

DAFTAR ISI

DAFTAR ISI.....	i
BAB I PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	1
C. Tujuan dan Manfaat.....	1
BAB II PEMBAHASAN.....	2
A. Metode Analisis	2
1. Naive Bayes Classifier.....	2
B. Bahan dan Peralatan Riset	2
1. API Twitter	2
2. Google Collaboratory	2
3. Python	2
4. Numpy	3
5. Pandas	3
6. Seaborn	3
7. TextBlob.....	3
8. Natural Language Toolkit	3
9. Tweepy	3
C. Tahapan dan Hasil Analisis	3
Langkah 1 : Proses Install and Import Libraries	3
Langkah 2 : Proses Crawling Data (Pengumpulan Data)	4
Langkah 3: Cleaning Data (Pembersihan Data)	6
Langkah 4: Proses Pemodelan Data Menggunakan Textblob	8
Langkah 5: Klasifikasi Data Dengan Metode Naïve Bayes Classifier	9
Langkah 5: Proses Visualisasi Data	11
Langkah 7: Analisis Klasifikasi Menggunakan Metode Naive Bayes Classifier	13
BAB III PENUTUP.....	15
A. Simpulan	15
B. Saran	15
DAFTAR PUSTAKA.....	16

BAB I

PENDAHULUAN

A. Latar Belakang

Coronavirus Disease atau yang biasa disebut dengan Covid-19 telah mendapat perhatian khusus dari WHO dan pemerintah negara di seluruh dunia. Sejak pertengahan tahun 2020, vaksinasi gencar dilakukan untuk mengantisipasi dan mengurangi penyebaran virus secara masif. Namun, keberadaan virus ini masih menjadi kewaspadaan bagi seluruh umat manusia. Pasalnya, Virus Covid-19 terus bermutasi menjadi varian-varian baru di berbagai wilayah di dunia, termasuk di Indonesia.

Organisasi Kesehatan Dunia menaruh perhatian yang mendalam pada perkembangan virus mutasi ini. Dimulai dari Varian *Alpha* pada September 2020, *Beta* pada Mei 2020, *Gamma* pada November 2020, *Delta* pada Oktober 2020, *Lambda* pada Desember 2020, *Kappa* pada Oktober 2020, *Eta* pada Desember 2020, *Iota* pada November 2020, dan *Mu* pada Januari 2021. Keseluruhan varian ini memiliki gejala yang hampir sama dengan semua jenis virus corona lainnya. Varian terbaru dari virus Covid-19 ditemukan di Afrika Selatan pada November 2021, yaitu Varian *Omicron*. *Omicron* memiliki gejala yang lebih ringan dari varian sebelumnya. Meskipun demikian, Varian *Omicron* memiliki tingkat penyebaran lima kali lipat lebih cepat dari virus aslinya dan empat kali lebih cepat dari Varian *Delta*. Seiring berjalannya waktu, telah ditemukan kasus pertama Varian *Omicron* di berbagai negara. Hal ini tentu membuat publik memiliki kekhawatiran yang lebih tinggi setelah sebelumnya sempat mendapat angin segar karena kurva penyebaran yang mulai melandai. Melalui perkembangan IPTEK saat ini, berbagai opini masyarakat Indonesia tentu sangat banyak ditemukan melalui media sosial, terutama melalui platform Twitter.

Analisis sentimen adalah suatu analisis terhadap suatu peristiwa yang didasarkan pada opini dan sikap seseorang mengenai suatu objek. Analisis ini dilakukan dengan mengumpulkan data-data dari berbagai platform sosial media, surat kabar, atau media lainnya. Dengan dilakukannya analisis sentimen ini, kita dapat mengetahui bagaimana pendapat publik terhadap suatu objek. Sentimen untuk opini sendiri dibedakan menjadi tiga, yaitu opini positif, negatif, dan netral tergantung dari pendapat masing-masing individu. Hal ini dilakukan agar dapat menjadi parameter baik atau buruknya tanggapan publik tentang keberadaan Virus *Omicron* setelah ditemukan pertama kali di Afrika Selatan.

Maka dari itu, kami menyusun laporan praktikum yang berjudul “Analisis Sentimen Opini Publik mengenai Virus *Omicron* di Indonesia menggunakan Metode Naive Bayes Classifier pada Twitter”.

B. Rumusan Masalah

Bagaimana analisis sentimen opini publik mengenai Virus *Omicron* di Indonesia menggunakan metode Naive Bayes Classifier pada Twitter?

C. Tujuan dan Manfaat

Untuk mengetahui sentimen opini publik mengenai Virus *Omicron* di Indonesia menggunakan metode Naive Bayes Classifier pada Twitter.

BAB II PEMBAHASAN

A. Metode Analisis

1. Naive Bayes Classifier

Naive Bayes merupakan suatu metode pengklasifikasian probabilistik sederhana yang melakukan perhitungan terhadap sekumpulan probabilitas dengan cara menjumlahkan rekuensi dan kombinasi nilai dari *dataset* yang sudah tersedia. Metode ini dinilai lebih baik daripada model *classifier* lainnya karena hanya membutuhkan *data training* yang kecil untuk menentukan parameter dalam pengklasifikasian. Bentuk umum dari *Naive Bayes Classifier* adalah sebagai berikut,

$$P(X|Y) = \frac{P(Y|X) P(X)}{P(Y)}$$

Keterangan:

Y	: Data dengan kelas yang belum diketahui
X	: Hipotesis data
$P(X Y)$: Probabilitas hipotesis X berdasarkan kondisi Y
$P(Y)$: Probabilitas Y
$P(Y X)$: Probabilitas hipotesis Y berdasarkan kondisi X
$P(X)$: Probabilitas hipotesis X

B. Bahan dan Peralatan Riset

1. API Twitter

Application Programming Interface dapat digunakan untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda dalam waktu yang bersamaan. Pada praktikum ini, API Twitter yang digunakan berhubungan dengan “*Omicron*”. Baik postingan Twitter berupa tagar maupun pencarian dengan kata kunci “*Omicron*”. Selanjutnya, data tersebut akan diolah serta dimodelkan untuk menarik suatu kesimpulan.

2. Google Collaboratory

Google Collaboratory merupakan sebuah *tools* yang dibuat dengan *environment Jupyter* dan mendukung semua *library* dalam pengembangan AI. *Tools* ini berbasis *cloud* dan gratis untuk tujuan penelitian. *Google Collab* memiliki keunggulan dalam hal aksesibilitas karena dapat diakses secara gratis dan tidak perlu melakukan konfigurasi apapun. Akan tetapi, jika ingin menambah *library* baru, diharuskan untuk menginstall *library package*-nya.

3. Python

Python merupakan salah satu bahasa pemrograman yang cocok digunakan untuk *data mining*, *scripting language*, dan sebagainya. Hal ini disebabkan oleh kode yang dituliskan akan di-*compile* menjadi *bytecode* dan dieksekusi. Selain itu, *Python* mempunyai struktur fungsi, *class*, modul, dan *packages* serta konsep *Object Oriented Programming* yang kuat.

4. Numpy

Numerical Python atau *Numpy* merupakan *library python* yang memiliki fokus pada *scientific computing*. *Numpy* dapat membentuk sebuah objek *N-dimensional array* yang mengonsumsi memori lebih kecil dan *runtime* yang lebih cepat.

5. Pandas

Python Data Analyst Library adalah *library open source* pada *Python* yang memiliki lisensi BSD. *Pandas* digunakan untuk membersihkan data agar dapat dianalisis lebih mudah. Misalnya, data diubah dalam bentuk daftar tabel.

6. Seaborn

Seaborn adalah *library* visualisasi data *open source* yang juga memiliki lisensi BSD. Pustaka ini dibangun dengan dasar pustaka *matplotlib*. *Seaborn* dapat memberikan visualisasi data yang menarik dan indah serta sederhana.

7. TextBlob

TextBlob merupakan *library Python* yang menyediakan API sederhana untuk melakukan *Natural Language Process*. *TextBlob* digunakan dalam analisis sentimen, klasifikasi, terjemahan, penandaan *part-of-speech*, ekstraksi frasa kata benda, dan lain-lain.

8. Natural Language Toolkit

Natural Language Toolkit merupakan *library Python* yang digunakan dalam pemodelan teks sebelum digunakan pada *machine learning*.

9. Tweepy

Tweepy merupakan *library Python* untuk mengakses API dari platform Twitter. *Tweepy* digunakan untuk mendapatkan data dari Twitter berupa *tweet* berdasarkan kata kunci yang digunakan dalam pencarian dengan mudah. Setelah memasukkan kata kunci, semua *tweet* yang mengandung kata kunci tersebut akan muncul.

C. Tahapan dan Hasil Analisis

Langkah 1 : Proses Install and Import Libraries

Sebelum melakukan analisis, kita perlu menginstall *library textblob* dan *tweepy* terlebih dahulu menggunakan perintah **!pip install** di Notebook Jupyter Kita.

```
!pip install tweet-preprocessor
!pip install google_trans_new
!pip install tweepy
!pip install textblob
!pip install wordcloud
!pip install nltk
```

Kemudian, *import library* yang akan digunakan dalam proyek analisis sentimen ini.

```
from wordcloud import WordCloud, STOPWORDS
from datetime import timedelta, datetime
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
import matplotlib.pyplot as plt
from tweepy import OAuthHandler
from textblob import TextBlob
import preprocessor as p
import seaborn as sns
import pandas as pd
import numpy as np
import tweepy
import csv
import re
import nltk
nltk.download('punkt')
```

Sebelum autentikasi, kita membutuhkan API dari Twitter yang dapat diibaratkan sebagai gerbang untuk mendapatkan *tweet*. Kita harus *apply* terlebih dahulu apabila belum memiliki *account* Twitter *developer* ke website developer Twitter atau dengan tautan <https://developer.twitter.com/en>. Untuk mendapatkan *account* Twitter *developer*, biasanya membutuhkan satu atau dua hari, atau terkadang lebih, agar aplikasi kita ditinjau oleh Twitter.

Langkah 2 : Proses Crawling Data (Pengumpulan Data)

Proses *crawling* (pengambilan data) yang kami lakukan pada proyek ini adalah dengan menggunakan *library* *tweepy*. Adapun sebelum melakukan pengambilan data dilakukan beberapa proses yaitu:

1. Autentikasi API Twitter menggunakan *Consumer Key*, *Consumer Secret*, *Access Token*, dan *Access Token Secret* yang didapatkan dari registrasi akun developer pada <https://developer.twitter.com/en/application>.

```
consumer_key = "T1lfLLj8Csy6hVWniInr8E21S"
consumer_secret = "zdWyOB28A3NQGFCA80ULZKt37IHMy6IqWvf15xzodPwKB65kZ0"
access_token = "886237353366376448-fjm0ghiJrMtEp4YUDwY5kjUseeA9g7z"
access_token_secret = "RJ2AIPJ3WyPbWUeLZEdDv1EoMGQx7u9Us7AwUBLIDhCwM"
```

Pada bagian inilah *key* dan token tersebut akan kita gunakan. *Access token* dan *key* kita *input* ke setiap bagian pada *consumerKey*, *consumerSecret*, *accessToken*, dan *accessTokenSecret*. Setelah itu, kita sudah dapat mengambil *tweet* dari *database* Twitter, karena API Twitter sudah kita dapatkan melalui *access token* dan *key* tersebut.

2. Setelah melakukan proses autentikasi, langkah selanjutnya yang kita lakukan adalah membuat fungsi proses pengambilan data Twitter tersebut. Adapun dalam fungsi tersebut (*scraptweets*) langkah awal yang dilakukan yaitu membuat kumpulan kolom dengan variabel *db_tweets* sebagai acuan *field* data yang akan kita masukkan. Kolom-kolom tersebut sendiri terdiri atas:

- *username* : nama dari orang yang melakukan *tweet* (postingan)

- *tweetcreatedts* : waktu *tweet* (postingan) dibuat
- *text* : isi pesan *tweet* (postingan)

```

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

def scraptweets(search_words, date_since, date_until):

    db_tweets = pd.DataFrame(columns=['username', 'tweetcreatedts', 'text'])

    tweets = tweepy.Cursor(
        api.search, q=search_words, lang="id",
        since=date_since, until=date_until, tweet_mode='extended').items(1000)

    tweet_list = [tweet for tweet in tweets]

    for tweet in tweet_list:
        username = tweet.user.screen_name
        tweetcreatedts = tweet.created_at

        try:
            text = tweet.retweeted_status.full_text
        except AttributeError:
            text = tweet.full_text

        ith_tweet = [username, tweetcreatedts, text]

        db_tweets.loc[len(db_tweets)] = ith_tweet

    print('Proses Scrapping Selesai Dengan Jumlah Data', len(db_tweets))
    filename = 'omicron.csv'
    db_tweets.to_csv(filename, index=False)

today = datetime.today().strftime("%Y-%m-%d")
last_week = datetime.today() - timedelta(7)
last_week = last_week.strftime("%Y-%m-%d")

last_month = datetime.today() - timedelta(30)
last_month = last_month.strftime("%Y-%m-%d")

search_words = "#Omicron OR #OmicronVariant OR #OmicronArtmadanOnline OR #Omicronindia OR #covid19omicron OR #virusomicron"
date_since = last_month
date_until = today

scraptweets(search_words, date_since, date_until)

```

3. Selanjutnya di dalam fungsi tersebut dilakukan pemanggilan data pada API Twitter sesuai dengan kata kunci dan rentang waktu yang telah ditentukan. Data diambil dengan rentang waktu satu bulan terakhir terhitung dari tanggal 16 Desember 2021 dan kami berhasil mengumpulkan sekitar 1000 data *Tweet* mengenai “*Omicron*”.

```

today = datetime.today().strftime("%Y-%m-%d")
last_week = datetime.today() - timedelta(7)
last_week = last_week.strftime("%Y-%m-%d")

last_month = datetime.today() - timedelta(30)
last_month = last_month.strftime("%Y-%m-%d")

search_words = "#Omicron OR #OmicronVariant OR #OmicronArtmadanOnline OR #Omicronindia OR #covid19omicron OR #virusomicron"
date_since = last_month
date_until = today

scraptweets(search_words, date_since, date_until)

Proses Scrapping Selesai Dengan Jumlah Data 1000

```

4. Melakukan perulangan untuk data yang telah diambil dan mengumpulkannya dalam sebuah *array* untuk digabungkan nantinya dalam sebuah file .csv yang akan diolah sesuai

kolom-kolom yang telah ditentukan sebelumnya.

```
data = pd.read_csv('omicron.csv')
data.head(10)
```

	username	tweetcreatedts	text
0	__aichaaa	2021-12-16 23:59:47	Omicron sudah ditemukan di Indonesia, jangan p...
1	rdkxogus	2021-12-16 23:59:46	Omicron sudah ditemukan di Indonesia, jangan p...
2	kjokkn	2021-12-16 23:59:42	Namun, disisi lain.. ringannya gejala #omicron...
3	doddooo7	2021-12-16 23:59:36	Omicron sudah ditemukan di Indonesia, jangan p...
4	breeze_moment	2021-12-16 23:59:25	Omicron sudah ditemukan di Indonesia, jangan p...
5	manjikro	2021-12-16 23:59:24	Namun, disisi lain.. ringannya gejala #omicron...
6	manjikro	2021-12-16 23:59:15	Data menunjukkan puncak grafik #omicron bisa d...
7	manjikro	2021-12-16 23:59:08	Saat ini lebih dari 4000 pasien #omicron di ra...
8	manjikro	2021-12-16 23:58:50	Jumlah kasus #omicron di Afrika Selatan juga s...
9	violetaaday	2021-12-16 23:58:49	Omicron sudah ditemukan di Indonesia, jangan p...

```
data.sample(n=5)
```

	username	tweetcreatedts	text
950	heydianac	2021-12-16 17:18:14	Omicron sudah ditemukan di Indonesia, jangan p...
476	raden_hayati	2021-12-16 22:13:30	Namun, disisi lain.. ringannya gejala #omicron...
360	lsvl_lla	2021-12-16 22:51:08	Omicron sudah ditemukan di Indonesia, jangan p...
643	ruangyellow	2021-12-16 20:20:33	Omicron sudah ditemukan di Indonesia, jangan p...
305	maendzr	2021-12-16 23:04:35	Omicron sudah ditemukan di Indonesia, jangan p...

Langkah 3: Cleaning Data (Pembersihan Data)

Pada proyek ini kami melakukan *cleaning data* (pembersihan data) dengan menggunakan *library tweet-preprocessor*. Adapun langkah-langkah yang dilakukan dalam proses *cleaning data* (pembersihan data) adalah :

1. Mengecek apakah ada data kosong dari data yang sudah didapat dari proses crawling (pengambilan data), data yang dicek pada proses ini adalah :
 - username : nama dari orang yang melakukan *tweet* (postingan)
 - tweetcreatedts : waktu *tweet* (postingan) dibuat
 - text : isi pesan *tweet* (postingan)

```
data.isnull().sum()
```

```
username      0
tweetcreatedts 0
text          0
dtype: int64
```

2. Langkah selanjutnya adalah proses *tokenizing text*. *Tokenizing* adalah proses memisahkan kata berdasarkan spasi yang ditemukan. Proses ini bertujuan untuk

mempermudah menganalisis setiap kata-kata dari setiap kalimat pada data *tweet* yang sudah dikumpulkan.

```
import googletrans
from googletrans import Translator
from googletrans import LANGUAGES

def preprocessing_data(x):
    return p.clean(x)

def tokenize_data(x):
    return p.tokenize(x)

data['tweet_clean'] = data['text'].apply(preprocessing_data)
data['tweet_clean'] = data['tweet_clean'].apply(tokenize_data)
data = data.drop_duplicates()

data.to_csv("D:\\KULIAH\\Semester 5\\Analisis Big Data\\UAS\\omicron\\omicron_preprocessing.csv", header=True, index=False)
```

3. Selanjutnya, menerjemahkan setiap kata dari kalimat yang sudah dipisahkan menggunakan proses *tokenizing* dari Bahasa Indonesia menjadi Bahasa Inggris. Hal ini dilakukan karena *library* yang kami gunakan untuk menganalisis data menggunakan dasar Bahasa Inggris. *Library* yang kami gunakan untuk menerjemahkan bahasa adalah *Google Translate*.

```
translator = Translator()

# df = pd.read_csv("omicron_preprocessing.csv", encoding='latin-1')
data.columns = ['username', 'tweetcreatedts', 'text', 'tweet_clean']
# print(df)

tweet_english = []

for element in data['tweet_clean']:
    tweet_english.append(translator.translate(element).text)

# print(tweet_english)

data['tweet_english'] = tweet_english

data.to_csv("D:\\KULIAH\\Semester 5\\Analisis Big Data\\UAS\\omicron\\omicron_english.csv", header=True, index=False)
```

4. Langkah selanjutnya adalah proses *stemming* data. Proses *stemming* digunakan untuk menemukan kata dasar dengan cara menghilangkan kata imbuhan.

```
ps = PorterStemmer()

def stemming_data(x):
    return ps.stem(x)

data['tweet_english'] = data['tweet_english'].apply(stemming_data)
```


Langkah 4: Proses Pemodelan Data Menggunakan Textblob

Setelah melakukan *crawling data* dan *wrangle data*, kami melakukan pemodelan data menggunakan *TextBlob*. Adapun langkah-langkah yang kami lakukan dalam proses pemodelan data adalah:

1. Langkah pertama data yang telah *crawling* (dikumpulkan) dan *cleaning* (dibersihkan) kami simpan kedalam file .csv untuk kami gunakan dalam melakukan pemodelan data di *Google Collaboratory*. Kita memanggil data yang telah dikumpulkan :

data.head(10)

	username	tweetcreatedts	text	tweet_clean	tweet_english
0	...achaaa	2021-12-16 23:59:47	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron has been found in Indonesia, don't pan...
1	rskoogus	2021-12-16 23:59:46	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron has been found in Indonesia, don't pan...
2	ljokko	2021-12-16 23:59:42	Namun, disisi lain... ringannya gejala #Omicron...	Namun, disisi lain... ringannya gejala dan peru...	However, on the other hand... the symptoms of...
3	dodddoo7	2021-12-16 23:59:36	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron has been found in Indonesia, don't pan...
4	breaze_mohawf	2021-12-16 23:59:25	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron has been found in Indonesia, don't pan...
5	marjiko	2021-12-16 23:59:24	Namun, disisi lain... ringannya gejala #Omicron...	Namun, disisi lain... ringannya gejala dan peru...	However, on the other hand... the symptoms of...
6	marjiko	2021-12-16 23:59:15	Data menunjukkan puncak grafik #Omicron bisa d...	Data menunjukkan puncak grafik bisa dicapai da...	Data shows the peak of the graph can be achiev...
7	marjiko	2021-12-16 23:59:08	Saat ini lebih dari 4000 pasien #Omicron di ra...	Saat ini lebih dari pasien di rawat di RS Afr...	At present more than patients are treated at S...
8	marjiko	2021-12-16 23:58:53	Jumlah kasus #Omicron di Afrika Selatan juga s...	Jumlah kasus di Afrika Selatan juga sudah menc...	The number of cases in South Africa has abso...
9	violetasday	2021-12-16 23:58:49	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron sudah ditemukan di Indonesia, jangan p...	Omicron has been found in Indonesia, don't pan...

2. Setelah kita memanggil data, kita akan melakukan analisis terhadap data *tweet* tersebut menggunakan *TextBlob*. Dalam analisis ini, kami membagi kategori data *tweet* menjadi 3 yaitu Positif, Negatif, dan Netral. Dalam proses ini, kita mendapatkan berapa banyak data *tweet* yang dimiliki dan dari data *tweet* tersebut kita menganalisis data tersebut masuk ke dalam kategori yang telah kita sediakan.

```
data_tweet = list(data['tweet_english'])
polaritas = 0

status = []
total_positif = total_negatif = total_netral = total = 0

def klasifikasi(polaritas):
    if polaritas > 0.0:
        return 'Positif'
    elif polaritas > 0.0 == 0.0:
        return 'Netral'
    else:
        return 'Negatif'

for i, tweet in enumerate(data_tweet):
    analysis = TextBlob(tweet)
    polaritas += analysis.polarity

    if analysis.sentiment.polarity > 0.0:
        total_positif += 1
        status.append('Positif')
    elif analysis.sentiment.polarity == 0.0:
        total_netral += 1
        status.append('Netral')
    else:
        total_negatif += 1
        status.append('Negatif')

    total += 1

status = pd.DataFrame({'klasifikasi': status})
data['klasifikasi'] = status
data.tail()

print(f'Hasil Analisis Data:\nPositif = {total_positif}\nNetral = {total_netral}\nNegatif = {total_negatif}')
print(f'\nTotal Data : {total}')
```

Adapun hasil yang didapatkan dari klasifikasi menggunakan *TextBlob* tersebut adalah :

- Positif : 109
- Netral : 29
- Negatif : 862

Hasil Analisis Data:

Positif = 109

Netral = 29

Negatif = 862

Total Data : 1000

Langkah 5: Klasifikasi Data Dengan Metode Naïve Bayes Classifier

Kami melakukan klasifikasi data dengan metode *Naive Bayes Classifier* menggunakan library *Natural Language Toolkit* atau disingkat NLTK. Berikut merupakan langkah-langkah dalam melakukan klasifikasi data:

1. Langkah pertama kami menyiapkan data yang akan kami proses untuk dilakukan pengklasifikasian data, data-data yang kami ambil adalah :
 - `username` : nama dari orang yang melakukan *tweet* (postingan)
 - `tweetcreatedts` : waktu *tweet* (postingan) dibuat
 - `text` : isi pesan *tweet* (postingan)
 - `tweet_clean` : isi pesan *tweet* yang sudah melalui proses pembersihan data

```
dataset = data.drop(['username', 'tweetcreatedts', 'text', 'tweet_clean'], axis=1, inplace=False)
dataset = [tuple(x) for x in dataset.to_records(index=False)]
```

2. Selanjutnya kami mengambil *dataset* untuk kemudian dijadikan *data training* (data yang digunakan untuk melatih algoritma) dengan tujuan *data training* ini dapat digunakan untuk membantu melakukan prediksi. Pada langkah ini, kami mengambil hasil dari metode pengklasifikasian sebelumnya untuk kemudian kami kelompokkan ulang pada variabel baru berdasarkan labelnya dan kami ambil secara acak sampel dari masing-masing labelnya. Pada langkah ini, kami mendapatkan hasil pada variabel `train_set` dan merupakan *data training* kami yang diambil dari hasil proses data sebelumnya.

```

import random

set_positif = []
set_negatif = []
set_netral = []

for n in dataset:
    if(n[1] == 'Positif'):
        set_positif.append(n)
    elif(n[1] == 'Negatif'):
        set_negatif.append(n)
    else:
        set_netral.append(n)

set_positif = random.sample(set_positif, k=int(len(set_positif)/2))
set_negatif = random.sample(set_negatif, k=int(len(set_negatif)/2))
set_netral = random.sample(set_netral, k=int(len(set_netral)/2))

train = set_positif + set_negatif + set_netral

train_set = []

for n in train:
    train_set.append(n)

```

3. Setelah itu kami melakukan penghitungan akurasi dari variabel `train_set` menggunakan *Naive Bayes Classifier* dari library *TextBlob*. Hasil akurasinya sampai dengan nilai 0,913 atau 91% yang artinya hasil prediksinya sudah sangat baik.

```

from textblob.classifiers import NaiveBayesClassifier
cl = NaiveBayesClassifier(train_set)
print('Akurasi Test:', cl.accuracy(dataset))

```

Akurasi Test: 0.913

4. Selanjutnya, melakukan pelabelan data ulang dengan menggunakan metode *Naive Bayes Classifier*. Kami mengambil data *tweet_english* yaitu data *tweet_clean* yang diterjemahkan ke bahasa inggris dan memasukkannya ke dalam variabel `data_tweet`. Dalam proses perulangan, kami melakukan analisis untuk mengklasifikasi data ketiga kelas, yaitu positif, negatif, dan netral serta menghitung jumlah masing-masing klasifikasinya dan total keseluruhan data yang diuji. Adapun hasil yang didapatkan dari klasifikasi menggunakan metode *Naive Bayes Classifier* tersebut adalah :
 - Positif : 100
 - Negatif : 785
 - Netral : 115

```

data_tweet = list(data['tweet_english'])
polaritas = 0

status = []
total_positif = total_negatif = total_netral = total = 0

for i, tweet in enumerate(data_tweet):
    analysis = TextBlob(tweet, classifier=cl)

    if analysis.classify() == 'Positif':
        total_positif += 1
    elif analysis.classify() == 'Netral':
        total_netral += 1
    else:
        total_negatif += 1

    status.append(analysis.classify())
    total += 1

status = pd.DataFrame({'klasifikasi_bayes': status})
data['klasifikasi_bayes'] = status

print(f'\nHasil Analisis Data:\nPositif = {total_positif}\nNetral = {total_netral}\nNegatif = {total_negatif}')
print(f'\nTotal Data : {total}')

```

Hasil Analisis Data:
 Positif = 100
 Netral = 115
 Negatif = 785

 Total Data : 1000

Langkah 5: Proses Visualisasi Data

Visualisasi data dalam topik kami menggunakan *chart pie* yang terdiri dari 3 bagian yaitu Sentimen Positif, Negatif, serta Netral. Selain itu, kami juga menggunakan *word cloud* untuk melihat kata apa saja yang paling banyak muncul dari data yang telah diolah.

Berikut merupakan visualisasi dari hasil data yang sudah kami klasifikasikan menggunakan metode *Naive Bayes Classifier* menggunakan fungsi *chart pie* yang melakukan pengkodean untuk letak perdata serta posisi label sesuai data yang ada. Kami menggunakan diagram lingkaran persen dengan warna biru mewakili data positif, warna orange mewakili data negatif, dan hijau mewakili warna netral.

```

def show_pie(label, data, legend_title):
    fig, ax = plt.subplots(figsize=(8, 10), subplot_kw=dict(aspect='equal'))

    labels = [x.split()[-1] for x in label]

    def func(pct, allvals):
        absolute = int(pct/100.*np.sum(allvals))
        return "{:.1f}% ({:d})".format(pct, absolute)

    wedges, texts, autotexts = ax.pie(data, autopct=lambda pct: func(pct, data),
                                     textprops=dict(color="w"))

    ax.legend(wedges, labels,
              title= legend_title,
              loc="center left",
              bbox_to_anchor=(1, 0, 0.5, 1))

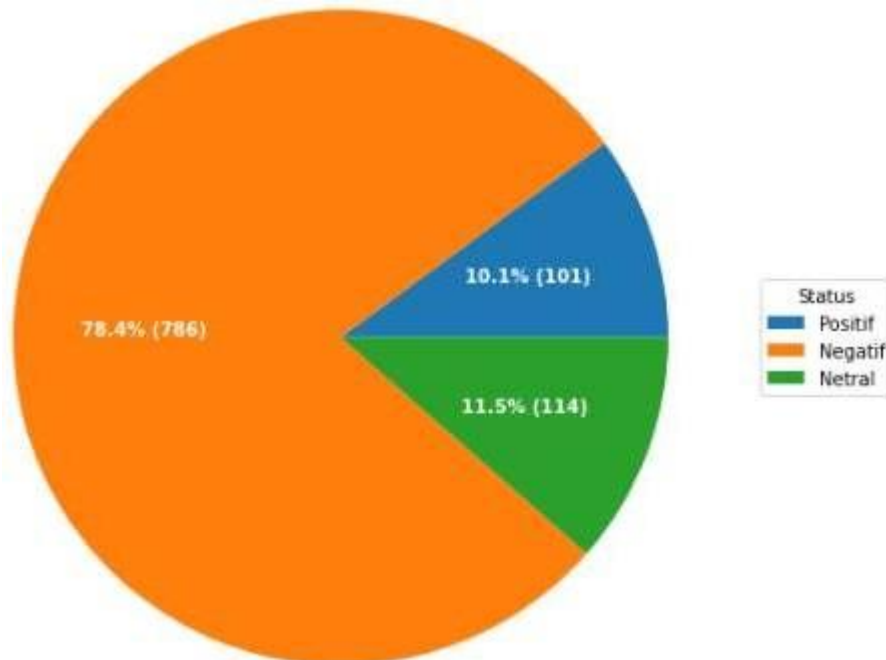
    plt.setp(autotexts, size=10, weight="bold")
    plt.show()

```

```
label = ['Positif', 'Negatif', 'Netral']
count_data = [total_positif+1, total_negatif+1, total_netral]

show_pie(label, count_data, "Status")
```

Sehingga, hasil dari proses pemodelan *chart pie* tersebut adalah sebagai berikut,



Dilihat dari visualisasi data berbentuk *chart pie* di atas, dapat disimpulkan bahwa mayoritas masyarakat menanggapi mengenai Virus *Omicron* dengan negative.

Selain itu, pengolahan *chart* tersebut menggunakan *library wordcloud*, *matplotlib*, dan *seaborn* untuk melakukan visualisasi data. Adapun proses menampilkannya, yaitu dengan membuat fungsi `plot_cloud`. Setelah itu, kita bagi setiap kata menjadi *array* untuk diolah oleh *wordcloud* agar dapat ditampilkan kata yang paling banyak muncul. Berikut proses visualisasinya,

```
def plot_cloud(wordcloud):
    plt.figure(figsize=(12, 8))
    plt.imshow(wordcloud)
    plt.axis("off");
```

```
all_words = ' '.join([tweets for tweets in data['tweet_english']])
wordcloud = WordCloud(width = 3000, height = 2000, random_state=3, background_color='white', colormap='Set2', collocations=False, stopwords = STOPWORDS)
plot_cloud(wordcloud)
```

Sehingga, hasil dari proses pemodelan *wordcloud* tersebut adalah sebagai berikut,



Langkah 7: Analisis Klasifikasi Menggunakan Metode Naive Bayes Classifier

Setelah melakukan klasifikasi, kita dapat melakukan pengecekan perubahan data yang dilakukan saat menggunakan metode *Naive Bayes Classifier* dengan membandingkan perbedaan pada kolom klasifikasi sebelumnya menggunakan kode berikut,

```
data_eval = [tuple(x) for x in data.to_records(index=False)]

for n in data_eval:
    if n[5] != n[6]:
        print(f'Text: {n[3]}\nClassifier: {n[5]}\nClassifier Bayes: {n[6]} \n')
```

Adapun beberapa contoh klasifikasi yang diubah, yaitu,

Text: Namun, disisi lain.. ringannya gejala dan penularan yg lebih cepat malah diartikan beberapa ilmuwan ini merupakan akhir dari pandemi, karena virus melemah dan malah akan membuat peningkatan kekebalan alami.. kita lihat saja bagaimana di UK ya.

.

Classifier: Negatif

Classifier Bayes: Netral

Text: Jumlah kasus di Afrika Selatan juga sudah mencapai persen. Menggantikan Delta . Lambat laun Omicron akan menyebar ke seluruh Dunia. Kita lihat saja UK dalam beberapa hari sudah menjadi puluhan ribu kasus. persen varian di London saat ini sudah Omicron.

Classifier: Negatif

Classifier Bayes: Netral

Text: Terkait , apakah Indonesia kecolongan lagi?Bagaimana tidak kecolongan, aturan karantina perjalanan luar negeri saja bimbang. Terakhir, malah ada pengecualian untuk orang-orang terhormat

Classifier: Negatif

Classifier Bayes: Netral

Text: Kawan Kabinet, mari simak arahan Presiden perkembangan terkini COVID-19 terkait dengan terdeteksinya varian Omicron di Indonesia.

Classifier: Positif

Classifier Bayes: Netral

Text: Mestinya tidak ada privilege apapun untuk karantina, apalagi ada ancaman . Lha saja tidak perlu pilih-pilih, mau hinggap di tubuh pejabat atau orang biasa

Classifier: Negatif

Classifier Bayes: Netral

Text: Indonesia telah mengonfirmasi kasus pertama varian virus Corona Omicron. Hal tersebut disampaikan oleh Menteri Kesehatan RI Budi Gunadi Sadikin pada Kamis (16/12).

Classifier: Positif

Classifier Bayes: Netral

Metode Naive Bayes Classifier pada praktikum ini memiliki tingkat akurasi sebesar 0.913 atau setara dengan 91%. Ada perbedaan dari klasifikasi sebelumnya, yaitu,

- 100 sentimen positif (+9 data)
- 115 sentimen netral (-86 data)
- 785 sentimen negatif (+77 data)

Maka, dapat disimpulkan bahwa mayoritas publik menanggapi virus “*Omicron*” dengan negatif.

BAB III

PENUTUP

A. Simpulan

Pada hasil pembahasan diatas, dapat kita lihat bahwa data-data yang diubah memiliki sentimen yang tepat sesuai isi teks. Oleh karena itu, dapat ditarik kesimpulan bahwa metode *Naives Bayes Classifier* cukup akurat jika diterapkan dalam klasifikasi sentimen analisis pada praktikum ini.

B. Saran

Bertitik tolak dari simpulan di atas, analisis sentimen menggunakan *Naive Bayes Classifier* memerlukan tingkat ketelitian yang tinggi. Maka, disarankan agar sering melakukan *trial error* dalam mengolah data, mulai dari data mentah hingga proses analisisnya.

DAFTAR PUSTAKA

- Suryono, Sigit, Ema Utami, dan Emha Taufiq Luthfi., “Klasifikasi Sentimen pada Twitter dengan Naive Bayes Classifier,” *Jurnal Ilmiah Bidang Teknologi, ANGKASA*, 2018, vol. 10, 89-96.
- Saleh, Alfa, “Penerapan Data Mining dengan Metode Klasifikasi Naive Bayes untuk Memprediksi Kelulusan mahasiswa dalam Mengikuti English Proficiency Test (Studi Kasus : Universitas Potensi Utama),” *Konferensi Nasional Sistem Informasi*, 2015, vol. 2015.
- Aggarwal, Charu C., *Data Classification Algorithms and Applications*, New York, USA: CRC Press Taylor & Francis Group, 2015.
- Pranita, E. (2021, November 30). *Kompas.com*. Diambil kembali dari Kompas Website: <https://www.kompas.com/sains/read/2021/11/30/190300723/10-varian-covid-19-beserta-gejalanya-dari-alpha-hingga-omicron?page=all>
- Safitri, E. (2021, Desember 10). *detikNews*. Diambil kembali dari detikNews Website: <https://news.detik.com/berita/d-5862236/terungkap-awal-mula-kasus-pertama-omicron-di-indonesia>