

MULTI CHANNEL ACCESS
“TEXT CLASSIFICATION UNTUK MENGANALISIS SENTIMEN
MASYARAKAT TERHADAP LAYANAN INTERNET INDIHOME
MENGGUNAKAN PYTHON BERBASIS ALGORIT=MA NAÏVE BAYES”



Oleh:

Shilta Inda Qurroti A'yun Achmadi (1905551002)

TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS UDAYANA

2021

BAB I

PENDAHULUAN

Bab I merupakan pendahuluan dari laporan berjudul “Text Classification Untuk Menganalisis Sentimen Masyarakat Terhadap Layanan Internet Indihome Menggunakan Python Berbasis Algoritma Naïve Bayes”. Pendahuluan ini berisi latar belakang dari percobaan yang dilakukan. Penjelasan dapat dilihat sebagai berikut.

1.1 Latar Belakang

Seiring berkembangnya teknologi informasi, kini kita dapat dengan mudah mengakses segala informasi yang ada di manapun. Hampir semua orang menggunakan perangkat yang terhubung dengan internet. Untuk tersambung ke internet kita memerlukan koneksi internet yang biasanya bisa kita beli pada provider penyedia layanan internet. Salah satu provider layanan internet adalah Indihome, pengguna Indihome kini sudah mencapai 8,3 juta pelanggan, jumlah itu tentunya tidak sedikit, namun dibalik banyaknya pengguna Indihome, banyak terdapat komentar mengenai layanan Indihome yang masih kurang di berbagai sosial media.

Dari masalah di atas, penulis tertarik untuk menganalisis sentimen masyarakat terhadap layanan Indihome berdasarkan komentar masyarakat di media sosial Twitter. *Sentiment analysis* adalah proses penggunaan *text analytics* untuk mendapatkan berbagai sumber data dari internet dan beragam platform media sosial. Tujuannya adalah untuk memperoleh opini dari pengguna yang terdapat pada platform tersebut. Twitter adalah jaringan informasi *real-time* yang menghubungkan penggunanya dengan cerita, ide, pendapat dan berita terbaru tentang apa saja yang dianggap menarik oleh banyak orang. Twitter adalah salah satu jaringan sosial dan layanan mikroblogging yang memungkinkan penggunanya untuk mengirim dan membaca pesan hingga 140 karakter yang disebut tweet. Penulis mengambil data dari tweet mengenai indihome. Nantinya data tersebut akan dianalisis hingga dapat mengetahui bagaimana tanggapan seseorang mengenai layanan Indihome tersebut.

BAB II

GAMBARAN UMUM SISTEM

Bab II berisikan gambaran umum dari percobaan yang akan dilakukan. Berikut adalah penjabaran dari gambaran umum percobaan yang akan dilakukan oleh penulis.

2.1 Gambaran Umum Grabbing Data

Grabbing data dilakukan menggunakan *library open source* bernama *snsrape*. *Snsrape* merupakan *library* yang berbasis *python* untuk mengambil data dari sosial media, baik Instagram, twitter, facebook, dan lain sebagainya. Pengambilan data menggunakan keyword “Indihome” dan mengambil sebanyak 100 *tweet* dari twitter. Setelah data berhasil diambil, data berupa teks tersebut dilakukan pembersihan atau *cleansing* dengan menghilangkan karakter-karakter yang tidak diperlukan, juga menghilangkan emoji, *link*, maupun hastag. Setelah data teks bersih, dilakukan penyimpanan ke *file* berupa CSV. Data ini lah yang nantinya digunakan untuk menganalisis sentimen.

2.2 Gambaran Umum Sentiment Analysis

Sentimen analisis dibuat menggunakan *library* *TextBlob*, data yang telah dilakukan *cleansing* atau dibersihkan dari karakter-karakter yang tidak digunakan akan dimasukkan kedalam *file* CSV, kemudian data tersebut akan disimpan atau *diimport* ke dalam *file* MySQL. Langkah selanjutnya adalah melakukan *polarity* dan *subjectivity*, namun sebelumnya karena *tweet* yang diambil menggunakan bahasa Indonesia, terlebih dahulu harus dilakukan translate kedalam bahasa Inggris karena *library* *textblob* hanya bisa digunakan pada *text* yang menggunakan bahasa Inggris. Setelah itu, melakukan *polarity* untuk menentukan nilai dari sebuah *tweet*, nantinya dari nilai tersebut kita dapat menentukan apakah *tweet* tersebut bersifat positif, *negative*, ataupun netral. Kemudian, di paling akhir akan dilakukan uji coba dengan memasukkan sebuah *text*, lalu *textblob* akan melakukan klasifikasi *text* tersebut berdasarkan data training yang telah dilakukan sebelumnya, apakah *tweet* tersebut bersifat *negative*, positif, ataupun netral.

BAB III

PEMBAHASAN

Bab III merupakan pembahasan dari gambaran umum yang sudah dijelaskan pada Bab II sebelumnya. Berikut adalah pembahasannya.

3.1 Grabbing Data

Grabbing data dilakukan menggunakan *library* *snsrape*, *text editor* yang digunakan adalah Visual Studio Code dengan Jupyter Notebook. Menggunakan *library* *snsrape* tidak memerlukan API Twitter. Langkah-langkah untuk mengambil data adalah sebagai berikut.

```
import os
from datetime import date
from textblob import TextBlob
import pandas as pd
import mysql.connector as mysql
import re
import nltk
import csv
nltk.download('words')
words = set(nltk.corpus.words.words())
```

Kode Program 3.1 Mengimport File dari Module

Melakukan *import file* yang diperlukan dari *module* yang telah diinstall pada tahap sebelumnya. *Library* yang digunakan adalah *textblob*, *pandas*, *mysql.connector*, dan *library* lainnya. Lakukan *import file* agar *library* dapat digunakan pada program.

```
search_term = "indihome"
from_date = "2021-09-10"
end_date = "2021-11-10"
max_results = 100
```

Kode Program 3.2 Inisialisasi Keyword dan Tanggal Pencarian

Inisialisasi *variable* *search_term* untuk *keyword* yang digunakan mencari *tweet* yang diperlukan dalam sentiment analisis, *from_date* berguna sebagai tanggal awal pencarian *tweet*, *end_date* adalah tanggal_akhir dari

pencarian *tweet*, dan `max_results` adalah maksimal dari *tweet* yang akan diambil. *Keyword* yang digunakan adalah “indihome” dengan pencarian awal pada tanggal 10 September 2021 hingga 10 November 2021 dengan maksimal pencarian sebanyak 100 *tweets* mengenai Indihome.

```
tweets = 'snsrcape --jsonl --max-results {} --since {}
twitter-search "{} until:{}"> result-
tweets.json'.format(max_results, from_date, search_term,
end_date)
os.system(tweets)
df = pd.read_json('result-tweets.json', lines=True)
```

Kode Program 3.3 Melakukan Grabbing Data

Setelah inisialisasi semua *variable* yang dibutuhkan untuk melakukan *grabbing* data, langkah selanjutnya adalah melakukan *grabbing* data yang dibantu menggunakan *library* dari *snsrcape*. Data yang didapatkan akan disimpan kedalam *file* json dengan nama *result-tweets.json*. isi dari *file* json tersebut adalah sebagai berikut.



Kode Program 3.4 File Hasil Grabbing Data

Gambar di atas merupakan *file* json yang digunakan untuk menyimpan data *tweet* yang telah diambil, data *tweet* tersebut masih terdapat banyak *field* seperti *url*, nama pengguna, dan lain sebagainya, serta data dari *tweet* yang masih mengandung karakter-karakter yang tidak diperlukan harus dilakukan *cleansing*

terlebih dahulu. Untuk melakukan *cleansing* dilakukan dengan melakukan *remove* karakter seperti *link*, *hashtag*, dan lainnya. Proses *cleansing* dapat dilihat pada kode program di bawah ini.

```
def cleaner(tweets):
    tweets = " ".join(re.sub("@[A-Za-z0-9+]|([^\0-9A-Za-z
\t])|(\w+:\w+\/\S+)", " ", tweets).split())
    tweets = " ".join(re.sub(r"http\S+", " ",
tweets).split())
    tweets = " ".join(w for w in
nltk.wordpunct_tokenize(tweets))
    return tweets
df['content'] = df['content'].map(lambda x: cleaner(x))
df.to_csv('data.csv') #specify location
for row in df['content'].iteritems():
    print(row)
```

Kode Program 3.5 Cleansing Data

Cleansing data dilakukan untuk menghapus karakter-karakter yang tidak diperlukan. Menghapus karakter-karakter tersebut dengan membuat fungsi bernama *cleaner* dan melakukan *return* berupa data yang sudah dibersihkan, lalu menyimpannya ke dalam *file* json.

```
(0, 'AktivitasTanpaBatas InternetnyaIndonesia BerjuangTanpaBatas IndiHome')
(1, 'Jangan terlalu cepat menghakimi tanpa pernah mau mengerti sudut pandang yang lain Untuk urusan jaringan internet percayakan pad
(2, 'oh God i really HATE indihome')
(3, 'AktivitasTanpaBatas InternetnyaIndonesia BerjuangTanpaBatas IndiHome')
(4, 'tolong dong ini kenapa yah Apa ada maintenance')
(5, 'Siap Kak Rose Selamat menikmati layanan IndiHome dan menjalani aktivitas hari ini Semoga harinya menyenangkan Terima kasih Esha
(6, 'Dear bayar mahal kecepatannya wifinya yang tinggi tp tetep aja lemot wifinya')
(7, 'Halo Kak Ariya Terima kasih atas apresiasinya ya Apabila ada pertanyaan lain seputar layanan IndiHome silakan hubungi kami kemb
(8, 'Indihome astaga')
(9, 'Halo Kak Rose Terima kasih sudah berpartisipasi meramaikan kuis serta atas apresiasinya Semoga Kakak beruntung ya Stay safe Sta
(10, 'Indihome lg trouble ya')
(11, 'ujan gini indihome bertingkah kuota pake acara abis segala WKWKWKWK')
(12, 'kenapa internet nya lemot kalau di tv buat youtube viu sama weetv muternya lama padahal kalau di hp sama laptop lancar interne
(13, 'Hai Sobat Bisa sekali nih Sobat nantinya langsung dibantu dari sini bisa juga registrasi pasang baru IndiHome melalui aplikasi
(14, 'Halo Kak Ratih Terima kasih banyak untuk apresiasinya semoga IndiHome selalu membantu dan mempermudah menjalani aktivitasnya S
(15, 'Customer service IndiHome oke banget gercep amp care TOP deh')
(16, 'Happy pake IndiHome jaringan bagus walau masuk pedalaman Thanks')
(17, 'Tentunya dong Kak Rayad Terima kasih atas apresiasinya ya Kak Selamat menikmati layanan IndiHome dan menjalani aktivitas hari
(18, 'Alhamdulillah jaringan mantul WFH lancar Makaziiiiii IndiHome')
(19, 'quenaja hip hip horrayyy udh keluar pemenangnya keren indihome')
(20, 'IndiHome luar biasa sangat lancar untuk WFH')
(21, 'Pagi Kak Nina Ada dong Kak Kami memiliki layanan IndiHome Smart Camera paket sewa Rp25 00 0 00 bulan PPN 10 Apakah Kakak bermi
(22, 'Indihome koneksinya lancar terus keren deh')
(23, 'Pagi IndiHome ada CCTV ga')
(24, 'Indihome mati kok gak viral yah kayak yg di Jakarta')
```

Kode Program 3.6 Melakukan Pencarian

Gambar di atas merupakan hasil *return* dari data yang sudah dibersihkan, yaitu hanya data konten saja, setelah bersih, data akan disimpan ke dalam *storage* menggunakan database MySQL.

```
df = pd.read_csv('data.csv', usecols=['content'])
df.to_csv('data_konten.csv') #specify location
df.head()
```

Gambar 3.1 Menemukan Tweet

Lakukan *read file* csv dengan menggunakan kolom *content* saja, karena untuk analisis hanya menggunakan data *tweet* atau data konten. Kita coba lakukan visualisasi data menggunakan dataframe, seperti berikut.

	content
0	AktivitasTanpaBatas InternetnyaIndonesia Berju...
1	Jangan terlalu cepat menghakimi tanpa pernah m...
2	oh God i really HATE indihome
3	AktivitasTanpaBatas InternetnyaIndonesia Berju...
4	tolong dong ini kenapa yah Apa ada maintenance

Kode Program 3.7 Inisialisasi Maksimal Tweet

Gambar di atas merupakan *tweet* yang telah dibersihkan atau dilakukan *cleansing*. Data yang digunakan hanya data *content* atau *tweets* saja. Setelah itu, data tersebut akan dilakukan penyimpanan ke database MySQL.

3.2 Data Storage

Storage dilakukan untuk menyimpan data hasil *cleansing* kedalam sebuah *database* SQL yaitu menggunakan *database* MySQL. Proses untuk menyimpan data tersebut adalah sebagai berikut.

```
tweetData = pd.read_csv('data_konten.csv', index_col=False)
tweetData.head()

tweetData = mysql.connect(host='localhost', user='root',
passwd='', db='indihome_data')
cursor = tweetData.cursor()

csv_data = csv.reader(open('data_konten.csv'))
next(csv_data, None)
```

```

for row in csv_data:
    cursor.execute('INSERT INTO data(id, tweet) VALUES(%s, %s)', row)
tweetData.commit()
cursor.close()
print('Done')

```

Kode Program 3.8 Melakukan Storage Data

Data yang sebelumnya menggunakan csv akan diubah menjadi bentuk MySQL *database*, cara mengubahnya dengan memanfaatkan *library* pymysql, csv, dan juga pandas. Pandas membantu untuk melakukan *read* data, kemudian pymysql digunakan untuk melakukan koneksi ke mysql server secara *local*. Dari data csv kemudan dilakukan *insert* ke tabel yang sudah dibuat.

+ Options	
id	tweet
0	AktivitasTanpaBatas InternetnyaIndonesia BerjuangT...
1	Jangan terlalu cepat menghakimi tanpa pernah mau m...
2	oh God i really HATE indihome
3	AktivitasTanpaBatas InternetnyaIndonesia BerjuangT...
4	tolong dong ini kenapa yah Apa ada maintenance
5	Siap Kak Rose Selamat menikmati layanan IndiHome d...
6	Dear bayar mahal kecepatannya wifinya yang tinggi ...
7	Emang ni indihome semalem bikin kzl kzl kzl
8	Halo Kak Ariya Terima kasih atas apresiasinya ya A...
9	Indihome astaga
10	Halo Kak Rose Terima kasih sudah berpartisipasi me...

Gambar 3.2 Hasil Storage Data ke MySQL

Data yang sudah dimasukkan ke dalam *database* MySQL akan terlihat seperti gambar di atas, terdapat 2 *field* yaitu *id* sebagai nomor *tweet* dan *field* *tweet* untuk menyimpan konten *tweet*.

3.3 Training Data

Training data dilakukan untuk menambahkan label ke data yang sudah kita dapatkan, fungsinya agar nantinya program dapat melakukan klasifikasi berdasarkan data yang sudah *ditraining* atau dilatih. Proses dari *training* data adalah sebagai berikut.


```
def translate(x):
    try:
        blob=TextBlob(x)
        if(blob.detect_language() != 'en'):
            return (str(blob.translate(to = 'en')))
    except:
        return None

df=pd.read_csv('data_konten.csv')
df['content'] = df.content.astype(str) ##converting column
to string where translation is required
df['translated'] = df['content'].apply(lambda x:
translate(x)) #Translation happening
df.to_csv('translated.csv') #specify location
```

Kode Program 3.9 Melakukan Translate Data

Library textblob hanya bisa digunakan untuk *text* yang menggunakan bahasa inggris, namun *tweet* yang diambil kebanyakan menggunakan bahasa Indonesia, jadi harus dilakukan *translate* terlebih dahulu, textblob juga sudah menyediakan fungsi *translate* yang dapat digunakan untuk melakukan *translate* kedalam bahasa Inggris. Data yang *ditranslate* diambil dari *data_konten.csv*, kemudian hasil dari *translate* tersebut disimpan ke dalam csv dengan nama *translated.csv*.

```
df = pd.read_csv('translated.csv', usecols=['translated'])
df.to_csv('translated_tweets.csv') #specify location
df.head()
```

Kode Program 3.10 Melakukan Storage Data

Setelah disimpan, kita akan mengambil data hanya menggunakan kolom “*translated*” yaitu kolom yang menyimpan hasil *tweet* yang sudah dilakukan *translate*. Kemudian kita coba untuk melihat isi dari datanya sebagai berikut.

	translated
0	Internet Unlimited ActivitiesIndonesia Struggl...
1	Don't be too quick to judge without ever wanti...
2	NaN
3	Internet Unlimited ActivitiesIndonesia Struggl...
4	please, why is this, is there any maintenance

Gambar 3.3 Hasil Storage Data ke MySQL

Gambar di atas merupakan hasil *translate tweet* dari bahasa Indonesia ke bahasa Inggris. Jika data sudah berhasil menghasilkan bahasa Inggris, maka siap untuk dilakukan analisis *sentiment*. Proses analisis *sentiment* dapat dilihat pada kode program berikut.

```
def getSubjectivity(text):
    return TextBlob(text).sentiment.subjectivity

def getPolarity(text):
    return TextBlob(text).sentiment.polarity

df['translated'] = df.translated.astype(str)
df['Subjectivity'] = df['translated'].apply(lambda x:
getSubjectivity((x)))
df['Polarity'] = df['translated'].apply(lambda x:
getPolarity(x))

def getAnalysis(score):
    if score < 0:
        return 'Negative'
    elif score == 0:
        return 'Neutral'
    else:
        return 'Positive'
df['analisis'] = df['Polarity'].apply(getAnalysis)
df.to_csv('analisis.csv') #specify location
df
```

Kode Program 3.11 Melakukan Storage Data

Melakukan analisis *sentiment* memanfaatkan *library* *textblob* dengan menggunakan fungsi `getSubjectivity` dan `getPolarity`. Dari data *translated* kemudian dilakukan *apply* untuk mendapatkan nilai *subjektivitas* dan juga *polarity*. Kemudian dari nilai tersebut akan dilakukan penilaian jika *score* atau nilai kurang dari 0 maka *tweet* tersebut bersifat *negative*, jika sama dengan 0 maka netral dan selain itu akan bernilai positif. Kemudian membuat satu kolom untuk menyimpan hasil analisis dan disimpan kedalam file csv bernama *analisis.csv*. Kita coba untuk melakukan visualisasi data dengan *dataframe* yang dapat dilihat pada gambar berikut.

	translated	Subjectivity	Polarity	analisis
0	Internet Unlimited ActivitiesIndonesia Struggl...	0.000000	0.000000	Neutral
1	Don't be too quick to judge without ever wanti...	0.500000	0.333333	Positive
2	nan	0.000000	0.000000	Neutral
3	Internet Unlimited ActivitiesIndonesia Struggl...	0.000000	0.000000	Neutral
4	please, why is this, is there any maintenance	0.000000	0.000000	Neutral
...
95	It's raining like this, it's nice to lie down ...	0.684921	0.370032	Positive
96	Hello, why is it that at this time of the day ...	0.000000	0.000000	Neutral
97	I swear it's really bikes	0.200000	0.200000	Positive
98	nan	0.000000	0.000000	Neutral
99	Hello Sis Almadina, Thank you for your appreci...	0.533333	0.600000	Positive

Gambar 3.4 Hasil Storage Data ke MySQL

Gambar di atas merupakan hasil data *subjectivity*, *polarity* dan analisis yang telah dilakukan pada kode program sebelumnya. terdapat 3 kolom yaitu *translated* yang merupakan *tweet*, *subjectivity*, *polarity* dan juga hasil dari analisisnya.

```
df = pd.read_csv('analisis.csv')
df[['translated', 'analisis']].to_csv('traintest.csv',
index=False, header=False) #specify location
df.head()
```

Kode Program 3.12 Melakukan Storage Data

Data tersebut kemudian disimpan untuk nanti dijadikan sebagai data latihan. Data yang disimpan adalah data translated dan juga analisis, kemudian juga menghapus index dan header dari data. Langkah selanjutnya adalah melakukan uji coba teks klasifikasi.

3.4 Text Classification

Langkah terakhir adalah melakukan uji coba atau *testing* untuk menguji apakah program dapat melakukan uji coba teks klasifikasi. Teks klasifikasi memanfaatkan *module* NaiveBayesClassifier.

```
from textblob.classifiers import NaiveBayesClassifier

filename = 'traintest.csv'
def loadCsv(filename):
    separated = []
    with open(filename, 'r') as f:
        file = csv.reader(f)
        my_list = list(file)
        for x in my_list:
            convert_tuple = tuple(x)
            separated.append(convert_tuple)
    return separated

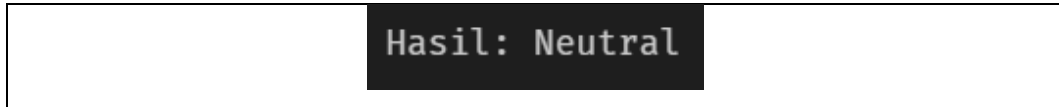
def classifier(dataset, word):
    cl = NaiveBayesClassifier(dataset)
    test = cl.classify(word)
    return test

cek = classifier(loadCsv(filename), "Indihome best")
print("Hasil:", cek)
```

Kode Program 3.13 Melakukan Storage Data

Kode program di atas merupakan cara untuk melakukan teks klasifikasi menggunakan NaiveBayesClassifier yang ada pada *library* textblob. Data *training* yang digunakan mengambil dari *file* csv bernama traintest.csv. kemudian membuat fungsi untuk klasifikasi bernama classifier yang mempunyai dua parameter yaitu dataset untuk mengambil dataset *training* dan word untuk teks

yang akan dilakukan teks klasifikasi. Setelah itu membuat *variable* cek dan memanggil fungsi `classifier` dengan memasukkan parameter yang diperlukan. Kemudian, cetak hasil klasifikasi dengan menggunakan perintah `print`. Hasil dari klasifikasi “Neutral”

A screenshot of a terminal window. The terminal has a light gray background. In the center, there is a dark gray rectangular box containing the text "Hasil: Neutral" in a white, monospaced font.

Gambar 3.5 Hasil Storage Data ke MySQL

Setelah dilakukan klasifikasi, cetak hasil klasifikasi dengan menggunakan perintah `print`. Hasil dari klasifikasi adalah “Neutral”.