ISSN: 2715-2766
Vol. 6 No: 2, Juli 2025

Penggunaan Algoritma Naive Bayes dan Support Vector Machine dalam Analisis Sentimen Pengguna X terhadap Karakter Luffy pada Animasi One Piece

FajarAgung
Universitas Buana Perjuangan
Karawang, Indonesia
if17.fajaragung@mhs.ubpkarawang.ac.id

Yana Cahyana
Universitas Buana Perjuangan
Karawang, Indonesia
yana.cahyana@ubpkarawang.ac.id

Sutan Faisal
Universitas Buana Perjuangan
Karawang, Indonesia
sutan.faisal@ubpkarawang.ac.id

***Abstract*—** Beragam platform media sosial, salah satunya seperti X, membantu mempercepat proses penyebaran informasi dan menyampaikan perspektif serta pendapat dari berbagai komunitas. Dalam penelitian ini, penulis menggunakan pendekatan kuantitatif, yaitu proses pencarian data dalam jejaring sosial X untuk mengumpulkan informasi publik tentang pendapat dan perasaan orang-orang yang menantikan episode wujud Sun God Nika dalam animasi One Piece. Sebanyak 3.121 data dari media sosial X, yang dikumpulkan dari 27 Desember hingga 20 Januari 2024, diperoleh menggunakan teknik crawling dengan pengembangan program sederhana berbasis Python. Hasil labelling menunjukkan 1.191 sentimen positif, 187 sentimen negatif, dan 1.200 sentimen netral. Nilai akurasi untuk algoritma Naive Bayes sebesar 84%, sedangkan nilai akurasi untuk algoritma Support Vector Machine sebesar 87%.

Kata kunci — Data mining, Naive Bayes, Support Vector Machine, Luffy, One Piece

I. PENDAHULUAN

Pesatnya perkembangan media sosial di zaman serba digital membuat banyak penggunanya menuliskan berbagai pendapat terhadap suatu hal yang sedang atau akan terjadi. Beragamnya media sosial, layaknya X, mempercepat proses penyebaran informasi dan memberikan sudut pandang ataupun pendapat berbagai komunitas [1]. Pengguna media sosial menjadikan X sebagai wadah bertukar mengenai kegemarannya terhadap karya animasi Jepang One Piece, yang menjadi animasi terpopuler dan salah satu trending X pada setiap halaman komik maupun episode animasinya. Informasi media sosial X bermanfaat dalam menguak pendapat serta perasaan publik yang menantikan episode wujud Sun God Nika animasi One Piece dalam pertarungan karakter Monkey D. Luffy dengan Kapten Bajak Laut Binatang Buas Kaido yang membangkitkan kekuatan dalam tubuh Luffy menjadi wujud Sun God Nika.

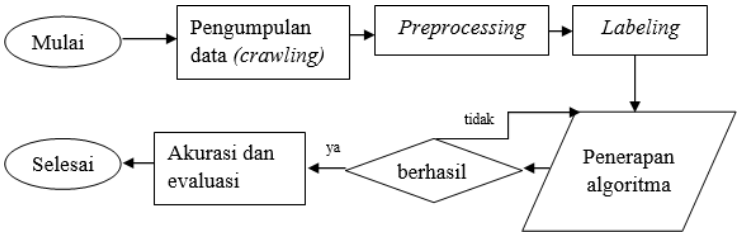
Terdapat beberapa penelitian yang telah dipublikasi pada penggunaan algoritma machine learning yang beragam dalam analisis sentimen. Dalam penelitian analisis sentimen pada komentar YouTube terhadap GeNose dengan Naive Bayes yang dilakukan [2], memperoleh hasil analisis sentimen menggunakan metode Naive Bayes tentang opini pengguna terhadap GeNose dengan dataset pada kolom komentar YouTube dengan nilai akurasi sebesar 90,70%, yang berisi komentar negatif. Jumlah nilai recall yang berhasil didapatkan sebesar 87,60%, sedangkan jumlah nilai precision untuk identifikasi prediksi positif yang ada pada kelas negatif menghasilkan jumlah persentase sebesar 93,39%. Berdasarkan hasil tersebut, dapat disimpulkan bahwa GeNose tidak maksimal dalam meminimalisir COVID-19. Adapun penelitian yang sama dilakukan oleh [3] terkait pandangan masyarakat di media sosial Twitter terhadap pemilihan umum pada tahun 2019 menunjukkan hasil nilai akurasi 85%. Penelitian lain yang dilakukan [4] menunjukkan hasil akurasi menggunakan algoritma Support Vector Machine memiliki akurasi yang lebih baik daripada Naïve Bayes dengan margin yang sangat kecil, dengan hasil masing-masing sebesar 97% dan 98%.

Penelitian yang dilakukan oleh [5] diperoleh hasil penerapan algoritma Naive Bayes dengan RapidMiner tools menunjukkan tingkat akurasi sebesar 76,92%, presisi sebesar 100%, recall sebesar 57,14%, dan nilai AUC 0,881 yang hampir sama dengan 1, menunjukkan bahwa hasil dari algoritma Naive Bayes dapat digunakan dengan baik untuk membuat keputusan tentang tingkat kepuasan pembelajaran online. Metode text mining dengan bantuan RapidMiner digunakan untuk mengklasifikasikan jurnal ilmu komputer berdasarkan pembagian pada Web Science memperoleh hasil penelitian bahwa klasifikasi menggunakan Naïve Bayes memiliki performa yang sangat baik dibandingkan dengan metode SVM. Metode Naïve Bayes mendapatkan nilai terbaik pada ukuran recall, ketepatan, F-measure, dan akurasi. Berdasarkan nilai-nilai tersebut, metode Naive Bayes dapat digunakan untuk otomatisasi pengklasifikasian jurnal lainnya [6].

Berdasarkan hasil penelitian sebelumnya, peneliti melakukan analisis sentimen menggunakan berbagai objek dan variabel, seperti respon pengguna media sosial X terhadap wujud Sun God Nika dari karakter Luffy dalam animasi One Piece menggunakan Naive Bayes dan Support Vector Machine. Peneliti berharap dapat mendukung penelitian ini dan meningkatkan pengetahuan lintas budaya untuk membantu penggemar memahami identitas budaya, nilai-nilai, dan norma yang tercermin pada karakter One Piece.

II. METODE PENELITIAN

Terdapat beberapa langkah yang digunakan dalam penelitian ini, yaitu pengumpulan data, preprocessing, labeling, penerapan algoritma, dan hasil analisis. Lebih lengkapnya dijelaskan di Gambar 1.



Gambar 1 *Flowchart* Prosedur Penelitian

A. Pengumpulan Data

Proses pengumpulan data menggunakan teknik crawling dengan program sederhana berbasis bahasa Python sehingga pengambilan data menjadi lebih efisien. Data teks diambil dari media sosial X, dengan jumlah data teks yang didapatkan sebanyak 3.121 dari rentang waktu 27 Desember 2023 – 20 Januari 2024.

B. Preproessing

Setelah pengumpulan data, selanjutnya adalah data preprocessing, yaitu tahap mempersiapkan teks yang dapat diolah lebih lanjut [7]. Pada tahapan ini, dilakukan pengubahan data yang telah berhasil dikumpulkan sehingga menjadi format yang diperlukan. Berikut adalah tahap yang dilakukan dalam melakukan preprocessing:

1. *Cleaning*
   Pada proses ini, data dibersihkan dari karakter dan emotikon yang tidak penting dalam data kolom full_text.
2. *Case Folding*
   Dalam tahap ini, huruf besar akan diubah menjadi lowercase, yang bertujuan agar teks pada model klasifikasi menjadi seragam untuk mencegah terjadinya kesalahan tokenize.
3. *Tokenize*
   Pada langkah ini, setiap kata pada dataset dipecah menjadi kata-kata berdasarkan spasi.
4. *Stopword Removal*
   Pada langkah ini, kata sambung dalam data dan kata keterangan yang diabaikan dalam data akan dihilangkan.
5. *Stemming*
   Langkah ini, kalimat dipecah menjadi kata dasar dengan menghapus kata imbuhan menggunakan library Sastrawi. Proses ini dilakukan agar kalimat yang disampaikan secara berbeda memiliki makna yang sama.
6. *Normalisasi*
   Pada langkah normalisasi, kalimat yang memiliki arti tertentu diganti dengan kalimat yang sebenarnya, sehingga sistem menjadi lebih efektif untuk memecah kalimat.

C. Pelabelan

Pada titik ini, penilaian komputasi dilakukan menggunakan VaderSentiment dengan dukungan estimasi intensitas. Akibatnya, kerangka kerja ini sebanding dengan penilaian manusia pada data. Pelabelan adalah proses yang digunakan untuk mengevaluasi perasaan positif, negatif, dan netral terhadap pendapat yang terkandung dalam data teks [8].

D. Penerapan Algoritma *Naive Bayes*

Pada tahap ini, library Multinomial Naive Bayes digunakan. Metode ini menghasilkan akurasi tinggi dan hanya membutuhkan sedikit data latihan untuk menentukan parameter yang diperlukan untuk memproses klasifikasi. Data yang telah dilabelkan akan dibagi menjadi dua, yaitu latih dan uji. Untuk memasuki tahap pelatihan model Naive Bayes, data teks dari masing-masing bagian ini akan diubah menjadi vektor fitur.

E. Penerapan Algoritma *Support Vector Machine*

Penerapan Support Vector Machine menggunakan kernel linear. Data yang telah melalui proses pelabelan akan dibagi menjadi dua, yaitu latih dan uji. Selanjutnya, data diproses dengan transformasi menjadi vektor TF-IDF untuk memasuki tahap inisialisasi model Support Vector Machine dan pelatihan model menggunakan data latih. Setelah itu, dilakukan pengujian pada data uji yang sudah di-vektor.

F. Hasil Akurasi dan Evaluasi

Model algoritma yang telah diterapkan akan dilakukan tahap evaluasi untuk menilai kinerja dari setiap model machine learning. Berdasarkan pengujian model yang sudah ditetapkan, maka akan didapatkan hasil akurasi, confusion matrix, dan classification report yang berisikan nilai precision, nilai recall, nilai F1-score, dan support [9].

III. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Dataset diperoleh dengan teknik crawling menggunakan pengembangan program sederhana berbasis Python yang diberi nama Tweet Harvest, sehingga pengambilan data menjadi efisien. Data teks diambil dari media sosial X dalam bentuk CSV (Comma Separated Values). Data teks memiliki 12 atribut seperti pada Gambar 2. Data teks yang diambil berjumlah 3.121 dari rentang waktu 27 Desember 2023 – 20 Januari 2024.

	Column1	id_str	full_text	quote_count	reply_count	retweet_count	favorite_count	lang	user_id_str	conversation_id_str	username	tweet_url
0	Fri Jan 19 16:02:39 +0000 2024	1,74838E+18	@ZS_Roronoa Walawei, semoga gak sampe kebawa m...	0	1	0	0	in	1,25282E+18	1,74837E+18	Tetehtehanget	https://twitter.com/Tetehtehanget/status/17483...
1	Fri Jan 19 09:29:09 +0000 2024	1,74828E+18	Jaket Hoodie Zipper jumper custom anime One Pi...	0	1	0	0	in	608370684	1,74573E+18	RIng_of_fire17	https://twitter.com/RIng_of_fire17/status/1748...
2	Fri Jan 19 02:23:52 +0000 2024	1,74817E+18	CEPAT COPOT PP LUFFY GEAR 5 MU ANJING, ketikan...	0	0	0	0	in	1,5545E+18	1,74817E+18	inturani	https://twitter.com/inturani/status/1748169372...
3	Fri Jan 19 01:26:35 +0000 2024	1,74816E+18	lagi lagi opini tolol wibu pp luffy gear 5, ga...	0	0	0	0	in	1,16594E+18	1,74816E+18	Saa_lla	https://twitter.com/Saa_lla/status/17481554596...
4	Fri Jan 19 01:24:26 +0000 2024	1,74815E+18	hahhaa sakit anuing luffy gear 5 yg satu ini, ...	0	0	0	0	in	1,12612E+18	1,74815E+18	ghinaaniich	https://twitter.com/ghinaaniich/status/1748154...

Gambar 2 Dataset Sebelum *Preprocessing*

B. *Preprcessing*

Pada tahapan ini, data yang digunakan adalah kolom full_text (Gambar 3), yang kemudian dilakukan perubahan data menjadi format yang diperlukan. Berikut langkah-langkah yang dilakukan dalam tahap preprocessing:

	full_text
0	@ZS_Roronoa Walawei, semoga gak sampe kebawa m...
1	Jaket Hoodie Zipper jumper custom anime One Pi...
2	CEPAT COPOT PP LUFFY GEAR 5 MU ANJING, ketikan...
3	lagi lagi opini tolol wibu pp luffy gear 5, ga...
4	hahhaa sakit anuing luffy gear 5 yg satu ini, ...

Gambar 3 Kolom *Fulltext*

1. *Cleaning*

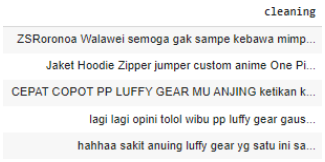
Pada tahapan cleaning, langkah awal yang dilakukan adalah mengimpor library regex, string, dan NLTK (Natural Language Toolkit), yang membantu menyediakan alat untuk memproses bahasa data yang tidak diperlukan dengan menghilangkan URL, emoji, nomor, hashtag, mention, spasi yang berlebihan, dan simbol.

```
import re
import string
import nltk

def remove_URL(tweet):
    url = re.compile(r'https?://\S+|www\.\S+')
    return url.sub(r'', tweet)

def remove_html(tweet):
    html = re.compile(r'<.*?>')
    return html.sub(r'', tweet)

def remove_emoji(tweet):
    emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F"
        u"\U0001F300-\U0001F5FF"
        u"\U0001F680-\U0001F6FF"
        u"\U0001F1E0-\U0001F1FF"
                           "]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', tweet)

def remove_numbers(tweet):
    tweet = re.sub(r'\d+', '', tweet)
    return tweet

def remove_symbols(tweet):
    tweet = re.sub(r'[^a-zA-Z0-9\s]', '', tweet)  # Menghapus semua simbol
    return tweet

df['cleaning'] = df['full_text'].apply(lambda x: remove_URL(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_html(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_emoji(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_symbols(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_numbers(x))

df.head()
```

Gambar 4 Proses *Cleaning*

ISSN: 2715-2766
Vol. 6 No: 2, Juli 2025

Setelah dilakukan proses cleaning, data yang dihasilkan tidak lagi terdapat URL, emoji, nomor, hashtag, mention, spasi yang berlebihan, dan simbol. Data yang telah melalui proses cleaning akan dipisah dengan nama kolom cleaning seperti pada Gambar 5.



Gambar 5 Kolom *Cleaning*

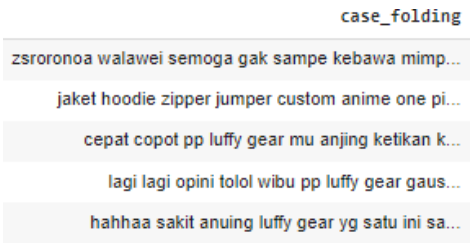
2. Case Folding

Setelah melalui proses cleaning, semua kata pada kolom cleaning diubah menjadi huruf kecil tanpa pengecualian dengan menggunakan fungsi lower, seperti pada Gambar 6.

```
def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower()
        return lowercase_text
    else:
        return text

df['case_folding'] = df['cleaning'].apply(case_folding)
df.head()
```

Gambar 6 Proses *Case Folding*

Setelah dilakukan proses case folding, tidak akan terdapat lagi huruf kapital dalam data yang dihasilkan. Data yang telah melalui proses case folding akan dipisah dengan nama kolom case_folding (Gambar 7).



Gambar 7 Kolom *Case Folding*

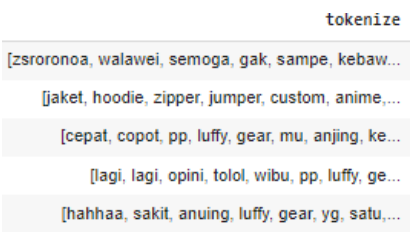
3. *Tokenize*

Hasil dari proses case folding selanjutnya, string input yang terdapat pada kolom case_folding dipotong menjadi kata-kata yang menyusunnya berdasarkan spasi dan white spaces menggunakan fungsi split, seperti pada Gambar 8.

```
def tokenize(text):
    tokens = text.split()
    return tokens

df['tokenize'] = df['case_folding'].apply(tokenize)
df.head()
```

Gambar 8 Proses *Tokenization*

Output yang dihasilkan berupa kata-kata yang sudah terpotong berdasarkan kata-kata yang menyusunnya. Data yang telah melalui proses tokenisasi akan dipisah dengan nama kolom tokenize (Gambar 9).



Gambar 9 Kolom *Tokenize*

ISSN: 2715-2766
Vol. 6 No: 2, Juli 2025

4. *Stopword Removal*

Hasil dari proses tokenisasi selanjutnya, kata-kata preposisi, kata penghubung, objek yang selalu muncul, atau kata-kata yang tidak memiliki makna namun sering muncul akan dihapus. Instalasi library NLTK (Natural Language Toolkit) Bahasa Indonesia merupakan langkah awal yang dilakukan, seperti pada Gambar 10.

```
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('indonesian')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
def remove_stopwords(text):
    return [word for word in text if word not in stop_words]

df['Filtering/stopword removal'] = df['tokenize'].apply(lambda x: remove_stopwords(x))

df.head()
```

Gambar 10 Proses *Stopword Removal*

Data yang telah melalui proses stopword removal menghasilkan data yang sudah tidak mengandung kata-kata preposisi, penghubung, objek yang selalu muncul, ataupun kata-kata yang tidak memiliki makna namun sering muncul. Data yang dihasilkan dari proses stopword removal akan dipisah dengan nama kolom filtering/stopword removal (Gambar 11).

Filtering/stopword removal
[zsroronoa, walawei, semoga, gak, sampe, kebaw...
[jaket, hoodie, zipper, jumper, custom, anime,...
[cepat, copot, pp, luffy, gear, mu, anjing, ke...
[opini, tolol, wibu, pp, luffy, gear, gausah, ...
[hahhaa, sakit, anuing, luffy, gear, yg, sapa,...

Gambar 11 Kolom *Filtering/Stopword*

5. *Steaming Data*

Data yang dihasilkan dari proses stopword removal kemudian akan diproses bahasa alaminya, menghapus kata-kata yang memiliki imbuhan infiks dan sufiks, serta mengubahnya ke bentuk dasar. Instalasi library Sastrawi merupakan langkah awal yang dilakukan pada proses ini, seperti pada Gambar 12.

```
!pip install Sastrawi

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
```

```
Collecting Sastrawi
  Downloading Sastrawi-1.0.1-py2.py3-none-any.whl (209 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 209.7/209.7 kB 3.8 MB/s eta 0:00:00
Installing collected packages: Sastrawi
Successfully installed Sastrawi-1.0.1
```

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

def stem_text(text):
    return [stemmer.stem(word) for word in text]

df['stemming_data'] = df['Filtering/stopword removal'].apply(lambda x: ' '.join(stem_text(x)))

df.head()
```

Gambar 12 Proses *Stemming Data*

Data yang sudah diubah ke bentuk dasar melalui proses stemming dengan bantuan library Sastrawi kemudian akan dipisah dengan nama kolom stemming_data (Gambar 13).

stemming_data
zsroronoa walawei moga gak sampe bawa mimpi ya...
jaket hoodie zipper jumper custom anime one pi...
cepat copot pp luffy gear mu anjing keti kotor...
opini tolol wibu pp luffy gear gausah didenger...
hahhaa sakit anuing luffy gear yg sapa tau mba...

Gambar 13 Kolom Stemming Data

ISSN: 2715-2766
Vol. 6 No: 2, Juli 2025

6. Normalisasi

Pada proses ini, kata-kata yang memiliki makna tertentu dalam data yang telah dihasilkan melalui proses stemming diganti menjadi kata-kata yang sebenarnya menggunakan fungsi text_normalization, seperti pada Gambar 14.

```
def text_normalization(text):
    normalized_text = text
    return normalized_text

df['normalized_text'] = df['stemming_data'].apply(text_normalization)
df.head()
```

Gambar 14 Proses Normalisasi

Untuk melihat hasil dari masing-masing proses, maka dilakukan print dataset seperti pada Gambar 15. Data akan disimpan menggunakan format CSV dengan nama SUNGODNIKA Preprocessing untuk kemudian dilakukan proses pelabelan.

	full_text	cleaning	case_folding	tokenize	Filtering/stopword removal	stemming_data	normalized_text
0	@ZS_Roronoa Walawei, semoga gak sampe kebawa m...	ZSRoronoa Walawei semoga gak sampe kebawa mimp...	zsroronoa walawei semoga gak sampe kebawa mimp...	['zsroronoa', 'walawei', 'semoga', 'gak', 'sam...	['zsroronoa', 'walawei', 'semoga', 'gak', 'sam...	zsroronoa walawei moga gak sampe bawa mimpi ya...	zsroronoa walawei moga gak sampe bawa mimpi ya...
1	Jaket Hoodie Zipper jumper custom anime One Pi...	Jaket Hoodie Zipper jumper custom anime One Pi...	jaket hoodie zipper jumper custom anime one pi...	['jaket', 'hoodie', 'zipper', 'jumper', 'custo...	['jaket', 'hoodie', 'zipper', 'jumper', 'custo...	jaket hoodie zipper jumper custom anime one pi...	jaket hoodie zipper jumper custom anime one pi...
2	CEPAT COPOT PP LUFFY GEAR 5 MU ANJING, ketikan...	CEPAT COPOT PP LUFFY GEAR MU ANJING ketikan k...	cepat copot pp luffy gear mu anjing ketikan k...	['cepat', 'copot', 'pp', 'luffy', 'gear', 'mu'...	['cepat', 'copot', 'pp', 'luffy', 'gear', 'mu'...	cepat copot pp luffy gear mu anjing keti kotor...	cepat copot pp luffy gear mu anjing keti kotor...
3	lagi lagi opini tolol wibu pp luffy gear 5, ga...	lagi lagi opini tolol wibu pp luffy gear gaus...	lagi lagi opini tolol wibu pp luffy gear gaus...	['lagi', 'lagi', 'opini', 'tolol', 'wibu', 'pp...	['opini', 'tolol', 'wibu', 'pp', 'luffy', 'gea...	opini tolol wibu pp luffy gear gausah didenger...	opini tolol wibu pp luffy gear gausah didenger...
4	hahhaa sakit anuing luffy gear 5 yg satu ini, ...	hahhaa sakit anuing luffy gear yg satu ini sa...	hahhaa sakit anuing luffy gear yg satu ini sa...	['hahhaa', 'sakit', 'anuing', 'luffy', 'gear',...	['hahhaa', 'sakit', 'anuing', 'luffy', 'gear',...	hahhaa sakit anuing luffy gear yg sapa tau mba...	hahhaa sakit anuing luffy gear yg sapa tau mba...

Gambar 15 Data Sesudah *Preprocessing*

C. Pelabelan

Proses pemberian label dilakukan untuk menilai sentimen positif, negatif, dan netral pada data. Pada proses ini, data yang digunakan diambil dari kolom stemming_data, seperti pada Gambar 16.

	stemming_data
0	zsroronoa walawei moga gak sampe bawa mimpi ya...
1	jaket hoodie zipper jumper custom anime one pi...
2	cepat copot pp luffy gear mu anjing keti kotor...
3	opini tolol wibu pp luffy gear gausah didenger...
4	hahhaa sakit anuing luffy gear yg sapa tau mba...

Gambar 16 Kolom *Stemming* Data

Langkah awal yaitu dengan menginstal VaderSentiment, kemudian sistem akan menganalisis sentimen yang terkandung dalam data dan memberikannya skor. Skor yang dihasilkan dicetak kemudian dimasukkan ke dalam kolom bernama compound_score dan sistem akan memberikan label negatif, positif, dan netral, seperti pada Gambar 17.

```
df.loc[df['Compound_Score'] < 0, 'Sentiments'] = 'Negatif'
df.loc[df['Compound_Score'] == 0, 'Sentiments'] = 'Netral'
df.loc[df['Compound_Score'] > 0, 'Sentiments'] = 'Positif'

df.head(5)
```

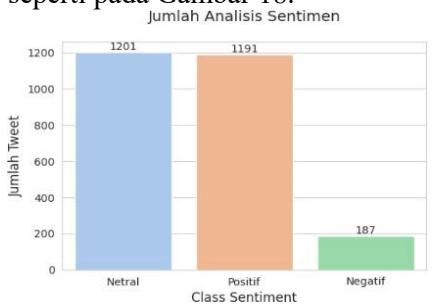
a

	stemming_data	Compound_Score
0	zsroronoa walawei moga gak sampe bawa mimpi ya...	0.0000
1	jaket hoodie zipper jumper custom anime one pi...	0.5859
2	cepat copot pp luffy gear mu anjing keti kotor...	0.0000
3	opini tolol wibu pp luffy gear gausah didenger...	0.0000
4	hahhaa sakit anuing luffy gear yg sapa tau mba...	0.4404

b

Gambar 17 Hasil Scoring Stemming Data

Setelah menilai sentimen berdasarkan skor, kemudian data akan divisualisasikan untuk melihat jumlah data sentimen berdasarkan kelas nilai sentimen. Langkah awal yang dilakukan pada proses ini adalah mengimpor pandas, matplotlib, dan seaborn. Kemudian, output yang dihasilkan berupa visualisasi diagram batang dengan jumlah sentimen masing-masing kelas berdasarkan jumlah tweet, seperti pada Gambar 18.

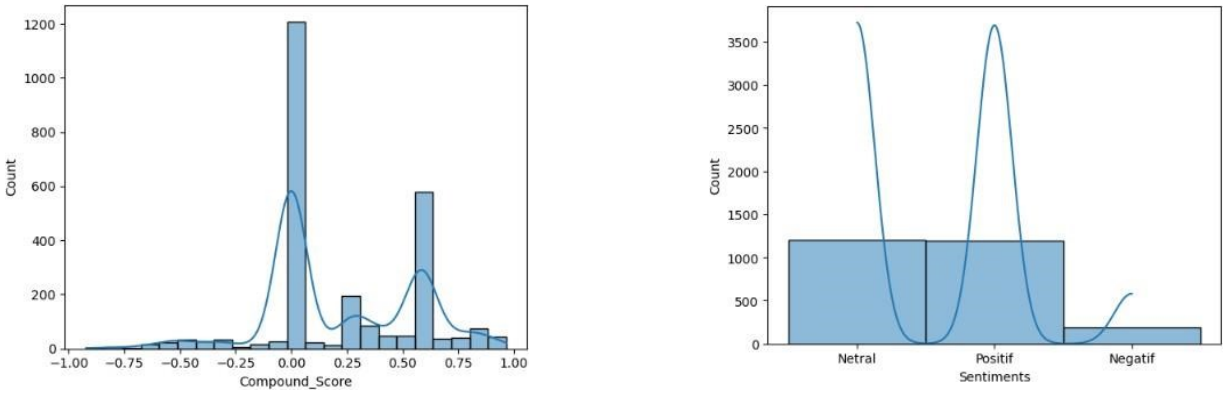


Gambar 18 Hasil Visualisasi Data Sentimen

ISSN: 2715-2766
Vol. 6 No: 2, Juli 2025

D. Penerapan Algoritma *Naive Bayes*

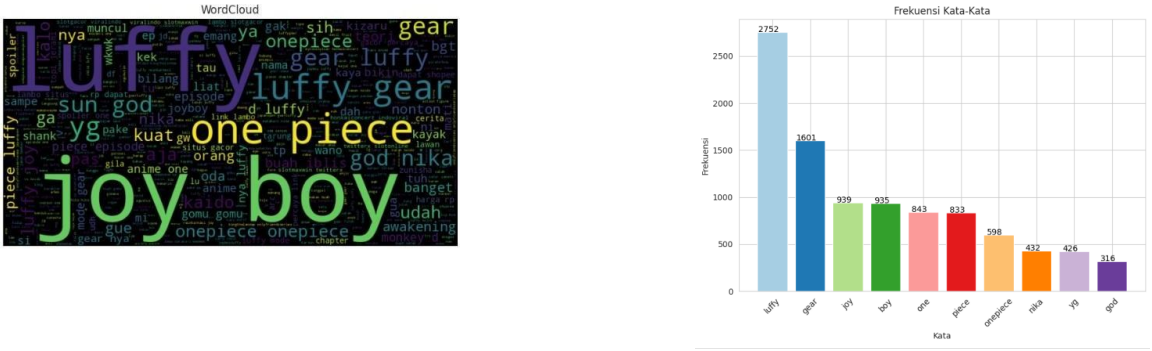
Pada proses perhitungan Naive Bayes dilakukan dengan menggunakan Google Colab dan bahasa Python. Langkah awal yang dilakukan yaitu dengan mengecek data deskriptif dari dataset, mengimpor matplotlib dan seaborn untuk memvisualisasikan distribusi kolom numerik. Setelah melakukan visualisasi distribusi kolom numerik, langkah selanjutnya adalah memvisualisasikan distribusi sentimen menggunakan library yang sama yang digunakan pada visualisasi distribusi kolom numerik, seperti pada Gambar 19.



a **b**

Gambar 19 Hasil Visualisasi Distribusi

Kemudian, langkah selanjutnya adalah menghitung jumlah sentimen positif, negatif, dan netral dari keseluruhan data berdasarkan jumlah tweet. Setelah itu, dibuat wordcloud dan frekuensi kata-kata yang muncul pada teks menggunakan library pandas, numpy, dan matplotlib. Data teks diambil dari kolom stemming_data. Output yang dihasilkan yaitu visualisasi wordcloud dan frekuensi kata-kata, seperti pada Gambar 20.



a **b**

Gambar 20 Hasil Wordcloud

Proses penerapan algoritma Naive Bayes dilakukan dengan menginstal library scikit-learn, kemudian mengimpor library pandas, train_test_split, CountVectorizer, Multinomial Naive Bayes, accuracy_score, classification_report, dan confusion_matrix. Data yang digunakan dibagi dua, yaitu dataset uji dan latih, sebelum data diubah menjadi vektor fitur. Jumlah data latih sebesar 80% dengan jumlah 2.062 data, dan 20% data uji yang berjumlah 516 data. Data yang telah dibagi kemudian diubah menjadi vektor fitur dengan menggunakan fit_transform. Selanjutnya, data yang telah diubah menjadi vektor fitur akan dicetak untuk melihat hasil vektor fitur data latih dan vektor fitur data uji, seperti pada Gambar 21 (a) dan (b).

```
print('Hasil Ekstraksi Fitur:')
print('----------------------')
print('Vektor Fitur Data Latih:')
print(X_train_vectorized.toarray())
print('\nVektor Fitur Data Uji:')
print(X_test_vectorized.toarray())
```

a

```
Hasil Ekstraksi Fitur:
----------------------
Vektor Fitur Data Latih:
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

Vektor Fitur Data Uji:
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

b

Gambar 21 Hasil Ekstrasi Vector Fitur

Tahap selanjutnya adalah mengevaluasi untuk mendapatkan hasil akurasi dan nilai confusion matrix dengan memasukkan y_test dan predictions. Hasil evaluasi data uji berupa classification report dengan nilai akurasi sebesar 0,84 atau 84% dan confusion matrix, seperti pada Gambar 22.

```
Accuracy: 0.84

Classification Report:
              precision    recall  f1-score   support

     Negatif       1.00      0.28      0.43        40
      Netral       0.82      0.90      0.86       221
     Positif       0.85      0.88      0.86       255

    accuracy                           0.84       516
   macro avg       0.89      0.68      0.72       516
weighted avg       0.85      0.84      0.83       516

Confusion Matrix:
[[ 11  12  17]
 [  0 198  23]
 [  0  31 224]]
```

Gambar 22 Hasil Evaluasi Data Uji Menggunakan Algoritma *Naive Bayes*

E. Penerapan Algoritma *Support Vector Machine*

Proses penerapan dilakukan dengan mengimpor library train_test_split, TfidfVectorizer, CountVectorizer, Support Vector Classification, accuracy_score, classification_report, dan confusion_matrix. Kemudian, data dibagi dua menjadi 70% data latih dengan jumlah 1.804 data dan 30% data uji dengan jumlah 774 data. Data yang telah dibagi kemudian diubah menjadi vektor TF-IDF menggunakan TfidfVectorizer.

Selanjutnya, tahap inisialisasi model Support Vector Classification menggunakan kernel linear, kemudian model dilatih menggunakan data latih yang sudah divektorisasi. Setelah melatih model dengan data latih, model akan diuji menggunakan data uji yang sudah divektorisasi dan mencetak contoh hasil prediksi pada data pengujian. Akurasi performa model pada data uji kemudian dihitung dengan memasukkan classification report, y_test, dan y_pred dicetak untuk melihat nilai akurasi sebesar 0,87 atau 87%, seperti pada Gambar 23.

```
              precision    recall  f1-score   support

     Negatif       0.85      0.27      0.41        62
      Netral       0.83      0.99      0.90       344
     Positif       0.93      0.87      0.90       368

    accuracy                           0.87       774
   macro avg       0.87      0.71      0.74       774
weighted avg       0.88      0.87      0.86       774

Confusion Matrix:
[[ 17  23  22]
 [  1 341   2]
 [  2  47 319]]
```

Gambar 23 Hasil Evaluasi Data Uji Menggunakan Algoritma *Support Vector Machine*

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat disimpulkan bahwa jumlah analisis sentimen positif pengguna X terhadap karakter Luffy sebesar 1.191 data atau sebesar 46%, negatif 187 data atau sebesar 7%, dan netral 1.200 data atau sebesar 47%. Sentimen netral lebih besar dari sentimen positif dengan selisih 1%. Serta, untuk penerapan algoritma Naive Bayes pada data uji didapatkan nilai akurasi sebesar 0,84 atau sebesar 84%. Sedangkan untuk penerapan algoritma Support Vector Machine, hasil performa model menunjukkan nilai akurasi sebesar 0,87 atau 87%.

DAFTAR PUSTAKA

[1] M. A. Kausar, A. Soosaimanickam, and M. Nasar, "Public Sentiment Analysis on Twitter Data during COVID-19 Outbreak," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 2, pp. 415–422, 2021, doi: 10.14569/IJACSA.2021.0120252.

[2] M. Jonathan and Y. Nataliani, "Analisis Sentimen Penilaian Masyarakat Indonesia terhadap GeNose pada Komentar Youtube Menggunakan Metode NaÃ¯ve Bayes," *J. Mat. dan Apl.*, vol. 11, no. 01, 2022, [Online]. Available: https://ejournal.unsrat.ac.id/index.php/decartesian/article/view/38339.

[3] S. Juanita, "Analisis Sentimen Persepsi Masyarakat Terhadap Pemilu 2019 Pada Media Sosial Twitter Menggunakan Naive Bayes," *J. Media Inform. Budidarma*, vol. 4, no. 3, p. 552, 2020, doi: 10.30865/mib.v4i3.2140.

[4] N. Pavitha *et al.*, "Movie recommendation and sentiment analysis using machine learning," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 279–284, 2022, doi: 10.1016/j.gltp.2022.03.012.

[5] Aminatuzzuhriyyah and N. Nafisah, "Klasifikasi Tingkat Kepuasan Mahasiswa Terhadap Pembelajaran Secara Daring Menggunakan Algoritma Naïve Bayes," *Techno Xplore J. Ilmu Komput. dan Teknol. Inf.*, vol. 6, no. 2, pp. 61–67, 2021, doi: 10.36805/technoxplore.v6i2.1377.

[6] S. Widaningsih, U. Suryakancana, A. Suheri, and U. Suryakancana, "Klasifikasi Jurnal Ilmu Komputer Berdasarkan Pembagian Web of," vol. 2018, no. March, pp. 23–24, 2018.N. L. W. S. R. Ginantra, C. P. Yanti, G. D. Prasetya, I. B. G. Sarasvananda, and I. K. A. G. Wiguna, "Analisis Sentimen Ulasan Villa di Ubud Menggunakan Metode Naive Bayes, Decision Tree, dan K-NN," *J. Nas. Pendidik. Tek. Inform.*, vol. 11, no. 3, pp. 205–215, 2022, doi: 10.23887/janapati.v11i3.49450.

[7] M. Ghiassi and S. Lee, "A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach," *Expert Syst. Appl.*, vol. 106, pp. 197–216, 2018, doi: 10.1016/j.eswa.2018.04.006.

[8] A. Kulkarni, D. Chong, and F. A. Batarseh, *Foundations of data imbalance and solutions for a data democracy*. Elsevier Inc., 2020.