



INNOVATIVE: Journal Of Social Science Research

Volume 4 Nomor 5 Tahun 2024 Page 6145-6158

E-ISSN 2807-4238 and P-ISSN 2807-4246

Website: <https://j-innovative.org/index.php/Innovative>

## Penerapan Algoritma *Support Vector Machine* Untuk Analisis Sentimen Data Ulasan Aplikasi *Binance* Pada *Google Play Store*

Anandyta Sakti Alamsyah<sup>1✉</sup>, Saiful Nur Budiman<sup>2</sup>, Rizki Dwi Romadhona<sup>3</sup>

Universitas Islam Balitar

Email: [ananditasakti@gmail.com](mailto:ananditasakti@gmail.com)<sup>1✉</sup>

### Abstrak

Penelitian ini dilatar belakangi oleh pentingnya pemahaman sentimen pengguna terhadap aplikasi Binance dan analisis sentimen merupakan alat yang efektif untuk mencapai tujuan tersebut. Metode yang digunakan meliputi pengumpulan data ulasan dari Google Play Store pada rentang waktu tertentu, preprocessing teks untuk membersihkan data dari noise dan normalisasi teks, serta pembagian data menjadi data latih dan data uji. Selanjutnya, model SVM dilatih menggunakan data latih dan dievaluasi menggunakan data uji dan confusion matrix. Hasil eksperimen menunjukkan bahwa model SVM mencapai akurasi sebesar 87,24%. Evaluasi lebih lanjut mengungkapkan precision sebesar 85%, recall sebesar 87%, dan f1-score sebesar 85%.

Kata Kunci: *Support Vector Machine, Analisis Sentimen, Ulasan Aplikasi, Google Play Store*

### Abstract

This research is motivated by the importance of understanding user sentiment towards the Binance application, with sentiment analysis being an effective tool to achieve this goal. The methods used include collecting review data from Google Play Store over a specific period, text preprocessing to clean the data from noise and normalize the text, and splitting the data into training and testing sets. Subsequently, the SVM model was trained using the training data and evaluated using the test data and confusion matrix. Experimental results show that the SVM model achieved an accuracy of 87.24%. Further evaluation revealed a precision of 85%, a recall of 87%, and an F1-score of 85%.

Keyword: *Support Vector Machine, Sentiment Analysis, Application Reviews, Google Play Store*

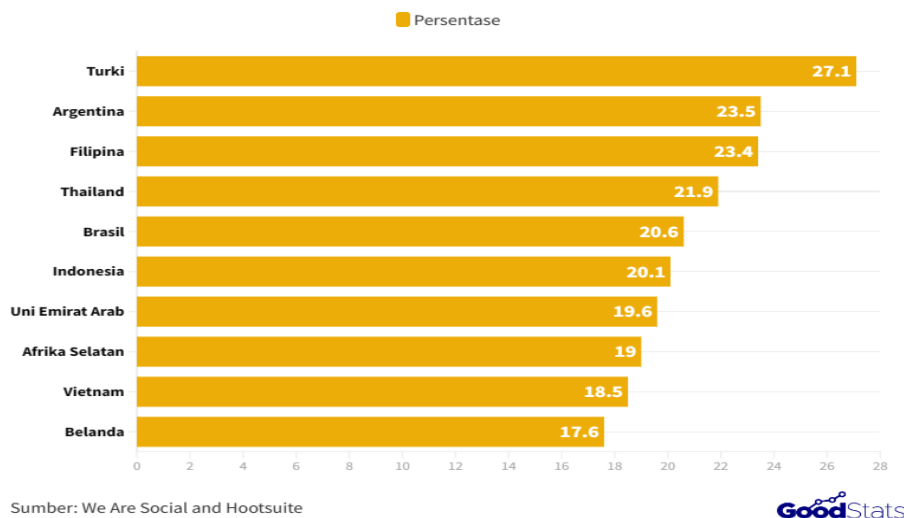
## PENDAHULUAN

Saat ini, banyak investor di Indonesia menggunakan berbagai metode investasi, termasuk berinvestasi dalam *Cryptocurrency*, selain opsi investasi lain seperti emas dan instrumen investasi lainnya. *Cryptocurrency*, sebagai bentuk mata uang digital, berbeda dengan mata uang konvensional yang memiliki bentuk fisik seperti koin atau uang kertas. Mata uang kripto adalah mata uang yang dibuat dari rangkaian kode, yang disebut *blockchain*. Karena tidak memiliki bentuk fisik, mata uang kripto terdiri dari rangkaian kode digital. Ilmuwan komputer dan matematika Amerika David Chaum menemukan algoritma khusus yang kemudian menjadi dasar enkripsi website dan transfer mata uang elektronik modern (Priambodo, 2022).

Dalam era digital yang semakin berkembang, aplikasi finansial semakin menjadi bagian penting dari kehidupan sehari-hari pengguna. Salah satu platform perdagangan *Cryptocurrency* yang paling terkenal adalah *Binance*, yang menyediakan layanan untuk perdagangan berbagai mata uang *Cryptocurrency*. Dalam konteks ini, keberhasilan suatu aplikasi tidak hanya ditentukan oleh fungsionalitas teknisnya tetapi juga oleh persepsi dan pengalaman pengguna. Oleh karena itu, penting untuk memahami sentimen pengguna terhadap aplikasitersebut, dan analisis sentimen merupakan alat yang efektif untuk mencapai tujuan tersebut.

Dari gambar dibawah, Indonesia menempati urutan keenam di dunia dalam kepemilikan mata uang kripto. Pada Januari 2023, dilaporkan bahwa 20,1 persen dari pengguna internet yang berusia 16-24 tahun di Indonesia memiliki aset kripto. Meskipun demikian, negara dengan kepemilikan mata uang kripto terbesar di dunia adalah Turki, dengan 27,1 persen. Di urutan kedua, Argentina dan Filipina memiliki kepemilikan sebesar 23,5 persen dan 23,4 persen masing-masing (Kemp, 2023).

### Daftar negara pemilik mata uang kripto terbesar Januari 2023



Gambar 1. Daftar Negara Pemilik Mata Uang Cryptocurrency

Sumber: (Naurah, 2023)

Namun, pemerintah menargetkan ekonomi digital nasional mencapai Rp2.228 triliun (kurs: Rp15.265 miliar). Ekonomi Indonesia dapat menembus US\$ 146 miliar pada tahun 2025. Dengan angka itu, diproyeksikan bahwa Indonesia akan memiliki ekonomi terbesar di Asia Tenggara (Naurah, 2023).

*Binance*, sebagai salah satu platform perdagangan *Cryptocurrency* terbesar di dunia, telah mengalami pertumbuhan yang pesat sejak didirikan pada tahun 2017. Platform ini tidak hanya menyediakan fasilitas perdagangan, tetapi juga berbagai fitur lain seperti staking, lending, dan beragam mata uang *Cryptocurrency* yang dapat diperdagangkan. Keberhasilan *Binance* tidak hanya bergantung pada keamanan teknisnya tetapi juga pada cara pengguna merespons fitur-fitur dan layanannya (Paters, 2023).

Seiring dengan pesatnya pertumbuhan penggunaan aplikasi finansial seperti *Binance*, ulasan dan testimoni pengguna di platform distribusi aplikasi, seperti *Google Play Store*, menjadi semakin signifikan. Ulasan pengguna mencakup berbagai aspek, termasuk antarmuka pengguna, kecepatan transaksi, keamanan, dan fitur-fitur lainnya. Analisis sentimen terhadap ulasan-ulasan ini dapat memberikan wawasan yang berharga bagi pengembang dan penyedia layanan, memungkinkan mereka untuk merespons kebutuhan dan preferensi pengguna dengan lebih baik.

Analisis sentimen merupakan proses mengenali dan mengkategorikan opini yang masih dalam bentuk teks ke dalam sentimen positif, negatif atau netral. Dalam konteks ini, penerapan analisis sentimen dapat mencakup evaluasi, penilaian, emosi, pendapat, atau sikap terhadap berbagai aspek seperti produk, tokoh, organisasi, layanan, isu, atau peristiwa yang terjadi di lingkungan masyarakat. Pentingnya analisis sentimen juga terkait erat

dengan interaksi dengan masyarakat, mengingat sumber informasi diperoleh melalui media sosial di mana masyarakat berperan sebagai pengguna aktif.

Dengan adanya analisis sentimen, pengambilan keputusan dapat didukung oleh data yang lebih kontekstual dan akurat, membantu para pengembang dan pengelola platform untuk menghadapi berbagai tantangan yang muncul di tengah persaingan yang semakin ketat di dunia aplikasi finansial. Oleh karena itu, penelitian ini bertujuan untuk menerapkan algoritma *Support Vector Machine* untuk melakukan analisis sentimen terhadap ulasan aplikasi *Binance* di *Google Play Store*. Cara untuk mengevaluasi performa model klasifikasi adalah dengan membandingkan nilai yang diprediksi oleh model dengan nilai sebenarnya dari data. *Confusion matrix* merupakan alat evaluasi yang digunakan untuk masalah klasifikasi dalam *machine learning*, di mana outputnya dapat terdiri dari dua kelas atau lebih.

*Confusion matrix* adalah sebuah tabel yang membandingkan kelas yang diprediksi dengan kelas aktual dari suatu contoh data. Fungsi utamanya adalah untuk mengukur kinerja model klasifikasi dengan memberikan informasi tentang seberapa baik model tersebut dalam mengklasifikasikan data. Melalui *confusion matrix*, kita dapat menghitung berbagai matrix evaluasi seperti akurasi, presisi, recall, dan f-measure. Pada dasarnya, *confusion matrix* digunakan untuk menganalisis performa model machine learning dalam klasifikasi dengan membandingkan prediksi model dengan nilai aktual dari data. *Confusion matrix* digunakan untuk mengukur kinerja. Tabel *Confusion Matrix* terdiri dari empat gabungan nilai prediksi dan nilai aktual. True Positif, True Negatif, False Positif, dan False Negatif adalah empat hasil klasifikasi dari matriks confusion.

## METODE PENELITIAN

Penelitian ini dilakukan di Universitas Islam Balitar (UNISBA) yang beralamat di Jalan Imam Bonjol No. 16 Sananwetan, Kota Blitar. Dalam penelitian ini pusat datanya adalah ulasan aplikasi di google play store. Metode yang diterapkan dalam penelitian ini adalah deskriptif kualitatif. Data yang dimanfaatkan dalam penelitian ini diambil melalui proses *web scraping* menggunakan bahasa pemrograman *Python*, kemudian disusun sedemikian rupa agar dapat dengan mudah diolah. Data yang dikumpulkan terdiri dari ulasan dan penilaian yang diberikan oleh pengguna aplikasi tersebut. Adapun metode pengumpulan data yang digunakan dalam penelitian ini yaitu observasi, *data mining*, dan studi pustaka. Tahap-tahap penelitian yaitu pengumpulan data, data hasil *scrapping*, *case folding*, *tokenizing*, *filtering*, *streaming*, pembobotan TF-IDF, hasil pembobotan TF-IDF, penerapan metode SVM, hasil klasifikasi sentimen, dan terakhir hasil pengujian *confussion matrix*.

## HASIL DAN PEMBAHASAN

Hasil

### a. Preprocessing

Setelah dilakukan tahap pengambilan data selanjutnya preprocessing. Dalam *natural language processing* (NLP), informasi yang dieksplorasi terdiri dari data yang tidak terstruktur atau memiliki format yang acak. Oleh karena itu, diperlukan proses untuk mengubah data tersebut menjadi bentuk yang terstruktur. Pada tahap ini ada empat pengolahan mulai *Case folding*, *tokenizing*, *filtering* dan *stemming*. Kemudian didapatkan hasil berupa data ulasan yang sudah dibersihkan sehingga lebih ringkas.

### a. Pelabelan Data

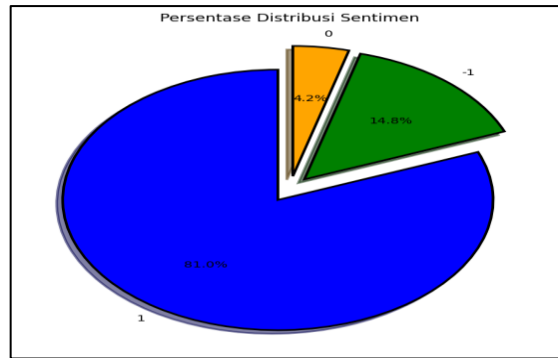
Pelabelan data adalah proses memberikan label atau kategori kepada data mentah sehingga dapat digunakan dalam model machine learning. Fungsi pelabelan data sangat penting dalam berbagai aplikasi machine learning dan analisis data. Label digunakan untuk memvalidasi dan mengevaluasi kinerja model. Setelah model dilatih, data yang dilabeli dapat digunakan untuk menguji seberapa akurat model dalam membuat prediksi. seperti akurasi, presisi, recall, dan F1-score dihitung dengan membandingkan prediksi model dengan label yang benar.

```
1  sentimen = []
2  for index, row in data.iterrows():
3      if row['score'] > 3 :
4          sentimen.append(1)
5      elif row['score'] == 3:
6          sentimen.append(0)
7      else:
8          sentimen.append(-1)
9  data['sentiment'] = sentimen
10 data.head()
```

Gambar 4. 1 Kode program pelabelan data

Kode ini menambahkan kolom sentimen ke dalam dataframe berdasarkan nilai kolom score, di mana nilai sentimen dihitung sebagai 1 untuk sentimen positif (score > 3), 0 untuk sentimen netral (score = 3), dan -1 untuk sentimen negatif (score < 3). Proses ini berguna untuk mengkategorikan ulasan berdasarkan sentimen.

Kemudian sentimen dihitung dan dari proses sebelumnya didapatkan hasil berupa sentimen positif sebesar 81.0%, sentimen negatif 14.8% dan sentimen netral 4.2%.



Gambar 2. Persentase Distribusi Sentimen

b. *Case folding*

Pada tahap case folding ini data ulasan akan diolah untuk menghilangkan perbedaan antara penggunaan huruf kapital dan huruf kecil dalam teks, sehingga penulisan menjadi lebih konsisten dan teratur.

```

1  def cleaning_text(text):
2      text = text.lower()
3      text = re.sub(r'https?://\S+', '', text)
4      text = re.sub(r'@\S+', '', text)
5      text = re.sub(r#\S+', '', text)
6      text = re.sub(r'\s+', ' ', text)
7      text = re.sub(r'\d+', '', text)
8      text = re.sub(r'^\w\s', '', text)
9      text = re.sub(r'\'\w+', '', text)
10     return text
11
12  def cleaning_emoji(text):
13     emoji_pattern = re.compile("[
14         u"\U0001F600-\U0001F64F"
15         u"\U0001F300-\U0001F5FF"
16         u"\U0001F680-\U0001F6FF"
17         u"\U0001F700-\U0001F77F"
18         u"\U0001F780-\U0001F7FF"
19         u"\U0001F800-\U0001F8FF"
20         u"\U0001F900-\U0001F9FF"
21         u"\U0001FA00-\U0001FA6F"
22         u"\U0001FA70-\U0001FAFF"
23         u"\U0001F000-\U0001F0FF"
24         u"\U0001F200-\U0001F2FF"
25         u"\U0001F400-\U0001F4FF"
26         u"\U0001F500-\U0001F5FF"
27         u"\U0001F600-\U0001F64F"
28         "]" + ", flags=re.UNICODE)
29     return emoji_pattern.sub(r'', text)
30
31  def case_folding(text):
32     text = text.lower()
33     return text
34
35  data['content'] = data['content'].apply(cleaning_text)
36  data['content'] = data['content'].apply(cleaning_emoji)
37  data['content'] = data['content'].apply(case_folding)
38  data.head()

```

Gambar 3. Kode Program *Case Folding*

Dari gambar 3 melakukan pembersihan teks dalam tiga langkah yaitu '*def cleaning\_text(text):*' membersihkan teks, '*def cleaning\_emoji(text):*' menghapus emoji, dan '*def case\_folding(text):*' melakukan case folding. Program ini membersihkan teks ulasan dalam kolom '*content*' dengan menghapus URL, mention, hashtag, spasi berlebih, angka,

karakter non-alfanumerik dan emoji, serta mengubah semua teks menjadi huruf kecil.

Tabel 4. Proses *Case Folding*

Sebelum	Sesudah
Binance bagus saya kasi bintang 5 saya sangat nyaman dan puas dengan apk ini Binance is the best	binance bagus saya kasi bintang saya sangat nyaman dan puas dengan apk ini binance is the best

c. *Tokenizing*

Pada tahap selanjutnya data diolah dengan tokenizing untuk mengubah data yang awalnya berbentuk kalimat menjadi per kata.

```
1 from nltk.tokenize import word_tokenize
2 from nltk.tokenize import RegexpTokenizer
3 regexp = RegexpTokenizer('\w+')
4 data['content_token'] = data['content'].apply(regexp.tokenize)
5 data.head(5)
```

Gambar 5. Kode Program *Tokenizing*

Program tersebut berfungsi memecah setiap kalimat atau teks dalam kolom '*content*' menjadi kata-kata individual. '*\w+*' memastikan bahwa hanya urutan karakter alfanumerik (kata) yang diambil, mengabaikan tanda baca dan karakter non-alfanumerik lainnya.

Tabel 6. Proses *Tokenizing*

Sebelum	Sesudah
yah sayang banget umur gabisa belajar crypto tolong dong ya minimum umurnya jangan	yah, sayang, banget, umur, gabisa, belajar, crypto, tolong, dong, ya, minimum, umurnya, jangan

d. *Filtering*

Digunakan untuk menghilangkan kata-kata yang tidak diperlukan atau tidak berarti dalam suatu teks. Tujuan dari filtering adalah untuk mengurangi noise dan mengoptimalkan hasil analisis teks yang akan dilakukan.

```
1 import nltk
2 nltk.download('stopwords')
3 from nltk.corpus import stopwords
4
5 stopwords = nltk.corpus.stopwords.words("indonesian")
6 my_stopwords = ['binance']
7 stopwords.extend(my_stopwords)
8
9 data['content_token'] = data['content_token'].apply(lambda x: [item for item in x if item not in stopwords])
10 data.head(5)
```

Gambar 7. Kode program *Filtering*

Kode di atas melakukan pembersihan tambahan pada teks dengan menghapus kata-kata umum (stopwords) dari teks yang telah ditokenisasi. *stopwords = nltk.corpus.stopwords.words("indonesian")* berfungsi memuat daftar stopwords untuk

bahasa Indonesia. `my_stopwords = [' ... '] stopwords.extend(my_stopwords)` berfungsi Menambahkan kata ke dalam daftar stopwords. Kata ini dianggap tidak memberikan informasi yang berguna dalam analisis. Kemudian baris berikutnya digunakan untuk Menghapus semua stopwords dari token yang sudah dibersihkan. Dilakukan dengan menggunakan fungsi lambda yang memfilter kata-kata dalam setiap daftar token.

Tabel 8. Proses *Filtering*

Sebelum	Sesudah
yah, sayang, banget, umur, gabisa, belajar, crypto, tolong, dong, ya, minimun, umurnya, jangan	'sayang', 'banget', 'umur', 'gabisa', 'belajar', 'crypto', 'tolong', 'ya', 'minimun', 'umurnya'

#### e. *Stemming*

Pada proses ini kata-kata akan diubah menjadi kata dasar menggunakan *library sastrawi*. Sastrawi adalah library yang digunakan untuk melakukan stemming pada teks dalam bahasa Indonesia. Stemming adalah proses mengurangi kata-kata turunan menjadi bentuk dasar atau kata dasarnya.

```

1  !pip install Sastrawi
2
3  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
4
5  factory = StemmerFactory()
6  stemmer = factory.create_stemmer()
7
8  data['stemmed'] = data['content_token'].apply(lambda x: [stemmer.stem(y) for y in x])
9  data.head(5)

```

Gambar 9. Kode program *Stemming Sastrawi*

Kode ini digunakan untuk melakukan stemming pada kata-kata di dalam sebuah kolom dataframe dengan memanfaatkan library Sastrawi, yang khusus dirancang untuk bahasa Indonesia. Hasil stemming kemudian disimpan dalam kolom baru pada dataframe, sehingga memungkinkan analisis teks yang lebih mudah dan konsisten.

Tabel 1. Proses *Stemming*

Sebelum	Sesudah
'sayang', 'banget', 'umur', 'gabisa', 'belajar', 'crypto', 'tolong', 'ya', 'minimun', 'umurnya'	'sayang', 'banget', 'umur', 'bisa', 'ajar', 'crypto', 'tolong', 'ya', 'minim', 'umur'

#### b. Visualisasi Data Sentimen

Visualisasi berfungsi melihat pola-pola yang mungkin tersembunyi di dalam data mentah. Sebagai contoh, lonjakan besar dalam sentimen negatif pada suatu waktu dapat mengindikasikan adanya masalah spesifik yang perlu diperhatikan. Wordcloud menampilkan kata-kata yang paling sering muncul dalam teks dengan ukuran yang lebih besar, sehingga dengan mudah dapat dilihat kata-kata kunci atau topik utama dari teks

tersebut.

#### a. Sentimen Positif

Sentimen positif adalah hasil dari proses klasifikasi yang mengelompokkan ulasan ke dalam kategori positif menggunakan analisis sentimen. Ulasan positif ini diidentifikasi berdasarkan seberapa sering istilah tertentu muncul dalam ulasan tersebut. Berikut adalah visualisasi output dari ulasan positif yang diekstraksi berdasarkan informasi yang dianalisis dari ulasan yang ditulis oleh pengguna. Berikut ini adalah skrip untuk menunjukkan kata-kata yang paling sering muncul dalam kelas sentimen positif, serta hasil dari *wordcloud* tersebut dapat dilihat pada Gambar 10.

```
1 data_p=data[data['sentiment']==1]
2 all_words_lem = ' '.join([word for word in data_p['text_string']])
3
4 %matplotlib inline
5 import matplotlib.pyplot as plt
6 from wordcloud import WordCloud
7
8 wordcloud = WordCloud(background_color='white', width=800, height=500, random_state=21, max_font_size=130).generate(all_words_lem)
9
10 plt.figure(figsize=(20, 10))
11 plt.imshow(wordcloud, interpolation='bilinear')
12 plt.axis('off');
```

Gambar 10. Kode program visualisasi sentimen positif



Gambar 11. Wordcloud sentimen positif

#### b. Sentimen Negatif

Ulasan dengan sentimen negatif merupakan hasil dari pengelompokan data ke dalam kelas negatif menggunakan analisis sentimen. Ulasan negatif ini diidentifikasi berdasarkan seberapa sering kata-kata tertentu muncul dalam ulasan tersebut. Berikut adalah hasil visualisasi *wordcloud* dari ulasan-ulasan negatif yang diekstraksi dari informasi yang diperoleh dari ulasan yang ditulis oleh pengguna *Binance*.

```
1 data_neg=data[data['sentiment']==-1]
2 all_words_lemneg = ' '.join([word for word in data_neg['text_string']])
3
4 %matplotlib inline
5 import matplotlib.pyplot as plt
6 from wordcloud import WordCloud
7
8 wordcloud = WordCloud(background_color='white', width=800, height=500, random_state=21, max_font_size=130).generate(all_words_lemneg)
9
10 plt.figure(figsize=(20, 10))
11 plt.imshow(wordcloud, interpolation='bilinear')
12 plt.axis('off');
```

Gambar 12. Kode program visualisasi sentimen Negatif



uji. Weighted average menunjukkan precision sebesar 85%, recall 87%, dan f1-score 85%. Untuk perhitungan precision, recall dan f1-score sebagai berikut.

- Precision Positif

$$Precision = \frac{TP}{TP + FP} * 100\%$$

$$Precision = \frac{19}{19 + 10} = \frac{19}{29} * 100\%$$

$$Precision = 66\%$$

- Precision Negatif

$$Precision = \frac{TN}{TN + FN} * 100\%$$

$$Precision = \frac{316}{316 + 39} = \frac{316}{355} * 100\%$$

$$Precision = 89\%$$

- Recall Positif

$$Recall = \frac{TP}{TP + FN} * 100\%$$

$$Recall = \frac{19}{19 + 39} = \frac{19}{58} * 100\%$$

$$Recall = 33\%$$

- Recall Negatif

$$Recall = \frac{TN}{TN + FP} * 100\%$$

$$Recall = \frac{316}{316 + 10} = \frac{316}{326} * 100\%$$

$$Recall = 97\%$$

- F1-Score Positif

$$F1 = 2 * \frac{(Recall * Precision)}{(Recall + Precision)}$$

$$F1 = 2 * \frac{(33 * 66)}{(33 + 66)} = \frac{2176}{99} = 44\%$$

- F1-Score Negatif

$$F1 = 2 * \frac{(Recall * Precision)}{(Recall + Precision)}$$

$$F1 = 2 * \frac{(97 * 89)}{(97 + 89)} = \frac{8633}{186} = 93\%$$

Untuk menghitung weighted average, kita perlu mengetahui jumlah sampel di setiap kelas. Dari data di atas, kita memiliki:

- Total data negatif:  $316 + 10 = 326$
  - Total data positif:  $19 + 39 = 58$
  - Total sampel:  $326 + 58 = 384$
- Precision

$$Precision = \frac{Precision\ Negatif * Jumlah\ Negatif + Precision\ Positif * Jumlah\ Positif}{Jumlah\ Total\ Sampel}$$

$$\text{Precision} = \frac{89 * 326 + 66 * 58}{384}$$

$$\text{Precision} = \frac{29014 + 3828}{384} = 85\%$$

- Recall

$$\text{Recall} = \frac{\text{Recall Negatif} * \text{Jumlah Negatif} + \text{Recall Positif} * \text{Jumlah Positif}}{\text{Jumlah Total Sampel}}$$

$$\text{Recall} = \frac{97 * 326 + 33 * 58}{384}$$

$$\text{Recall} = \frac{31622 + 1914}{384} = 87\%$$

- F1-Score

$$F1 = \frac{F1 \text{ Negatif} * \text{Jumlah Negatif} + F1 \text{ Positif} * \text{Jumlah Positif}}{\text{Jumlah Total Sampel}}$$

$$F1 = \frac{93 * 326 + 44 * 58}{384}$$

$$F1 = \frac{30318 + 2552}{384} = 85\%$$

Jadi, berdasarkan perhitungan di atas, kita mendapatkan Precision 85%, Recall 87% dan F1-score 85%

```

1 from sklearn.metrics import confusion_matrix
2
3 cm = confusion_matrix(y_test, y_pred)
4 accuracy = accuracy_score(y_test, y_pred)
5
6 print('Confusion matrix\n\n', cm)
7 print('\nTrue Positives (TP) = ', cm[0, 0])
8 print('\nTrue Negatives (TN) = ', cm[1, 1])
9 print('\nFalse Positives (FP) = ', cm[0, 1])
10 print('\nFalse Negatives (FN) = ', cm[1, 0])
11 print('\nAccuracy = ', accuracy)

```

Gambar 16. Kode program *Confusion Matrix*

Kode program tersebut digunakan untuk menghitung dan menampilkan *confusion matrix* serta akurasi model klasifikasi. *Confusion matrix* memberikan informasi mengenai kinerja model dengan mengkategorikan prediksi ke dalam empat kategori: True Positives (TP), True Negatives (TN), False Positives (FP), dan False Negatives (FN). Indeks 0 dan 1 dalam matriks digunakan untuk mengakses elemen spesifik yang merepresentasikan jumlah masing-masing kategori tersebut. Dengan menghitung jumlah masing-masing kategori ini, kita bisa mendapatkan gambaran lebih jelas tentang kinerja model, termasuk seberapa baik model dalam memprediksi kelas positif dan negatif. Hasilnya bisa dilihat pada gambar 16.

```

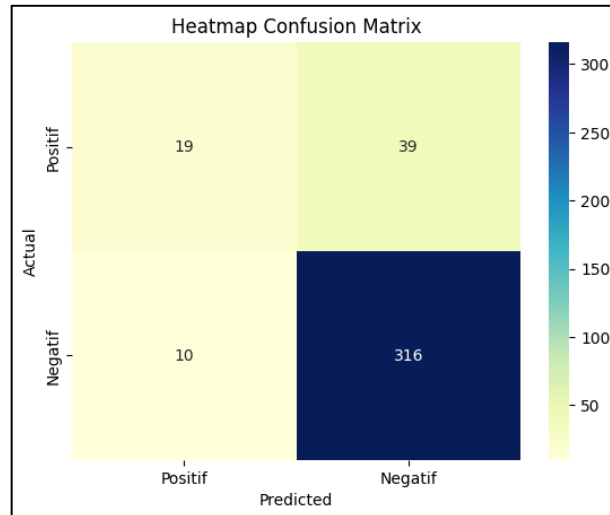
Confusion matrix

[[ 19  39]
 [ 10 316]]

True Positives (TP) = 19
True Negatives (TN) = 316
False Positives (FP) = 39
False Negatives (FN) = 10
Accuracy = 0.8723958333333334

```

Gambar 17. Hasil *Confusion Matrix*



Gambar 18. Visualisasi *Confusion Matrix*

Tabel 2. Hasil *Confusion Matrix*

Aktual	Prediksi	
	Positif	Negatif
Positif	19	39
Negatif	10	316
Akurasi	87,24%	

Tabel menunjukkan pada data ulasan pengguna Binance terdapat 316 data negatif yang benar diprediksi termasuk ke sentimen negatif dan 10 data negatif yang diprediksi termasuk ke dalam sentimen positif, kemudian terdapat 19 data positif yang benar diprediksi termasuk ke dalam sentimen positif dan 39 data positif yang diprediksi termasuk ke dalam sentimen negatif. Untuk perhitungan akurasi SVM sebagai berikut.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} * 100\%$$

$$Accuracy = \frac{19 + 316}{19 + 39 + 316 + 10} * 100\%$$

$$Accuracy = \frac{335}{384} * 100\%$$
$$Accuracy = 87,24\%$$

## Pembahasan

Berdasarkan hasil penelitian ini menghitung pembahasan hasil analisis yang dilakukan menunjukkan bahwa dari proses pengumpulan data menggunakan teknik web scrapping, diperoleh 500 ulasan pengguna aplikasi *Binance*. Analisis dalam penelitian ini dilakukan menggunakan *Google Collaboratory* dengan bahasa pemrograman Python. Tahapan dalam analisis ini meliputi *preprocessing* teks, pembobotan kata, klasifikasi data dengan Support Vector Machine, dan pengujian menggunakan confusion matrix.

Dari hasil pengujian confusion matrix terhadap klasifikasi data, diperoleh bahwa 19 data positif berhasil terprediksi dengan benar sebagai sentimen positif, dan 39 data positif terprediksi salah sebagai sentimen negatif. Selain itu, terdapat 316 data negatif yang berhasil terprediksi dengan benar sebagai sentimen negatif, sementara 10 data negatif terprediksi salah sebagai sentimen positif. Kemudian menghasilkan akurasi sebesar 87,24% dengan menunjukkan precision sebesar 85%, recall 87%, dan f1-score 85%.

## SIMPULAN

Berdasarkan analisis dan hasil penelitian yang telah dilakukan dari proses analisis sentimen pengguna aplikasi *Binance* menggunakan algoritma *Support vector machine*, beberapa kesimpulan dapat diambil sebagai berikut :

1. Model *Support Vector Machine* yang digunakan dalam penelitian ini menunjukkan akurasi sebesar 87,24%, menunjukkan bahwa model memiliki kinerja yang baik dalam memprediksi sentimen ulasan pengguna aplikasi Binance.
2. Dari hasil pengujian confusion matrix, diperoleh 19 data positif yang benar terprediksi sebagai sentimen positif dan 39 data positif yang terprediksi salah sebagai sentimen negatif. Selain itu, 316 data negatif berhasil terprediksi dengan benar sebagai sentimen negatif, sementara 10 data negatif terprediksi salah sebagai sentimen positif. Hal ini menunjukkan bahwa model lebih baik dalam memprediksi sentimen positif dibandingkan sentimen negatif.

Dari hasil proses web scraping tersebut didapatkan bahwa Dari 500 sampel ulasan didapatkan gambaran data ulasan *Binance* memiliki data kelas positif sebanyak 405 buah ulasan dan kelas negatif sebanyak 74 ulasan.

## DAFTAR PUSTAKA

- Kemp, S. (2023). *DIGITAL 2023: GLOBAL OVERVIEW REPORT*. Datareportal.Com.  
<https://datareportal.com/reports/digital-2023-global-overview-report>
- Naurah, N. (2023). *Indonesia Masuk Jajaran Negara Pemilik Kripto Terbesar di Dunia*. Goodstats.Id. <https://goodstats.id/article/indonesia-masuk-jajaran-negara-pemilik-kripto-terbesar-di-dunia-ugkz6>
- Paters, K. (2023). *No Title*. Investopedia.Com.  
<https://www.investopedia.com/terms/b/binance-exchange.asp>
- Priambodo, G. (2022). Tinjauan Yuridis Terhadap Investasi Aset Kripto Berdasarkan Peraturan Badan Pengawas Perdagangan Berjangka Komoditi Nomor .... *Lex Crimen*, 7(3). <https://ejournal.unsrat.ac.id/index.php/lexcrimen/article/view/44360>