

BAB 3

METODE PENELITIAN

Penelitian ini merupakan penelitian analisis sentimen positif dan negatif dari Twitter. Penelitian ini menggunakan metode SVM dengan mengambil data *tweet* yang didapat dari media sosial Twitter yang berkaitan dengan NFT, yaitu Token NFT, Opensea, dan Metaverse dengan menggunakan Snsrape yang di eksekusi menggunakan *tools* Google colab. Pada penelitian ini menggunakan data *training* sebanyak 800 data yang sudah di beri label secara manual, dengan rincian 400 data *tweet* yang bernilai positif dan 400 data *tweet* yang bernilai negatif. Pada data testing menggunakan 200 data *tweet* dengan rincian 100 data berlabel positif, dan 100 data berlabel negatif. Penelitian ini dilakukan beberapa tahap pengolahan data, antara lain *preprocessing*, pelabelan manual, *data training*, dan *data testing* yang digunakan untuk menganalisis sentimen pada *tweet* yang berkaitan dengan NFT. Berikut ini adalah bahan, alat, dan jalan nya penelitian analisis sentimen NFT, serta langkah-langkah penelitian untuk menyelesaikan proses analisis sentimen dari data *tweet*.

3.1 BAHAN DAN ALAT PENELITIAN

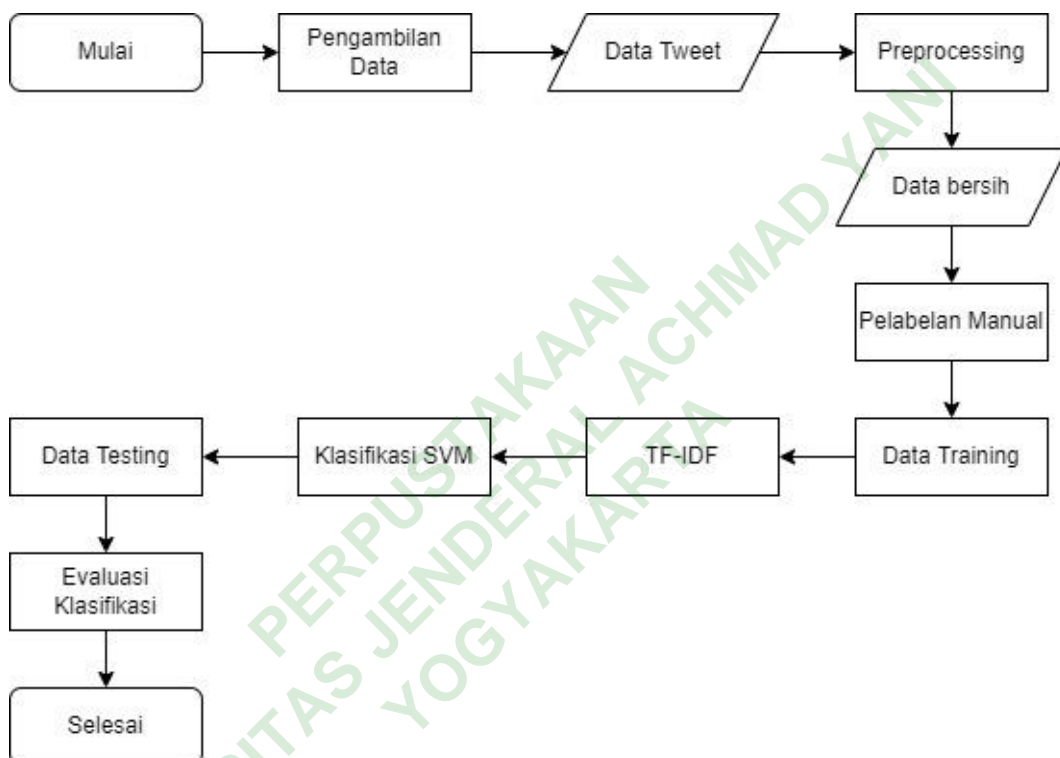
Bahan yang akan digunakan dalam penelitian ini adalah data *tweet* pada Twitter yang berhubungan dengan NFT, yaitu Token NFT, Opensea, dan Metaverse

Pada penelitian ini, alat yang akan digunakan yaitu komputer yang memiliki spesifikasi cukup untuk mempraktikkan sistem operasi dan perangkat lunak pengembangan serta koneksitas Internet. Sistem Operasi dan program-program aplikasi yang akan digunakan pada penelitian ini yaitu:

1. Sistem Operasi: Windows 10 64 bit
2. Google Colab
3. Sublime Text 1.0.0.1

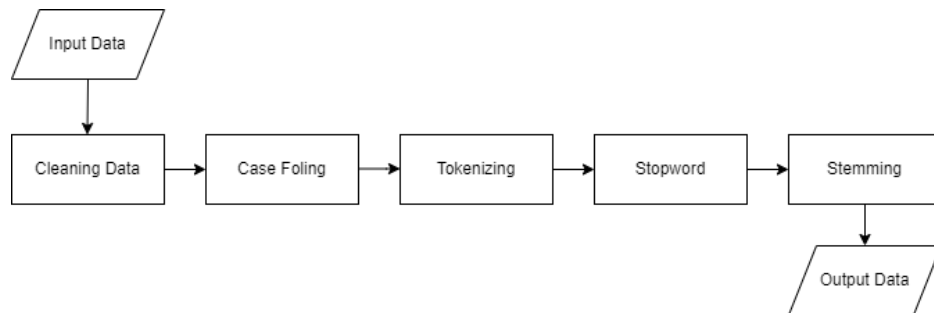
3.2 JALAN PENELITIAN

Penelitian ini merupakan analisis sentimen positif dan negatif menggunakan data yang diambil dari Twitter. Penelitian ini menggunakan metode SVM, dengan mengumpulkan opini berupa tweet yang berkaitan dengan NFT. Flowchart untuk Alur penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Flowchart Alur Penelitian

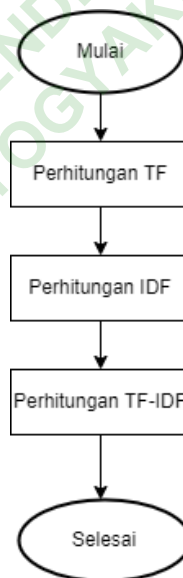
Pada gambar 3.1 menampilkan Flowchart jalan penelitian analisis sentimen ini. Tahapan yang dilakukan pada proses *Preprocessing* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Flowchart *Preprocessing*

Pada gambar 3.2 diatas menampilkan alur yang dilakukan pada tahap preprocessing, yaitu antara lain *cleaning data*, *case folding*, *tokenizing*, *stopword*, dan *stemming*. Setelah tahap *preprocessing* dilakukan, akan menghasilkan output data berupa data yang bersih tanpa ada simbol simbol yang tidak diperlukan.

Setelah melakukan proses yang ada, dilakukan tahap untuk perhitungan TF-IDF, alur untuk perhitungan TF-IDF dapat dilihat pada Gambar 3.3.



Gambar 3.3 Alur Perhitungan TF-IDF

3.2.1 Pengambilan Data

Pengambilan data adalah proses pengambilan data dari twitter untuk mendapatkan *tweet* yang berhubungan dengan NFT, menggunakan teknik Snsrape yang dijalankan menggunakan Google Colab. Data *tweet* yang diambil, merupakan data *tweet* yang berkaitan dengan kata kunci NFT, yaitu Token NFT, Opensea, dan Metaverse selama periode 01 Januari 2022- 31 Mei 2022. Library untuk pengambilan data dapat dilihat pada kode dibawah ini.

```
pip install snsrape
```

Kode diatas menampilkan kode library untuk pengambilan data *tweet* dengan menginstall snsrape menggunakan pip. Tahap yang dilakukan untuk pemanggilan library dapat dilihat dibawah ini.

```
import snsrape.modules.twitter as sntwitter
import pandas as pd
```

Diatas menampilkan pemanggilan library dengan dilakukan import untuk dapat digunakan pada tahap selanjutnya. Kode yang digunakan untuk pengambilan data *tweet* pada Twitter dapat dilihat pada kode dibawah ini.

```
maxTweets = 30000

tweets_list2 = []

for i,tweet in enumerate(sntwitter.TwitterSearchScrapper('Token NFT since
:2022-01-01 until:2022-07-31 lang:id').get_items()):
    if i>maxTweets:
        break
    tweets_list2.append([tweet.date, tweet.id, tweet.content, tweet.user
name])
```

Pada kode diatas menampilkan kode yang digunakan untuk pengambilan data *tweet* pada Twitter. Fungsi `maxTweets` diatas digunakan untuk mengatur maksimal data *tweet* yang akan diambil. Fungsi `tweet in enumerate` digunakan

untuk mencari kata kunci, dan periode tweet yang akan diambil. Kode untuk ekspor data tweet dapat dilihat dibawah ini.

```
tweets_df2.to_csv('datatokennft.csv', sep=',', index=False)
```

Pada kode diatas menampilkan fungsi kode untuk ekspor data Tweet kedalam file csv. Hasil dari data tweet yang berhasil di ekspor dapat dilihat pada tabel 3.1.

Tabel 3.1 Tabel data Tweet

No	Data Tweet
1.	Tren menyimpan karya dalam aset sertifikat digital "non-fungible token" atau NFT ikut melanda media massa. NFT digadang-gadang menjadi sekoci baru bisnis media. Namun, tantangan juga membayangi tren itu. #Polhuk #Kompas57
2.	Bill Gates menyebut, mata uang kripto dan aset digital non-fungible token (NFT) adalah palsu. Tren aset digital, menurut pendiri Microsoft tersebut, hanya berdasarkan greater fool theory alias teori kebodohan yang besar. #Video
3.	NFT alias non-fungible token tren di Indonesia karena Ghozali Everyday meraup miliaran rupiah berkat aset digital berupa foto diri (selfie). Transaksi NFT diprediksi semakin besar, dengan beragam utilitas seperti album musik dan sertifikat rumah
4.	Semenjak fenomena Ghozali Everyday mencuat, banyak masyarakat Indonesia yang kini ikut-ikutan berbisnis non-fungible token (NFT). #GhozaliJuraganNFT #GhozaliEveryday #NFT #lampostco
5.	Seiring dengan kesuksesan pemuda bernama Ghozali yang berhasil menjual koleksi swafoto miliknya dalam bentuk Non-Fungible Token (NFT) mencapai harga miliaran, banyak orang menjual produk nyeleneh lainnya di marketplace OpenSea.
6.	@dickycatty @vngnc @yourbae kalo lo ga ngerti gimana susahnya perjuangan arap memperkenalkan nft di indonesia mending lo diem. Izzy tu belum ada apa-apanya dibandingkan Arap.

7.	@NFTIndonesia_ Masih, min. Udh hampir 7 bulan, bikin aset nft. Ini my latest work. Thanks. https://t.co/mz176M4wk7
8.	@Saintsuffers_ @NFTIndonesia_ mengerikan
9.	@idnft_ Thanks for my team in @HarmonyNFT_ID makasih udah saling support dan berbagi hal-hal baru di dunia NFT, semoga Harmony NFT Indonesia bisa berkembang dan dikenal luas di Indonesia 🤍
10.	@favll_ @d0uxcalins @infotwitwor_ Orang Indonesia itu kebanyakan misused dalam segala aspek, contohnya penggunaan kata "lol" Dulu yang parah itu sempet trending opensea/tempat jual beli aset digital, dipake buat lapak dagang baju 😊

Pada Tabel 3.1 menunjukkan bahwa hasil dari pengambilan data ini masih mengandung komponen yang perlu dihilangkan, karena tidak digunakan pada pembuatan analisis sentimen. Komponen yang tidak digunakan dalam beberapa *tweet* ini akan dilakukan *preprocessing data* yang bertujuan untuk mendapatkan hasil data yang sesuai.

3.2.2 Preprocessing Data

Preprocessing data adalah proses pengolahan data yang dilakukan untuk memperbaiki data teks yang belum diperbaiki. Kode library yang digunakan pada tahap *preprocessing* data berupa *pandas* untuk mengatur data mentah menjadi terstruktur, *numpy* untuk mengolah data numerik, *nltk* untuk membantu pemrosesan teks.

```
import pandas as pd
import numpy as np
import nltk
import string
import emoji
import re
from pandas import DataFrame
from nltk.corpus import stopwords
    from nltk.tokenize import word_tokenize
```

1. Text Cleaning

Text cleaning merupakan tahap yang digunakan untuk menghapus html & url, *emoji*, *mention*, *hashtag*, dan menghilangkan simbol atau huruf yang tidak relevan. Kode yang digunakan untuk tahap *text cleaning* dapat dilihat dibawah ini

```
def url_remove(tweet):
    if not tweet:
        return ''
    t = re.sub(r'http\S+', '', tweet)
    return t

def punc_remove(tweet):
    if not tweet:
        return ''
    t = re.sub(r'^\w\s', '', tweet)
    return t

def rt_remove(tweet):
    if not tweet:
        return ''
    t = re.sub(r'RT[\s]+', '', tweet)
    return t

def number_remove(tweet):
    if not tweet:
        return ''
    t = re.sub('[0-9]+', '', tweet)
    return t

def slang_remove(tweet):
    if not tweet:
        return ''
    t = re.sub(r'\\n', " ", tweet)
    return t

def regex_remove(tweet):
    if not tweet:
        return ''
    t = re.sub("b'", " ", tweet)
    return t

def remove_user(tweet):
```

```

if not tweet:
    return ''
t = re.sub('@[^\s]+', '', tweet)
return t

def emoji_remove(tweet):
    if not tweet:
        return ''
    return emoji.replace_emoji(tweet, replace='').strip()

def hashtag_remove(tweet):
    reg = "#(\w+:\w+\/S+)"
    return re.sub(reg, " ", tweet)

cleaned = []

def clean_text(tweet):
    for i in tweet:
        tweet_raw = re.sub("[\n\r\t\xa0]", " ", str(i)).strip()
        if tweet_raw:
            cleaned.append(url_remove(punc_remove(number_remove(remove_use
r(regex_remove(hashtag_remove(rt_remove(emoji_remove(slang_remove(
tweet_raw))))))))))
        else:
            cleaned.append("")
    clean_text(data["Text"])

```

Pada kode diatas merupakan kode yang digunakan pada tahap cleaning data untuk menghapus simbol – simbol yang tidak diperlukan. Hasil dari tahap *cleaning* data dapat dilihat pada Tabel 3.2.

Tabel 3.2 Data *tweet* bersih

No	Data <i>Tweet</i>
1.	Tren menyimpan karya dalam aset sertifikat digital "non-fungible token" atau NFT ikut melanda media massa. NFT digadang-gadang menjadi sekoci baru bisnis media. Namun, tantangan juga membayangi tren itu.
2.	Bill Gates menyebut, mata uang kripto dan aset digital non-fungible token (NFT) adalah palsu. Tren aset digital, menurut pendiri Microsoft tersebut, hanya berdasarkan greater fool theory alias teori kebodohan yang besar.

3.	NFT alias non-fungible token tren di Indonesia karena Ghozali Everyday meraup miliaran rupiah berkat aset digital berupa foto diri (selfie). Transaksi NFT diprediksi semakin besar, dengan beragam utilitas seperti album musik dan sertifikat rumah
4.	Semenjak fenomena Ghozali Everyday mencuat, banyak masyarakat Indonesia yang kini ikut-ikutan berbisnis non-fungible token (NFT).
5.	Seiring dengan kesuksesan pemuda bernama Ghozali yang berhasil menjual koleksi swafoto miliknya dalam bentuk Non-Fungible Token (NFT) mencapai harga miliaran, banyak orang menjual produk nyeleneh lainnya di marketplace OpenSea.
6.	kalo lo ga ngerti gimana susahnya perjuangan arap memperkenalkan nft di indonesia mending lo diem. Izzy tu belum ada apa-apanya dibandingkan Arap.
7.	Masih, min. Udh hampir 7 bulan, bikin aset nft. Ini my latest work. Thanks.
8.	mengerikan
9.	Thanks for my team in makasih udah saling support dan berbagi hal-hal baru di dunia NFT, semoga Harmony NFT Indonesia bisa berkembang dan dikenal luas di Indonesia
10.	Orang Indonesia itu kebanyakan misused dalam segala aspek, contohnya penggunaan kata "lol" Dulu yang parah itu sempet trending opensea/tempat jual beli aset digital, dipake buat lapak dagang baju

Pada Tabel 3.2 menampilkan data *tweet* yang sudah dilakukan tahap *cleaning* data, sehingga mendapatkan data tweet yang bersih dan tidak mengandung html & url, *emoji*, *hashtag*, dan *mention*.

2. Tokenizing

Tokenizing adalah proses untuk memecah kalimat-kalimat menjadi kata. Kode yang digunakan pada proses *tokenizing* dapat dilihat dibawah ini.

```
cleaned = []
```

```
def clean_text(tweet):
    for i in tweet:
        tweet_raw = re.sub("[\n\r\t\xa0]", " ", str(i)).strip()
```

```

        if tweet_raw:
            cleaned.append(url_remove(punc_remove(number_remove(remove_use
r(regex_remove(hashtag_remove(rt_remove(emoji_remove(slang_remove(
            tweet_raw))))))))))
        else:
            cleaned.append("")
clean_text(data["Text"])

data['tweet'] = cleaned
data.head(10)

```

3. Case Folding

Case Folding adalah proses untuk menyeragamkan karakter pada data (*lowercase* atau *uppercase*). Kode yang digunakan pada tahap *Case Folding* dapat dilihat dibawah ini.

```

def lowercase():
    lower_word = data['tweet'].str.lower()
    return lower_word

lower_tweet = lowercase()

print(lower_tweet)

```

Hasil dari tahap *Lowercase* dapat dilihat dibawah ini pada Tabel 3.3.

Tabel 3.3 Hasil *Lowercase*

No	Data Tweet
1.	tren menyimpan karya dalam aset sertifikat digital "non-fungible token" atau nft ikut melanda media massa. nft digadang-gadang menjadi sekoci baru bisnis media. namun, tantangan juga membayangi tren itu.
2.	bill gates menyebut, mata uang kripto dan aset digital non-fungible token (nft) adalah palsu. tren aset digital, menurut pendiri microsoft tersebut, hanya berdasarkan greater fool theory alias teori kebodohan yang besar.
3.	nft alias non-fungible token tren di indonesia karena ghozali everyday meraup miliaran rupiah berkat aset digital berupa foto

	diri (selfie). transaksi nft diprediksi semakin besar, dengan beragam utilitas seperti album musik dan sertifikat rumah
4.	semenjak fenomena ghozali everyday mencuat, banyak masyarakat indonesia yang kini ikut-ikutan berbisnis non-fungible token (nft).
5.	seiring dengan kesuksesan pemuda bernama ghozali yang berhasil menjual koleksi swafoto miliknya dalam bentuk non-fungible token (nft) mencapai harga miliaran, banyak orang menjual produk nyeleneh lainnya di marketplace opensea.
6.	kalo lo ga ngerti gimana susahnya perjuangan arap memperkenalkan nft di indonesia mending lo diem. izzy tu belum ada apa-apanya dibandingkan arap.
7.	masih, min. udh hampir 7 bulan, bikin aset nft. ini my latest work. thanks.
8.	mengerikan
9.	thanks for my team in makasih udah saling support dan berbagi hal-hal baru di dunia nft, semoga harmony nft indonesia bisa berkembang dan dikenal luas di indonesia
10.	orang indonesia itu kebanyakan misused dalam segala aspek, contohnya penggunaan kata "lol" dulu yang parah itu sempet trending opensea/tempat jual beli aset digital, dipake buat lapak dagang baju

Pada Tabel 3.3 diatas menampilkan hasil data yang sudah dilakukan proses case folding, dengan menggunakan fungsi *lowercase* yaitu mengubah keseluruhan tulisan menjadi huruf kecil.

4. Stopword

Stopword adalah proses yang digunakan untuk menghilangkan kata yang tidak memiliki makna. Kode yang digunakan pada tahap *stopword* dapat dilihat dibawah ini.

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
```

```
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
```

```
stopwords = factory.get_stop_words()
print(stopwords)
```

Hasil dari proses *Stopword* dapat dilihat pada Tabel 3.4.

Tabel 3.4 Hasil Library Sastrawi

Hasil kata dari Library Sastrawi
<pre>['yang', 'untuk', 'pada', 'ke', 'para', 'namun', 'menurut', 'antara', 'dia', 'dua', 'ia', 'seperti', 'jika', 'jika', 'sehingga', 'kembali', 'dan', 'tidak', 'ini', 'karena', 'kepada', 'oleh', 'saat', 'harus', 'sementara', 'setelah', 'belum', 'kami', 'sekitar', 'bagi', 'serta', 'di', 'dari', 'telah', 'sebagai', 'masih', 'hal', 'ketika', 'adalah', 'itu', 'dalam', 'bisa', 'bahwa', 'atau', 'hanya', 'kita', 'dengan', 'akan', 'juga', 'ada', 'mereka', 'sudah', 'saya', 'terhadap', 'secara', 'agar', 'lain', 'anda', 'begitu', 'mengapa', 'kenapa', 'yaitu', 'yakni', 'daripada', 'itulah', 'lagi', 'maka', 'tentang', 'demi', 'dimana', 'kemana', 'pula', 'sambil', 'sebelum', 'sesudah', 'supaya', 'guna', 'kah', 'pun', 'sampai', 'sedangkan', 'selagi', 'sementara', 'tetapi', 'apakah', 'kecuali', 'sebab', 'selain', 'seolah', 'seraya', 'seterusnya', 'tanpa', 'agak', 'boleh', 'dapat', 'dsb', 'dst', 'dll', 'dahulu', 'dulunya', 'anu', 'demikian', 'tapi', 'ingin', 'juga', 'nggak', 'mari', 'nanti', 'melainkan', 'oh', 'ok', 'seharusnya', 'sebetulnya', 'setiap', 'setidaknya', 'sesuatu', 'pasti', 'saja', 'toh', 'ya', 'walau', 'tolong', 'tentu', 'amat', 'apalagi', 'bagaimanapun']</pre>

Tabel 3.4 diatas merupakan hasil dari tahap *Stopword*, yaitu data yang tidak memiliki makna yang nantinya akan dihilangkan. Kode yang digunakan untuk menghilangkan kata yang tidak memiliki makna, dapat dilihat dibawah ini.

```
def removeStopWords(tweet):
    clean_word_list = [word for word in tweet.split() if word not in stop
words]
    return clean_word_list

stopwords_tweet = lower_tweet.apply(removeStopWords)

print(stopwords_tweet)
```

5. Stemming

Stemming merupakan proses untuk merubah kata menjadi kata dasar. Kode yang digunakan pada proses *Stemming* dapat dilihat dibawah ini.

```

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in stopwords_tweet:
    for term in document:
        if term not in term_dict:
            term_dict[term] = " "

print(len(term_dict))
print("-----")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":", term_dict[term])

print(term_dict)
print("-----")

def get_stemmed_term(document):
    return [term_dict[term] for term in document]

stem_tweet = stopwords_tweet.apply(get_stemmed_term)

print(stem_tweet)

```

6. Normalisasi

Normalisasi adalah proses yang digunakan untuk mengubah menjadi kalimat yang baku. Kode yang digunakan pada proses normalisasi dapat dilihat dibawah ini.

```

normalizad_word = pd.read_excel("normalisasi_fix.xlsx")

normalizad_word_dict = {}

```

```

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

def normalized_term(document):
    return [normalizad_word_dict[term] if term in normalizad_word_dict e
lse term for term in document]

normal_tweet = stem_tweet.apply(normalized_term).str.join(" ")

print(normal_tweet)

```

Pada kode diatas merupakan proses normalisasi dilakukan perubahan yang tidak baku menjadi baku sesuai dengan data yang ada. Contoh kata yang diproses pada tahap normalisasi dapat dilihat pada Tabel 3.5.

Tabel 3.5 Daftar kata Normalisasi

No	Sebelum	Sesudah
1.	tak	Tidak
2.	slalu	Selalu
3.	aja	Saja
4.	tu	Itu
5.	d	Di
6.	dah	Sudah
7.	Onok	Ada
8.	Gaiso	Tidak bisa
9.	Tak	Tidak
10.	Tau	Tahu

Tabel 3.6 Hasil tahap *preprocessing*

No	Data hasil <i>preprocessing</i>
1.	tren menyimpan karya dalam aset sertifikat digital "non-fungible token" atau nft ikut melanda media massa. nft digadang-gadang menjadi sekoci baru bisnis media. namun, tantangan juga membayangi tren itu.
2.	bill gates menyebut, mata uang kripto dan aset digital non-fungible token (nft) adalah palsu. tren aset digital, menurut pendiri microsoft tersebut, hanya berdasarkan greater fool theory alias teori kebodohan yang besar.
3.	nft alias non-fungible token tren di indonesia karena ghozali everyday meraup miliaran rupiah berkat aset digital berupa foto diri (selfie). transaksi nft diprediksi semakin besar, dengan beragam utilitas seperti album musik dan sertifikat rumah
4.	semenjak fenomena ghozali everyday mencuat, banyak masyarakat indonesia yang kini ikut-ikutan berbisnis non-fungible token (nft).
5.	seiring dengan kesuksesan pemuda bernama ghozali yang berhasil menjual koleksi swafoto miliknya dalam bentuk non-fungible token (nft) mencapai harga miliaran, banyak orang menjual produk nyeleneh lainnya di marketplace opensea.
6.	kalo lo ga ngerti gimana susahnya perjuangan arap memperkenalkan nft di indonesia mending lo diem. izzy tu belum ada apa-apanya dibandingkan arap.
7.	masih, min. udh hampir 7 bulan, bikin aset nft. ini my latest work. thanks.
8.	Mengerikan
9.	thanks for my team in makasih udah saling support dan berbagi hal-hal baru di dunia nft, semoga harmony nft indonesia bisa berkembang dan dikenal luas di indonesia
10.	orang indonesia itu kebanyakan misused dalam segala aspek, contohnya penggunaan kata "lol" dulu yang parah itu sempet trending opensea/tempat jual beli aset digital, dipake buat lapak dagang baju

Pada Tabel 3.6 diatas menampilkan tapel hasil dari tahap *preprocessing*, yang berisi data *tweet* yang sudah bersih yang berarti sudah tidak ada url atau html, tidak ada *emoji*, maupun simbol simbol yang tidak diperlukan.

3.2.3 Pelabelan Manual

Pada proses pelabelan ini dilakukan dengan melabeli kata atau kalimat secara manual, yang bertujuan untuk menganalisis lebih lanjut mengenai sifatnya yang positif atau negatif. Berdasarkan data *tweet* yang sudah dilakukan *Preprocessing* lalu dilakukan pemberian label secara manual. Pada tahap pelabelan manual dari 25.623 data *tweet* mendapatkan data sebanyak 800 *tweet*, dengan rincian 400 data *tweet* bersifat positif dan 400 data *tweet* bersifat negatif. Data ini nantinya akan digunakan untuk *training data*, Contoh *tweet* yang sudah diberi label secara manual dapat dilihat pada Tabel 3.7.

Tabel 3.7 Tabel pelabelan manual

No	Kelas	Skor	<i>Tweet</i>
1.	Positif	1	di jadikan nft saja terus di jual di open sea siapa tahu laku terus jadi cuan
2.	Positif	1	main opensea memang banyak cuan sih
3.	Positif	1	waaa gilee kerennn banget
4.	Positif	1	opensea umum fitur aman baru lindung guna tipu nft
5.	Positif	1	belakang nft nonfungible token jadi salah satu topik ramai bincang utama sejak muncul fenomena ghozali everyday nft bahkan nilai potensi bantu seniman kreator jual karya nft hakcipta
6.	Negatif	0	punya banyak tanah metaverse tapi ngegembel dunia ternyata
7.	Negatif	0	biasa tengah jadi tren jadi incar jahat dunia maya masuk populer nonfungible token nft
8.	Negatif	0	dagang terus sampai muka tidak urus

9.	Negatif	0	investor koleksi nonfungible token nft hilang uang atas tipu lebih juta usd miliar rupiah
10	Negatif	0	agak aneh dengan metode jual offer ini

Dari Tabel 3.7 diatas didapatkan informasi bahwa kelas positif diberi label dengan nilai 1, dan kelas negatif diberi label dengan nilai 0. Pelabelan manual dilakukan untuk memberikan nilai sentimen mengenai sifatnya yang positif atau negatif, yang selanjutnya akan dihitung nilai akurasi.

3.2.4 Training Data

Training data adalah proses *training* pada data menggunakan metode SVM. Tahap proses *training* ini diawali dengan mengekstraksi data menggunakan TF-IDF (*term frequency-inverse document frequency*). Selanjutnya dilakukan proses *training* data untuk menghasilkan model klasifikasi yang dapat digunakan untuk melakukan sentimen secara otomatis. Kode library untuk SVM dapat dilihat dibawah ini.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

from matplotlib import pyplot as plt
%matplotlib inline
```

Pada kode diatas menunjukan library yang digunakan pada metode SVM, Kode yang digunakan untuk pemanggilan data *training* dapat dilihat dibawah ini.

```
df = pd.read_excel ('data_training.xlsx')
df=pd.DataFrame(df)
df=df.dropna()
df
```

Pada kode diatas menampilkan kode yang digunakan untuk pemanggilan data *training* yang telah di beri label dan skor yang disimpan pada file excel. Hasil dari pemanggilan data *training* dapat dilihat pada Gambar 3.4.

No	Kelas	Skor	Tweet	
0	1.0	positif	1.0	accenture persen pimpin usaha yakin metaverse ...
1	2.0	positif	1.0	aduh guna twitter kabar bagus ini twitter lunc...
2	3.0	positif	1.0	aduh terima kasih banyakbanyak collect mas
3	4.0	positif	1.0	akan buat metaverse versi borobudur borobudur ...
4	5.0	positif	1.0	akhir bisa buka opensea
...
795	796.0	negatif	0.0	ya sudah aku tidak urus
796	797.0	negatif	0.0	yakin tidak salah metaverse bang
797	798.0	negatif	0.0	yang penting hati tidak dekil
798	799.0	negatif	0.0	yang sudah main metaverse sini main dukun asta...
799	800.0	negatif	0.0	zaman sudah canggih era metaverse semacam beli...

800 rows × 4 columns

Gambar 3.4 Hasil pemanggilan data *training*

Pada Gambar 3.4 diatas adalah hasil pemanggilan data *training* yang telah diberi label secara manual. Contoh dokumen yang nantinya akan digunakan untuk perhitungan TF-IDF dapat dilihat pada Tabel 3.8.

Tabel 3.8 Tabel Term

Dokumen (d)	Kalimat
d1	agak aneh dengan offer ini.
d2	biasa tengah jadi tren jadi incar jahat dunia maya masuk populernonfungible token nft
d3	new post investasi nonfungible token nft terlalu resiko belabelain pansos ntaaar nanggiis

d4	Nonfungible token nft sedang naik daun nft sendiri bilang salah satu cara dapat cuan lebih banyak internet.
----	---

Pada Tabel 3.8 tabel term terdapat 4 dokumen yang berlabel d1 berisikan dokumen 1, d2 berisikan dokumen 2, d3 berisikan dokumen 3, dan d4 berisikan dokumen 4 kalimat dari *tweet* yang nantinya akan digunakan untuk perhitungan IDF secara manual. Perhitungan TF dapat dilihat pada Tabel 3.9.

Tabel 3.9 Tabel perhitungan TF

Term	d1	d2	d3	d4	Df
Agak	1				1
Aneh	1				1
dengan	1				1
Offer	1				1
Ini	1				1
Biasa		1			1
tengah		1			1
jadi		1			1
tren		1			1
incar		1			1
jahat		1			1
dunia		1			1
maya		1			1
masuk		1			1
populer		1			1
nonfungible		1	1	1	3
token		1	1	1	3
nft		1	1	1	3
new			1		1
post			1		1
investasi			1		1

terlalu			1		1
resiko			1		1
belain			1		1
pansos			1		1
ntar			1		1
nangis			1		1
sedang				1	1
naik				1	1
daun				1	1
sendiri				1	1
bilang				1	1
salah				1	1
satu				1	1
cara				1	1
dapat				1	1
cuan				1	1
lebih				1	1
banyak				1	1
internet				1	1

Pada Tabel 3.9 diatas memiliki data yang digunakan untuk perhitungan TF, yang memiliki persamaan (1) pada setiap kata. Perhitunagn IDF dapat dilihat pada Tabel 3.10.

Tabel 3.10 Perhitungan IDF

Term	df	idf	Idf (n=4)	Idf(n=800)
Agak	1	1	0,60206	2,903089
Aneh	1	1	0,60206	2,903089
Dengan	1	1	0,60206	2,903089
Offer	1	1	0,60206	2,903089
Ini	1	1	0,60206	2,903089
Biasa	1	1	0,60206	2,903089

Tengah	1	1	0,60206	2,903089
Jadi	1	1	0,60206	2,903089
Tren	1	1	0,60206	2,903089
Incar	1	1	0,60206	2,903089
Jahat	1	1	0,60206	2,903089
Dunia	1	1	0,60206	2,903089
Maya	1	1	0,60206	2,903089
Masuk	1	1	0,60206	2,903089
Populer	1	1	0,60206	2,903089
nonfungible	3	0,33333	0,12493	2,425968
Token	3	0,33333	0,12493	2,425968
Nft	3	0,33333	0,12493	2,425968
New	1	1	0,60206	2,903089
Post	1	1	0,60206	2,903089
Investasi	1	1	0,60206	2,903089
Terlalu	1	1	0,60206	2,903089
Resiko	1	1	0,60206	2,903089
Belain	1	1	0,60206	2,903089
pansos	1	1	0,60206	2,903089
Ntar	1	1	0,60206	2,903089
nangis	1	1	0,60206	2,903089
sedang	1	1	0,60206	2,903089
Naik	1	1	0,60206	2,903089
Daun	1	1	0,60206	2,903089
sendiri	1	1	0,60206	2,903089
Bilang	1	1	0,60206	2,903089
Salah	1	1	0,60206	2,903089
Satu	1	1	0,60206	2,903089
Cara	1	1	0,60206	2,903089
Dapat	1	1	0,60206	2,903089
Cuan	1	1	0,60206	2,903089

Lebih	1	1	0,60206	2,903089
banyak	1	1	0,60206	2,903089
internet	1	1	0,60206	2,903089

Tabel 3.10 menunjukkan perhitungan IDF dengan rumus persamaan (1), yang dihitung secara manual. Setelah menghitung nilai TF dan nilai IDF secara manual, selanjutnya dilakukan perhitungan TF-IDF secara manual, dapat dilihat pada Tabel 3.11.

Tabel 3.11 Perhitungan TF-IDF Manual

Term(kata)	d1	d2	d3	d4
Agak	0,60206			
Aneh	0,60206			
Dengan	0,60206			
Offer	0,60206			
Ini	0,60206			
Biasa		0,60206		
Tengah		0,60206		
Jadi		0,60206		
Tren		0,60206		
Incar		0,60206		
Jahat		0,60206		
Dunia		0,60206		
Maya		0,60206		
Masuk		0,60206		
Populer		0,60206		
nonfungible		0,12493	0,12493	0,12493
Token		0,12493	0,12493	0,12493
Nft		0,12493	0,12493	0,12493
New			0,60206	
Post			0,60206	

Investasi			0,60206	
Terlalu			0,60206	
Resiko			0,60206	
Belain			0,60206	
pansos			0,60206	
ntar			0,60206	
nangis			0,60206	
sedang				0,60206
naik				0,60206
daun				0,60206
sendiri				0,60206
bilang				0,60206
salah				0,60206
satu				0,60206
cara				0,60206
dapat				0,60206
cuan				0,60206
lebih				0,60206
banyak				0,60206
internet				0,60206

Perhitungan klasifikasi TF-IDF dilakukan untuk menghasilkan perhitungan pada pembobotan kata dalam setiap dokumen secara otomatis pada google colab. Kode perhitungan TF-IDF dapat dilihat dibawah ini.

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
d1 = "agak aneh dengan offer ini."
```

```
d2 = "biasa tengah jadi tren jadi incar jahat dunia maya masuk populerno  
nfungible token nft"
```

```
d3 = "new post investasi nonfungible token nft terlalu resiko belabelain  
pansos ntaaar nangiiis"
```

```
d4 = "Nonfungible token nft sedang naik daun nft sendiri bilang salah sa
tu cara dapat cuan lebih banyak internet."
```

```
vect = TfidfVectorizer()
X = vect.fit_transform([d1, d2, d3, d4])
```

```
X.toarray()
```

```
[(0.0, 'agak'), (0.0, 'aneh'), (0.25271028564649684,
'banyak'), (0.0, 'belabelain'), (0.0, 'biasa'),
(0.25271028564649684, 'bilang'), (0.25271028564649684,
'cara'), (0.25271028564649684, 'cuan'),
(0.25271028564649684, 'dapat'), (0.25271028564649684,
'daun'), (0.0, 'dengan'), (0.0, 'dunia'),
```

Gambar 3.5 Hasil perhitungan TF-IDF pada Google Colab

Pada Gambar 3.5 diatas merupakan hasil dari perhitungan TF-IDF yang dilakkukan perhitungan pada sistem di Google Colab. Kode yang digunakan untuk perhitungan *confusion matrix* dapat dilihat dibawah ini.

```
from sklearn.model_selection import ShuffleSplit
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
X = df.Tweet
y = df.Kelas

ss = ShuffleSplit(n_splits=10, test_size=0.2)
sm = SMOTE()

accs = []
f1s = []
cms = []

for train_index, test_index in ss.split(X):

    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
```



```

y_train, y_test = y.iloc[train_index], y.iloc[test_index]

X_train_vect = vect.fit_transform(X_train)
X_test_vect = vect.transform(X_test)

X_train_res, y_train_res = sm.fit_resample(X_train_vect, y_train)

SVM.fit(X_train_res, y_train_res)
y_pred = SVM.predict(X_test_vect)

accs.append(accuracy_score(y_test, y_pred))
f1s.append(f1_score(y_test, y_pred, average='weighted'))
cms.append(confusion_matrix(y_test, y_pred))

print("\nAverage accuracy across folds: {:.2f}%".format(sum(accs) / len(
accs) * 100))
print("\nAverage F1 score across folds: {:.2f}%".format(sum(f1s) / len(f
1s) * 100))
print("\nAverage Confusion Matrix across folds: \n {}".format(sum(cms) /
len(cms)))

y_pred

```

Pada Kode diatas digunakan untuk perhitungan Confusion Matrix pada *training* data. Perhitungan ini menghasilkan informasi perbandingan hasil klasifikasi yang dilakukan pada sistem, sehingga dapat diketahui hasil dari kinerja pada model yang telah dibuat.

```

import os
import pickle
from sklearn.pipeline import Pipeline
from sklearn.svm import SVC
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import LabelEncoder

x = df.Tweet
y = df.Kelas

Encoder = LabelEncoder()

```

```

Train_Y = Encoder.fit_transform(y)

Tfidf_vect = TfidfVectorizer()
Tfidf_vect.fit(x)
Train_X_Tfidf = Tfidf_vect.transform(x)

SVM = SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
text_clf=SVM.fit(Train_X_Tfidf,Train_Y)

files = open('klasifikasi_SVM_.pickle', 'wb')
pickle.dump(text_clf, files)
files.close()

print('Proses Training SVM Selesai!')

```

Pada kode diatas merupakan kode yang digunakan untuk menyimpan file dengan format pickle agar dapat menyimpan dan membaca data dari sebuah file, yang nantinya akan digunakan pada tahap klasifikasi SVM.

3.2.5 Metode Support Vector Machine

Klasifikasi dengan metode SVM ini, pada tahap sebelumnya sudah dilakukan pemrosesan data dengan menghasilkan file pickle. Pada kode ini digunakan untuk pemanggilan file pickel, data ini nantinya akan digunakan untuk perhitungan *k-fold* perujian data. Kode pemanggilan file pickle dapat dilihat dibawah ini.

```

model = open('SVM_classifier.pickle', 'rb')
svm_classifier = pickle.load(model)
svm_classifier

```

Kode dibawah ini merupakan kode yang digunakan untuk menghitung mengujian *k-fold Accuracy* dan *f-measure*.

```

fig, (ax1, ax2) = plt.subplots(2, 1, sharex=True, figsize=(16,9))

acc_scores = [round(a * 100, 1) for a in accs]
f1_scores = [round(f * 100, 2) for f in f1s]

```

```

x1 = np.arange(len(acc_scores))
x2 = np.arange(len(f1_scores))

ax1.bar(x1, acc_scores)
ax2.bar(x2, f1_scores, color='#559ebf')

# Place values on top of bars
for i, v in enumerate(list(zip(acc_scores, f1_scores))):
    ax1.text(i - 0.25, v[0] + 2, str(v[0]) + '%')
    ax2.text(i - 0.25, v[1] + 2, str(v[1]))

ax1.set_ylabel('Accuracy (%)')
ax1.set_title('SVM')
ax1.set_ylim([0, 100])

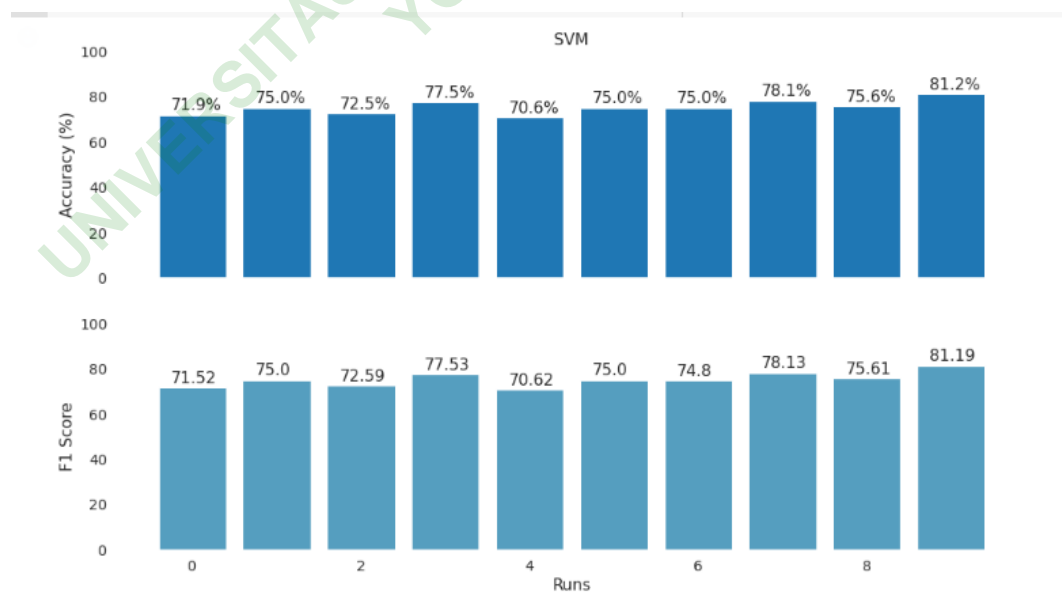
ax2.set_ylabel('F1 Score')
ax2.set_xlabel('Runs')
ax2.set_ylim([0, 100])

sns.despine(bottom=True, left=True)

plt.show()

```

Hasil dari menghitung mengujian *k-fold Accuracy* dan *f-measure* dapat dilihat pada Gambar 3.6.



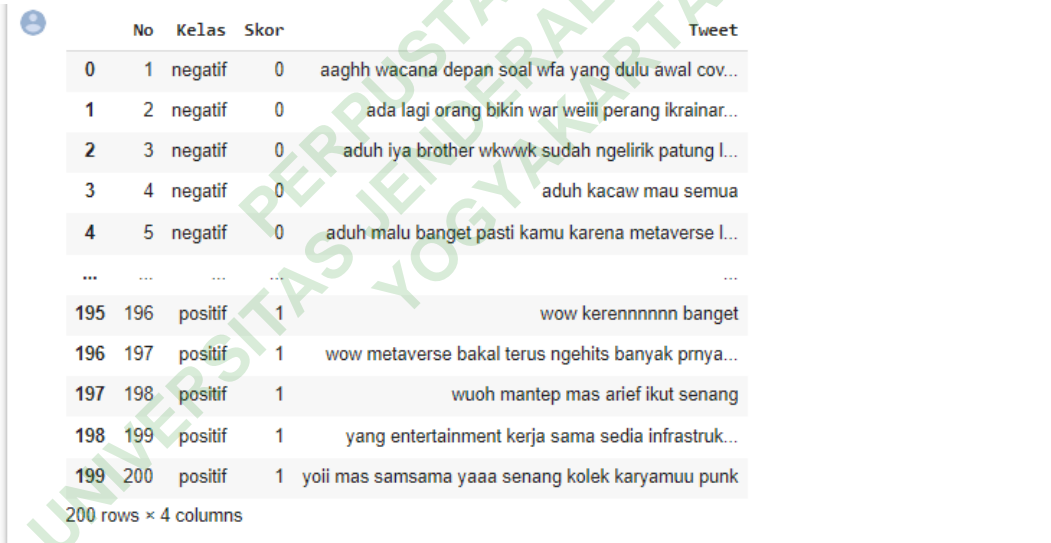
Gambar 3.6 Grafik *k-fold Accuracy* dan *f-measure*

Pada gambar 3.6 diatas menampilkan grafik nilai *accuracy* dan *f-measure* yang sudah diuji sebanyak 10 kali pengujian, untuk melihat hasil akurasi tertinggi. Menghasilkan nilai paling tinggi yaitu pada pengujian ke 10 dengan nilai 81.2% untuk *Accuracy* dan nilai 81.19 untuk *f-measure*.

3.2.6 Testing

```
df = pd.read_excel ('data_testing.xlsx')
df=pd.DataFrame(df)
df=df.dropna()
df
```

Pada kode diatas merupakan kode yang digunakan untuk pemanggilan data *testing* yang telah disimpan sebelumnya dalam bentuk file excel. Hasil tampilan data *testing* dapat dilihat pada Gambar 3.7.



No	Kelas	Skor	Tweet
0	negatif	0	aaghh wacana depan soal wfa yang dulu awal cov...
1	negatif	0	ada lagi orang bikin war weiii perang ikrainar...
2	negatif	0	aduh iya brother wkwwk sudah ngelirik patung I...
3	negatif	0	aduh kacaw mau semua
4	negatif	0	aduh malu banget pasti kamu karena metaverse I...
...
195	positif	1	wow kerennnnnn banget
196	positif	1	wow metaverse bakal terus ngehits banyak prnya...
197	positif	1	wuoh mantep mas arief ikut senang
198	positif	1	yang entertainment kerja sama sedia infrastruk...
199	positif	1	yoi mas samsama yaaa senang kolek karyamu punk

200 rows × 4 columns

Gambar 3.7 Hasil pemanggilan data *Testing*

Pada Gambar 3.7 diatas menampilkan hasil dari pemanggilan data testing yang berisikan data *tweet* yang telah diberikan label positif maupun negatif dan skor. Data testing ini berisikan 200 data *tweet*, dengan rincian 100 data berlabel positif dan 100 data berlabel negatif. Data testing ini nantinya akan digunakan untuk tahap perhitungan *confusion matrix* data *testing*, kode perhitungannya dapat dilihat dibawah ini.

```

from sklearn.metrics import accuracy_score, f1_score, confusion_matrix

print("Accuracy: {:.2f}%".format(accuracy_score(y_test, y_pred) * 100))
print("\nF1 Score: {:.2f}".format(f1_score(y_test, y_pred, average='weighted') * 100))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))

```

Accuracy: 81.25%

F1 Score: 81.19

Confusion Matrix:
[[61 20]
[10 69]]

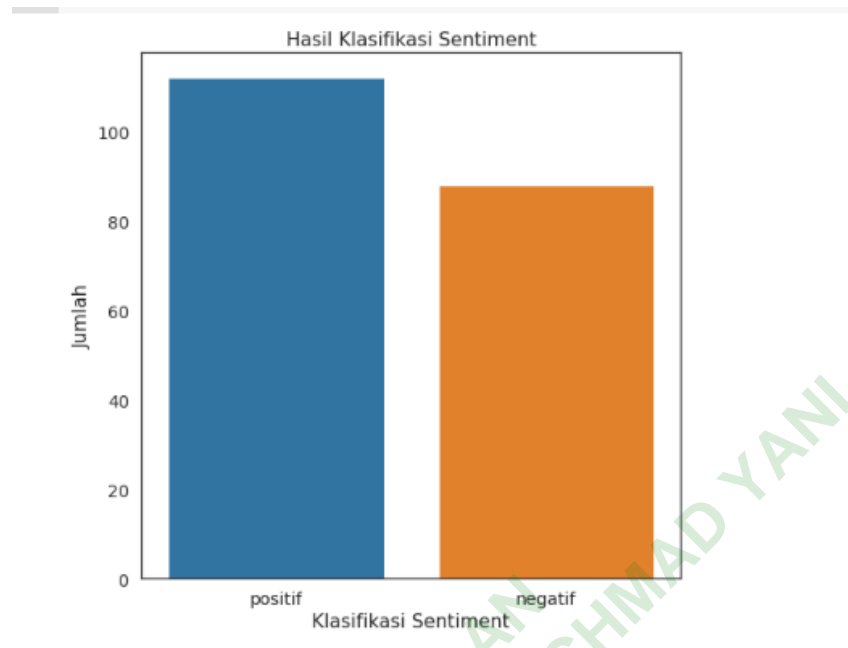
Pada kode diatas menghasilkan data *accuracy* sebesar 81.25% dan *f-measure* sebesar 81.19. Pada perhitungan *Confusion Matrix* dari data testing dengan hasil TP sebesar 61, FN sebesar 20, FP sebesar 10, TN sebesar 69. Kode yang digunakan untuk menampilkan grafik dilihat dibawah ini.

```

import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(8, 8))
sentiment = ['positif', 'negatif']
jumlah_sentiment = [106, 94]
sns.barplot(sentiment, jumlah_sentiment)
ax.set_ylabel('Jumlah')
ax.set_xlabel('Klasifikasi Sentiment')
ax.set_title('Hasil Klasifikasi Sentiment')
plt.show()

```

Pada kode diatas digunakan untuk menampilkan grafik klasifikasi data testing, dengan jumlah sentimen positif sebanyak 112 data, dan jumlah sentimen negatif sebanyak 88 data. Tampilan grafik dapat dilihat pada gambar 3.8.



Gambar 3.8 Grafik klasifikasi sentimen data *training*

Pada gambar 3.8 menampilkan grafik klasifikasi sentimen dari data *training* dengan label berwarna biru menunjukkan sentimen positif berjumlah 112 data, dan label berwarna oranye menunjukkan sentimen negatif berjumlah 88 data.

```
accuracy = (TP+TN)/(TP+TN+FP+FN)
print('Accuracy =', accuracy)
```

Pada kode diatas menunjukkan perhitungan akurasi pada data testing menunjukkan nilai *accuracy* sebesar 74%.

```
recall = (TP) / (TP + FN)
print('Recall =', recall)
```

Pada kode diatas merupakan perhitungan *recall* yang menghasilkan data sebesar 72%.

```
precision = (TP) / (TP+FP)
precision = (TP) / (TP+FP)
```

Pada kode di atas merupakan kode untuk melakukan perhitungan *precision*, setelah dilakukan perhitungan dengan rumus yang digunakan menghasilkan data yaitu sebesar 77%.

```
F1_Score = 2 * (recall*precision) / (recall + precision)
print('F1Score =', F1_Score)
```

Pada kode di atas merupakan kode yang digunakan untuk melakukan perhitungan *F1 score*. Pada perhitungan ini di dapatkan dari rata rata nilai dari *recall* dan *precision*.

3.2.7 Klasifikasi

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.rcParams["figure.figsize"] = [8,8]
plt.rcParams["figure.autolayout"] = True
x = ['Positif', 'Negatif',]
y = [7683,16970]
percentage = [7683,16970]
ax = sns.barplot(x=x, y=y)
patches = ax.patches
for i in range(len(patches)):
    x = patches[i].get_x() + patches[i].get_width()/2
    y = patches[i].get_height()+.5
    ax.annotate('{:}'.format(percentage[i]), (x, y), ha='center')

plt.title('Klasifikasi Sentimen \n')
plt.xlabel('Jenis Klasifikasi')
plt.ylabel('Jumlah')
plt.show()
```

Pada kode di atas merupakan kode untuk menampilkan grafik klasifikasi sentimen, dengan persamaan x = positif dan negatif, persamaan y = 7683 ber label positif, dan 16970 ber label negatif.