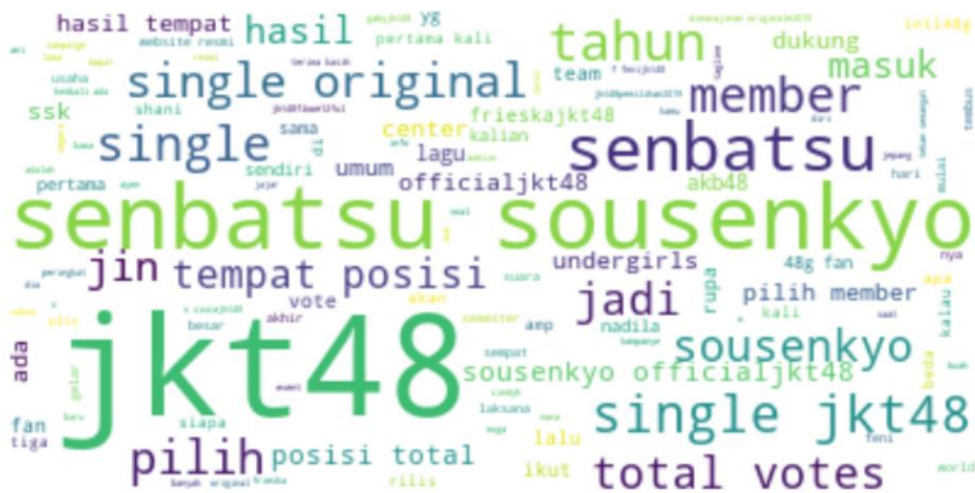
NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

TGL PRAKTIKUM : 27 SEPTEMBER 2024

A word cloud for the JKT48 Senbatsu Sosenkyo. The words are arranged in a circular pattern. The most prominent words are 'senbatsu' and 'sosenkyo' in large blue font. Other words include 'jkt48' in large green font, 'single' in blue, 'yupi' in blue, 'champions' in blue, 'tomodachi' in green, 'ini' in green, 'masuk' in green, 'reward' in green, 'akan' in green, 'analisa' in green, 'game' in blue, 'member' in blue, 'rival' in green, 'pilih' in green, 'pov' in blue, 'hasil' in green, 'ada' in green, 'masa' in green, 'congrats' in green, and 'shani' in green.

Cluster 3



Program Studi S1 Teknologi Sains Data
Fakultas Teknologi Maju & Multidisiplin
Universitas Airlangga
2024

BAB : SOCIAL NETWORK ANALYSIS
MINGGU KE- : TUJUH (7)
NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
TGL PRAKTIKUM : 27 SEPTEMBER 2024

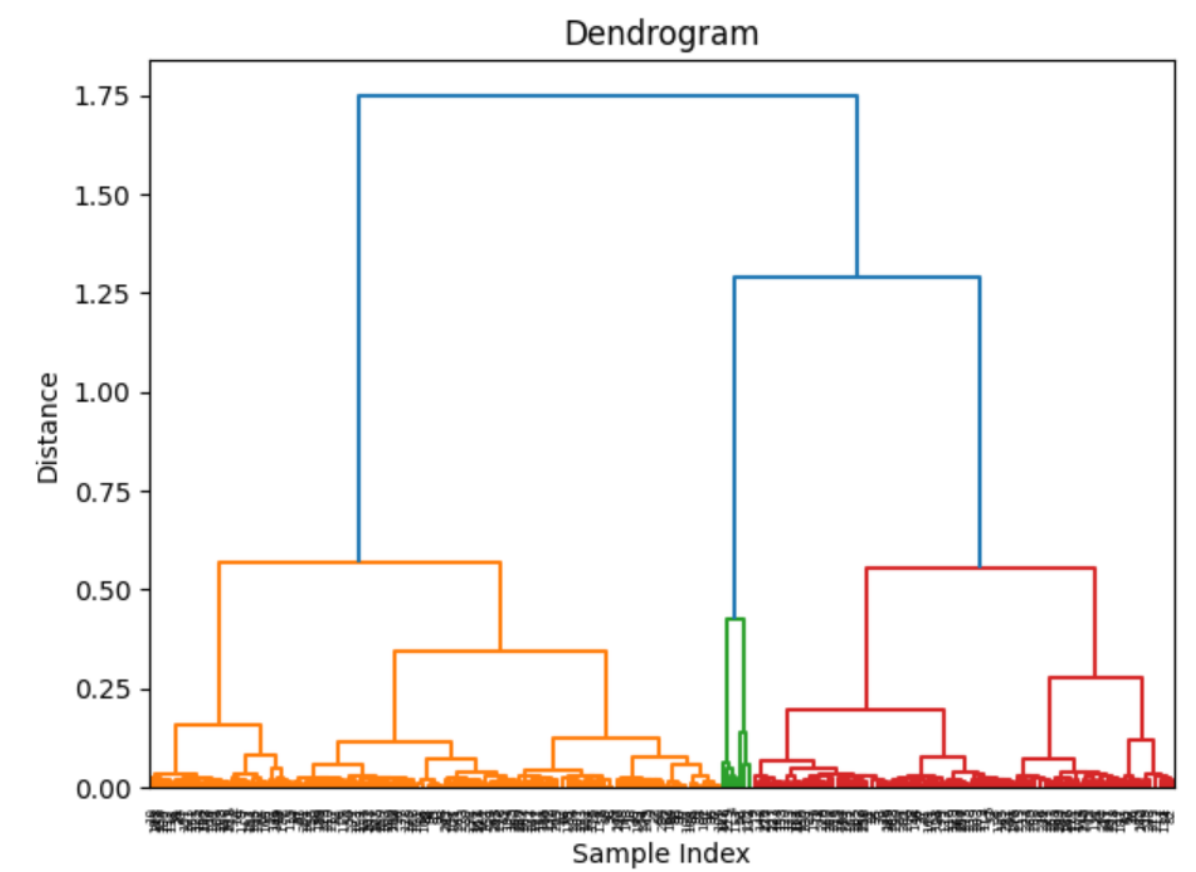
Hierarchical Clustering

Selain KMeans Clustering, dilakukan pula Hierarchical Clustering untuk melihat perbandingan Silhouette Score-nya.

```
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

# Create a Linkage matrix
linked = linkage(doc_vectors, method='ward')

# Plot the dendrogram
plt.figure(figsize=(7,5))
dendrogram(linked)
plt.title('Dendrogram')
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()
```



Visualisasi dendogram dilakukan untuk melihat jumlah cluster yang tepat. Dengan cutoff pada distance 0.75 dapat ditentukan jumlah cluster sebanyak 3.

BAB : SOCIAL NETWORK ANALYSIS

MINGGU KE- : TUJUH (7)

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

TGL PRAKTIKUM : 27 SEPTEMBER 2024

```
from sklearn.cluster import AgglomerativeClustering

hierarchical_cluster = AgglomerativeClustering(n_clusters=3, metric='euclidean', linkage='ward')
cluster_labels = hierarchical_cluster.fit_predict(doc_vectors)

print("Cluster Labels:")
print(cluster_labels)

Cluster Labels:
[0 1 0 0 2 1 0 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 0 1 1
 1 0 0 0 1 1 1 1 0 2 0 0 0 0 1 0 2 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1
 1 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 2 2 2 1 0 1 0 1 0 1 1 1 1 0 1 1 0 1 2 1 1 1 1 0 0 0 0 0 0 0 1 0 1 1 2 0
 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 1 0 1 0 0
 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 1 1 0 1 1 1 0 2 0 1
 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 1 1 1 0 1 1 1
 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0]
```

```
silhouette_avg = silhouette_score(doc_vectors, cluster_labels)
print(f'Silhouette Score: {silhouette_avg}')

Silhouette Score: 0.5402483344078064
```

Setelah menjalankan Hierarchical Clustering, dapat dilihat bahwa silhouette score menghasilkan nilai yang sama dengan KMeans Clustering.

- 3. Buatlah social network dari data tersebut (boleh berupa follow network atau reply/interaction network)

Untuk melakukan social network diperlukan membuat graph dengan menggunakan library networkx. Di sini penulis akan membuat interaction network. Hasil dari social network apabila dijadikan dalam dataframe akan memuat source, target, dan weight. Source sendiri merupakan username dari user yang membuat tweet tersebut. Target merupakan user yang dimention oleh source. Edge akan terbuat apabila terdapat mention antara source dan target. Kemudian weightnya akan bertambah setiap source dan target berinteraksi. Dengan kata lain, weight adalah jumlah mention dari source dan target.

```
import pandas as pd
import networkx as nx
import re
import matplotlib.pyplot as plt

G_mention = nx.DiGraph()

# Loop through each row to extract mentions and create the mention graph with weights
for index, row in df.iterrows():
    author = f"@{row['username']}"
    text = row['full_text']

    # Extract mentioned users using regex
    mentions = re.findall(r'@(\w+)', text)

    # Add edges to the graph with weights
    for mention in mentions:
        target = f"@{mention}"
        if G_mention.has_edge(author, target):
            G_mention[author][target]['weight'] += 1 # Increment the weight if edge exists
        else:
            G_mention.add_edge(author, target, weight=1) # Add new edge with initial weight 1
```

PRAKTIKUM DATA MINING II

BAB : SOCIAL NETWORK ANALYSIS

MINGGU KE- : TUJUH (7)

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

TGL PRAKTIKUM : 27 SEPTEMBER 2024

```
edges_df = pd.DataFrame([(u, v, d['weight']) for u, v, d in G_mention.edges(data=True)],
                        columns=['source', 'target', 'weight'])

print(edges_df)
```

	source	target	weight
0	@DenalizeID	@Daneella_JKT48	2
1	@InterindahFC	@C_IndahJKT48	1
2	@Mixcam48	@Levi_JKT48	1
3	@Mixcam48	@Regie_JKT48	1
4	@HandikaNovilinh	@vvotanet	1
..
198	@AllanMoech	@A_MutheJKT48	1
199	@AllanMoech	@N_AurelJKT48	1
200	@AllanMoech	@A_ChristyJKT48	1
201	@rdwnagam	@Nadila_JKT48	1
202	@ENUT96	@Nadila_JKT48	1

[203 rows x 3 columns]

4. Ukur centrality measures dari network tersebut, dan interpretasikan

```
from operator import itemgetter

graph_centrality = nx.degree_centrality(G_mention)
max_de = max(graph_centrality.items(), key=itemgetter(1))

graph_closeness = nx.closeness_centrality(G_mention)
max_clo = max(graph_closeness.items(), key=itemgetter(1))

graph_betweenness = nx.betweenness_centrality(G_mention, normalized=True, endpoints=False)
max_bet = max(graph_betweenness.items(), key=itemgetter(1))

# Output the node with maximum centrality
print(f'Max Degree Centrality: {max_de}')
print(f'Max Closeness Centrality: {max_clo}')
print(f'Max Betweenness Centrality: {max_bet}')
```

Max Degree Centrality: ('@AllanMoech', 0.16080402010050251)
Max Closeness Centrality: ('@officialJKT48', 0.05527638190954774)
Max Betweenness Centrality: ('@jawapos', 7.613826709304096e-05)

- a. Max Degree Centrality
- Nilai degree centrality menunjukkan banyaknya koneksi yang terhubung pada suatu node. Nilai yang tinggi menunjukkan user yang sering dimention atau sering berinteraksi dengan user lain. Dalam hal ini user @AllanMoech memiliki nilai degree centrality tertinggi yaitu sebesar 0.1608. Nilai yang tinggi menunjukkan bahwa user tersebut sering berinteraksi dengan user lain.
- b. Max Closeness Centrality

BAB : SOCIAL NETWORK ANALYSIS
MINGGU KE- : TUJUH (7)
NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
TGL PRAKTIKUM : 27 SEPTEMBER 2024

Closeness centrality mengukur seberapa dekat suatu user dengan user lainnya. Nilai closeness centrality yang tinggi menunjukkan bahwa user tersebut mudah untuk menyampaikan informasi kepada user lain. Dalam network ini, user @officialJKT48 memiliki nilai closeness centrality tertinggi yaitu sebesar 0.0553. Nilai tertinggi tersebut didukung oleh hal di mana user tersebut adalah akun official dari JKT48 yang pastinya banyak diikuti oleh user-user dalam network ini.

c. Max Betweenness Centrality

Betweenness centrality menunjukkan user yang berperan sebagai jembatan dalam suatu network. Dalam network ini user @jawapos memiliki nilai yang sangat kecil meskipun nilainya sudah yang tertinggi. Berarti user tersebut tidak terlalu sering menjembatani informasi antara user. Namun nilai tertinggi yang dimiliki user @jawapos mungkin didukung oleh fakta di mana user jawa pos merupakan laman berita di Indonesia.

5. Visualisasikan network tersebut dan interpretasikan

a. Menggunakan Python

```
pos = nx.spring_layout(G)
f, ax = plt.subplots(figsize=(10, 10))
plt.style.use('ggplot')
nodes = nx.draw_networkx_nodes(G_mention, pos, alpha=0.8)
nodes.set_edgecolor('k')
nx.draw_networkx_labels(G_mention, pos, font_size=10)
nx.draw_networkx_edges(G_mention, pos, width=1.0, alpha=0.5)

plt.title("User Mentions Graph")
plt.axis('off')
plt.show()
```

```
# Visualizing the mention network
pos = nx.spring_layout(G_mention, k=0.6, iterations=50)

f, ax = plt.subplots(figsize=(6, 6))
plt.style.use('ggplot')

# Draw nodes and edges with varying sizes based on weights
nx.draw_networkx_nodes(G_mention, pos, alpha=0.8, node_size=200, node_color='lightblue')
nx.draw_networkx_edges(G_mention, pos, width=[G_mention[u][v]['weight'] for u, v in G_mention.edges()], alpha=0.6)
nx.draw_networkx_labels(G_mention, pos, font_size=6)

plt.title("Mention Network with Weighted Edges")
plt.axis('off')
plt.show()
```

BAB : SOCIAL NETWORK ANALYSIS

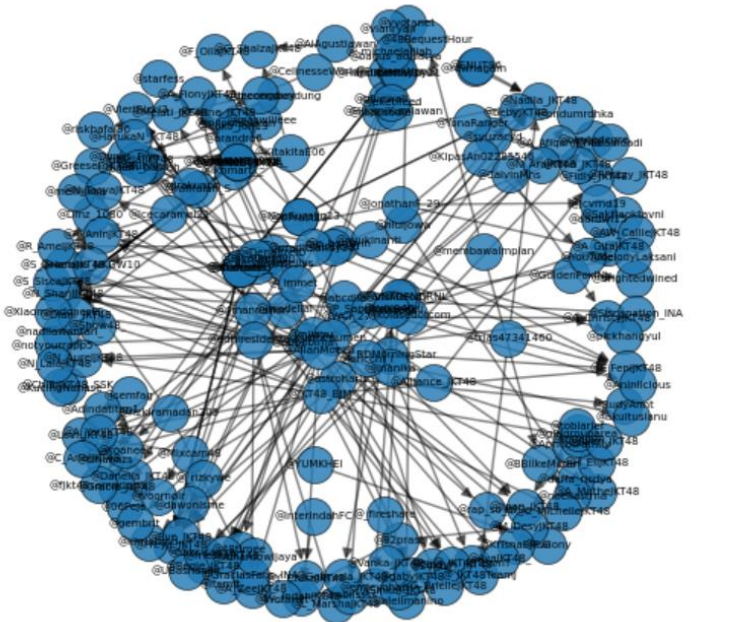
MINGGU KE- : TUJUH (7)

NAMA : AMALIKA ARI ANINDYA

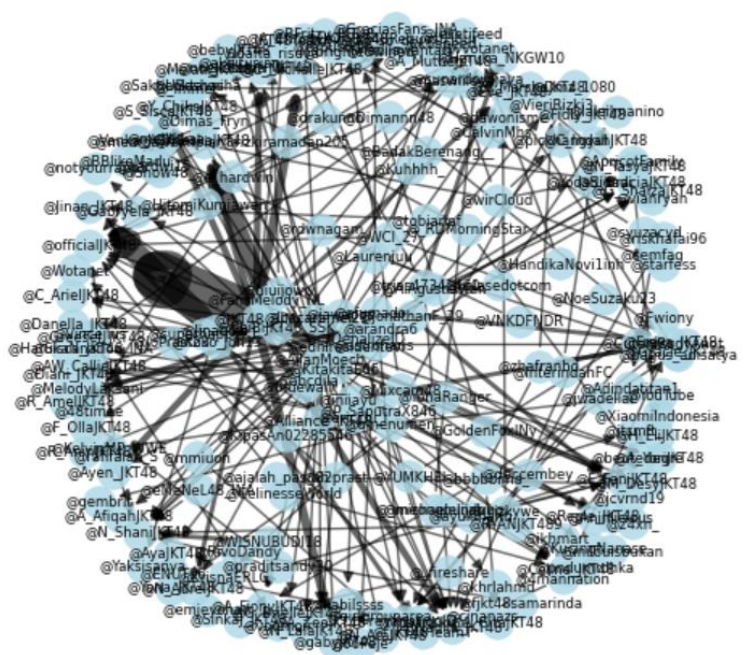
NIM : 164221029

TGL PRAKTIKUM : 27 SEPTEMBER 2024

User Mentions Graph



Mention Network with Weighted Edges



BAB : SOCIAL NETWORK ANALYSIS
MINGGU KE- : TUJUH (7)
NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
TGL PRAKTIKUM : 27 SEPTEMBER 2024

Terlihat @AllanMoech merupakan user dengan label terbesar, hal ini sesuai dengan Max Degree Centrality yang telah dihitung bahwa user tersebut memiliki interaksi terbesar dengan user lain. Dari visualisasi juga dapat dilihat selain user @AllanMoech, terdapat user @JKT48_BJM, @officialJKT48, dan @Nadila_JKT48 yang sering berinteraksi dengan user lain. Selain itu, garis yang menghubungkan antara @JKT48_BJM dan @officialJKT48 terlihat sangat tebal. Hal ini menunjukkan bahwa user @JKT48_BJM sering berinteraksi atau *me-mention* user @officialJKT48.

6. Buatlah kesimpulan kumpulan tweet tersebut beserta interaksi yang terjadi (contoh: siapa pemantik diskusi, siapa tokoh dalam diskusi tersebut, adakah kubu yang berlawanan, adakah broker diantara kubu tersebut, dsb)

Apabila dilihat dari hasil visualisasi sosial network analysis dengan bantuan *software* Gephi, dapat dilihat pada node dan edge berwarna hijau toska bahwa @AllanMoech mendominasi topik ‘JKT48 Senbatsu Sousenkyo’. Jika dilihat lebih seksama, hampir semua akun yang disebut oleh user Allan merupakan member dari JKT48 itu sendiri. Contohnya adalah @Celine_JKT48, @M_DesyJKT48, @A_YoriJKT48, @Eve_JKT48, @F_FeniJKT48, @R_AmelJKT48, dan masih banyak lagi. Jika dilihat dari topik dan user yang disebut, kemungkinan user Allan memberikan live report dengan memberikan hasil senbatsu sousenkyo (pemilihan member) dan menyebut (*mention*) anggota-anggota JKT48 yang lolos. Kemungkinan kedua adalah user Allan mereply akun anggota JKT48 dengan menyebutkan keyword yang dipakai.

Selain Allan, ada pula user @JKT48_BJM yang memiliki *pattern* yang mirip. User tersebut memiliki garis yang sangat tebal kepada @officialJKT48 yang merupakan akun resmi dari grup JKT48. User tersebut juga menyebut member JKT48 dengan frekuensi sering yang dapat dilihat dari garis berwarna merah muda pada visualisasi. Member JKT48 yang sering disebut adalah Michelle, Ayen, Gaby, Sisca, Cindy, dan Aurel. Dapat dilihat juga bahwa akun @officialJKT48 masih membawa warna pink dari @JKT48_BJM yang sesuai dengan fakta bahwa akun @officialJKT48 memiliki nilai Closeness Centrality yang tinggi.