

Analisis Sentimen Mengenai Gangguan Bipolar Pada Twitter Menggunakan Algoritma Naïve Bayes

Oriza Sativa Dinauni Silaen¹, Herlawati^{1,*}, Rasim¹

¹ Program Studi Informatika Fakultas Ilmu Komputer, Universitas Bhayangkara Jakarta Raya

*email: herlawati@ubharajaya.ac.id

DOI: <https://doi.org/10.31603/komtika.v6i2.8198>

Received: 21-11-2022, Revised: 28-11- 2022, Accepted: 29-11-2022

ABSTRACT

Bipolar disorder is one of the most common mental health disorders in the world. To find out public sentiment regarding bipolar disorder, sentiment analysis is carried out through social media to analyse positive or negative sentiments with the aim of maintaining positive sentiment towards the problem of bipolar disorder. Twitter is a social media that is often used to exchange information, discuss, and even express emotions. Emotions of Twitter users can be called sentiment. Sentiment analysis is also carried out to see opinions or tendencies towards an opinion. Opinion tendencies can be in the form of positive or negative sentiments. The data used in this study uses the bipolar keyword. There are 2177 tweets data that were successfully obtained in the crawling process using API key access from Twitter developers, after which the data will be processed using pre-processing. The comparison of the presentations obtained is 70.92% expressing a negative opinion and 29.08% expressing a positive opinion. The results of the analysis in this study using the naive Bayes algorithm is with an accuracy value of 92.110092%

Keywords: Sentiment Analysis, Twitter, Naïve Bayes, Bipolar.

ABSTRAK

Gangguan bipolar merupakan salah satu gangguan kesehatan mental yang banyak terjadi di dunia. Untuk mengetahui sentimen masyarakat mengenai gangguan bipolar maka dilakukan analisis sentimen melalui sosial media untuk menganalisis sentimen positif ataupun negatif dengan tujuan untuk mempertahankan sentiment positif terhadap masalah gangguan bipolar. Twitter merupakan sosial media yang sering kali digunakan untuk bertukar informasi, berdiskusi, bahkan mengungkapkan emosi. Emosi para pengguna Twitter dapat disebut dengan sentimen. Analisis sentimen juga dilakukan untuk melihat pendapat atau kecenderungan terhadap suatu opini. Kecenderungan opini dapat berupa sentimen positif maupun negatif. Data yang digunakan dalam penelitian ini menggunakan keyword bipolar. Terdapat 2177 data tweets yang berhasil diperoleh pada proses crawling menggunakan API key access yang berasal dari Twitter developer, setelah itu data akan di proses menggunakan preprocessing. Perbandingan presentasi yang didapatkan adalah sebesar 70,92 % mengungkapkan opini negatif dan 29,08 % mengungkapkan opini positif. Hasil analisis pada penelitian ini menggunakan algoritma Naïve Bayes adalah dengan nilai akurasi sebesar 92,110092%.

Kata-kata kunci: Bipolar, Analisis Sentimen, Twitter, Algoritma Naïve Bayes.

PENDAHULUAN

Di dunia kesehatan mental masih menjadi satu permasalahan yang signifikan. Terdapat lebih dari 1 dari 7 remaja berusia 10-19 tahun di dunia yang hidup dengan diagnosis gangguan mental dan setiap tahunnya terdapat 46.000 tindakan bunuh diri. Salah satu gangguan mental yang banyak diagnosis pada masyarakat saat ini adalah gangguan bipolar [1].

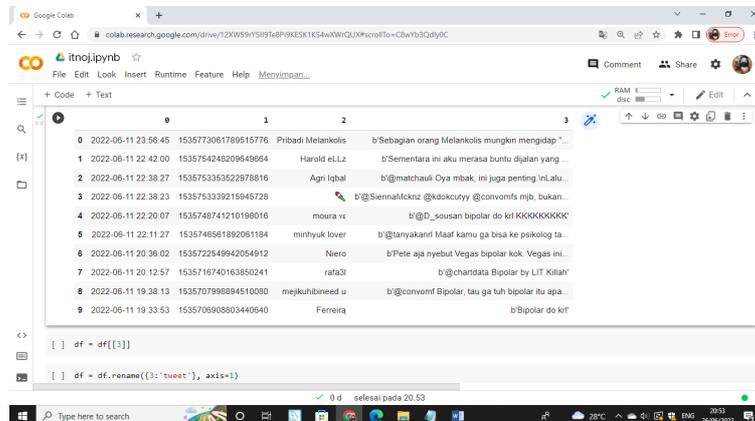
Gangguan bipolar merupakan salah satu bentuk gangguan kesehatan mental yang ditandai dengan sering kalinya mengalami perubahan suasana hati yang ekstrim dan fluktuatif dalam

kurun waktu tertentu, seperti perubahan suasana hati dari yang sedih tiba-tiba menjadi sangat bahagia ataupun sebaliknya dari yang sangat bahagia menjadi sedih. Gangguan bipolar muncul disebabkan dari banyak faktor yang muncul secara bersama-sama sehingga memicu timbulnya penyakit bipolar [1].

Media sosial merupakan sarana yang digunakan untuk melakukan aktivitas secara virtual seperti berbagi informasi dan berdiskusi. Bagi penggunanya, media sosial juga digunakan sebagai tempat untuk menunjukkan eksistensi diri yang dapat berpengaruh pada cara pandang, gaya hidup, serta budaya suatu bangsa. *Twitter* merupakan salah satu aplikasi media sosial yang paling banyak digunakan masyarakat dunia termasuk masyarakat Indonesia pada saat ini dan penggunaannya dapat dilakukan dengan menggunakan komputer atau perangkat mobile tanpa batas ruang dan waktu. Pengguna *Twitter* dapat mengekspresikan opininya dengan tulisan dalam maksimal 240 karakter biasa yang disebut dengan *tweet*. Isi *tweet* ini menjadi sumber data apabila diolah dengan benar dan tepat sehingga dapat menghasilkan berbagai informasi yang berguna, misalnya dapat digunakan sebagai pengambilan keputusan atau digunakan untuk meningkatkan persepsi seseorang terhadap suatu hal yang baru diketahui [2]. Jadi, *Twitter* digunakan masyarakat untuk tempat menuangkan sesuatu hal baik berupa hal positif, negatif, maupun netral yang dapat diaplikasikan terhadap penelitian analisis sentiment mengenai gangguan bipolar. Dalam penelitian ini dilakukan untuk menganalisis sentimen positif dan negatif pada *tweet* persepsi masyarakat mengenai gangguan bipolar menggunakan algoritma *Naïve Bayes* karena memiliki tingkat performa akurasi yang lebih baik. *Tweet* yang akan diklasifikasikan berdasarkan sifat positif, negatif, maupun netral dan data akan diambil menggunakan *API (Application Programming Interface)* dengan kata kunci bipolar. Setelah *tweet* berhasil diambil maka selanjutnya akan dilakukan proses data mining (*text mining*) dan data akan diuji keakurasiannya menggunakan *Confusion Matrix*.

Penelitian-penelitian terdahulu yang membahas tentang analisis sentimen dijelaskan sebagai berikut: **Pertama**, penelitian yang membahas mengenai analisis sentimen dompet elektronik meliputi GoPay, OVO, dan DANA pada media sosial twitter menggunakan metode *Naïve Bayes Classifier* yang bertujuan memberikan rekomendasi kepada masyarakat aplikasi dompet mana yang tepat untuk digunakan dalam kegiatan jual beli dikehidupan sehari-hari [3],[4]. **Kedua**, Penelitian ini akan membahas mengenai bentuk tanggapan masyarakat terkait dengan obesitas yang diambil dari setiap *tweet* dengan bertujuan untuk memperluas pemahaman mengenai literasi kesehatan melalui media sosial terkait obesitas melewati media sosial dalam meningkatkan pemahaman dan kemampuan mempercayai informasi kesehatan guna mempertahankan sentiment positif yang ada [5]. **Ketiga**, Penelitian ini membahas mengenai pemanfaatan media sosial banyak digunakan untuk mengetahui banyaknya komentar contohnya dengan penggunaan *Twitter*. Dalam penelitian ini digunakan algoritma *Naïve Bayes* untuk melakukan analisis media sosial terhadap reaksi masyarakat pada RUU Cipta Kerja [6][7][2][8]. **Keempat**, Penelitian ini membahas mengenai penggunaan metode *Naïve Bayes* untuk menambahkan fitur konversi ikon emoji karena diduga akan meningkatkan keakurasiannya jika dilakukan pengklasifikasian tanpa menambah penambahan fitur [9] [10]. **Kelima**, Penelitian ini membahas mengenai penggunaan metode *K-Nearest Neighbor* dengan bertujuan mendeteksi tingkat depresi dan mengetahui tingkat keakurasiannya. [11][12].

Setelah mengetahui kode akses selanjutnya akan dilanjutkan pada tahap mengkoneksikan pemrograman menggunakan *syntax* dengan memasukkan kode akses dan kata kunci “bipolar”. Pada Gambar 3. merupakan data *tweets* yang merupakan hasil dari *crawling* yang terdiri dari waktu, id, *username*, dan *tweets*.



Gambar 3. Hasil Crawling
Sumber: Hasil Penelitian (2022)

b. Preprocessing

Data *tweets* yang sudah didapat dari *Twitter* telah disimpan dalam bentuk *csv* yang belum berstruktur. Oleh karena itu, untuk mendapatkan data yang terstruktur diperlukan proses perubahan data. Pada tahap preprocessing ini akan dilakukan pembersihan data. Berikut merupakan tahapan proses *text preprocessing*:

1. Case Folding dan Remove Number

Pada algoritma 1 menunjukkan melakukan proses untuk merubah huruf besar menjadi huruf kecil. Pada algoritma 2 menunjukkan *script* yang akan digunakan untuk melakukan proses *cleansing* dengan melakukan proses pembersihan teks dari *tab*, *new line*, *back slice*, *mention*, *link*, *hashtag*, *URL*. Pada algoritma 3 menunjukkan proses pembersihan teks dari angka.

Script yang digunakan untuk melakukan proses *case folding* sebagai berikut:

Algoritma 1. Script Case Folding

1. `df['tweet'] = df['tweet'].str.lower()`
2. `df.head()`

Algoritma 2. Cleansing

1. `#cleansing`
2. `import string`
3. `import re`
4. `def remove_tweet_spesial(text):`
5. `text = text.replace(" ", "").replace("\n", "").replace("\t", "").replace("!", "")`
6. `text = text.encode('ascii', 'replace').decode('ascii')`
7. `text = ''.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/S+)", "", text).split())`
8. `return text.replace("http://", "").replace("https://", "")`
9. `df['tweet'] = df['tweet'].apply(remove_tweet_spesial)`
10. `df['tweet'] = df['tweet'].str.slice(2)`
11. `df.head()`

Algoritma 3. Remove Number

1. `#remove number`
2. `def remove_number(text):`
3. `return re.sub(r" d+", "", text)`
4. `df['tweet'] = df['tweet'].apply(remove_number)`
5. `df.head()`

2. Remove Punctuation, Remove Single Char, dan Tokenizing

Pada algoritma 4 menunjukkan proses pembersihan teks dari tanda baca. Pada algoritma 5 menunjukkan proses pembersihan teks dari huruf tunggal atau tidak berbentuk kata. Pada algoritma 6 menunjukkan proses pemisahan teks menjadi bentuk potongan kata-kata. *Script* yang digunakan untuk melakukan proses *remove punctuation* dan proses *Remove Single Char* serta proses *tokenizing* sebagai berikut:

Algoritma 4. Remove Punctuation

```
1. def remove_punctuation(text):  
2. return text.translate(str.maketrans("", "", string.punctuation))  
3. df['tweet'] = df['tweet'].apply(remove_punctuation)  
4. df.head()
```

Algoritma 5. Remove Single Char

```
1. #remove single char  
2. def remove_single_char(text):  
3. return re.sub(r"[a-zA-Z]b", "", text)  
4. df['tweet'] = df['tweet'].apply(remove_single_char)  
5. df.head()
```

Algoritma 6. Tokenizing

```
1. #tokenizing  
2. from nltk.tokenize import word_tokenize  
3. def word_tokenize_wrapper(text):  
4. return word_tokenize(text)  
5. df['tweet'] = df['tweet'].apply(word_tokenize_wrapper)  
6. df.head()
```

3. Filtering

Pada tahap ini melakukan proses penyaringan teks dengan bantuan kamus *stopword* yang tersedia dalam *library NLTK* dan tambahan kata yang dibuat secara manual. *Script* yang digunakan untuk melakukan proses *filtering* sebagai berikut:

Algoritma 7. Filtering

```
1. from nltk.corpus import stopwords  
2. list_stopwords = stopwords.words('indonesian')  
3. list_stopwords.extend(['pas', 'ya', 'mjb', 'sih', 'uya', 'is', 'oya', 'Oya', 'xb', 'to', 'xre', 'x', 'kacem', 'keum', 'kea', 'wts', 'deh', 'loh', 'kl', 'do', 'oiya', 'oya', 'nih', 'ok', 'ah', 'xad', 'xd', 'xa', 'xe', 'ri', 'xv', 'yg', 'xm', 'kalo', 'pt', 'gue', 'bilang', 'xf', 'kkkkkkkkk'])  
4. txt_stopword = pd.read_csv('stopwords.csv', names = ["stopwords"], header = None)  
5. list_stopwords.extend(txt_stopword["stopwords"][0].split(" "))  
6. list_stopwords = set(list_stopwords)  
7. def stopwords_removal(words):  
8. return [word for word in words if word not in list_stopwords]  
9. df['tweet'] = df['tweet'].apply(stopwords_removal)  
10. df.head()
```

c. Ekstraksi Data

Data *tweet* yang telah melalailah proses *preprocessing* yang berbentuk kata akan di konversi menjadi bentuk numerik. Proses ekstraksi kata dilakukan menggunakan *TF-IDF*. *Script* yang digunakan untuk melakukan proses pembobotan dengan menggunakan *TF-IDF* sebagai berikut:

Algoritma 8. TF-IDF

```
1. from sklearn.feature_extraction.text import CountVectorizer  
2. from sklearn.feature_extraction.text import TfidfTransformer  
3. cvec = CountVectorizer(stop_words='english', min_df=1, max_df=.5, ngram_range=(1,2))  
4. cvec  
5. cvec = CountVectorizer(stop_words='english', min_df=.0025, max_df=.5, ngram_range=(1,2))  
6. cvec.fit(review_text_list)  
7. len(cvec.vocabulary_)  
8. transformer = TfidfTransformer()  
9. transformed_weights = transformer.fit_transform(cvec_counts)  
10. transformed_weights  
11. weights = np.asarray(transformed_weights.mean(axis=0)).ravel().tolist()  
12. weights_df = pd.DataFrame({'term': cvec.get_feature_names(), 'weight': weights})  
13. weights_df.sort_values(by='weight', ascending=False).head(20)
```

Dalam tahap pembobotan menggunakan *TF-IDF*, dilakukan contoh perhitungan menggunakan 4 contoh data. pembobotan menggunakan *TF-IDF* secara manual terdapat pada Tabel 1.

Tabel 1. Contoh Pembobotan Manual

Data A	melankolis mengidap mood swing gangguan bipolar													
Data B	dasar bipolar najis berobat													
Data C	bipolar mood swing drastis sedih sedih banget senang senang banget													
Data D	bangga banget anxiety bipolar dari info google sinting													
Term	TF				df	N/df	(-)				Bobot (w) = tf x idf			
	A	B	C	D			A	B	C	D	A	B	C	D
melankolis	1	0	0	0	1	4	0,602059991	0,60206	0	0	0	0	0	
mengidap	1	0	0	0	1	4	0,602059991	0,60206	0	0	0	0	0	
mood	1	0	1	0	2	2	0,301029996	0,30103	0	0,30103	0	0	0	
swing	1	0	1	0	2	2	0,301029996	0,30103	0	0,30103	0	0	0	
gangguan	1	0	0	0	1	4	0,602059991	0,60206	0	0	0	0	0	
bipolar	1	1	1	1	4	1	0	0	0	0	0	0	0	
dasar	0	1	0	0	1	4	0,602059991	0	0,60206	0	0	0	0	
najis	0	1	0	0	1	4	0,602059991	0	0,60206	0	0	0	0	
berobat	0	1	0	0	1	4	0,602059991	0	0,60206	0	0	0	0	
drastis	0	0	1	0	1	4	0,602059991	0	0	0,60206	0	0	0	
sedih	0	0	2	0	2	2	0,301029996	0	0	0,60206	0	0	0	
banget	0	0	2	0	2	2	0,301029996	0	0	0,60206	0	0	0	
senang	0	0	2	1	3	1,4	0,146128036	0	0	0,292256	0,146128	0	0	
bangga	0	0	0	1	1	4	0,602059991	0	0	0	0	0,60206	0	
anxiety	0	0	0	1	1	4	0,602059991	0	0	0	0	0,60206	0	
dari	0	0	0	1	1	4	0,602059991	0	0	0	0	0,60206	0	
info	0	0	0	1	1	4	0,602059991	0	0	0	0	0,60206	0	
google	0	0	0	1	1	4	0,602059991	0	0	0	0	0,60206	0	
sinting	0	0	0	1	1	4	0,602059991	0	0	0	0	0,60206	0	

Sumber: Hasil Penelitian (2022)

d. **Pelabelan Data**

Pelabelan data hasil *crawling* dan telah melalui tahapan *preprocessing* telah dikumpulkan menggunakan *library python* dengan menggunakan *script* pada Gambar 4. Proses pelabelan kelas sentimen pada data *tweet* dilakukan secara otomatis dengan menggunakan *coding* pelabelan berdasarkan kamus *lexicon*. Konsep pelabelan pada data *tweet* ini terbagi menjadi dua kelas sentimen yaitu kelas sentimen positif dan kelas sentimen negatif. Nilai sentimen 0 akan dinyatakan sebagai kelas negatif, sedangkan nilai sentimen 1 dinyatakan sebagai kelas positif.

```

Algoritma 9. Label
1. #label = []
2. for index, row in res_df.iterrows():
3. if row["sentimen"] >= 0:
4. label.append(1)
5. else:
6. label.append(0)
7. res_df["label"] = label
8. res_df = res_df.drop(columns=['sentimen'])
9. res_df.head()
    
```

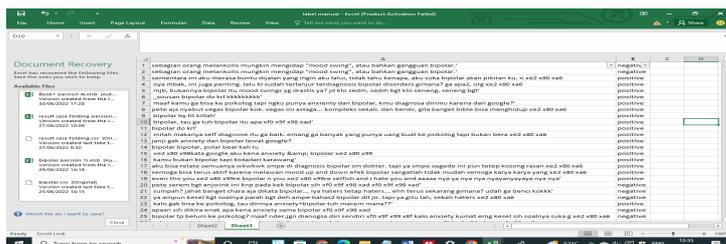
Negatif	Positif	Jumlah
1538	639	2177
70,65%	29,35%	100%

Gambar 4. Contoh Hasil Pelabelan dan Hasil Klasifikasi

Sumber: Hasil Penelitian (2022)

Berdasarkan hasil percobaan didapatkan bahwa hasil akhir dari pelabelan dengan menggunakan *library python* sejumlah 2177 data *tweet* adalah 1538 *tweet* digolongkan kelas negatif dan 639 *tweet* digolongkan kedalam kelas positif.

Sebagai perbandingan terhadap hasil data yang dilabeli dengan menggunakan pemrograman *python*, dalam penelitian ini juga dilakukan *labelling* secara manual. Proses *labelling* manual yang dilakukan dapat dilihat pada gambar 5.



Negatif	Positif	Jumlah
1725	452	2177
79,23%	20,77%	100%

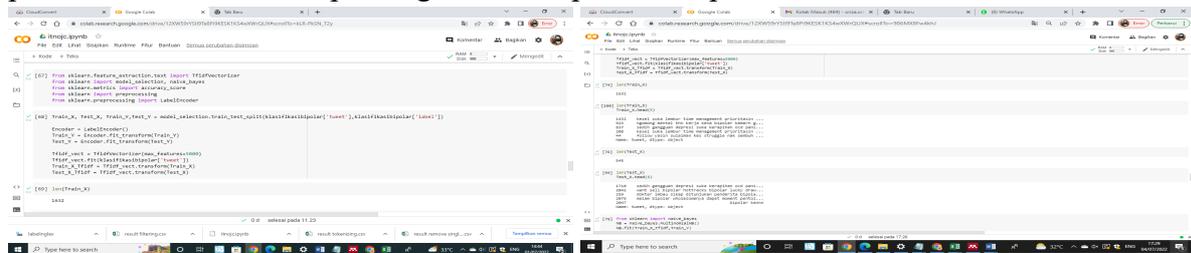
Gambar 5. Contoh Hasil Labelling Manual dan Hasil Klasifikasi Manual

Sumber: Hasil Penelitian (2022)

Didapatkan hasil akhir dari pelabelan dengan cara manual sebanyak 2177 data *tweet* adalah 1725 *tweet* digolongkan ke kelas negatif dan 452 *tweet* digolongkan ke kelas positif.

e. **Penerapan Klasifikasi Naïve Bayes**

Setelah melakukan *preprocessing data, labelling*, dan ekstraksi data, selanjutnya dilakukan pembuatan model yang akan digunakan untuk melakukan klasifikasi. Proses ini dilakukan dengan menggunakan pemrograman Python yang bernama *scikit-learn* yang berguna untuk proses klasifikasi. Proses pembagian data dapat dilihat pada Gambar 6.



Gambar 6. Pembagian Data Latih dan Data Uji dan Tampilannya
Sumber: Hasil Penelitian (2022)

Pada tahap klasifikasi dilakukan dengan perhitungan probabilitas antar kalimat setiap kata agar dapat menghasilkan dengan jelas prediksi data yang dimasukkan. Data uji yang digunakan sejumlah 545 data digunakan untuk menguji model klasifikasi yang dibuat dengan menggunakan algoritma *Naïve Bayes*. Adapun penerapan proses klasifikasi dan *script* disajikan pada Gambar 7.



No.	Komentar	Kategori
1.	Senang sedih penderita bipolar mengalami ketakutan	Negatif
2.	Bipolar akibat faktor genetic tanggapan ahli psikolog	Positif
3.	aktif bipolar	Positif

Gambar 7. Script Klasifikasi Naive Bayes dan Contoh Data Latih
Sumber: Hasil Penelitian (2022)

Berdasarkan hasil pembagian data latih dan data uji, dilanjutkan membuat perhitungan manual menggunakan algoritma *Naïve Bayes* menggunakan 3 contoh data latih yang telah melalui tahap *preprocessing*.

Dibuat suatu model probabilitas dengan menggunakan persamaan pada kajian teori [13].

$$P(v_{positif}) = \frac{2}{3} \quad P(v_{negatif}) = \frac{1}{3} \quad (1)$$

f. **Pengujian Model Klasifikasi Naïve Bayes**

Untuk mengetahui performa dari algoritma *Naïve Bayes*, maka dilakukan pengujian terhadap model yang telah dibuat. Hasil klasifikasi dinyatakan dalam bentuk *confusion matrix*. Pada dasarnya *confusion matrix* memuat informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya. Pengujian model klasifikasi dengan menggunakan *library python sklearn.metric* yang memuat *confusion_matrix* dan

visualisasi dengan sumber terbuka dibuat menggunakan pustaka *matplotlib* seperti disajikan pada gambar 8.



Gambar 8. Script Confusion Matrix dan Visualisasi Confusion Matrix
Sumber: Hasil Penelitian (2022)

Gambar 8 merupakan *script* menjalankan perintah tampilan untuk visualisasi *confusion matrix*, dan menunjukkan hasil visualisasi *Confusion Matrix*. Setelah diketahui *Confusion Matrix* dari model, diperoleh nilai akurasi dari model klasifikasi *Naïve Bayes* yang telah dibuat menggunakan *sklearn.metrics* dengan melakukan *import accuracy score* yang disediakan oleh *library scikit.learn*. Akurasi yang diperoleh dari perhitungan *matrix* yaitu sebesar 92,11009174311927%.

g. **Evaluasi Model**

Evaluasi model dilakukan setelah pengujian model. *Confusion Matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh model klasifikasi dengan hasil klasifikasi sesungguhnya. Hasil evaluasi model dengan *confusion matrix* disajikan pada Tabel 2.

Tabel 2. Hasil Confusion Matrix

		Predict Class	
		Positif	Negatif
Actual Class	Positif	149	12
	Negatif	31	353

Sumber: Hasil Penelitian (2022)

Tabel 2 memperlihatkan *Confusion Matrix* berupa 353 data sebagai negatif dan 149 data sebagai positif. Berdasarkan hasil pengujian model klasifikasi *Naïve Bayes* sebelumnya, nilai akurasi pada keseluruhan dapat dihitung sebesar 92,11009174311927%. *Script* menghitung nilai akurasi sebagai berikut:

```

Algoritma 10. Hitung nilai akurasi
1. from sklearn import naive_bayes
2. NB = naive_bayes.MultinomialNB()
3. NB.fit(Train_X_Tfidf, Train_Y)
4. predictions_NB = NB.predict(Test_X_Tfidf)
5. print("NB Accuracy Score -> ", accuracy_score(predictions_NB, Test_Y)*100)
    
```

Perhitungan akurasi manual dari perhitungan matriks adalah sebagai berikut.

$$\text{Akurasi} = \frac{\text{True Negatif} + \text{True Positif}}{\text{Total Data yang di Uji}} \times 100\% = \frac{502}{545} \times 100\% = 92,1100092\%$$

$$\text{Akurasi} = \frac{\text{True Negatif} + \text{True Positif}}{\text{Total Data yang di Uji}} \times 100\% = \frac{502}{545} \times 100\% = 92,1100092\%$$

Akurasi menyatakan seberapa besar tingkat akurat model yang telah dibuat dapat mengklasifikasi data dengan benar. Akurasi diperoleh dari perhitungan rasio prediksi benar dengan keseluruhan data. Dengan mengetahui besarnya nilai akurasi pada kinerja keseluruhan sistem dapat dinyatakan tingkat kemampuan sistem dalam mencari ketepatan antara informasi yang diinginkan pengguna dengan jawaban yang diberikan sistem. Akurasi yang diperoleh dalam sistem klasifikasi ini adalah sebesar 92,1100092%.

Untuk performa metode klasifikasi dari setiap kelas dapat dilihat dari nilai *precision*, *recall*, dan *f1-score* yang akan ditampilkan pada Tabel 3.

Tabel 3. Nilai Precision, Recall, Fi-Score

<i>precision</i>	<i>recall</i>	<i>f1-score</i>
0,9254	0,8277	0,8738

Sumber: Hasil Penelitian (2022)

h. **Visualisasi**

Hasil sentiment diperoleh setelah dilakukan tahap klasifikasi. *Wordcloud* merupakan bentuk visualisasi dari data teks yang menggambarkan kumpulan kata yang banyak terdapat dalam sebuah analisis teks. *Wordcloud* ditampilkan berdasarkan dua pengelompokan yaitu:

1) *Wordcloud* Positif

Script untuk menampilkan *wordcloud* positif sebagai berikut:

Algoritma 11. *Wordcloud* positif

```
1. #wordcloud positif
2. from wordcloud import WordCloud
3. from wordcloud import ImageColorGenerator
4. all_text_positive = ''.join(str(word)for word in strippedpos)
5. wordcloud= WordCloud(max_font_size=260, max_words=50,width=700,height=700,mode='RGBA',background_color='white').
generate(all_text_positive)
6. plt.figure(figsize=(15,8))
7. plt.imshow(wordcloud, interpolation='bilinear')
8. plt.axis("off")
9. plt.margins(x=0,y=0)
10. plt.show()
```

2) *Wordcloud* Negatif

Script untuk menampilkan *wordcloud* negatif sebagai berikut:

Algoritma 12. *Wordcloud* negatif

```
#wordcloud negatif
from wordcloud import WordCloud
from wordcloud import ImageColorGenerator
all_text_negative = ''.join(str(word)for word in strippedneg)
wordcloud= WordCloud(max_font_size=260, max_words=50,width=700,height=700,mode='RGBA',background_color='white').generate(al
l_text_negative)
plt.figure(figsize=(15,8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.margins(x=0,y=0)
plt.show()
```

Berdasarkan *script* tersebut dihasilkan tampilan *wordcloud* yang terdiri dari kata yang banyak muncul dalam penelitian yang disajikan pada Gambar 9.

- [5] F. F. Mailo and L. Lazuardi, “Analisis Sentimen Data Twitter Menggunakan Metode Text Mining Tentang Masalah Obesitas di Indonesia,” *Journal of Information Systems for Public Health*, vol. 4, no. 1, pp. 28–36, 2019.
- [6] D. Ayu Wulandari, R. Rohmat Saedudin, R. Andreswari, P. S. Studi, and S. Informasi, “Analisis Sentimen Media Sosial Twitter Terhadap Reaksi Masyarakat Pada Ruu Cipta Kerja Menggunakan Metode Klasifikasi Algoritma Naive Bayes Analysis of Twitter Social Media Sentiment on the Public’S Reaction To the Drafts of Job Creation Law Using the Cla,” vol. 8, no. 5, pp. 9007–9016, 2021.
- [7] V. A. Permadi, “Analisis Sentimen Menggunakan Algoritma Naive Bayes Terhadap Review Restoran di Singapura,” *Jurnal Buana Informatika*, vol. 11, no. 2, p. 140, 2020, doi: 10.24002/jbi.v11i2.3769.
- [8] Herlawati, R. T. Handayanto, D. Setiyadi, and E. Retnoningsih, “Corpus Usage for Sentiment Analysis of a Hashtag Twitter,” *Proceedings of 2019 4th International Conference on Informatics and Computing, ICIC 2019*, no. May 2021, 2019, doi: 10.1109/ICIC47613.2019.8985772.
- [9] F. V. Sari and A. Wibowo, “Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi,” *Jurnal SIMETRIS*, vol. 10, no. 2, pp. 681–686, 2019.
- [10] H. Herlawati, R. T. Handayanto, P. D. Atika, F. N. Khasanah, A. Y. P. Yusuf, and D. Y. Septia, “Analisis Sentimen Pada Situs Google Review dengan Naïve Bayes dan Support Vector Machine,” *Jurnal Komtika (Komputasi dan Informatika)*, vol. 5, no. 2, pp. 153–163, 2021, doi: 10.31603/komtika.v5i2.6280.
- [11] A. P. Tirtopangarsa and W. Maharani, “Sentiment Analysis of Depression Detection on Twitter Social Media Users Using the K-Nearest Neighbor Method,” pp. 247–258, 2021.
- [12] A. P. Giovani, A. Ardiansyah, T. Haryanti, L. Kurniawati, and W. Gata, “Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi,” *Jurnal Teknoinfo*, vol. 14, no. 2, pp. 116–124, 2020, doi: 10.33365/jti.v14i2.679.
- [13] Fadila, W. I. Rahayu, and M. H. K. Saputra, *Penerapan Metode Naive Bayes dan Skala Likert Pada Aplikasi Prediksi Kelulusan Mahasiswa*. Kreatif, 2020.



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)
