162012133068_Nicholas Juan Kalvin_Week 4

September 13, 2022

1 Tugas Praktikum Minggu 4

1.0.1 Nicholas Juan Kalvin P. | 162012133068

2 Text Mining

2.1 Tugas Praktikum:

Download data teks dari halaman: https://raw.githubusercontent.com/ruzcmc/medmon/main/jawapos19012021.c

Buatlah wordcloud dan most common word barplot, interpretasikan hasilnya!

```
<li>Lakukan <i>clustering</i> dengan menggunakan fitur TF-IDF</li>
<li>Buat visualisasi clusternya dan lakukan interpretasi terhadap hasil tersebut!</li>
```

[ ]:
```
# Modules and Libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import nltk
import re
import string
import nlp_id
import seaborn as sns

from nltk.probability import FreqDist
from collections.abc import Sequence

from nlp_id.stopword import StopWord
from nlp_id.lemmatizer import Lemmatizer
from nlp_id.tokenizer import Tokenizer

from wordcloud import WordCloud
```

Selain dari library yang biasa digunakan untuk melakukan data mining di Python, digunakan beberapa library baru yang bekerja spesifik untuk memproses teks. re atau regular expressionadalah salah satu library Python yang paling dikenal, digunakan untuk berbagai macam prosesing teks dari mengurangi tanda baca hingga mengganti bagian dari kata.

nlp-id adalah library NLP khusus untuk Bahasa Indonesia yang memiliki fitur untuk Stopwords, Lemmatizing, Tokenizing dan lain-lain. nlp-id mirip seperti Sastrawi, tetapi dengan komunitas yang

jauh lebih aktif, akan terdapat beberapa hal yang nlp-id dapat lakukan lebih baik dari Sastrawi

Wordcloud adalah library untuk membuat wordcloud

2.2 Step 1: Dataset Loading

[ ]:
```
df = pd.read_csv('text.csv')

df.drop(columns = ['Unnamed: 0', 'published', 'link'], inplace=True)
df.head(10)
```

[ ]:
```
                                                title  \
0             Taiwan Batalkan Festival Besar Lampion
1  Sebut Bali Ramah LGBT, Kristeb Gray Ditangkap …
2  Uji klinis Awal Sebut Vaksin Covid-19 Rusia 10…
3  Masyarakat Harus Disiplin Terapkan Protokol Ke…
4  Belasan Motor Bodong dari Jawa Gagal Diselundu…
5  Citilink Layani Penerbangan Padang-Medan Empat…
6  40 Ton Beras Diangkut dari Sulteng dengan Tol …
7  Kolaborasi KPK-BPN-PLN Riau Selamatkan Aset Ta…
8  Hukuman Jerinx Didiskon 4 Bulan dalam Putusan …
9  Kelebihan Listrik Sumsel Disalurkan untuk Jamb…

                                              content  \
0  elasa membatalkan festival besar selama libura…
1  h Hukum dan HAM Bali melakukan upaya hukum ter…
2  s kesehatan konsumen di Rusia, Rospotrebnadzor…
3  ologi Fakultas Kesehatan Masyarakat Universita…
4  it sepeda motor bodong atau tanpa dilengkapi s…
5  ar Udara Internasional Kualanamu (KNIA) di Del…
6  kut Logistik Tol Laut KM Kendhaga Nusantara 13…
7  tara PT PLN (Persero), Komisi Pemberantasan Ko…
8  de Ary Astina alias Jerinx mendapat keringanan…
9  rgi listrik dari pembangkit listrik di wilayah…

                                              summary
0  Taiwan pada Selasa membatalkan festival besar …
1  KANTOR Wilayah Hukum dan HAM Bali melakukan up…
2  Badan pengawas kesehatan konsumen di Rusia, Ro…
3  PAKAR epidemiologi Fakultas Kesehatan Masyarak…
4  SEKITAR 16 unit sepeda motor bodong atau tanpa…
5  Otoritas Bandar Udara Internasional Kualanamu …
6  Kapal pengangkut Logistik Tol Laut KM Kendhaga…
7  Kolaborasi antara PT PLN (Persero), Komisi Pem…
8  TERDAKWA I Gede Ary Astina alias Jerinx mendap…
9  Kelebihan energi listrik dari pembangkit listr…
```

Kolom published dan link saya hapus, karena tidak terdapat nilai yang berguna untuk melakukan

prosesing teks

2.3 Step 2: Text Cleaning

Beberapa langkah pembersihan teks yang dilakukan pada dataset ini adalah:

Mengubah menjadi huruf kecil

Menghilangkan URL dan unsur HTML

Menghilangkan Emoji dan Emoticons

Menghilangkan tanda baca

Menghilangkan stopwords

Lematisasi (PySastrawi)

2.3.1 Lowercasing

[ ]:
```
for (column, textval) in df.iteritems():
    df[column] = df[column].str.lower()

df.head(5)
```

[ ]:
```
                                                title  \
0             taiwan batalkan festival besar lampion
1  sebut bali ramah lgbt, kristeb gray ditangkap …
2  uji klinis awal sebut vaksin covid-19 rusia 10…
3  masyarakat harus disiplin terapkan protokol ke…
4  belasan motor bodong dari jawa gagal diselundu…

                                              content  \
0  elasa membatalkan festival besar selama libura…
1  h hukum dan ham bali melakukan upaya hukum ter…
2  s kesehatan konsumen di rusia, rospotrebnadzor…
3  ologi fakultas kesehatan masyarakat universita…
4  it sepeda motor bodong atau tanpa dilengkapi s…

                                              summary
0  taiwan pada selasa membatalkan festival besar …
1  kantor wilayah hukum dan ham bali melakukan up…
2  badan pengawas kesehatan konsumen di rusia, ro…
3  pakar epidemiologi fakultas kesehatan masyarak…
4  sekitar 16 unit sepeda motor bodong atau tanpa…
```

2.3.2 Menghilangkan unsur URL/HTML

[ ]:
```
def bersih_text(text):
    text = str(text)
    text = re.sub("@[A-Za-z0-9_]+", "", text)  # Menghapus @<name> [mention␣
 ↪twitter]
    text = re.sub("#\w+", "", text)
    text = re.sub("\[.*?\]", "", text)
    text = re.sub("https?://\S+|www\.\S+", "", text)
    text = re.sub("<.*?>+", "", text)
    text = re.sub("[%s]" % re.escape(string.punctuation), "", text)
    # text = re.sub('\n', '', text)
    text = re.sub("\w*\d\w*", "", text)
    text = re.sub("\d+", "", text)
    text = re.sub("\s+", " ", text).strip()
    # text = re.sub('\n', '', text) jadi:
    text = text.replace("\n", " ")
    text = " ".join(text.split())
    return text

for column, content in df.iteritems():
    for index, value in enumerate(df[column]):
        df[column][index] = bersih_text(df[column][index])

df_clean = df.copy()
```

2.3.3 Tokenizing

[ ]:
```
tokenizer = Tokenizer()

for column, content in df.iteritems():
    for index, value in enumerate(df[column]):
        df[column][index] = tokenizer.tokenize(df[column][index])

df_token = df.copy()
```

Tokenizing dilakukan untuk memisahkan setiap kata, yang lalu dapat dilakukan lematisasi

[ ]:
```
df.head(5)
```

[ ]:
```
                                               title  \
0       [taiwan, batalkan, festival, besar, lampion]
1  [sebut, bali, ramah, lgbt, kristeb, gray, dita…
2  [uji, klinis, awal, sebut, vaksin, rusia, efek…
3  [masyarakat, harus, disiplin, terapkan, protok…
4  [belasan, motor, bodong, dari, jawa, gagal, di…
```

```
                                                  content  \
0  [elasa, membatalkan, festival, besar, selama, …
1  [h, hukum, dan, ham, bali, melakukan, upaya, h…
2  [s, kesehatan, konsumen, di, rusia, rospotrebn…
3  [ologi, fakultas, kesehatan, masyarakat, unive…
4  [it, sepeda, motor, bodong, atau, tanpa, dilen…

                                                  summary
0  [taiwan, pada, selasa, membatalkan, festival, …
1  [kantor, wilayah, hukum, dan, ham, bali, melak…
2  [badan, pengawas, kesehatan, konsumen, di, rus…
3  [pakar, epidemiologi, fakultas, kesehatan, mas…
4  [sekitar, unit, sepeda, motor, bodong, atau, t…
```

2.3.4 Lemmatizing

[ ]:
```
lemmatizer = Lemmatizer()

for column, content in df.iteritems():
    for index, value in enumerate(df[column]):
        df[column][index] = [lemmatizer.lemmatize(word) for word in␣
↪df[column][index]]
```

[ ]:
```
df.head(5)
```

[ ]:
```
                                                   title  \
0          [taiwan, batal, festival, besar, lampion]
1  [sebut, bal, ramah, lgbt, kristeb, gray, tangk…
2  [uji, klinis, awal, sebut, vaksin, rusia, efek…
3  [masyarakat, harus, disiplin, terap, protokol,…
4  [belas, motor, bodong, dari, jawa, gagal, selu…

                                                  content  \
0  [elasa, batal, festival, besar, lama, libur, t…
1  [h, hukum, dan, ham, bal, laku, upaya, hukum, …
2  [s, sehat, konsumen, di, rusia, rospotrebnadzo…
3  [ologi, fakultas, sehat, masyarakat, universit…
4  [it, sepeda, motor, bodong, atau, tanpa, lengk…

                                                  summary
0  [taiwan, pada, selasa, batal, festival, besar,…
1  [kantor, wilayah, hukum, dan, ham, bal, laku, …
2  [badan, awas, sehat, konsumen, di, rusia, rosp…
3  [pakar, epidemiologi, fakultas, sehat, masyara…
4  [sekitar, unit, sepeda, motor, bodong, atau, t…
```

Lematisasi memberikan output dataframe dimana isinya telah diambil kata akarnya.

2.3.5 Menghilangkan Stopwords

```
[ ]: stopword = StopWord()

all_content = []
for index, cvalue in enumerate(df['content']):
    all_content = all_content + cvalue

all_summary = []
for index, svalue in enumerate(df['summary']):
    all_summary = all_summary + svalue

all_title = []
for index, tvalue in enumerate(df['title']):
    all_title = all_title + tvalue

all_content_str = stopword.remove_stopword(' '.join(all_content))
all_summary_str = stopword.remove_stopword(' '.join(all_summary))
all_title_str = stopword.remove_stopword(' '.join(all_title))
```

2.4 Step 3: Wordcloud dan Histogram

2.4.1 Worcloud dari kolom konten

```
[ ]: wc = WordCloud(max_words=5000, margin=2).generate(all_content_str)

plt.axis("off")
plt.imshow(wc, interpolation='bilinear')
plt.show()
```

2.4.2 Worcloud dari kolom summary

```
[ ]: wc = WordCloud(max_words=5000, margin=2).generate(all_summary_str)

plt.axis("off")
plt.imshow(wc, interpolation='bilinear')
plt.show()
```

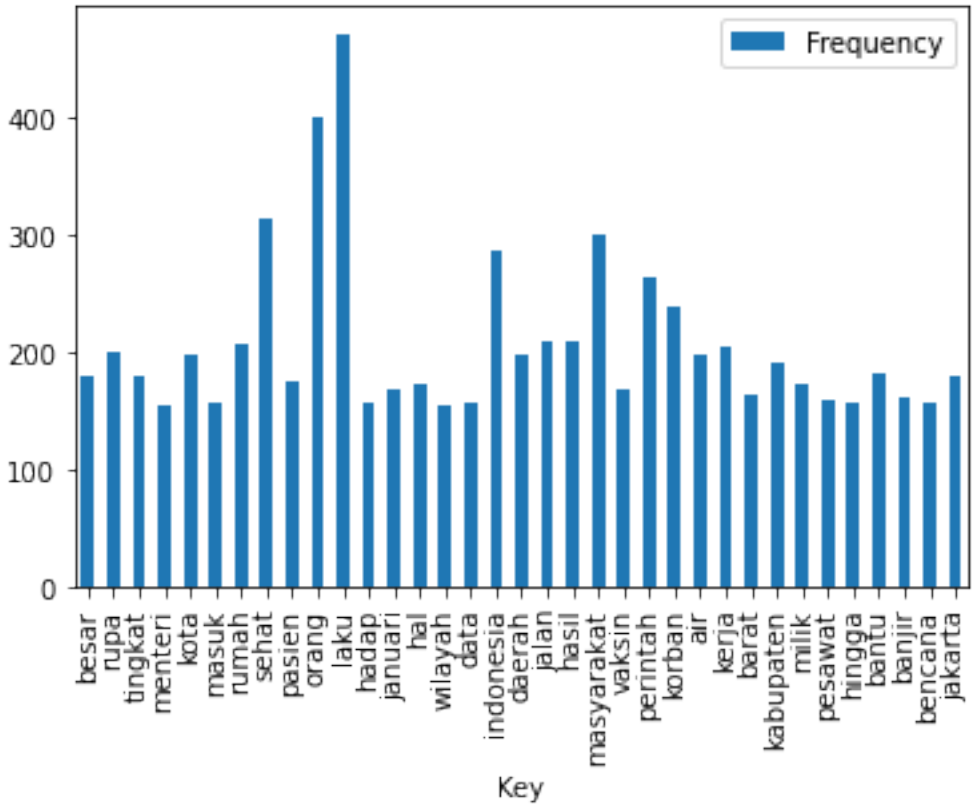
2.4.3 Worcloud dari kolom title

```
[ ]: wc = WordCloud(max_words=5000, margin=2).generate(all_title_str)

plt.axis("off")
plt.imshow(wc, interpolation='bilinear')
plt.show()
```

[ ]:
```
cfdist = FreqDist(all_content_str.split(sep=' '))
sfdist = FreqDist(all_summary_str.split(sep=' '))
tfdist = FreqDist(all_title_str.split(sep=' '))
```

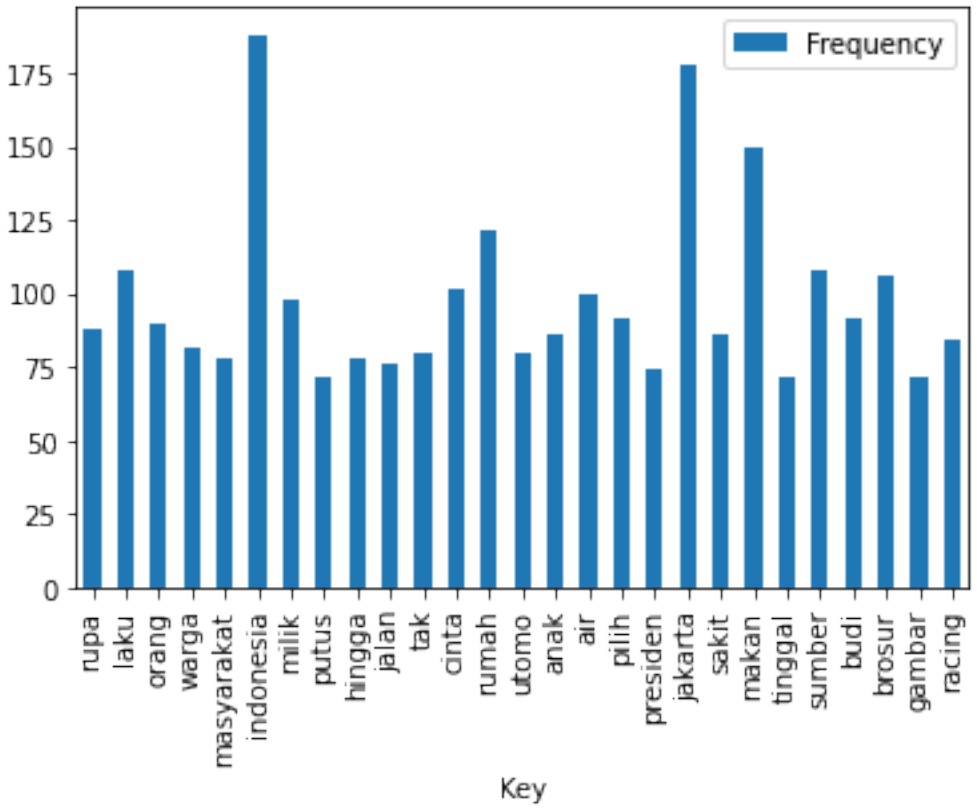
[ ]:
```
most_common_cfdist = cfdist.most_common(20)
most_common_sfdist = sfdist.most_common(20)
most_common_tfdist = tfdist.most_common(20)
```

[ ]:
```
df_freq_token_content = pd.DataFrame.from_dict(cfdist, orient='index')
df_freq_token_content.columns = ['Frequency']
df_freq_token_content.index.name = 'Key'

plt_df_freq_token_content =␣
↪df_freq_token_content[df_freq_token_content['Frequency'] > 150]
plt_df_freq_token_content.plot(kind='bar')
plt.show()
```

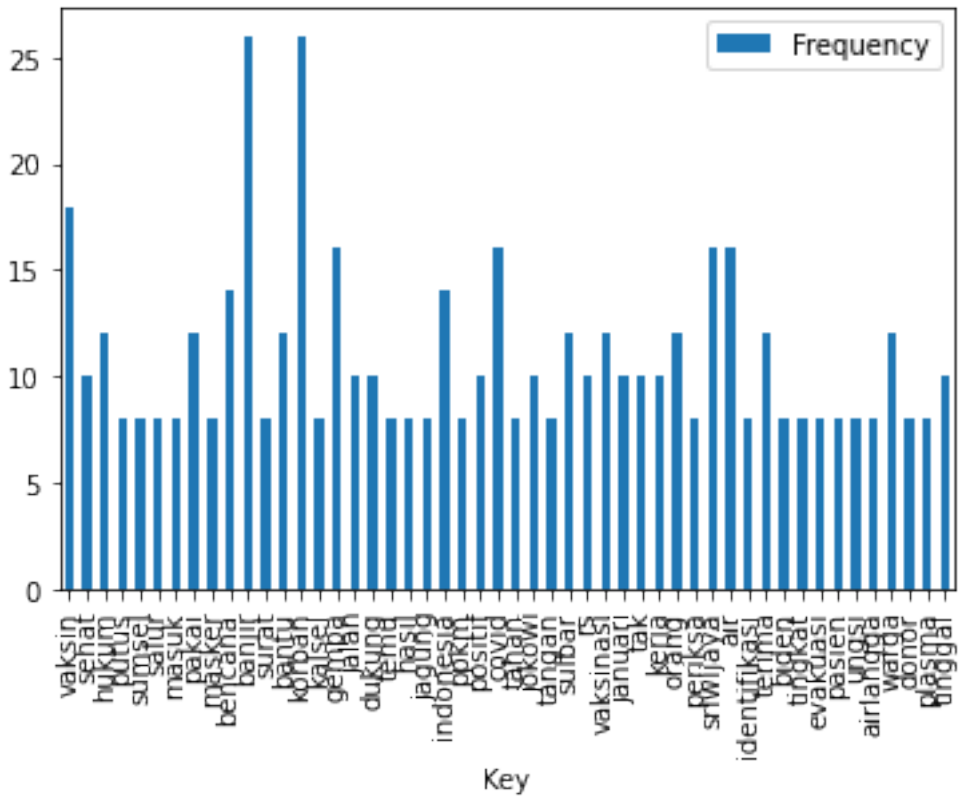


[ ]:
```
df_freq_token_summary = pd.DataFrame.from_dict(sfdist, orient='index')
df_freq_token_summary.columns = ['Frequency']
```

```
df_freq_token_summary.index.name = 'Key'

plt_df_freq_token_summary =␣
 ↪df_freq_token_summary[df_freq_token_summary['Frequency'] > 70]
plt_df_freq_token_summary.plot(kind='bar')
plt.show()
```



[ ]:
```
df_freq_token_title = pd.DataFrame.from_dict(tfdist, orient='index')
df_freq_token_title.columns = ['Frequency']
df_freq_token_title.index.name = 'Key'

plt_df_freq_token_title = df_freq_token_title[df_freq_token_title['Frequency']␣
 ↪> 7]
plt_df_freq_token_title.plot(kind='bar')
plt.show()
```



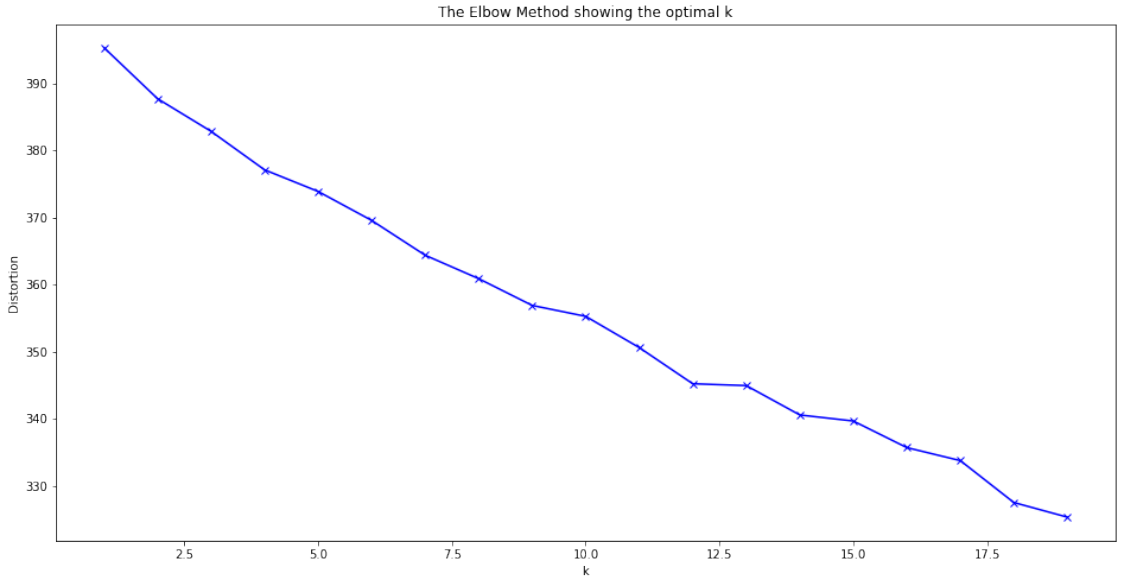
2.5 Step 4: TF-IDF Clustering

[ ]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
```

[ ]:
```
vectorizer = TfidfVectorizer(sublinear_tf=True, min_df=5, max_df=0.95)
X = vectorizer.fit_transform(df_clean['content'] + df_clean['summary'] +␣
 ↪df_clean['title'])
# X = vectorizer.fit_transform(df['content'][0])
tfidf_tokens = vectorizer.get_feature_names()
```

[ ]:
```
distortions = []
K = range(1,20)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(X)
    distortions.append(kmeanModel.inertia_)
plt.figure(figsize=(16,8))
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
```

```
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
clusters = kmeanModel.labels_
```



Blok kode ini digunakan untuk melakukan cluster sebanyak 20 kali, untuk mendapatkan plot elbow yang akurat

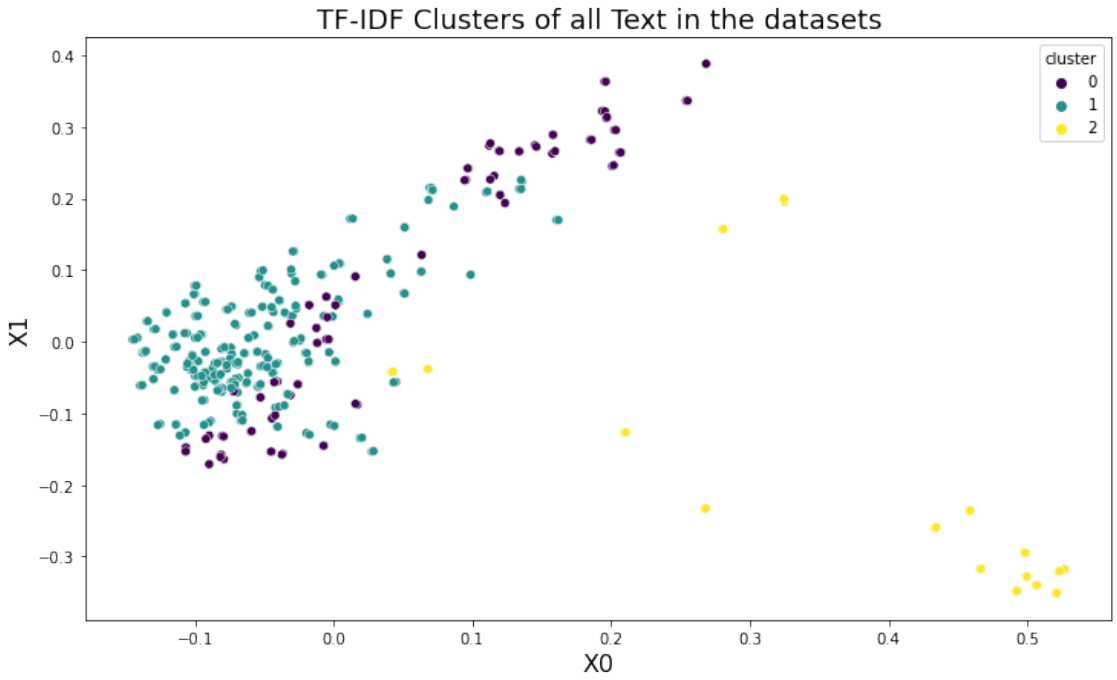
[ ]:
```
kmeanModel = KMeans(n_clusters=3)
kmeanModel.fit(X)
clusters = kmeanModel.labels_
```

Digunakan nilai cluster bernilai 3 yang diketahui dari plot elbow

[ ]:
```
pca = PCA(n_components=2, random_state=42)
pca_vecs = pca.fit_transform(X.toarray())
x0 = pca_vecs[:, 0]
x1 = pca_vecs[:, 1]

# assign clusters and PCA vectors to columns in the original dataframe
df['cluster'] = clusters
df['x0'] = x0
df['x1'] = x1
```

[ ]:
```
plt.figure(figsize=(12, 7))
plt.title("TF-IDF Clusters of all Text in the datasets", fontdict={"fontsize":␣
↪18})
plt.xlabel("X0", fontdict={"fontsize": 16})
```

```
plt.ylabel("X1", fontdict={"fontsize": 16})
sns.scatterplot(data=df, x='x0', y='x1', hue='cluster', palette="viridis")
plt.show()
```



Dari scatterplot diatas, diketahui benar adanya bahwa tiga cluster adalah nilai yang cukup untuk mewakkili dataset ini, dilihat dari titik-titiknya yang memiliki batasan jelas.

[ ]:
```
cluster0 = df.loc[pd.Index([0], name="cluster")]
wc = WordCloud(max_words=5000, margin=2).generate(stopword.remove_stopword(' '.
 ↪join(cluster0['content'][0])))

plt.axis("off")
plt.imshow(wc, interpolation='bilinear')
plt.show()
```

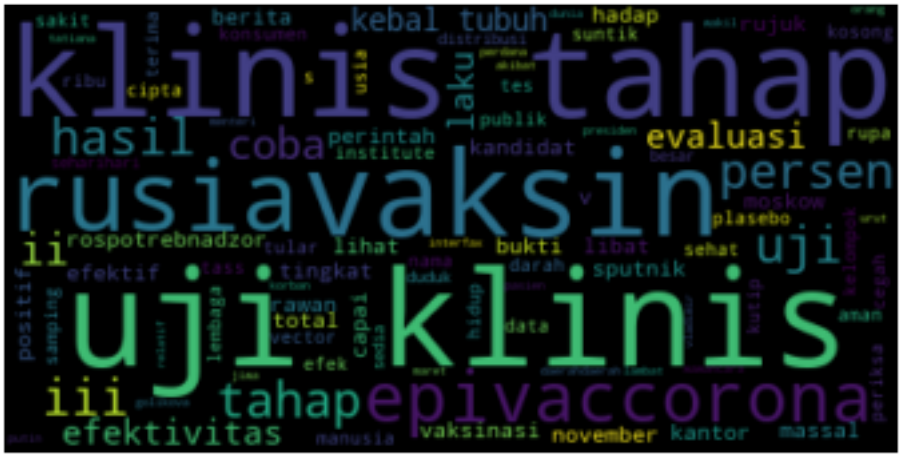
Menurut wordcloud yang dibuat berdasarkan cluster, cluster0 berisi semua kata yang memiliki hubungan dengan Covid-19 ### Taiwan, Festival, COVID

[ ]:
```
cluster0 = df.loc[pd.Index([1], name="cluster")]
wc = WordCloud(max_words=5000, margin=2).generate(stopword.remove_stopword(' '.
↪join(cluster0['content'][1])))

plt.axis("off")
plt.imshow(wc, interpolation='bilinear')
plt.show()
```

Sedangkan, cluster 1 berisi semua teks yang berhubungan dengan imigrasi, wna, dan kecelakaan ### Imigrasi, Kecelakaan, WNA

[ ]:

```
[ ]: cluster0 = df.loc[pd.Index([2], name="cluster")]
     wc = WordCloud(max_words=5000, margin=2).generate(stopword.remove_stopword(' '.
      ↪join(cluster0['content'][2])))

     plt.axis("off")
     plt.imshow(wc, interpolation='bilinear')
     plt.show()
```



Cluster terakhir yaitu cluster 2 berisi semua berita yang berhubungan juga dengan Covid, lebih spesifiknya adalah perkembangan vaksin covid ### Vaksin, Uji Klinis, Rusia