Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

Application of Support Vector Machine (SVM) Algorithm in Analyzing Public Sentiment in the Media Tiktok Related to Tour de Entete Event

Frich Nandra Mangi[1]*, Fajar Hariadi[2], Leonard Marten Doni Ratu[3]

[1,2,3] *Informatics Engineering Study Program, Wira Wacana Christian University Sumba*
nandramangi@gmail.com [1], fajar@unkriswina.ac.id [2], leonard.ratu@unkriswina.ac.id [3]*

Abstract

This study aims to analyze public sentiment on TikTok social media towards the implementation of the Tour De EnTeTe event using the Support Vector Machine (SVM) algorithm. The phenomenon of using social media as a space for public opinion provides an opportunity for local governments to evaluate the psychological and social impact of sports tourism activities in real-time. The research data was obtained through a crawling technique that produced 1,000 data entries, which were then reduced to 808 valid data after going through the text preprocessing stage. The analysis is done by classifying the data into positive and negative sentiments. The results showed that the SVM model was able to provide excellent performance with an accuracy rate of 89%, precision of 91%, and recall of 97%. The findings show a significant dominance of positive sentiment, reflecting the high enthusiasm and support of the public for the Tour De EnTeTe. Although there are constraints on language ambiguity that lead to some misclassification, overall the SVM algorithm has proven to be effective and accurate as an instrument for digital opinion analysis. This study concludes that the public perception of the event is very positive, so that it can be used as a strategic reference in the development of tourism promotion in East Nusa Tenggara in the future.

***Keywords**: Sentiment Analysis, Support Vector Machine, Tour De EnTeTe, Sport Tourism, Social Media, TF-IDF.*

1. Introduction

The growth of the digital era is very rapid and has a significant impact on various aspects of people's lives, including in the way they communicate and express their opinions [1]. In Indonesia, the development of social media also goes hand in hand with the increasing number of major events on a national and international scale that are the center of public attention, such as the Asian Games, Mandalika MotoGP, World Superbike (WSBK), to the G20 Summit in Bali. Not only at the national level, East Nusa Tenggara (NTT) also held a big event that attracted the attention of the wider community, namely the Tour De Entete.

Tour De EnTeTe is an international bicycle racing event held in East Nusa Tenggara (NTT), Indonesia, with the main goal of promoting sports tourism in the archipelago. This prestigious event is not only a competition for national and international teams, but also a means to display the natural beauty, cultural richness, and friendliness of the people of NTT to the world. During the race, participants were treated to spectacular scenery ranging from majestic mountains, stunning beaches, to vast expanses of savannah, making each stage not only a physical challenge, but also an unforgettable visual experience.

In the midst of the ongoing Tour De EnTeTe event, various dynamics of responses from the community have emerged that reflect various perspectives on the implementation of the activity. Social media is one of the main spaces for the public to express their opinions, both in the form of appreciation for the positive impact of events on regional promotion, as well as criticism of technical and managerial aspects that are considered to still need improvement. As one of the TikTok accounts from @EGHY NAMANG wrote and the impact was non-existent😌. Likewise on the account of @Xytrkl wrote The whole road is not OK.. Repair road infrastructure first.. new. Make a Tour of Sorts2. From @Gilberto account wrote coolnn coolnn👍. The public response conveyed through social media is very diverse, both positive and negative, showing that the event gets different responses according to the experience and perception of the community.

In line with the importance of understanding public perception of the Tour De EnTeTe event, sentiment analysis is a relevant approach to systematically identify people's perceptions and attitudes [2]. Sentiment analysis is a technique for evaluating emotions in digital texts to determine whether they are positive or negative [3]. One of the methods used in this study is the Support Vector Machine method which will be written as SVM [4]. SVM is a classification method that falls under the category of supervised learning [5].

2. Theoretical Foundations

2.1. Social Media

Social media invites anyone who is interested to participate by contributing in a quick and unlimited time. The term social media is composed of two words, namely media and social. Media means communication children, while the word social is interpreted as a social reality that each individual takes actions that contribute to society. So social media is an online medium where users can easily interact [6].

2.2. Public Opinion

Public opinion is the view, reaction, or attitude that develops among the public towards a certain issue, event, figure, or policy. Public opinion is formed through processes that involve social interaction, information exchange, personal experiences, and the influence of mass media and social media. In social and political contexts, public opinion has an important role because it can influence decision-making, policy making, and social dynamics in society. In addition, public opinion can be positive, negative, or neutral, depending on the collective perception formed on the issue being discussed [7].

2.3. Sentiment Analysis

Sentiment analysis is the process of determining the attitudes, opinions, or emotions contained in a text, whether they are positive, negative, or neutral. This process is usually applied to text-based data such as comments or reviews to find out the tendency of people's views on a certain issue or object. Sentiment analysis utilizes natural language processing (NLP) techniques to identify language patterns that reflect emotions. The results can be used to support decision-making in the fields of marketing, policy, and social research [8].

2.4. Text Mining

Text mining is the process of extracting valuable information from unstructured text data with the aim of finding specific patterns, knowledge, or relationships. This process involves several stages such as text cleaning, data transformation, and analysis using computational techniques. Text mining is widely used in various fields such as research, marketing, and decision-making to understand the information hidden in large amounts of text data [9].

2.5. Machine Learning

Machine learning (ML) is a branch of artificial intelligence that allows systems to learn from data and improve its performance without the need for explicit programming. Through this approach, models can recognize patterns, make predictions, or make decisions based on previous data experience. Machine learning is widely applied in various fields, such as product recommendations, fraud detection, sentiment analysis, to image and voice processing [10].

2.6. Natural Language Processing (NLP)

Natural Language Processing (NLP) is a branch of artificial intelligence that focuses on the interaction between computers and human natural language. Through NLP, computers can understand, process, and produce text and speech in human language in a more meaningful way. This technology is used in a variety of applications such as automatic translators, chatbots, sentiment analysis, and information retrieval systems [11].

2.7. Support Vector Machine (SVM) Algorithm

Support Vector Machine (SVM) is a machine learning algorithm used for classification and regression tasks by looking for the best hyperplane that can separate data into different classes. SVM works by maximizing the margins between data from different classes resulting in optimal separation. This algorithm is widely used because it is able to provide good performance on high-dimensional data and remain effective even when the amount of data is not very large [12].

3. Research Methodology

The research flow in this study involves several main stages, ranging from data collection to model evaluation using the Support Vector Machine (SVM) algorithm. The research process will run systematically and in a directed manner according to the concept that has been designed, which can be illustrated in the following diagram:

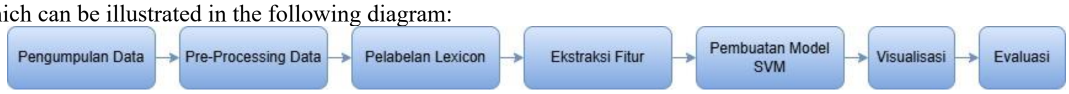


Fig. 1. Research Flow

3.1. Data Collection

Data collection in this study was carried out using Arify, which is a tool used to automatically retrieve data from social media. The data collected came from the TikTok platform, focusing on posts discussing Tour de EnTeTe activities. Data collection is done by searching for uploads that use hashtags such as #TourdeEnTeTe and other keywords related to the event. From this collection process, 1,000 data entries were obtained containing comments, video descriptions, and TikTok users' responses to Tour de EnTeTe activities. All of the data is then stored in CSV (Comma Separated Values) format so that it can be easily processed at the next stage of analysis.

3.2. Pre-Processing Data

The preprocessing stage is applied to prepare the text data before sentiment analysis is carried out. This process includes cleaning to remove irrelevant characters, case folding to equalize letter writing, normalization to convert non-standard words into standard, tokenizing to break text into tokens, stopword removal to remove common words that don't contribute to meaning, and stemming to return words to their basic form. This stage aims to improve the consistency of the data and the effectiveness of the sentiment classification process.

3.3. Labeling Lexicon

In this section, the data that has been obtained will be grouped or labeled into two parts, namely Negative and Positive based on the results of preprocessing. This labeling process is carried out using a lexicon dictionary approach.

3.4. TF-IDF calculation

Once the data has been cleaned and processed, the next stage is to convert the text into a numerical representation using the TF-IDF method. TF-IDF (Term Frequency-Inverse Document Frequency) is used to assess how important a word is in a given document relative to the entire document collection.

3.5. Model Creation of Support Vector Machine

Once the data is labeled, a Support Vector Machine (SVM) model will be built to classify sentiment into two positive and negative categories. SVM is an effective machine learning algorithm in text classification and is particularly suitable for data that has a lot of features and high dimensions. The model is trained using pre-processed training data, which covers 80% of the total data, and then tested using test data covering 20% of the total data to evaluate its performance.

3.6. Visualization

After the model is trained, the results of the analysis will be visualized in the form of a graph to make it easier to understand. A bar chart or pie chart will be used to show the distribution of sentiment (positive and negative).

3.7. Model Evaluation

Once the SVM model is built and the prediction is made using the One-vs-All (OvA) method, the next step is an evaluation to measure the model's performance. One method that is often used to assess the performance of classification models is to use a confusion matrix. The confusion matrix gives an idea of the success rate of the model in classifying the data correctly or incorrectly in each class.

4. Results and discussion

4.1. Data Collection

To crawl data from TikTok, users need to register and have an account on the Arify platform, which allows access to TikTok's API legally. After signing up, users must verify their account to gain full access to the API. Next, users need to set data capture criteria that are relevant to the research objective, such as choosing a specific video or hashtag. The crawling process then begins, and the retrieved data will be downloaded in JSON or CSV format. After successfully collecting 1000 data entries, the data is saved in CSV format for further analysis.

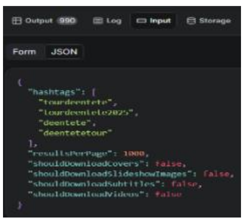


Fig. 2. Script Crawling Data

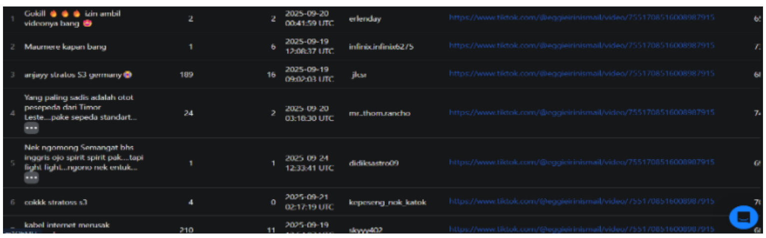


Fig. 3. Data Crawling Results

4.2. Pre-Processing Data

1) Cleaning

In the cleaning stage, data is cleaned to ensure that the text used in the analysis is free of irrelevant elements. Hashtags, URLs, emojis, symbols, numbers, and usernames are removed using functions that have been adapted to regular expressions (regex).

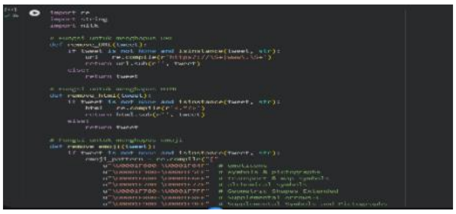


Fig. 4. Script Cleaning Data

index	text	cleaning
0	Best view kalo mau lihat cantiknya kelokan jalan di tengah perbukitan Sumba ✨ #explore #sumba #sumbatimur_ntt #piarakukuhills	Best view kalo mau lihat cantiknya kelokan jalan di tengah perbukitan Sumba explore sumba sumbatimurntt piarakukuhills
1	Salah satu jalan tercantik di Indonesia. 🥰 #indonesia #wonderfulindonesia #fyp〽️ #sumba #sumbaisland #palindipiarakuku #piarakukuhills	Salah satu jalan tercantik di Indonesia indonesia wonderfulindonesia fyp sumba sumbaisland palindipiarakuku piarakukuhills
2	alam NTT tidak ada habisnya semua indah #piarakukuhills #fy #fypage #fypppppppppppppppppppppp #c70 #c70indonesia #c70modifikasi #c70touring #silaturiding	alam NTT tidak ada habisnya semua indah piarakukuhills fy fypage fypppppppppppppppppppppp c cindonesia cmodifikasi ctouring silaturiding

Fig. 5. Cleaning Results

2) Case Folding

In the case folding stage, the text is changed to lowercase so that the analysis model does not distinguish between uppercase and lowercase letters. The case_folding() function checks whether the text is a string and converts it to lowercase using the .lower() method.

```
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(5)
```

Fig. 6. Script Case Folding

cleaning	case_folding
waingapusumbatimur deentetefyp	waingapusumbatimur deentetefyp
tour de entete stag melewati salah satu jalan tericonic yang ada di pulau sumba tanarara waingapu nttpride sumba ride rider tourdeentete	tour de entete stag melewati salah satu jalan tericonic yang ada di pulau sumba tanarara waingapu nttpride sumba ride rider tourdeentete

Fig. 7. Case Folding Results

3) Normalization

At the normalization stage, non-standard words are replaced with standard forms according to language rules using dictionaries downloaded from external sources. The replace_taboo_words() function is applied after the case folding stage to replace non-standard words in the text with the correct standard shape, record the replaced words, and generate a hash for each non-standard word. This process ensures consistency of language in the data for further analysis.

```
# Unduh dan baca kamus dari GitHub
url = "https://github.com/analysisdatasentiment/kamus_kata_baku/raw/main/kamuskatabaku.xlsx"
response = requests.get(url)
file_excel = BytesIO(response.content)
kamus_data = pd.read_excel(file_excel)

# Buat dictionary dari kamus
kamus_tidak_baku_dict = dict(zip(kamus_data['tidak_baku'], kamus_data['kata_baku']))

# Terapkan fungsi normalisasi
data[['normalisasi', 'Kata Baku', 'Kata Tidak Baku', 'Kata Tidak Baku Hash']] = data['case_folding'].apply(
    lambda x: pd.Series(replace_taboo_words(x, kamus_tidak_baku_dict))
)

# Ambil kolom yang relevan
df = pd.DataFrame(data[['text','cleaning','case_folding','normalisasi']])
df.head(5)
```

Fig. 8. Normalization Script

case_folding	normalisasi
waingapusumbatimur deentetefyp	waingapusumbatimur deentetefyp
tour de entete stag melewati salah satu jalan tericonic yang ada di pulau sumba tanarara waingapu nttpride sumba ride rider tourdeentete	tour deh entete stag melewati salah satu jalan tericonic yang ada di pulau sumba tanarara waingapu nttpride sumba ride rider tourdeentete

Fig. 9. Normalization Results

4) Tokenize

At the tokenization stage, the normalized text is broken down into word units or tokens using the tokenize() function. These tokens are stored in a new column on the dataset called "tokenize", facilitating further analysis such as word frequency, sentiment, or machine learning.

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

df['tokenize'] = df['normalisasi'].apply(tokenize)

df.head(5)
```

1 to 5 of 5 entries Filter

normalisasi	tokenize
waingapusumbatimur deentetefyp	waingapusumbatimur,deentetefyp
tour deh entete stag melewati salah satu jalan tericonic yang ada di pulau sumba tanarara waingapu nttpride sumba ride rider tourdeentete	tour,deh,entete,stag,melewati,salah,satu,jalan,tericonic,yang,ada,di,pulau,sumba,tanarara,waingapu,nttpride,sumba,ride,rider,tourdeentete

Fig. 10. Tokenize Results

5) Stopword
At the stopword removal stage, common words that are not relevant to analysis, such as conjunctions, are removed using the list of stopwords from the NLTK library. The remove_stopwords() function is applied to the tokens that have been separated, and the result is stored in a new "stopword removal" column on the dataset, ensuring only significant words are retained.

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

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

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

# Ubah hasil list jadi string
df['stopword removal'] = df['tokenize'].apply(
    lambda x: " ".join(remove_stopwords(x))
)

df.head(5)
```

Fig. 11. Script Stopword

tokenize	stopword removal
waingapusumbatimur,deentetefyp	waingapusumbatimur deentetefyp
tour,deh,entete,stag,melewati,salah,satu,jalan,tericonic,yang,ada,di,pulau,sumba,tanarara,waingapu,nttpride,sumba,ride,rider,tourdeentete	deh stag melewati salah jalan tericonic pulau tanarara waingapu nttpride ride rider

Fig. 12. Tokenize Results

6) Vote
At the voting stage, words are changed to basic forms using the Literary library. The stem_text() function is applied to text that has been cleaned of stopwords, and the result is stored in the "stemming_data" column for further analysis.

```
!pip install Sastrawi

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

Requirement already satisfied: Sastrawi in /usr/local/lib/python3.12/dist-packages (1.0.1)

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

def stem_text(text):
    words = text.split()  # Memecah teks menjadi kata-kata
    return [stemmer.stem(word) for word in words] # Melakukan stemming pada setiap kata


# Menerapkan fungsi stemming pada kolom 'stopword removal'
df['steming_data'] = df['stopword removal'].apply(lambda x: ' '.join(stem_text(x)))

# Menampilkan 5 baris pertama hasil stemming
df.head(5)
```

Fig. 13. Script Mood

stopword removal	stemming_data
waingapusumbatimur deentetefyp	waingapusumbatimur deentetefyp
deh stag melewati salah jalan tericonic pulau tanarara waingapu nttpride ride rider	deh stag lewat salah jalan tericonic pulau tanarara waingapu nttpride ride rider

Fig. 14. Voting Results

Before preprocessing, many words were irrelevant such as non-standard words, repetition of letters, language mixes, symbols, and stopwords that interfered with sentiment analysis. After preprocessing, word cloud becomes cleaner and more relevant, with structured words like "tour," "dentete," "code," and "stage." Preprocessing processes such as case folding, tokenizing, stopword removal, and word normalization improve data quality, allowing for more focused and accurate sentiment analysis.

Fig. 15. Wordcloud PreProcessing Results

4.3. Lexicon labeling

After the data is cleaned and processed with TF-IDF, labeling is carried out automatically using the Indonesian opinion dictionary Sentiment lexicon. Words in the dictionary are given a polarity score from -5 (negative) to +5 (positive). Words that are not in the dictionary are ignored. If the labeling total score is < 0, the label is negative, and if > 0, the label is positive.

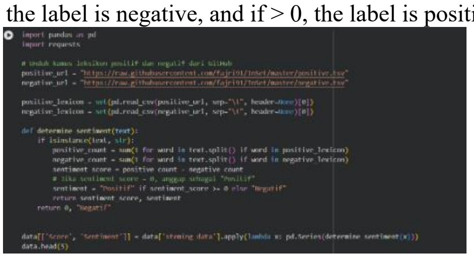


Fig. 16 Lexicon Labeling Script

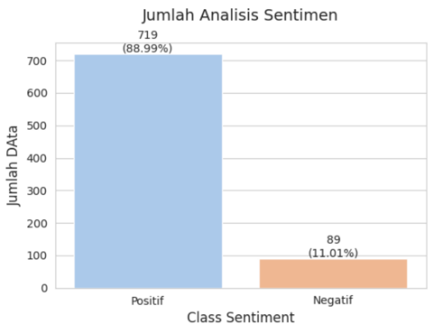
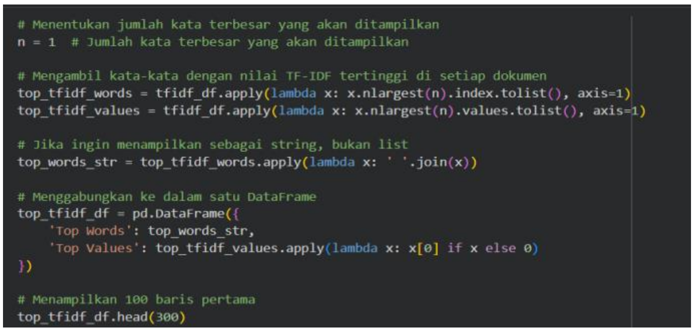


Fig. 17. Labeling graphics

The graph shows the distribution of positive and negative sentiment in the dataset. Positive sentiment dominated with 719 data (88.99%), while negative sentiment was only 89 data (11.01%). These significant differences indicate data imbalances, which can lead to the model being more biased towards the majority class. Therefore, data balancing techniques or model evaluation need to be considered to keep the classification results accurate and representative.

4.4. TF-IDF calculation

At the TF-IDF calculation stage, the TfidfVectorizer is used to calculate the importance of words in a document. The fit_transform() function is applied to the data after stemming to generate a TF-IDF matrix, which is then converted to a DataFrame. Next, the words with the highest TF-IDF value are selected using the apply() method and arranged in descending order, and then stored in a new DataFrame with their TF-IDF values.



```
# Menentukan jumlah kata terbesar yang akan ditampilkan
n = 1  # Jumlah kata terbesar yang akan ditampilkan

# Mengambil kata-kata dengan nilai TF-IDF tertinggi di setiap dokumen
top_tfidf_words = tfidf_df.apply(lambda x: x.nlargest(n).index.tolist(), axis=1)
top_tfidf_values = tfidf_df.apply(lambda x: x.nlargest(n).values.tolist(), axis=1)

# Jika ingin menampilkan sebagai string, bukan list
top_words_str = top_tfidf_words.apply(lambda x: ' '.join(x))

# Menggabungkan ke dalam satu DataFrame
top_tfidf_df = pd.DataFrame({
    'Top Words': top_words_str,
    'Top Values': top_tfidf_values.apply(lambda x: x[0] if x else 0)
})

# Menampilkan 100 baris pertama
top_tfidf_df.head(300)
```

Fig. 18. TF-IDF Calculation Script

4.5. Model Creation of Support Vector Machine

At this stage, a sentiment classification model was made using the Support Vector Machine (SVM) algorithm by utilizing the Term Frequency–Inverse Document Frequency (TF-IDF) extraction feature. The TF-IDF numerical representation is used so that text reviews can be processed by the classification algorithm. SVM works by finding the best hyperplane that separates data into two sentiment classes, i.e. positive and negative, through the maximum margin formed by support vectors. The training process was carried out by studying the patterns of connectivity between TF-IDF features and sentiment labels so that the model was able to recognize words associated with certain sentiments. The SVM algorithm was chosen because it is effective for high-dimensional text data and is able to produce good classification performance. The model formed is then used at the stage of testing and performance evaluation.

```
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.neural_network import MLPClassifier
import seaborn as sns
import matplotlib.pyplot as plt

# Initialize models
models = {
    "SVM": SVC(kernel='linear', random_state=42),
}

# Train models
results = {}
for model_name, model in models.items():
    model.fit(X_train_vec, y_train)
    y_pred = model.predict(X_test_vec)
    results[model_name] = {
        "accuracy": accuracy_score(y_test, y_pred),
        "classification_report": classification_report(y_test, y_pred, output_dict=True),
        "confusion_matrix": confusion_matrix(y_test, y_pred)
    }
```

Fig. 19. SVM Model Creation Script

Before the training process is carried out, the datasets that have gone through the preprocessing and labeling stages are divided into training data (training set) and test data (test set). Initially, 1,000 review data was collected from TikTok, but after text cleanup and duplication removal, the number of data worth analyzing was reduced to 808 entries. The net dataset was then divided using the train_test_split() function with a proportion of 80% for training data as many as 640 entries and 20% for test data as many as 160 entries. This division is carried out to test the model's performance on data that has never been seen before and to avoid the risk of overfitting.

```
import matplotlib.pyplot as plt


train_size = len(X_train)
test_size = len(X_test)

plt.figure(figsize=(6, 4))
bars = plt.bar(['Data Training', 'Data Testing'], [train_size, test_size], color=['blue', 'orange'])

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 1, f'{height} ({height / (train_size + test_size) * 100:.2f}%)',
             ha='center', va='bottom')

plt.title('Jumlah Data Latih dan Data Uji')
plt.xlabel('Jenis Data')
plt.ylabel('Jumlah Data')
plt.show()
```

Fig. 20. Data Sharing Script

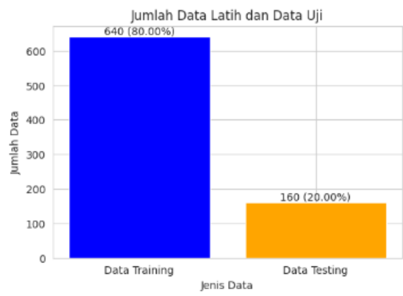


Fig. 21. Data Sharing Graph

4.6. Visualization

In the visualization stage, the confusion matrix is displayed in the form of a heatmap using the matplotlib and seaborn libraries. The values on each cell are displayed directly to make it easier to observe between the correct and false classification results. The horizontal axis shows the model's prediction results, while the vertical axis shows the actual value. Class labels such as Positive and Negative are added for clearer interpretation. With this visualization, researchers can see the model's prediction distribution and assess the classification performance more easily.

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

for model_name, result in results.items():
    fig, ax = plt.subplots(figsize=(6, 6))
    sns.heatmap(
        result["confusion_matrix"],
        annot=True,
        fmt='d',
        cmap="YlOrRd",
        cbar=False,
        xticklabels=['Negatif','Positif'],
        yticklabels=['Negatif','Positif'],
        ax=ax,
        square=True,
        linewidths=0.5
    )
    print(f"\nConfusion Matrix for {model_name}:")
    ax.set_title(f"{model_name} Confusion Matrix", fontsize=12)
    ax.set_xlabel("Predicted", fontsize=12)
    ax.set_ylabel("Actual", fontsize=12)


    plt.tight_layout()
    plt.show()
```

Fig. 22. Visualization Stage Scripts

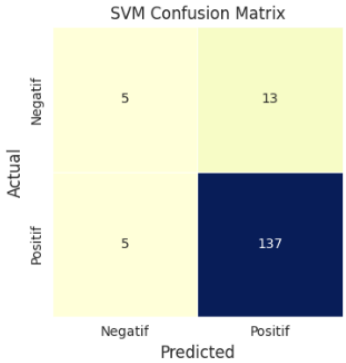


Fig. 23. Confusion Matrix Chart

The results of the visualization of the confusion matrix of the Support Vector Machine (SVM) model show that the model is quite good at classifying Positive and Negative sentiments. The model managed to predict 137 True Positive sentiments and 5 Negative sentiments. However, there were 13 Negative sentiments wrongly predicted as Positive (False Positive), and 5 Positive sentiments incorrectly predicted as Negative (False Negative). These errors are caused by factors such as language ambiguity, variations in opinion expressions, and data imbalances, where Positive sentiment is more dominant. To improve model performance, approaches such as data balancing, addition of linguistic features, and enhancement of training data can be considered.

4.7. Model Evaluation

Once the SVM model is trained, the next stage is evaluation using test data. Evaluations were conducted using the Accuracy, Precision, Recall, and F1-score metrics to measure the model's performance in classifying sentiment. In addition, the confusion matrix is used to see the prediction distribution and compare the results of the correct and incorrect classification in each class.

```
import matplotlib.pyplot as plt

accuracies = {model: result['accuracy'] for model, result in results.items()}
print("Accuracy Model SVM : ", accuracies)

Accuracy Model SVM :  {'SVM': 0.8875}

from IPython.display import display

for model_name, result in results.items():
    print(f"\nClassification Report for {model_name}:")

    report_df = pd.DataFrame(result['classification_report']).transpose()

    styled_df = report_df.style.background_gradient(cmap="coolwarm")
    styled_df = styled_df.format(precision=3)
    display(styled_df)
```

Fig. 24. Model Evaluation Script

	precision	recall	f1-score	support
Negatif	0.500	0.278	0.357	18.000
Positif	0.913	0.965	0.938	142.000
accuracy	0.887	0.887	0.887	0.887
macro avg	0.707	0.621	0.648	160.000
weighted avg	0.867	0.887	0.873	160.000

Fig. 25. Accuracy, Precision, Recall, F1-Score

Based on the results of the SVM model test, the Negative class showed low performance with a precision value of 0.500, a recall of 0.278, and an f1-score of 0.357 from a total of 18 data. This indicates that the model is less able to recognize the Negative sentiments that are a minority class. In contrast, the performance in the Positive class was much better with a precision of 0.913, a recall of 0.965, and an f1-score of 0.938 from 142 data, indicating excellent classification ability in the majority class. Overall, the model achieved an accuracy value of 0.887. The macro average section produces a precision of 0.707, a recall of 0.621, and an f1-score of 0.648, which indicates an imbalance in performance between classes. Meanwhile, the weighted average produced a precision value of 0.867, a recall of 0.887, and an f1-score of 0.873 because it was influenced by the dominance of the Positive class. In general, the model is quite effective in identifying Positive sentiment, but it is not optimal for Negative sentiment due to the imbalance of data distribution. Some approaches such as data balancing or cost-sensitive learning can be used to improve performance in minority classes.

5. Conclusion

Based on public sentiment analysis research on TikTok related to the Tour De EnTeTe event, the SVM algorithm showed excellent performance in classifying public opinion into positive and negative sentiments. Of the initial 1,000 entries, as many as 800 net data were used after the preprocessing process. The test results showed the dominance of positive sentiment, where 137 data were correctly classified as positive and 5 data as negative. The prediction errors that emerged (13 false positives and 5 false negatives) were influenced by language ambiguity, sarcasm, and class distribution imbalances. Nonetheless, SVM has been proven to be able to handle high-dimensional text data stably and accurately. Visualization through word cloud and confusion matrix also affirms the public's enthusiasm for the implementation of Tour De EnTeTe in East Nusa Tenggara, which can be a reference for local governments in the development of sports tourism in the future.

Acknowledgments

The author expresses his gratitude to Wira Wacana Sumba Christian University and the lecturers who have guided and provided knowledge during the process of compiling this work. The author also expressed his gratitude to parents for their support and prayers, as well as to friends who have helped and provided motivation.

References

[1] A. F. Habibah, "The Era of the Information Society as the Impact of New Media," *Journal of Business Technology and Information*, vol. 3, no. 2, pp.

350–363, 2021.
[2] C. Juditha, "Analysis and Public Opinion Polarization of the Phenomenon of 'Dark Indonesia' in Social Media: Sentiment Analysis and Public Opinion Polarization," *Journal of Communication*, pp. 157–170, 2025.
[3] H. Aulia *et al.*, "Mentality on Twitter Social Media Using," *Journal of Information Systems and Technology*, vol. 10, no. 2, pp. 75–81, 2024.
[4] I. S. K. Idris, Y. A. Mustofa, and I. A. Salihi, "Sentiment Analysis on the Use of the Shopee Application Using the Support Vector Machine (SVM) Algorithm," *Jambura Journal of Electrical and Electronics Engineering*, vol. 5, pp. 32–35, 2023.
[5] A. S. Ritonga and E. S. Purwaningsih, "Application of Support Vector Machine (Svm) Method in Classification of Smaw (Shield Metal Arc Welding) Welding Quality," *Edutic Scientific Journal*, vol. 5, no. 1, pp. 17–25, 2018.
[6] N. Syukerti and A. I. Mulyadi, "Social Media as a Media for Shifting Social Interaction for Adolescents," *Balayudha Communication Science*, vol. 2, no. 2, pp. 1–10, 2022.
[7] A. Setiadarma and M. Si, "The Relationship of Public Opinion and Public Relations," *Communal Science 2021*, vol. XXVI, no. 3, pp. 214–226, 2021.
[8] P. Arsi and R. Waluyo, "Sentiment Analysis On The Discussion Of Relocating I Ndonesia's Capital City Using The Support Vector Machine (SVM)," *Journal of Information Technology and Computer Science (JTIIK)*, vol. 8, no. 1, pp. 147–156, 2021, doi: 10.25126/jtiik.202183944.
[9] Muammar Khadapi, & Pakpahan, V. M. (2024). Analisis Sentimen Berbasis Jaringan LSTM dan BERT terhadap Diskusi Twitter tentang Pemilu 2024. *JUKI : Jurnal Komputer Dan Informatika*, *6*(2), 130–137. Retrieved from https://ioinformatic.org/index.php/JUKI/article/view/681
[10] A. Wijoyo *et al.*, "Machine Learning," *OKTAL : Journal of Computer Science and Science*, vol. 3, no. 2, pp. 375–380, 2024.
[11] M. I. Mubarok and M. Abdi, "Implementation of Natural Language Processing in the Design of Chatbot Applications in Fikti Umsu," *JATI (Student Journal of Informatics Engineering)*, vol. 8, no. 6, pp. 11992–12001, 2024.
[12] A. Wibowo and D. Laraswati, "Comparison of Decision Tree, Random Forest and SVM Algorithms for COVID-19 Prognosis," *IMTechno: Journal of Industrial Management and Technology*, vol. 5, no. 2, pp. 10–15, 2024.