

Nama : I Gede Laksmna Yudha

Email : laksmanayudha22@gmail.com

SENTIMENT ANALISIS MENGENAI OMNIBUS LAW PADA TWITTER

Omnibus berasal dari bahasa latin yang artinya ‘untuk semuanya’. Maka secara sederhana Omnibus Law dapat diartikan suatu undang – undang yang mengatur banyak hal. Akhir – akhir ini di Indonesia Omnibus Law RUU Cipta Kerja menjadi sorotan karena terdapat beberapa pasalnya yang dianggap kontroversial, RUU Cipta Kerja dinilai serikat buruh hanya mementingkan kepentingan investor. RUU Cipta Kerja hanya salah satu bagian dari omnibus law. Dalam omnibus law, terdapat tiga RUU yang siap diundangkan, antara lain RUU tentang Cipta Kerja, RUU tentang Ketentuan dan Fasilitas Perpajakan untuk Penguatan Perekonomian, dan RUU tentang Pengembangan dan Penguatan Sektor Keuangan (Kompas.com - 05/10/2020, 10:22 WIB) . Maka dari itu analisis sentiment ini bertujuan untuk mengetahui tanggapan masyarakat Indonesia mengenai Pengesahan dari Omnibus Law tersebut.

Proses pengambilan data dan pembersihan dilakukan dengan menjalankan file laksmanayudha22_tugas13.py. Selanjutnya untuk **proses analisis sentiment** dilakukan pada file final_project.ipynb

Pada topik kali ini data yang digunakan merupakan data tweet yang diambil menggunakan twitter API dimulai dari tanggal 18-11-2020 sampai 28-11-2020. Dari pengambilan tersebut diperoleh 1094 data tweet.

TOP & Most Data

Kali ini akan dicoba mencari data yang sering muncul sehingga bisa didapatkan gambaran awal mengenai kecenderungan opini masyarakat. Langkah awal yang dilakukan adalah mengambil semua data yang diperlukan dari database. Setelah itu menggabungkan clean tweet menjadi 1 paragraf untuk, kemudian melakukan wordcloud dan plot data.

```
import sqlite3
import pandas as pd
import numpy as np
from scipy import stats
import itertools
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
from wordcloud import WordCloud
```

```
import matplotlib.pyplot as plt
import tweepy

# ambil data dari database
conn = sqlite3.connect('laksmanayudha22_final.db')
cursor = conn.cursor()

query = """ SELECT tweetid, cleantweet FROM tweet; """
cursor.execute(query)
conn.commit()
all_data = cursor.fetchall()

cursor.close()
conn.close()

# pisahkan tweetid dan clean tweet
clean_tweet = []
tweet_id = []
for data in all_data:
    clean_tweet.append(data[1])
    tweet_id.append(data[0])

# word tokenize
data = [word_tokenize(tweet) for tweet in clean_tweet]

# multidimensional ke single dimension
data = list(itertools.chain(*data))

fqdist = FreqDist(data)
print(fqdist.most_common(20))

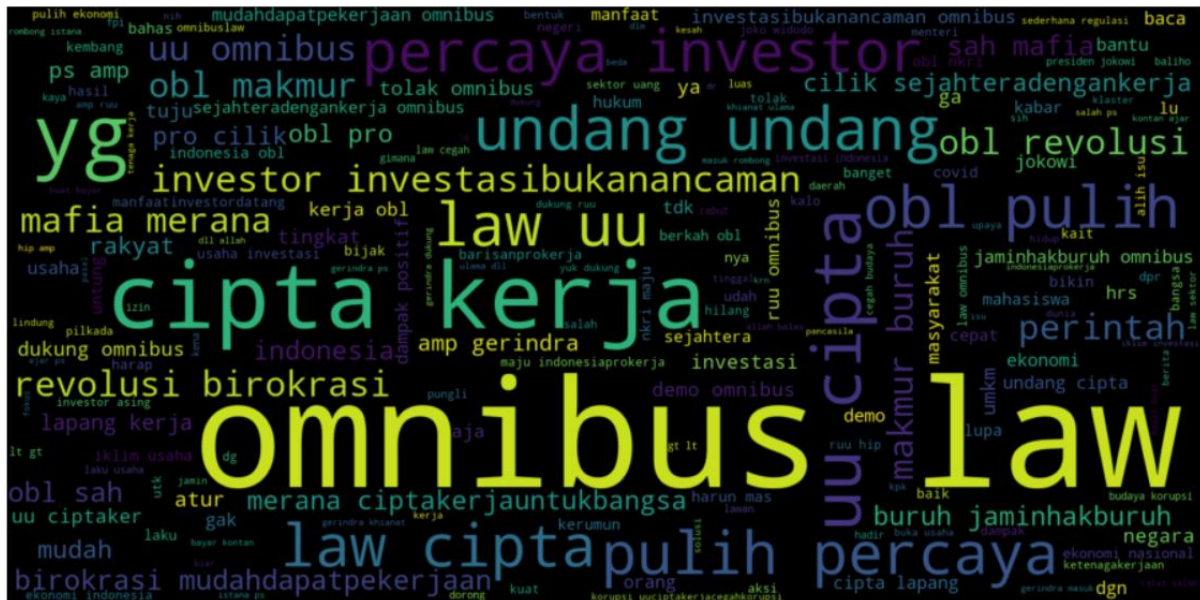
paragraf = ' '.join(data)
print(paragraf)

wordcloud = WordCloud(width=1600, height=800,
max_font_size=200).generate(paragraf)
plt.figure(figsize=(16,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Output :

```
[('omnibus', 1073), ('law', 1073), ('obl', 423), ('kerja', 360), ('uu', 330), ('cipta', 279), ('yg', 191), ('undang', 180), ('amp', 153), ('buruh', 143), ('indonesia', 139), ('investor', 125), ('usaha', 124), ('pulihan', 112), ('birokrasi', 106), ('investasi', 96), ('percaya', 96), ('revolusi', 95), ('sah', 93), ('investasibukanancaman', 90)]
```

20 kata yang sering muncul adalah omnibus, law, obl, kerja, revolusi, investasi bukan ancaman, dan seterusnya. Berikut gambar dari Word Cloud :



Gambar 1. Word Cloud

Dilihat dari WordCloud dan kata yang sering muncul dapat disimpulkan sementara omnibus law UU merupakan suatu investasi dan bukan ancaman. Kemudian UU ini dianggap lebih percaya terhadap investor daripada jaminan hak buruh.

Sentimen Analisis - Polarity

Untuk sentiment analisi awal menggunakan pendekatan polarity. Untuk melakukan hal tersebut pertama yang dibutuhkan adalah kumpulan kata positif dan negatif.

```
pos_list= open("./kata_positif.txt","r")
pos_kata = pos_list.readlines()
neg_list= open("./kata_negatif.txt","r")
neg_kata = neg_list.readlines()
```

Kemudian mencari kecenderungan setiap tweet dan melakukan plot.

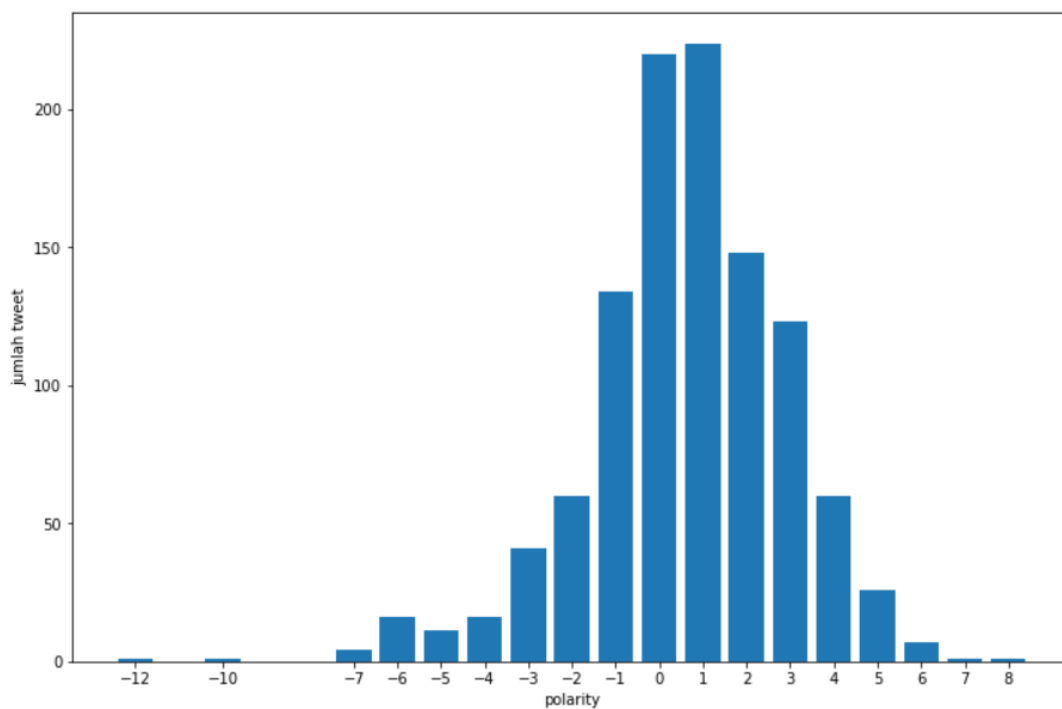
```

# loop kata pos neg
hasil = []
for tweet in clean_tweet:
    count_pos = 0
    count_neg = 0
    for kata_pos in pos_kata:
        if kata_pos.strip() in tweet:
            count_pos+=1
    for kata_neg in neg_kata:
        if kata_neg.strip() in tweet:
            count_neg+=1
    hasil.append(count_pos - count_neg)

labels, counts = np.unique(hasil, return_counts=True)
fig, ax = plt.subplots(figsize=[12,8])
ax.bar(labels, counts, align='center')
ax.set_xticks(labels)
ax.set_xlabel("polarity")
ax.set_ylabel("jumlah tweet")
plt.show()

```

Output :



Gambar 2. Perbandingan polarity dan jumlah tweet

Rules :

- Total Nilai > 0 , maka sentimen positif
- Total Nilai < 0 , maka sentimen negatif
- Total Nilai $= 0$, maka sentimen netral

Dari plot data diatas polarity paling banyak terjadi pada 0 dan 1. Sehingga sentiment yang dominan adalah positif.

```
In [8]: # rata - rata sentiment
print("Hasil rata - rata sentiment : ",np.mean(hasil))

Hasil rata - rata sentiment : 0.6160877513711152
```

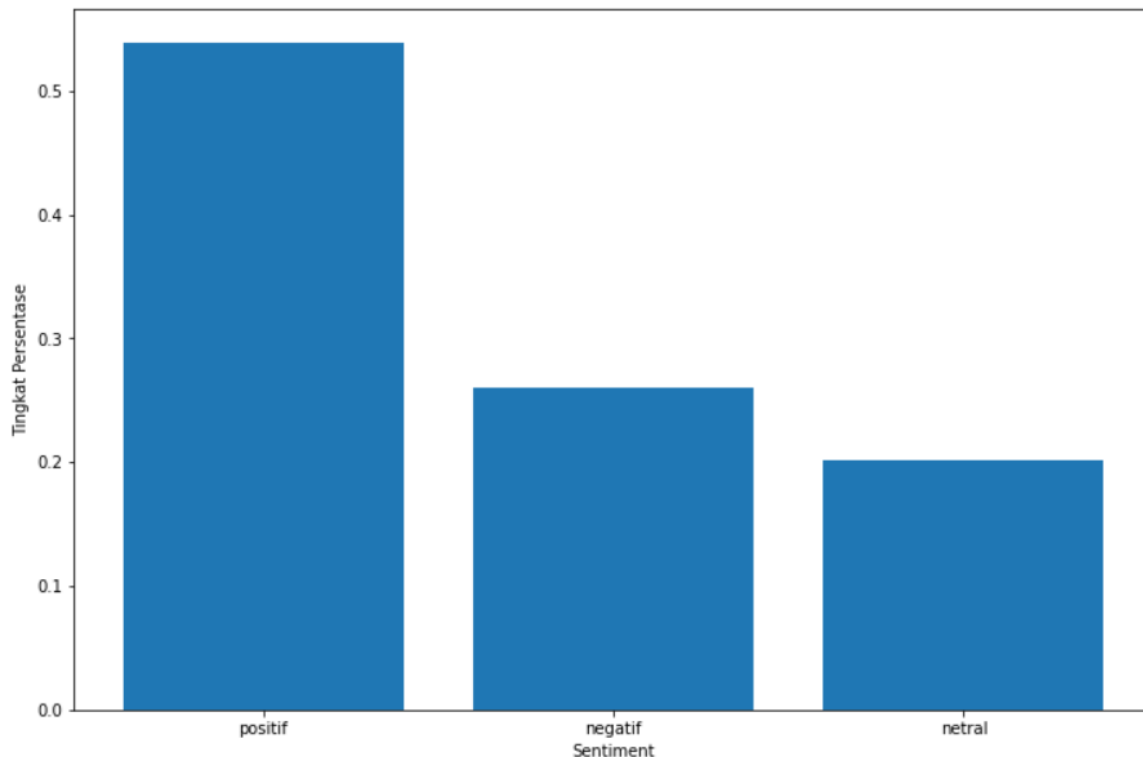
Dilihat dari gambar hasil diatas, rata – rata polarity adalah 0.6160877513711152 , ini berarti duntuk 1094 tweet kecenderungan tanggapan masyarakat positif terhadap bahasan Omnibus Law.

Kemudian langkah selanjutnya mengubah setiap sentiment sesuai dengan rules. Kemudian simpan ke dalam database.

```
# ubah sentiment
sentiments = []
for h in hasil:
    if h > 0:
        sentiments.append(1)
    elif h < 0 :
        sentiments.append(0)
    else :
        sentiments.append(2)
```

Tingkat Persentase Sentiment

Untuk mengetahui ini ambil sentiment dari database, ubah nilai sentiment menjadi nilai stringnya (misal 1 menjadi “positif”) kemudian hitung setiap nilainya, lalu plot. Maka akan dihasilkan :



Gambar 3. Tingkat Persentase Sentiment

Dengan nilai presentase :

```
positif    0.539305
negatif    0.259598
netral     0.201097
Name: sentiment, dtype: float64
```

Sentimen Analisis – Perbandingan Metode SVM, KNN, Naïve-Bayes

Data yang digunakan disini merupakan data yang telah diolah melalui pendekatan polarity. Dengan membagi data train sebesar 80 % dan test sebesar 20 %. Setelah dilakukan prediksi maka didapatkan akurasi dari setiap metode sebagai berikut :

Naïve-Bayes :

```
[[43  9  8]
 [ 8 69 37]
 [13 13 19]]
      precision    recall  f1-score   support

     0       0.67       0.72       0.69         60
     1       0.76       0.61       0.67        114
     2       0.30       0.42       0.35         45

 accuracy                   0.60         219
 macro avg       0.58       0.58       0.57         219
 weighted avg    0.64       0.60       0.61         219

nilai akurasinya adalah 0.5981735159817352
```

SVM :

```
[[ 45 11  4]
 [  7 102 5]
 [ 19 14 12]]
```

	precision	recall	f1-score	support
0	0.63	0.75	0.69	60
1	0.80	0.89	0.85	114
2	0.57	0.27	0.36	45
accuracy			0.73	219
macro avg	0.67	0.64	0.63	219
weighted avg	0.71	0.73	0.70	219

nilai akurasi adalah 0.726027397260274

KNN :

```
[[42 14  4]
 [14 95  5]
 [16 12 17]]
```

	precision	recall	f1-score	support
0	0.58	0.70	0.64	60
1	0.79	0.83	0.81	114
2	0.65	0.38	0.48	45
accuracy			0.70	219
macro avg	0.67	0.64	0.64	219
weighted avg	0.70	0.70	0.69	219

nilai akurasi adalah 0.7031963470319634

Nilai akurasi teringgi diperoleh oleh metode SVM yaitu 0.726027397260274. Kemudian selanjutnya akan dicoba pada data yang diambil pada 29 November 2020 dari pukul 00:00 hingga 12:20 WITA menggunakan metode SVM.

```
from laksmanayudha22_tugas13 import DataHandler, query, database, jumlah_
tweet, twitter_api
data_handler = DataHandler(database)
DataHandler.date_since = "2020-11-29"
data_handler.get_data(twitter_api, query, jumlah_tweet)
data_handler.save_sql()
data_handler.clean_data()

# ambil data
conn = sqlite3.connect('laksmanayudha22_final.db')
cursor = conn.cursor()
```

```

query = """ SELECT cleantweet FROM tweet WHERE createddate > '2020-11-29'
; """
cursor.execute(query)
conn.commit()
data29 = cursor.fetchall()

cursor.close()
conn.close()

# multidimensional ke single dimension
data29 = list(itertools.chain(*data29))

# ML
vectorizer = TfidfVectorizer (max_features=320)
x = vectorizer.fit_transform(df['cleantweet']).toarray()

# pisahkan data feature dan target
y = df.iloc[:, -1]

# data test
x_test = vectorizer.fit_transform(pd.Series(data29)).toarray()

# prediksi
model_s = svm.SVC(kernel='linear')
model_s.fit(x, y)

y_preds = model_s.predict(x_test)

dff = dff.replace({1: 'positif', 0: 'negatif', 2: 'netral'})
dff.value_counts(normalize=True)

# bar plot
data_plot = dff.value_counts(normalize=True)
label = data_plot.index
freq = data_plot.values

# plot data
fig, ax = plt.subplots(figsize=[12, 8])
ax.bar(label, freq)

#set label

```

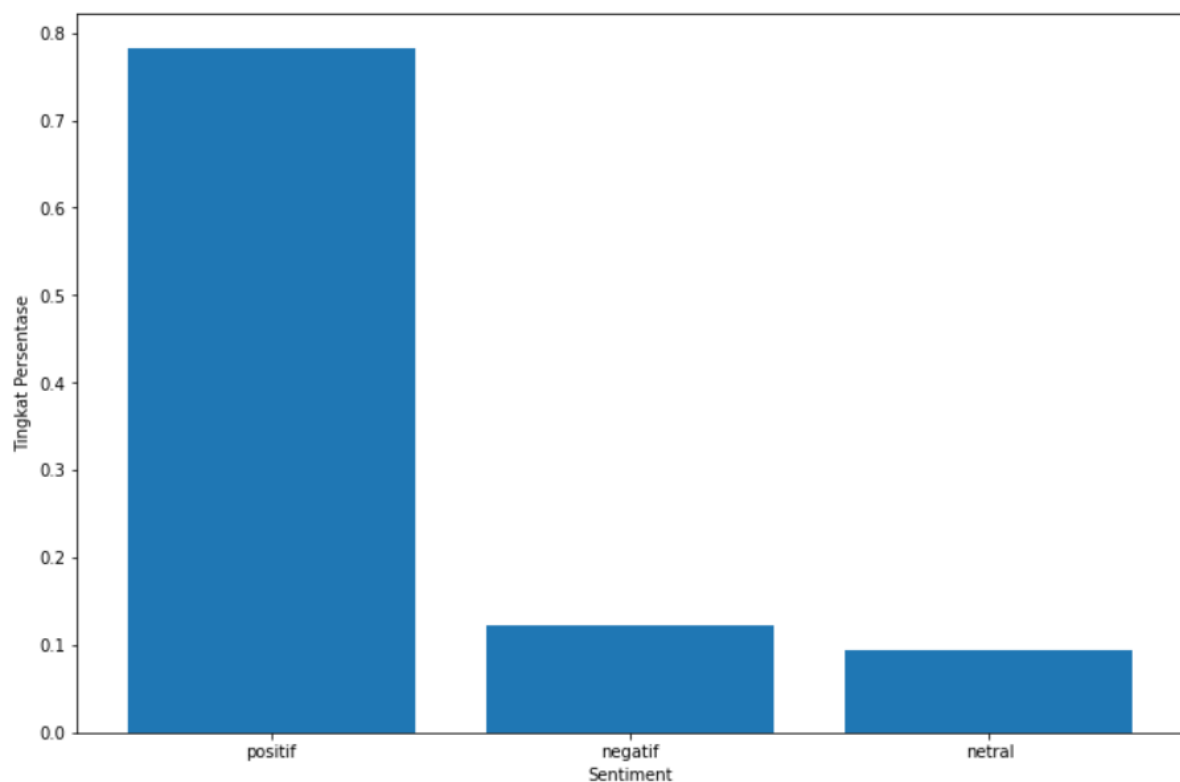


```
ax.set_xlabel('Sentiment')  
ax.set_ylabel('Tingkat Persentase')
```

Setelah dilakukan pengambilan data dan dilakukan prediksi terhadap data tersebut dan dilakukan plot, maka diperoleh :

Persentase sentiment :

```
positif    0.783019  
negatif    0.122642  
netral     0.094340  
dtype: float64
```



Gambar 4. Tingkat Persentase Sentiment Pada 29 November 2020

Dilihat dari gambar diatas, sampai saata ini sentiment masyarakat mengenai Omnibus Law masih cenderung ke arah yang positif.