BAB 3
METODE PENELITIAN

Penelitian ini adalah penelitian analisis sentimen positif, negatif dan netral para media sosial Twitter yang menggunakan metode Naive Bayes Classification dan Decision Tree. Dalam penelitian ini membutuhkan data *tweet* yang diperoleh dari Twitter yang berkaitan dengan PT. PLN, selanjutnya akan dilakukan proses *preprocessing* untuk mendapatkan informasi yang di inginkan. Data yang telah berhasil didapatkan dan diolah nantinya digunakan untuk memetakan informasi sentimen di Twitter mengenai PT. PLN sehingga memperoleh informasi yang sesuai mengenai pelayanan, keluhan dan tanggapan masyarakat mengenai kebijakan serta pelayanan PT. PLN.

Penelitian berawal dari latar belakang permasalahan yang ada, kemudian pengolahan data sehingga menghasilkan sentimen serta informasi yang tepat sesuai dengan yang di inginkan. Berikut ini adalah bahan, alat, dan jalanya penelitian yang digunakan dalam menganalisis sentimen PT. PLN menggunakan data *tweet*.

3.1 BAHAN DAN ALAT PENELITIAN

Bahan penelitian yang akan digunakan dalam penelitian ini adalah data dari *tweet, re-tweet* maupun komentar di Twitter yang berhubungan mengenai pelayanan, keluhan serta tanggapan masyarakat mengenai PT. PLN

Alat mencantumkan piranti-piranti yang dipakai untuk melakukan pengolahan atau pemberian perlakuan terhadap bahan penelitian. Sebagai contoh: Alat yang digunakan dalam penelitian ini adalah komputer dengan spesifikasi cukup untuk menjalankan sistem operasi dan perangkat lunak pengembangan serta koneksitas Internet.

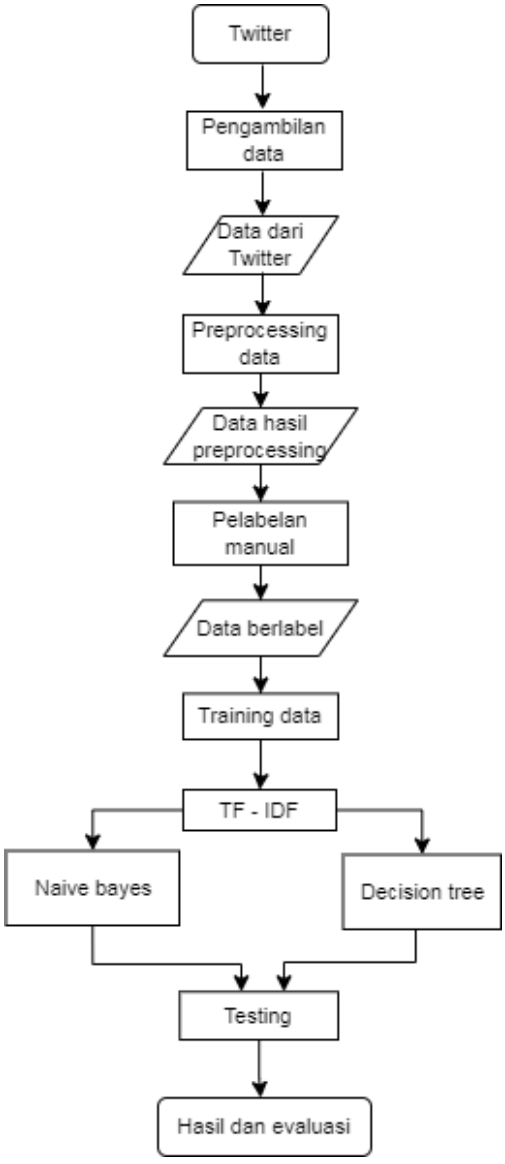
Sistem Operasi dan program-program aplikasi yang dipergunakan dalam dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi: Windows 7 Ultimate 64-bit.
2. Bahasa Pemrograman Python *3.7.3*

3. Microsoft Office Excel 2016.
4. Anaconda 3 64-bit.
5. Jupyter Notebook 6.0.1.
6. Sublime Text.

3.2 JALAN PENELITIAN

Penelitian ini menggunakan bahasa pemrograman Python, Anaconda 3 dan Jupyter Notebook untuk melakukan pengambilan data dari Twitter yang akan ditampilkan di Microsoft Office Excel kemudian akan dimodelkan dengan bantuan *library* pada bahasa pemrograman Python, jalan penelitian seperti pada Gambar 3.1.



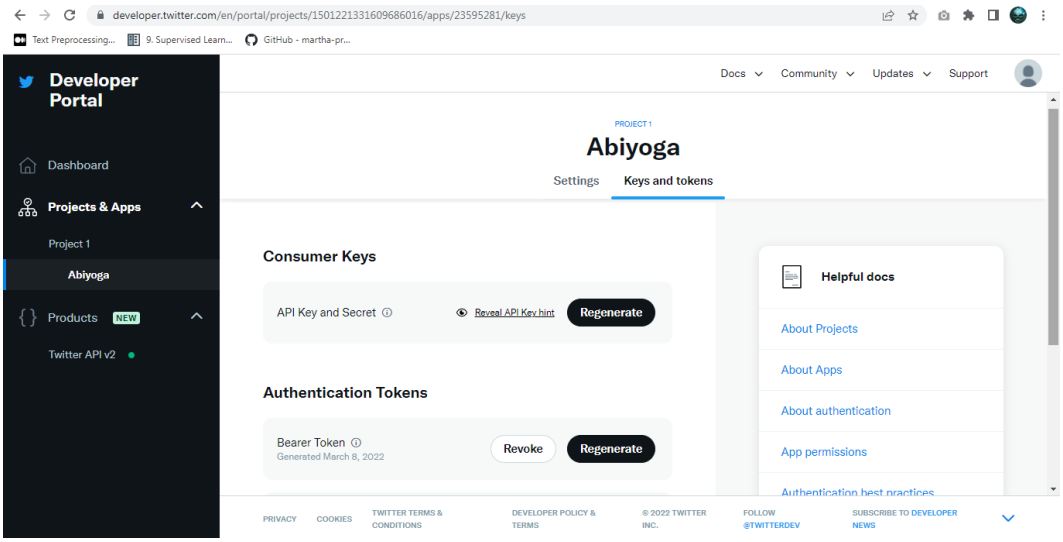
Gambar 3.1 Jalan Penelitian

Berikut merupakan tahapan- tahapan yang akan dilakukan dalam penelitian ini yaitu :

3.2.1 Pengambilan data

Pengambilan data merupakan tahap awal dalam penelitian ini. Dalam tahap ini data diambil dari Twitter mengenai *tweet* tentang PT. PLN dengan menggunakan Anaconda Prompt dengan editor Jupyter Notebook, kemudian data akan ditampilkan dengan Microsoft Excel dengan format .csv. Pengambilan data dilakukan mulai dari bulan 1 mei – 4 juni 2022 dan didapatkan sekitar 40.745 data.

Pengambilan data dilakukan dengan menggunakan API Twitter. API Twitter dilakukan untuk mendapatkan *consumer key*, *consumer secret*, *access token* dan *access token secret*. API berguna sebagai proses sinkronisasi sekaligus perizinan untuk mendapatkan data tweet yang akan dilakukan pengolahan lebih lanjut. Tampilan autentifikasi API yang disediakan Twitter dapat dilihat pada Gambar 3.2.



Gambar 3.2 Twitter Developer

Kemudian dilanjutkan dengan pengambilan data yang dilakukan pada jupyter notebook. Ada beberapa *library* yang digunakan dalam pengambilan data seperti pada Gambar 3.3.

Gambar 3.3 *Library* Pengambilan data

Library tersebut digunakan untuk melakukan pengambian data dari Twitter dengan menggunalan API Twitter. Data mentah yang telah diambil nantinya terdapat URL, *username*, tanggal, dan angka yang perlu untuk dilakukan pembersihan pada proses preprocessing. Berikut merupakan contoh data tweet yang telah di ambil seperti pada Tabel 3.1.

Tabel 3.1 Contoh Data Twitter

No	Tweet
1	2022-05-08 10:11:06+00:00,"b'Hujan besar di Solo sdh berhenti, tp listrik di tipes dsk masih oglangan. @infolistrikdjty @pln_123 https://t.co/Vb3Tgbum7r'"
2	2022-05-08 10:10:48+00:00,b'@pln_123 Sampai saat ini proses perubahan daya sepertinya belum dilaksanakan..apakah petugasnya masih libur? Atw prosesnya memang lama?'
3	2022-05-08 10:10:06+00:00,b'@pln_123 Min call centre PLN sby ga bisa dihubungi seharian . Aplikasi juga eror .ini saya mau lapor gimana? Listri\xe2\x80\xa6 https://t.co/swNuufTU4N'
4	2022-05-08 10:09:19+00:00,b'@astrinidudul Mohon maaf atas gangguan yang terjadi untuk kemudahan akses layanan PLN dapat melalui Aplikasi PLN M\xe2\x80\xa6 https://t.co/DHqWc3pS62'
5	2022-05-08 10:09:05+00:00,"b'Wifi kan bayarnya full sebulan, kalau telat bayar sehari di denda\nTrus tiap hari ada aja erornya gabisa dipake pern\xe2\x80\xa6 https://t.co/mt5GADXPpR'"
6	2022-05-08 10:08:06+00:00,"b'@pln_123 @infopln mohon dibantu, mati lampu sudah 1 jam lebih di Ds Tegalsari kelurahan Selokaton Kec Gondangrejo Kab Karanganyar'"
7	2022-05-08 10:03:50+00:00,"b'@meyruie @pln_123 Sama mbak, di belakang kampus uns juga gitu. Dikit2 mati lampu. Kl mati lampu lama pula. Dikirany\xe2\x80\xa6 https://t.co/d7X19KGXxH'"

8	2022-05-08 10:03:42+00:00,"b'@plnsolo123 selamat sore bpk/ibu, di wil jebres pln off sejak sebelum ashar. ada info penjelasannya?'"
9	2022-05-08 10:02:58+00:00,b'keknya dlu pas w mo kuliah dilarang tuh buat ngambil ujian mandiri alasannya karna uang gk ada dan inget bgt pas uj\xe2\x80\xa6 https://t.co/08TXeamETN'
10	2022-05-08 10:02:36+00:00,b'@pln_123 tolong ini mati listrik udh dari jam 3 kurang sampe jam segini blm nyala \xf0\x9f\x98\x92\n lokasi sepanjang mojosongo-won\xe2\x80\xa6 https://t.co/k15eXqNNuE'

3.2.2 Preprocessing data

Preprocessing data merupakan tahap pengolahan data teks yang baru saja di ambil dengan melakukan tahapan – tahapan untuk memperbaiki data teks yang belum sesuai. *Preprocessing* data bertujuan untuk mempermudah dalam melakukan analisis. *Preprocesing* dilakukan didalam sistem dengan *framework Flask* dan berikut beberapa *library* yang diperlukan seperti pada Gambar 3.4.

app.py | preprocessing.py | training_dc.html | Training.html

```
1  import re
2  import pandas as pd
3  from nltk.corpus import stopwords
4  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
5  from collections import OrderedDict
6  import tweepy
7  import csv
8  import sys
9  import numpy as np
10 import datetime
11 import string, emoji
12 from nltk.corpus import stopwords
13 from nltk.tokenize import word_tokenize
14 from nltk.probability import FreqDist
15
```

Gambar 3. 4 *Library* Preprocessing

Berikut merupakan beberapa tahapan dalam proses *preprocessing* :

1. ***Case Folding***

merupakan konversi dari bentuk awal menjadi bentuk standar (huruf kecil atau *lowercase*). Berikut merupakan kode yang digunakan untuk melakukan *case folding* seperti pada Gambar 3.5 :

```python
# Preprocessing
def lower(data):
    # Case Folding
    return data.lower()
```

Gambar 3.5 *Case Folding*

2. ***Punctuation removal***

Punctual removal yaitu penghapusan terhadap karakter khusus yang tidak memiliki nilai dalam melakukan analisis teks. Berikut merupakan kode yang digunakan untuk melakukan *Punctuation removal* seperti pada Gambar 3.6*:*

```python
def remove_punctuation(data):
    # Happy Emoticons
    emoticons_happy = set([
    ':-)', ':)', ';)', ':o)', ':]', ':3', ':c)', ':>', '=]', '8)', '=)', ':}',
    ':^)', ':-D', ':D', ':d', '8-D', '8D', 'x-D', 'xD', 'X-D', 'XD', '=-D', '=D',
    '=-3', '=3', ':-))', ":'-)", ":')", ':*', ':^*', '>:P', ':-P', ':P', 'X-P',
    'x-p', 'xp', 'XP', ':-p', ':p', '=p', ':-b', ':b', '>:)', '>;)', '>:-)',
    '<3'
    ])

    # Sad Emoticons
    emoticons_sad = set([
    ':L', ':-/', '>:/', ':S', '>:[', ':@', ':-(', ':[', ':-||', '=L', ':<',
    ':-[', ':-<', '=\\', '=/', '>:(', ':(', '>.<', ":'-(", ":'(", ':\\', ':-c',
    ':c', ':{', '>:\\', ';('
    ])

    # All emoticons (happy + sad)
    emoticons = emoticons_happy.union(emoticons_sad)

    data = ' '.join([word for word in data.split() if word not in emoticons])
    data = re.sub(r"b'\@[\w]*", ' ', data)
    data = re.sub(r'\@[\w]+', ' ', data)
    data = re.sub(r"b'[\w]*", ' ', data)
    data = re.sub(r'\w+:\/{2}[\d\w-]+(\.[\d\w-]+)*(?:(?:\/[^\s/]*))*', ' ', data)
    data = re.sub(r'https\:.*$', " ", data)
    data = re.sub(r'[^\w\s]+', ' ', data)
    data = re.sub(r'[0-9]+', ' ', data)
    data = re.sub(r'\$\w*', ' ', data)
    data = data.lower()
    return data
```

Gambar 3.6 *Punctuation removal*

3. *Tokenize*

Tokenize merupkan proses yaitu melakukan pemecahan atau pemisahan karakter menjadi token atau potongan kata tunggal. Berikut merupakan kode untuk melakukan proses *tokenize* seperti pada Gambar 3.7:

```
def tokenize(data):
    data = word_tokenize(data)
    data = " ".join([char for char in data if char not in string.punctuation])
    return data
```

Gambar 3.7 *Tokenize*

4. *Stopword removal*

Stopword removal yaitu pemisahan kata –kata penting dari hasil *tokenizing* dan merupakan kata – kata yang akan digunakan untuk mewakili dokumen. Berikut merupakan kode yg digunakan untuk melakukan *stopword* seperti pada Gambar 3.8.

```
def remove_stopwords(data):
    list_stopwords = (["b", "xef", "x", "xa","n", "xe", "xf","xb", "xad", "xd","xxxxxxx",
                       "xba", "xc", "k", "xcche", "xd xd xaa xd xd xd xd", "m", "t", "xbb", "f",
                       "xbf","xbd","xbc","xab", "xaa", "c""yg", "dg", "rt", "dgn", "ny", "d", "klo",
                       "kalo", "amp", "biar", "bikin", "bilang","gak", "ga", "krn", "nya", "nih", "sih",
                       "si", "tau", "tdk", "tuh", "utk", "ya", "jd", "jgn", "sdh", "aja", "n", "t",
                       "nyg", "hehe", "pen", "u", "nan", "loh", "rt", "tp","&amp", "yah", "w","x","xe","xa", "xc", "x
    list_stopwords = set(list_stopwords)
    data = ' '.join([word for word in data.split() if word not in list_stopwords])
    return data
```

Gambar 3.8 *Stopword removal*

5. *Normalization*

Normalization adalah sebuah proses untuk mengembalikan *term* atau kata kedalam bentuk yang baku, berikut merupakan kode yang digunakan untuk melakukan proses *normalization* seperti pada Gambar 3.9.

```
def normalized_term(data):
    for index, row in normalizad_word.iterrows():
        if row[0] not in normalizad_word_dict:
            normalizad_word_dict[row[0]] = row[1]

    data = ' '.join([normalizad_word_dict[term] if term in normalizad_word_dict else term for term in data.split
    return data
```

Gambar 3.9 *Normalization*

6. Stemming

Steming yaitu pengembalian kata menjadi bentuk dasarnya dengan cara penghilangan infleksinya. Berikut merupakan kode yang digunakan untuk melakukan proses *steming* seperti pada Gambar 3.10.

```
def stem_text(data):
    data = ' '.join([stemmer.stem(word) for word in data.split()])
    return data
```

Gambar 3.10 *Stemming*

Kode diatas berfungsi untuk merubah kata berdasarkan daftar kata yang telah ada di dalam file normalisasi.xlsx. Berikut merupakan contoh daftar kata yang ada di dalam file normalisasi.xlsx terdapat dalam Tabel 3.2.

Tabel 3.2 Daftar Kata Normalisasi

Before	***After***
Naek	Naik
Jg	Juga
Ttp	Tetap
Tp	Tetapi
Hp	Handphone
No	Nomor
Ga	Tidak
Gua	Saya
Gue	Saya
Pake	Pakai

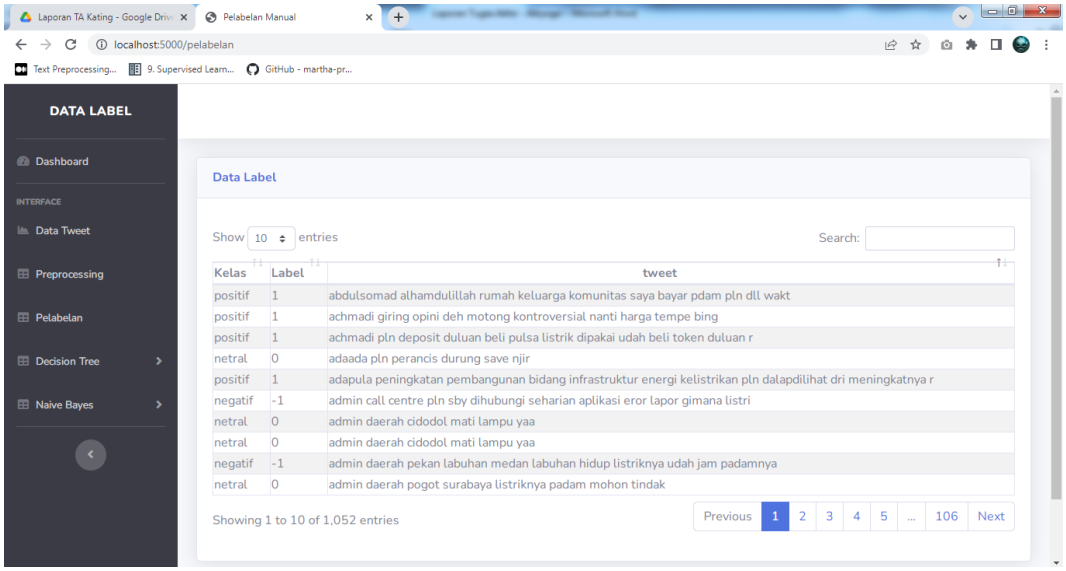
Hasil dari *preprocessing* adalah sebuah file excel dengan data yang sudah bersih, berikut merukan contoh data yang sudah bersih seperti pada Tabel 3.3.

Tabel 3.3 Hasil *Preprocessing*

Tweet
besar di solo sudah henti tetapi listrik di tipes dsk masih oglangan
admin call centre pln sby tidak bisa hubung hari aplikasi juga eror ini saya mau lapor gimana listri
mohon maaf atas ganggu yang jadi untuk mudah akses layan pln dapat lalu aplikasi pln
kan bayar full bulan kalau telat bayar hari di denda ntrus tiap hari ada saja eror gabisa dipake pern
mohon bantu mati lampu sudah jam lebih di desa tegalsari lurah selokaton kec gondangrejo kab karanganyar
sama mbak di belakang kampus uns juga gitu dikit mati lampu kalau mati lampu lama pula dikirany
selamat sore bapak ibu di wilayah jebres pln off sejak belum ashar ada info jelas
dulu pas saya mau kuliah larang tuh buat ngambil uji mandiri alas karena uang tidak ada dan ingat banget pas uj
tolong ini mati listrik udah dari jam kurang sampe jam gin belum nyala lokasi panjang mojosongo won
mohon bantu mati lampu sudah jam lebih di desa tegalsari lurah selokaton kec gondangrejo kab karanganyar

3.2.3 Pelabelan manual

Pelabelan manual adalah proses memberikan label pada kalimat ataupun kata yang ada pada dokumen sehingga dapat dilakukan analisis kalimat tersebut positif atau negatif, contoh pelabelan seperti pada Gambar 3.11 :



Gambar 3.11 Pelabelan Manual

Pelabelan manual dilakukan dengan menggunakan Microsoft Excel dengan data hasil *preprocessing.* Contoh data yang telah diberi label manual dapat dilihat pada Tabel 3.4.

Tabel 3.4 Pelabelan Manual

Kelas	Label	Tweet
Negatif	-1	hujan solo berhenti tetapi listrik tipes dsk oglangan
Netral	0	proses perubahan daya dilaksanakanapakah petugasnya libur atw prosesnya
Netral	0	proses perubahan daya dilaksanakanapakah petugasnya libur atw prosesnya
Netral	0	mohon maaf gangguan kemudahan akses layanan pln aplikasi pln m
Negatif	-1	admin call centre pln sby dihubungi seharian aplikasi eror lapor gimana listri
Negatif	-1	wifi bayarnya full sebulan telat bayar sehari denda trus erornya gabisa dipake pern
Netral	0	mohon dibantu mati lampu jam desa tegalsari kelurahan selokaton kec gondangrejo kab karanganyar
Negatif	-1	mbak kampus uns gitu dikit mati lampu kalau mati lampu dikirany
Netral	0	mohon maaf gangguan kemudahan akses layanan pln aplikasi pln m
Netral	0	selamat sore bapak ibu wilayah jebres pln off ashar info penjelasannya

3.2.4 Training data

Training data adalah proses melatih pada data dengan menggunakan metode Naive Bayes Classification dan Decision Tree. Nantinya data yang telah melewati proses latih akan digunakan untuk perhitungan probabilitas dari data berdasarkan data latih sebelumnya. Proses *training* data akan diawali dengan ekstraksi pada teks dengan menggunakan TF-IDF(*Term Frequency-Inverse Document Frequency*), Contoh perhitungan TF-IDF manual dapat dilihat pada Tabel 3.5

Tabel 3.5 Dokumen TF-IDF

Dokumen(d)	**Kalimat**
d1	padam sekarang padam payah
d2	boleh admin saya sudah dm akun yang ini ya terima kasih
d3	aku harus makan opor dengan suasana candle light dinner
d4	bisa malam takbir begini daerah rumah mati lampu pln cupu
d5	ini gimana sihhhhh ini mau lebaran kok tambah mati lampu

Tabel 3.5 merupakan contoh kalimat yang akan digunakan untuk menghitung TF-IDF sebagai contoh, terdapat 5 dokumen yaitu d1, d2, d3, d4 dan d5. Perhitungan TF menggunakan t atau *term* yaitu kata, d yaitu dokumen kemudian df yaitu banyaknya dokumen dimana suatu *term* muncul. Perhitungan TF dapat dilihat pada Tabel 3.6.

Tabel 3.6 Perhitungan TF

t	d1	d2	d3	d4	d5	df
admin		1				1
Aku			1			1
akun		1				1
begini				1		1
Bisa				1		1
boleh		1				1
candle			1			1
cupu				1		1

daerah				1		1
dengan			1			1
dinner			1			1
Dm		1				1
gimana					1	1
harus			1			1
Ini		1			1	1
kasih		1				1
Kok					1	1
lampu				1	1	2
lebaran					1	1
light			1			1
makan			1			1
malam				1		1
mati				1	1	2
Mau					1	1
opor			1			1
padam	1					1
payah	1					1
Pln				1		1
rumah				1		1
saya		1				1
sekarang	1					1
suasana			1			1
sudah		1				1
takbir				1		1
tambah					1	1
terima		1				1
yang		1				1

Tabel 3.6 menjelaskan pendistribusian setiap *term* pada setiap dokumen, perhitungann IDF menggunkaan komponen seperti *term*, df. IDF adalah tersedianya sebuah term dalam seluruh dokumen dan dihitung dengan N atau banyaknya doumen. Perhitungan IDF dapat dilihat pada Tabel 3.7.

Tabel 3.7 Perhitungan IDF

t	df	idf	idf(N=4)	idf(N=1000)
admin	1	1	0,60206	3
aku	1	1	0,60206	3
akun	1	1	0,60206	3
begini	1	1	0,60206	3
bisa	1	1	0,60206	3
boleh	1	1	0,60206	3
candle	1	1	0,60206	3
cupu	1	1	0,60206	3
daerah	1	1	0,60206	3
dengan	1	1	0,60206	3
dinner	1	1	0,60206	3
dm	1	1	0,60206	3
gimana	1	1	0,60206	3
harus	1	1	0,60206	3
ini	1	1	0,60206	3
kasih	1	1	0,60206	3
kok	1	1	0,60206	3
lampu	2	0,5	0,30103	2,69897
lebaran	1	1	0,60206	3
light	1	1	0,60206	3
makan	1	1	0,60206	3
malam	1	1	0,60206	3
mati	2	0,5	0,30103	2,69897
mau	1	1	0,60206	3
opor	1	1	0,60206	3
padam	1	1	0,60206	3
payah	1	1	0,60206	3
pln	1	1	0,60206	3
rumah	1	1	0,60206	3
saya	1	1	0,60206	3
sekarang	1	1	0,60206	3
suasana	1	1	0,60206	3
sudah	1	1	0,60206	3
takbir	1	1	0,60206	3
tambah	1	1	0,60206	3

terima	1	1	0,60206	3
yang	1	1	0,60206	3

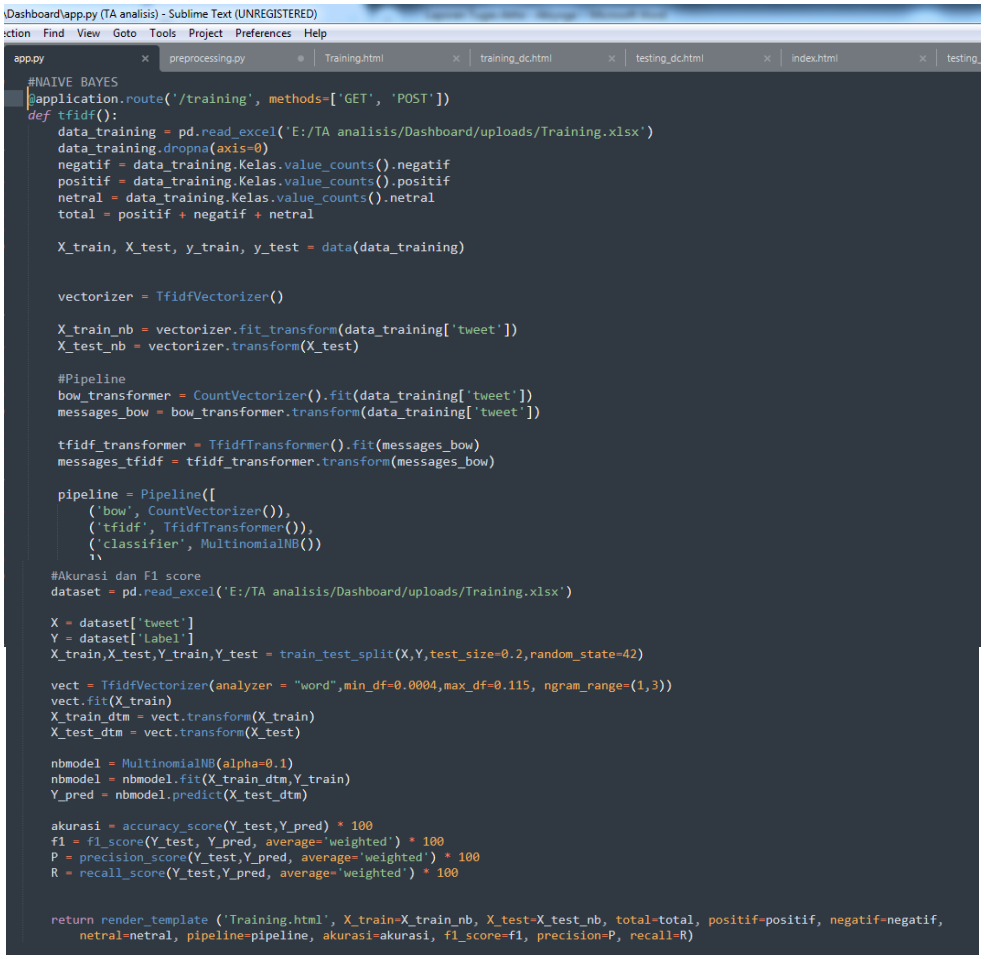
Tabel 3.7 menjelaskan perhitungan tentang idf yang dilakukan secara manual dengan rumus persamaan idf = $\log(\frac{N}{df})$, kemudian hasil perhitungan TF-IDF dapat dilihat dalam Tabel 3.8.

Tabel 3.8 Perhitungan TF-IDF

T	d1	d2	d3	d4	d5
admin		0,60206			
Aku			0,60206		
Akun		0,60206			
begini				0,60206	
Bisa				0,60206	
boleh		0,60206			
candle			0,60206		
Cupu				0,60206	
daerah				0,60206	
dengan			0,60206		
dinner			0,60206		
Dm		0,60206			
gimana					0,60206
Harus			0,60206		
Ini		0,60206			
Kasih		0,60206			
Kok					0,60206
lampu				0,30103	0,30103
lebaran					0,60206
Light			0,60206		
makan			0,60206		
malam				0,60206	
mati				0,30103	0,30103
Mau					0,60206
Opor			0,60206		
padam	0,60206				
payah	0,60206				
Pln				0,60206	
rumah				0,60206	
Saya		0,60206			

sekarang	0,60206				
suasana			0,60206		
sudah		0,60206			
takbir				0,60206	
tambah					0,60206
terima		0,60206			
Yang		0,60206			

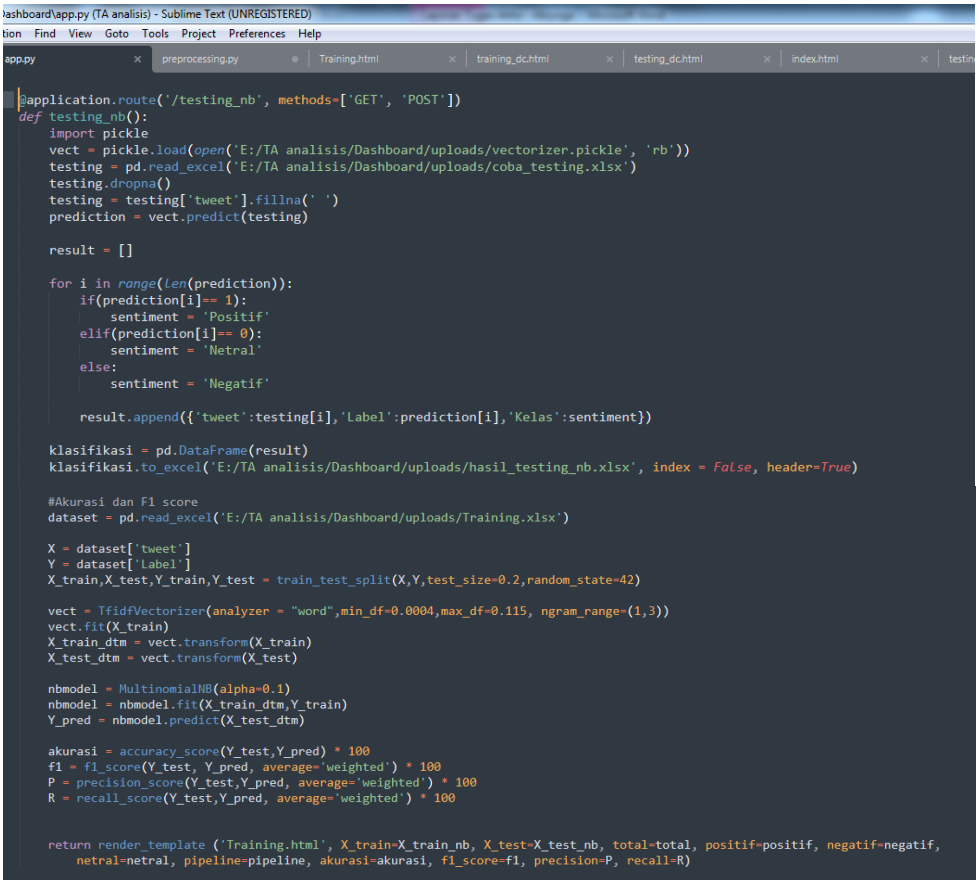
kemudian *Training data* agar dapat membuat model klasifikasi yang dapat digunakan untuk melihat sentimen dari data secara otomatis seperti pada Gambar 3.12.



Gambar 3.12 *Training* dan Model Klasifikasi

3.2.5 Testing

Testing adalah tahapan untuk mengetahui keakuratan dari pemodelan yang sudah dibangun pada tahap *training* untuk memprediksi label atau kelas dari data uji yang telah dibuat. Model yang telah didapatkan akan dihitung dengan menggunakan metode yang ada dalam *confusion matrix* untuk mengetahui persentase yang dilakukan pada setiap pengujian. Metode dalam *confusion matrix* meliputi beberapa tahap, Berikut merupakan kode *testing* seperti pada Gambar 3.13.



Gambar 3.13 *Testing*

1. *Accuracy* didapatkan dengan cara perbandingan antara data yang telah terklasifikasi benar dengan keseluruhan data.
2. *Precision* didapatkan dengan cara jumlah data yang diklasifikasi benar dibagi dengan total data yang diklasifikasi positif.

3. *Recall* digunakan untuk menunjukan seberapa besar persentase data yang terkategori positif yang terklasifikasi dengan benar oleh sistem.
4. *F- measure* digunakan untuk menghitung rata- rata antara *precision* dan *recall.*

Berikut merupakan kode perhitungan *confusion matrix* yang digunkan seperti pada Gambar 3.14.

```
#Akurasi dan F1 score
dataset = pd.read_excel('E:/TA analisis/Dashboard/uploads/hasil_testing_nb.xlsx')

X = dataset['tweet']
Y = dataset['Label']
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=42)

vect = TfidfVectorizer(analyzer = "word",min_df=0.0004,max_df=0.115, ngram_range=(1,3))
vect.fit(X_train)
X_train_dtm = vect.transform(X_train)
X_test_dtm = vect.transform(X_test)

nbmodel = MultinomialNB(alpha=0.1)
nbmodel = nbmodel.fit(X_train_dtm,Y_train)
Y_pred = nbmodel.predict(X_test_dtm)

akurasi = accuracy_score(Y_test,Y_pred) * 100
f1 = f1_score(Y_test, Y_pred, average='weighted') * 100
P = precision_score(Y_test,Y_pred, average='weighted') * 100
R = recall_score(Y_test,Y_pred, average='weighted') * 100
```

Gambar 3.14 *Confusion Matrix*

3.2.6 Hasil dan Evaluasi

Hasil dan Evaluasi adalah tahapan terakhir dimana hasil data yang telah diolah akan divisualisasikan menggunakan *dashboard* dengan *framework Flask* . Pada tampilan *dashboard* terdapat fitur untuk melihat hasil perbandingan dari kedua metode, jumlah data, data *training* dan dan data *testing*.