

***SENTIMENT ANALYSIS MOVIE MENGGUNAKAN ALGORITMA  
MULTINOMIAL NAÏVE BAYES***



**TUGAS UJIAN AKHIR SEMESTER 6 DATA MINING**

**Oleh :**

- 1. Hanan Dwi Wiranata (16090094)**
- 2. Ikhwanudin (16090041)**
- 3. Prieyudha Akadita Sustono (16090115)**

**POLITEKNIK HARAPAN BERSAMA  
TEGAL  
2019**

## 1. JUDUL

“*Sentiment analysis movie menggunakan algoritma multinomial naïve bayes*”

## 2. PENDAHULUAN

### 2.1 Latar Belakang

Hadirnya beberapa situs review film memberikan manfaat bagi mereka yang termasuk *pemilih* dalam menonton film. Seseorang yang sibuk tidak mau menghabiskan uang dan waktunya untuk menonton film yang jelek.

Selain itu, reaksi penonton di berbagai platform juga berguna bagi para kreator film itu sendiri supaya bisa melihat apa yang dikeluhkan dan yang diinginkan oleh audience.

Di sini, kami menggunakan dataset movie reaction dari imdb yang sudah memiliki label/class, anda bisa mengunduhnya di <https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences>. Dataset tersebut punya tiga item, imdb\_labelled.txt untuk sentiment analysis movie (ini yang kita gunakan), lalu yelp\_labelled.txt (resto review) dan amazon\_cells (perangkat seluler amazon).

Kita akan menggunakan python dan berbagai library pendukungnya. Kita juga akan menggunakan twitter untuk crawling data reaksi netizen terhadap suatu film sebagai testing, kita juga akan menggunakan tkinter untuk membuat GUI sederhana untuk

menangkap input kalimat yang anda masukkan untuk dicek sentiment-nya.

## **2.2 Tujuan Penelitian**

Adapun tujuan dilakukan penelitian adalah Mengolah dataset imdb review berlabel menjadi model prediction menggunakan algoritma multinomial naïve bayes untuk bisa membedakan tweet yang kita inputkan (yang tidak terdapat di dataset imdb review yang sudah berlabel) itu positif atau negatif serta menganalisa reaksi netizen twitter terhadap suatu film menggunakan model yang sudah dibuat.

## **2.3 Manfaat Penelitian**

Penelitian ini diharapkan dapat memberikan manfaat bagi :

- a. Menambah ilmu, pengalaman, dan pengetahuan pada bidang Teknologi Informasi khususnya dalam konsep data mining.
- b. Mengetahui sebuah tweet bersifat positif atau negatif dengan persentase akurasi yang besar karna menggunakan model prediction dari dataset berlabel aktual.
- c. Menambah pengetahuan tentang algoritma multinomial naïve

### 3. PEMBAHASAN

#### - Step pertama : Buat Project Python Baru

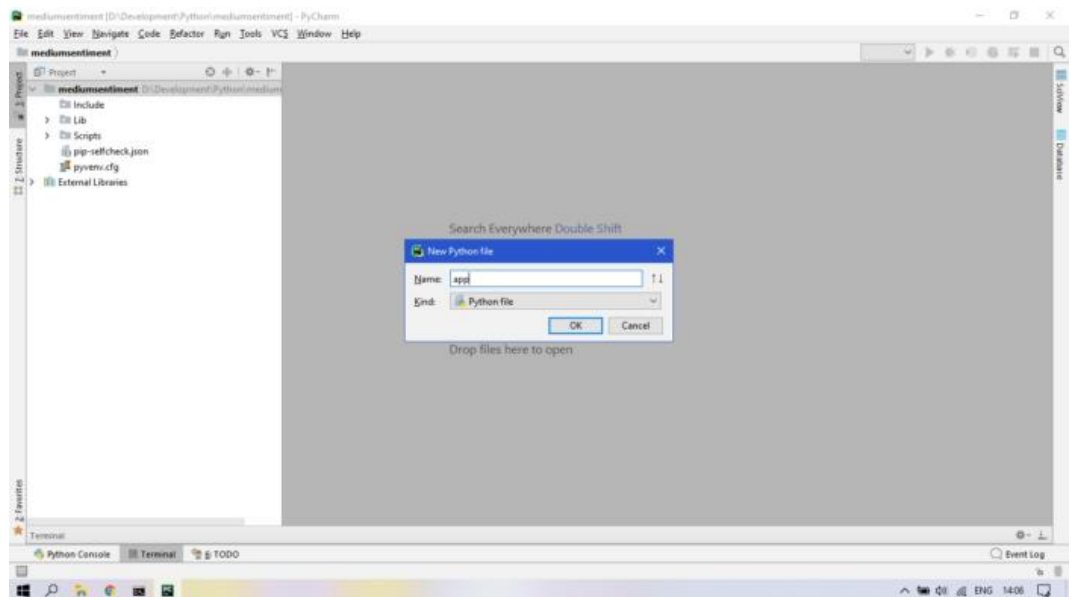
Untuk IDE sendiri kami menggunakan Pycharm Professional 2017. Buatlah Pure Python project lalu pilih create new virtual environment dan namai project anda. Selanjutnya, install-lah library-library berikut menggunakan syntax ***pip install NAMA\_LIBRARY***.

Contoh : *pip install sklearn*

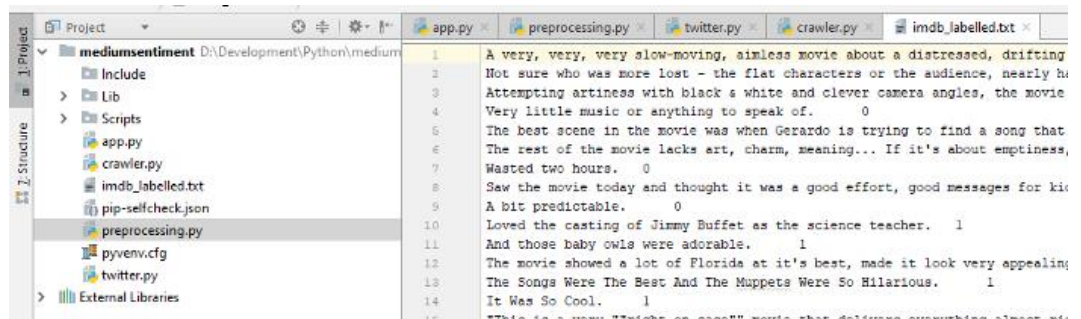
Library yang kita butuhkan:

- pandas
- tweepy
- nltk
- sklearn

Buat python script dengan nama app.py, buat lagi juga dengan nama preprocessing.py, lalu twitter.py lalu yang terakhir crawler.py



Download dataset `imdb_labelled` di atas lalu extract. Cukup ambil *imdb\_labelled.txt* saja lalu paste kan ke dalam project yang sudah kita buat tadi.



## - Step Kedua : Membuat Class Preprocessing Untuk Cleaning Data

Kita akan fokus menulis kode untuk data processing terlebih dahulu.

Editlah `preprocessing.py` menjadi seperti berikut :

```

import string
import re
from string import punctuation
from nltk.corpus import stopwords

class Preprocessing :
    def __init__(self):
        print("Initializing preprocessing...")
        pass

    def processTweet(self, tweet):
        tweet = re.sub(r'\&\w*;', '', tweet)
        tweet = re.sub('@[^\s]+', '', tweet)
        tweet = re.sub(r'\$\w*', '', tweet)
        tweet = tweet.lower()
        tweet = re.sub(r'https?:\/\/\.*\w*', '', tweet)
        tweet = re.sub(r'#\w*', '', tweet)
        tweet = re.sub(r'[' + punctuation.replace('@', '') + ']+', ' ', tweet)
        tweet = re.sub(r'\b\w{1,2}\b', '', tweet)
        tweet = re.sub(r'\s\s+', ' ', tweet)
        tweet = tweet.lstrip(' ')
        tweet = ''.join(c for c in tweet if c <= '\uFFFF')
        return tweet

    def text_process(self, raw_text):
        nopunc = [char for char in list(raw_text) if char not in
string.punctuation]
        nopunc = ''.join(nopunc)
        return [word for word in nopunc.lower().split() if word.lower() not in
stopwords.words('english')]

```

Fungsinya adalah untuk membersihkan data yang kita punya dari simbol-simbol dan karakter yang tidak diinginkan.

Selanjutnya kita edit twitter.py sehingga menjadi seperti ini:

```
import tweepy

class Twitter :
    def __init__(self):
        print("hehe")
        Pass

    def instance(self):
        #Gantilah consumer key,secret dan access key/secret sesuai dengan milik anda,
        #jika belum punya, anda bisa mengajukan ke twitter
        CONSUMER_KEY = "consumer_key_milik_anda"
        CONSUMER_SECRET = "consumer_secret_milik_anda"
        ACCESS_KEY = "access_key_milik_anda"
        ACCESS_SECRET = "access_secret_milik_anda"
        api = tweepy.OAuthHandler(consumer_key = CONSUMER_KEY, consumer_secret =
CONSUMER_SECRET)
        api.set_access_token(ACCESS_KEY, ACCESS_SECRET)
        return tweepy.API(api, wait_on_rate_limit=True,
wait_on_rate_limit_notify=True)
```

Gantilah empat value itu dengan milik anda sendiri. Jika anda belum memiliki API Key twitter, anda bisa mengajukannya via twitter developer. Prosesnya lumayan lama dan anda harus bersabar.

Selanjutnya, buat crawler.py supaya seperti berikut:

```
import tweepy

from preprocessing import Preprocessing
import csv
API = Twitter().instance()
```

```

waitQuery = 100
waitTime = 2.0
engineBlow = 1
Preprocessing = Preprocessing()
csvFile = open('movies_from_twitter.csv', 'w', encoding='utf-8')
csvWriter = csv.writer(csvFile)

def search() :
    global API, waitQuery, waitTime, engineBlow
    query = str(input("Search something : "))
    total_number = int(input("n : "))
    cursor = tweepy.Cursor(API.search, query + " -RT", tweet_mode = "extended", lang =
"en").items()
    count = 0
    error = 0
    secondcount = 0
    while secondcount < total_number:
        try:
            c = next(cursor)
            count += 1
            if count % waitQuery == 0:
                time.sleep(waitTime)
        except tweepy.TweepError:
            print("Sleeping...")
            time.sleep(60 * engineBlow)
            c = next(cursor)
        except StopIteration:
            break

        try:
            text_val = c._json['full_text']
            text_val = str(text_val).lower()
            text_val = Preprocessing.processTweet(text_val)
            if "rt" not in text_val:
                if len(text_val) != 0:
                    secondcount += 1
                    csvWriter.writerow([secondcount, str(text_val)])
                    print("[INFO] Getting a tweet : " + str(secondcount) + " = " +
text_val)
        except Exception as e:

```



```

        error += 1
        print('[EXCEPTION] Stream data: ' + str(e))

search()

```

Crawler.py ini akan kita gunakan untuk mengambil reaksi netizen terhadap suatu film nanti.

Oke, selanjutnya buatlah app.py supaya seperti berikut:

```

from sklearn.pipeline import Pipeline
import pandas as pd
from preprocessing import Preprocessing
from sklearn.externals import joblib
import nltk
import csv
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.naive_bayes import MultinomialNB

imdb_dataset = pd.read_csv("imdb_labelled.txt", sep="\t", header=None)
imdb_dataset.columns = ['text', 'label']
positives = imdb_dataset['label'][imdb_dataset.label == 1]
negatives = imdb_dataset['label'][imdb_dataset.label == 0]
COLNAMES = ["id", "text"]
nltk.download('stopwords')

def word_count(text):
    return len(str(text).split())

```

```

imdb_dataset["word_count"] = imdb_dataset["text"].apply(word_count)
print("Dataset loaded successfully!")

all_words = []
for line in list(imdb_dataset['text']):
    words = line.split()
    for word in words:
        all_words.append(word.lower())

dataset = imdb_dataset

####

dataset.to_pickle("dataset.p")
dataset_pickle = pd.read_pickle("dataset.p")
dataset_pickle['text'] = dataset_pickle['text'].apply(Preprocessing().processTweet)
dataset_pickle = dataset_pickle.drop_duplicates('text')
dataset_pickle.shape

eng_stop_words = stopwords.words('english')
dataset_pickle = dataset_pickle.copy()
dataset_pickle['tokens'] = dataset_pickle['text'].apply(Preprocessing().text_process)

bow_transformer =
CountVectorizer(analyzer=Preprocessing().text_process).fit(dataset_pickle['text'])
messages_bow = bow_transformer.transform(dataset_pickle['text'])
# print('Shape of Sparse Matrix: ', messages_bow.shape)
# print('Amount of Non-Zero occurrences: ', messages_bow.nnz)
print("Dataset dibersihkan!")
print("\nMulai train / test dengan perbandingan training 80% dan testing 20%")
# test nya hanya 20%, training nya 80%

```

```

X_train, X_test, y_train, y_test = train_test_split(imdb_dataset['text'],
imdb_dataset['label'], test_size=0.2)

pipeline = Pipeline([('bow', CountVectorizer(strip_accents='ascii', stop_words='english',
lowercase=True)),('tfidf', TfidfTransformer()), ('classifier', MultinomialNB()), ])

parameters = {'bow__ngram_range': [(1, 1), (1, 2)], 'tfidf__use_idf': (True,
False), 'classifier__alpha': (1e-2, 1e-3), }

grid = GridSearchCV(pipeline, cv=10, param_grid=parameters, verbose=1)
grid.fit(X_train, y_train)

# hasil ->
# print("\nModel: %f using %s" % (grid.best_score_, grid.best_params_))
# print('\n')
means = grid.cv_results_['mean_test_score']
stds = grid.cv_results_['std_test_score']
params = grid.cv_results_['params']
# for mean, stdev, param in zip(means, stds, params):
#     print("Mean: %f Stdev:(%f) with: %r" % (mean, stdev, param))

joblib.dump(grid, "model.pkl")
# buat test model
model_NB = joblib.load("model.pkl")

y_preds = model_NB.predict(X_test)

print('akurasi dari train/test split: ', str(accuracy_score(y_test, y_preds) * 100) + "%")
print('confusion matrix: \n', confusion_matrix(y_test, y_preds))
print(classification_report(y_test, y_preds))

# testing
model_NB = joblib.load("model.pkl")
sample_str = "Shazam movie is very bad. I cant watch it"

```

```

def label_to_str(x):
    if x == 0:
        return 'Negative'
    else:
        return 'Positive'

h = model_NB.predict([sample_str])
print("Kalimat: \n\n'{ }' \nhas a { } sentiment".format(sample_str, label_to_str(h[0])))

x = 0
text_ = [0] * len(imdb_dataset)
label_ = [0] * len(imdb_dataset)

for review in imdb_dataset['text']:
    predict = model_NB.predict([review])
    text_[x] = review
    label_[x] = predict[0]
    x += 1

print("write ke csv")
hehe = {"text": text_, "label": label_}
hehe2 = pd.DataFrame(data=hehe)
hehe2.to_csv('test_ulang_dataset.csv', header=True, index=False, encoding='utf-8')
hasil_test_ulang = pd.read_csv("test_ulang_dataset.csv", header='infer')
hasil_test_ulang.columns = ['text', 'label']

recheck_pos = hasil_test_ulang['label'][hasil_test_ulang.label == "Positive"]
recheck_neg = hasil_test_ulang['label'][hasil_test_ulang.label == "Negative"]

```

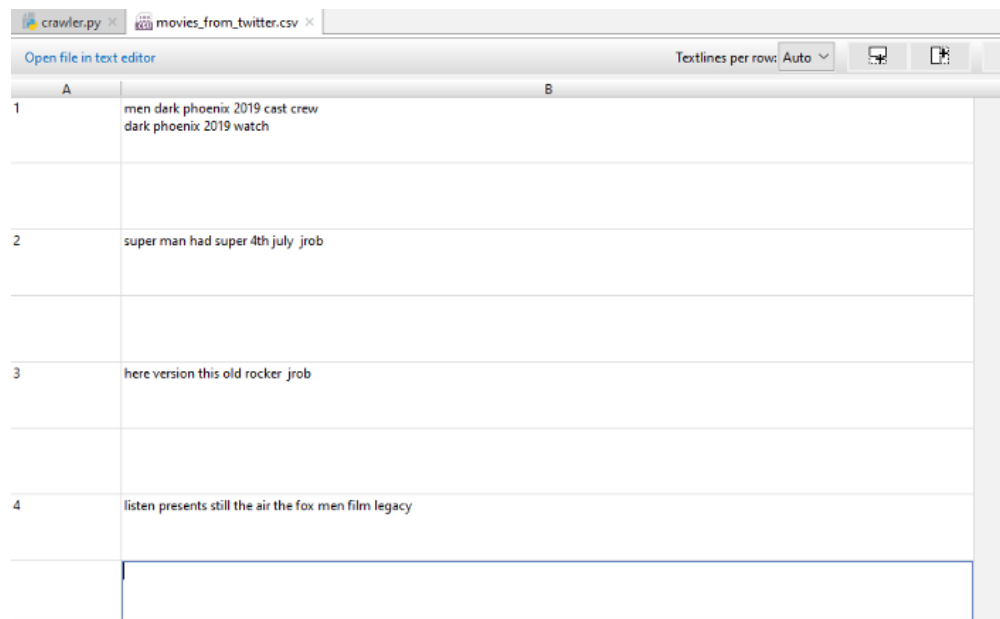
```
print("Hasil test ulang punya positive prediksi sebanyak : "+ str(len(recheck_pos)) +" dan negatif sebanyak " + str(len(recheck_neg)))
```

```
#anda harus menjalankan crawler.py terlebih dahulu untuk mendapatkan movues_twitter.py
print("test data from twitter")
from_twitter = pd.read_csv("movies_from_twitter.csv")
from_twitter.columns = ['id', 'tweet']
csvFile = open('movies_from_twitter_predict.csv', 'w', encoding='utf-8')
csvWriter = csv.writer(csvFile)
for tweet in list(from_twitter['tweet']):
    h = model_NB.predict([tweet])
    csvWriter.writerow([str(tweet), label_to_str(h[0])])
print("DONE!")
```

### - Step Ketiga : Jalankan Program

Pertama, jalankan terlebih dahulu **crawler.py** untuk mengambil data reaksi netizen terhadap suatu film di twitter. Jalankan dengan buka script crawler.py lalu klik “run crawler.py” atau melalui terminal dengan cara *python crawler.py*

Isikan query search dari crawler yang telah kita buat dan berapa jumlah tweet yang mau kita ambil. Sebagai contoh kami akan mencari hashtag #darkphoenix (film X-Men : Dark Phoenix) sebanyak 1000:



A	B
1	men dark phoenix 2019 cast crew dark phoenix 2019 watch
2	super man had super 4th july jrob
3	here version this old rocker jrob
4	listen presents still the air the fox men film legacy

Oke, selanjutnya kita run app.py dengan cara ketik *python app.py*. Kita menggunakan perbandingan 80% training dan 20% testing. Nilai classification\_report yang anda terima terima adalah nilai dari 20% testing tadi. Setelah anda menjalankan app.py akan ada file baru yang dibuat bernama test\_ulang\_dataset.csv yang mengetest ulang seluruh row dari dataset yg kita punya (bukan 20% lagi), hasilnya :

```
test_ulang_dataset.csv
text,label
"very, very, very slow-moving, aimless movie about a distressed, drifting young man. ",0
"Not sure who was more lost - the flat characters or the audience, nearly half of whom walked out. ",0
"Attempting artiness with black & white and clever camera angles, the movie disappointed - became even more ridiculous - as the act
Very little music or anything to speak of. ",0
The best scene in the movie was when Gerardo is trying to find a song that keeps running through his head. ",1
"The rest of the movie lacks art, charm, meaning... If it's about emptiness, it works I guess because it's empty. ",0
Wasted two hours. ",0
"Saw the movie today and thought it was a good effort, good messages for kids. ",1
A bit predictable. ",0
Loved the casting of Jimmy Buffet as the science teacher. ",1
And those baby owls were adorable. ",1
"The movie showed a lot of Florida at it's best, made it look very appealing. ",1
The Songs Were The Best And The Muppets Were So Hilarious. ",1
It Was So Cool. ",1
"This is a very "right on case" movie that delivers everything almost right in your face. ",1
"It had some average acting from the main person, and it was a low budget as you clearly can see. ",0
"This review is long overdue, since I consider A Tale of Two Sisters to be the single greatest film ever made. ",1
"I'll put this gem up against any movie in terms of screenplay, cinematography, acting, post-production, editing, directing, or any
It's practically perfect in all of them a true masterpiece in a sea of faux "masterpieces. ",1
*** The structure of this film is easily the most tightly constructed in the history of cinema. ",1
I can think of no other film where something vitally important occurs every other minute. ",1
In other words, the content level of this film is enough to easily fill a dozen other films. ",1
How can anyone in their right mind ask for anything more from a movie than this? ",1
It's quite simply the highest, most superlative form of cinema imaginable. ",1
Yes, this film does require a rather significant amount of puzzle-solving, but the pieces fit together to create a beautiful pictu
This short film certainly pulls no punches. ",0
Graphics is far from the best part of the game. ",0
This is the number one best TH game in the series. ",1
It deserves strong love. ",1
It is an insane game. ",1
There are massive levels, massive unlockable characters... it's just a massive game. ",1
Waste your money on this game.,,1
```

Hasilnya lumayan jika anda bandingkan dengan imdb\_labelled.txt. yang asli.

Selanjutnya, ada juga file model.pkl, model ini bisa kita gunakan untuk project lain (Nanti kami contohkan membuat GUI). Selain itu juga ada movies\_from\_twitter\_predict.csv, file tersebut berisikan data dari twitter yg kita ambil dengan crawler PLUS dengan prediksinya. Hasilnya seperti berikut :

```

movies_from_twitter_predict.csv ×
' clearly sign ' not into the men anymore when download decent bootleg weeks ago and still haven' watched ,Negative
stupid ending ,Negative
how new beginning when you died ,Positive
don like this scott either too young looking like ,Positive
having love ones always collateral damage the long run they your weakness main target for the enemy ,Positive
honest sad that earned less than considering cunt that one likes literally like not even actors like brie brie can you just
damn white lady ain die after all that ,Positive
took jean getting the understanding family learn joe control her powers for good ,Positive
omg the way she slinged her body thought white last was having exorcist hef neck just broke you can survive those types mov
white lady finally got powerful ,Positive
they should left storm for ,Negative
the soldiers are just hard headed ,Negative
charles admiring his wrongs finally too late take responsibility now ,Positive
these soldiers don even know who the good bad side they attacking everyone that shoot kill mentality bad ,Negative
finally scott the rescue ,Negative
the graphics veins jean face are dope doesn look too demonic excessive ,Negative
reminder check out our men franchise dark phoenix review which available itunes sticher spotify and podbean trashed talk po

```

Hasilnya tidak terlalu buruk untuk ukuran dataset yang kecil.

#### - **Step Keempat : Menggunakan Model Prediction Yang Sudah Dibuat Ke dalam Project Lain**

Buatlah sebuah file python baru bernama gui.py lalu buat filenya supaya seperti berikut :

```

import joblib
import tkinter as tk
from tkinter import messagebox

```



```

model = joblib.load("model.pkl")

def label_to_str(x):
    if x == 0:
        return 'Negatif'
    else:
        return 'Positif'

def btn_event():
    text = editText.get()
    print()
    h = model.predict([text])
    hasil = label_to_str(h[0])
    tk.messagebox.showinfo("Hasil", "Kalimat yang anda masukkan besentimen : "+hasil)

def hehe(text):
    print(text)

form = tk.Tk()
form.title("ydhnbw")

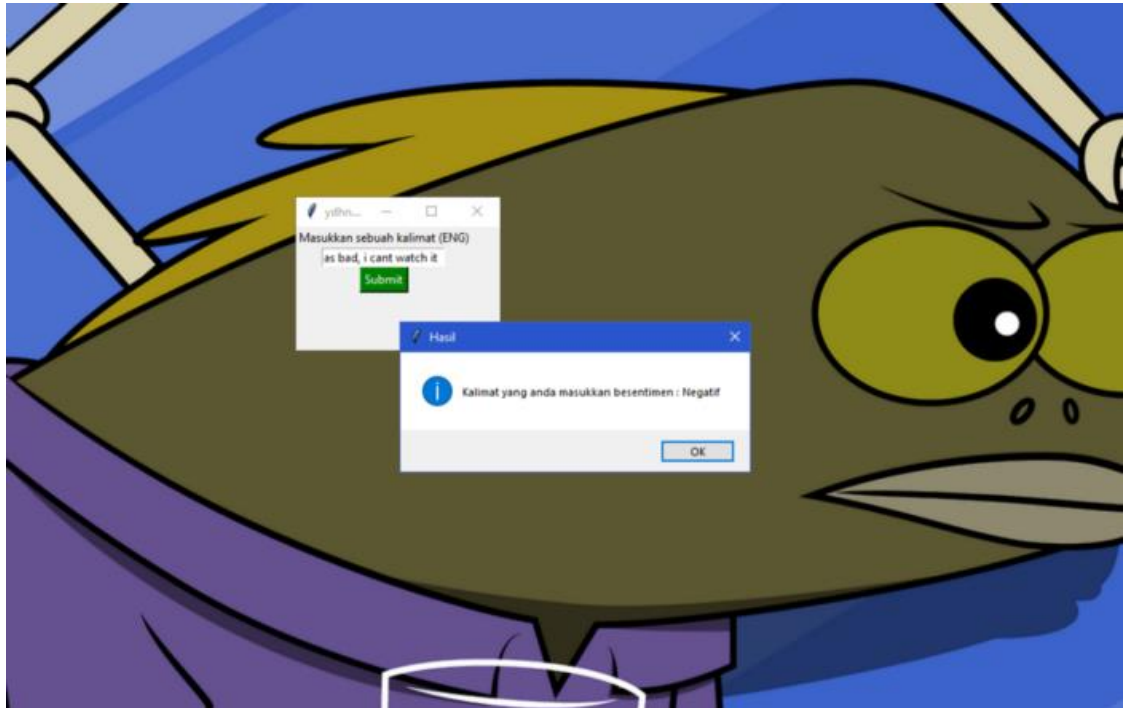
label1 = tk.Label(form, text = "Masukkan sebuah kalimat (ENG)")
editText = tk.Entry(form)
btn = tk.Button(form, text = "Submit", bg = "green", fg = "white", command = lambda :
btn_event())

label1.grid(row = 0)
editText.grid(row = 1)
btn.grid(row = 2)

form.mainloop()

```

Cobalah run gui.py yang anda buat tadi dan masukkan sebuah kalimat dalam bahasa inggris lalu klik submit dan lihat apa yang terjadi. Misalnya kami mengetikkan “*the movie was bad, i cant watch it*”



#### - Step Kelima : Visualisasi Sederhana

Kita perlu menginstall library matplotlib terlebih dahulu dengan cara ketik *pip install matplotlib==3.0.0*

Setelah itu buatlah sebuah python script baru bernama **visualization.py** lalu isi seperti ini

```
import matplotlib.pyplot as plt
import pandas as pd
```

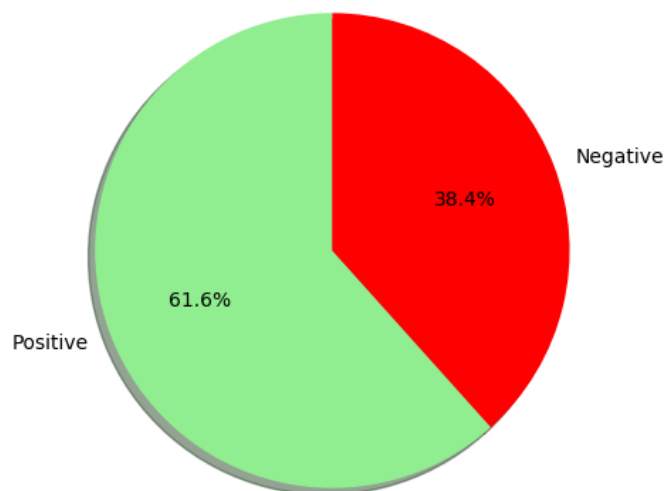
```

from_twitter_predicted = pd.read_csv("movies_from_twitter_predict.csv")
from_twitter_predicted.columns = ['tweet', 'label']
labels = 'Positive', 'Negative'
sizes = [len(from_twitter_predicted[from_twitter_predicted['label'] == "Positive"]),
len(from_twitter_predicted[from_twitter_predicted['label'] == "Negative"])]
colors = ['lightgreen', 'red']

# Plot
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=90)
plt.axis('equal')
plt.show()

```

Cobalah jalankan visualization.py , karena kami menggunakan PyCharm, maka tampilan visualisasi akan dimuat ke dalam SciView tab. Hasilnya seperti berikut :



Seluruh tahapan dan source code dalam laporan ini dimuat dalam artikel yang kami tulis di <https://medium.com/@yudhanewbie/sentiment-analysis-movie-menggunakan-algoritma-multinomial-naive-bayes-e0d9b9fe18bb>

#### - Step Keenam : Evaluasi

Berdasarkan dari 1000 row dataset yang kami punya (500 berlabel positif dan 500 berlabel negatif) dan dicocokkan dengan hasil prediksi menggunakan model yang sudah kami buat, kami menghitungnya dengan cara membuat script untuk mendapatkan nilai confusion matrix. Hasilnya seperti berikut:

```
True positive : 473
True negative : 464
False positive : 36
False negative : 27
Akurasi = 93.7%
Presisi = 92.92730844793712%
Recall = 94.6%
DONE!

Process finished with exit code 0
```

#### 4. KESIMPULAN

Banyak algoritme yang dapat digunakan untuk melakukan sentiment analysis dan multinomial naïve bayes adalah salah satunya. Nilai keakuratan dari model prediksi yang dibuat sangat bergantung bagaimana data yang kita punya dan bagaimana cara kita memprosesnya. Teknik evaluasi dan visualisasi juga penting untuk mengukur bagaimana performa dari model yang sudah kita buat dan juga bagaimana sebuah kumpulan data yang besar dapat dipahami oleh orang awam.