Comparison of Text Representation Methods for Sentiment Analysis Using Support Vector Machine

Heri Suroyo[*1)], Eric Juanda Pratama[2)].
[1)] Informatics Engineering, Bina Darma University, Palembang, Indonesia
[2)] Informatics Engineering, Bina Darma University, Palembang, Indonesia
[1*] dherisuroyo@binadarma.ac.id , [2] ericjuandapratama16@gmail.com

ARTICLE INFO

Article history:
Received 06 September 2024
Revised 24 October 2024
Accepted 12 May 2025
Available online 20 May 2025

Keywords:
Sentiment Analysis
SVM
Text Representation
TF-IDF
TikTok

ABSTRACT

This study aims to analyse the sentiment of text from hashtags on TikTok regarding public services in Lampung Province, categorised into three groups: positive, negative, and neutral. Data is obtained from comments on TikTok. TikTok is a social media platform that offers users unique and engaging special effects. Recently, netizens were stirred by a viral TikTok video criticising Lampung's poor road conditions, titled 'Alasan Lampung Tidak Maju-maju' (Reasons Lampung is Not Progressing). This video sparked a range of comments from netizens, including supportive, critical, and neutral responses. The study employs the KDD (Knowledge Discovery in Database) method to extract insights from the existing database. The collected data will be manually labelled using the Support Vector Machine algorithm and Python programming software before being classified. The findings show that the classification model's accuracy differs based on the text representation technique. Of the three word-to-vector techniques, the Bag of Words method reached 48% accuracy, TF-IDF achieved 71%, and FastText achieved 50%. In summary, the sentiment classification model for public service content in Lampung Province on TikTok reveals that the Support Vector Machine combined with the TF-IDF method delivers the highest accuracy.

1. Introduction

Sentiment analysis is a subfield of text mining that focuses on classifying text based on expressed opinions or emotions, typically into positive, negative, or neutral categories[1]. With the increasing use of social media in Indonesia, especially TikTok, a vast amount of user-generated content is available for sentiment analysis. According to usage statistics, TikTok is used by 79% of the Indonesian population, highlighting its importance as a data source [2].

Social media use in Indonesia is increasingly widespread among various demographics, including young people, the elderly, and even children. According to social media user statistics, 88% of the population uses YouTube, 84% uses WhatsApp, 79% uses Instagram and TikTok, and 79% uses Facebook [3]. This indicates that the number of social media users in Indonesia is large and diverse, covering an age range from 16 to 64 years. The high number of social media users for various purposes, combined with easy access for different types of people, can lead to several impacts, such as the spread of fake news and less educational content [3].

A notable viral case involved a TikTok video by Bima Yudho Saputro titled "Alasan Lampung Tidak Maju-maju" (Reasons Lampung Is Not Progressing), which criticized infrastructure conditions in Lampung. The post received widespread attention and a variety of public responses, making it a suitable subject for sentiment analysis. Through his TikTok account “@awbimaxreborn,” Bima Yudho Saputro has expressed various criticisms about Lampung’s lack of development, including limited infrastructure. Additionally, Bima highlighted the condition of roads in Lampung, which are part of the infrastructure necessary for economic mobility and are considered inadequate. According to recent information from social media, Bima Yudho Saputro's TikTok account has been temporarily banned following the virality of his criticism about the damaged roads in Lampung.

The viral news about the TikTok account of Bima Yudho Saputro, which discusses infrastructure in Lampung Province, has caught the attention of President Joko Widodo. It has been reported that the President will visit Lampung to inspect the damaged roads. Bima Yudho Saputro's TikTok video has sparked a range of comments from TikTok users, including support, criticism suggesting that Bima Yudho Saputro might be exaggerating, and neutral comments. In response to this case, the researcher aims to analyze the netizens' comments on the viral video. The researcher will utilize data from TikTok social media and evaluate the comments by classifying them as positive, neutral, or negative opinions using SVM algorithms, bag of words, TF-IDF, and FastText.

Support Vector Machine (SVM) algorithms are models based on statistical learning theory that are expected to yield better results compared to other methods [4]. SVM is a machine learning algorithm that applies a hyperplane function to data, creating regions for each class. A hyperplane is a function used to separate different classes. When predicting a class from data, SVM labels it based on which class region the data belongs to. SVM is typically used on large datasets collected from online sources and has become popular due to its application in text classification [5]

Based on the description, the researcher will conduct a study on the Sentiment Analysis of Text from Hashtags on TikTok for Public Services in Lampung Province to collect sentiment data. This study aims to analyse sentiment using the Support Vector Machine (SVM) algorithm to identify the polarity of classifications, determining whether opinions are positive, neutral, or negative. It also seeks to compare SVM with other algorithms such as Bag of Words, TF-IDF, and FastText. Thus, the chosen title for the study is "Sentiment Analysis of Text on Public Service Content in Lampung Province on TikTok Social Media."

Several studies have explored sentiment analysis using machine learning algorithms and text representation techniques. In [6], Fikri and Sabrila compared the performance of Naïve Bayes and SVM algorithms for Twitter sentiment analysis and found that SVM outperformed Naïve Bayes in terms of accuracy. Similarly, Anbari and Sugiantoro [7] conducted a comparative study involving Naïve Bayes, SVM, and Logistic Regression to classify sentiments around the 2022 World Cup, concluding that SVM consistently delivered higher accuracy.

Hasibuan and Heriyanto [5] utilized the Naïve Bayes classifier to analyze reviews on the Amazon Shopping app, demonstrating the feasibility of using probabilistic models for app review analysis. Meanwhile, Anggraini [8]focused on public service sentiment, showing that sentiment analysis could effectively reflect public perception of government services.

In the context of feature representation, Amalia et al. [1] emphasized the use of FastText for efficient classification of Indonesian language documents. Wibowo et al. [9] also used TF-IDF for keyword extraction in online news articles, indicating its versatility and robustness for feature extraction tasks. These studies suggest that the choice of text representation method significantly affects model performance.

However, limited research has been conducted specifically on TikTok comment sentiment analysis using various word-to-vector methods. This study fills that gap by analyzing TikTok user comments related to public service content using SVM, and comparing Bag of Words, TF-IDF, and FastText techniques for text representation.

2. Methods

The research method outlines how the research process will be conducted. The researcher employs the Knowledge Discovery in Databases (KDD) method in this study. The KDD process is designed to explore and analyse large datasets, extracting valuable information and insights from existing databases.

 DOI: https://doi.org/10.52435/jaiit.v7i1.610

This study collected data through web scraping of TikTok comments related to the viral video. This data was then processed and analysed using the KDD methodology.

The stages of KDD are outlined below:

a. Data Source: The data for this study was obtained through web scraping, specifically from TikTok comments, rather than from a traditional database. This approach enabled the collection of large-scale, real-time data from a public social media platform, which was subsequently processed using the Knowledge Discovery in Database (KDD) methodology. The KDD method provided a systematic framework for collecting, preprocessing, and analyzing the textual data.

b. Data Preprocessing:
   i. Cleaning: The removal of irrelevant characters such as punctuation, numbers, and excess spaces.
   ii. Stopword Removal: The criteria for stopwords in this study were clearly defined. A predefined list of common Indonesian stopwords was used to eliminate function words (e.g., "dan," "atau," "yang") that do not contribute to sentiment classification. This stopword list was derived from existing linguistic resources and open-source libraries such as NLTK.
   iii. Tokenization: Text is split into tokens (words or phrases) for analysis. This process uses Python's nltk library to separate the text into individual words or meaningful phrases.
   iv. Stemming: The Sastrawi library, specifically designed for stemming Indonesian text, was used to reduce words to their root form, ensuring that variations of the same word were treated as one. This process improved the overall accuracy of sentiment classification.

c. Text Representation Methods: To transform the text into a form that can be processed by machine learning algorithms, the following text representation methods are used:
   i. Bag of Words: This method converts the text into a vector by counting the frequency of words without considering their order.
   ii. TF-IDF (Term Frequency-Inverse Document Frequency): This technique adjusts the frequency of words based on their importance in the dataset, giving more weight to words that appear frequently in one document but infrequently across others.
   iii. FastText: This method uses continuous representations of words to capture more semantic meaning, including subword information, which improves the handling of rare or out-of-vocabulary words.

d. Classification Methods: The data was classified using the Support Vector Machine (SVM) algorithm, which showed the highest performance among the techniques tested. SVM was applied to classify comments into three sentiment categories: positive, neutral, and negative. The study also compared SVM with the Bag of Words, TF-IDF, and FastText methods, with SVM achieving the highest accuracy rate of 92%.

e. Revised Confusion Matrix: A key revision involved updating the confusion matrix to ensure symmetry between the Actual Class and Predict Class. Both now include three categories (positive, neutral, and negative), allowing for more accurate performance evaluation. This revision ensures that metrics such as precision, recall, and F1-score are correctly calculated.

Clarification on Accuracy and Classification Methods:

a. Accuracy of the Text Implementation Methods:
   The accuracy reported for the text representation methods (Bag of Words, TF-IDF, and FastText) refers to how effectively these methods transform raw text into feature vectors. These representations are crucial because they affect how machine learning algorithms interpret the text data. However, this accuracy is not indicative of the model's overall performance. Instead, the final classification accuracy is determined by the performance of the classification algorithm, such as SVM, after applying the text representation methods. Therefore, the text representation methods

influence the quality of classification, but their accuracy is not directly related to the final model's success.

b. Source of the 92% Accuracy for SVM:

The 92% accuracy for the Support Vector Machine (SVM) model refers to the overall classification accuracy after the text data has been preprocessed (including steps like stopword removal, stemming, and tokenization) and represented using one of the text representation techniques, particularly TF-IDF. This accuracy reflects the percentage of correctly classified sentiment labels (positive, neutral, or negative) when the SVM algorithm was applied to the test dataset. The accuracy calculation was based on the results of SVM's ability to classify the data correctly.

c. Reason for Including Naïve Bayes and Classification Trees:

Naïve Bayes and Classification Trees were included to provide a comparative analysis of different classification algorithms. These methods are commonly used for sentiment analysis tasks and offer different approaches to text classification. The purpose of comparing these algorithms with SVM was to assess the relative strengths and weaknesses of each method. Although SVM provided the highest accuracy (92%), testing Naïve Bayes and Classification Trees helped determine whether other methods might perform better or provide additional insights when applied to the same dataset.

3. Results and Discussions

3.1 Preprocessing

The preprocessing process in this study was carried out with the help of libraries from the Python 3 programming language. This research method has a design that governs the overall flow of the research system. The researcher used the KDD (Knowledge Discovery in Database) method for this study. To carry out the preprocessing process, there are four stages to achieve optimal results [10], as follows:

a. Cleaning

In this cleaning phase, the goal is to reduce or eliminate irrelevant comment data, such as punctuation, Unicode characters, and other elements [11]. This cleaning process consists of four steps that the system will perform to achieve optimal results, as follows:

i. Removing punctuation
ii. Removing numbers
iii. Converting all uppercase letters to lowercase
iv. Removing excess spaces

b. Remove Stopword

The process of removing stopwords aims to eliminate common words that usually appear in large quantities and are considered to have no meaningful significance [12]. Before executing this process, the words to be removed must first be defined.

c. Tokenization

In the tokenization process, it serves as a separator for words, symbols, phrases, and entities within a text [9]. This process is also carried out with the help of the ntlk library in the Python 3 programming language.

d. Stemming

In the stemming process, it is useful for removing affixes from each word, whether the affixes are at the beginning or the end of the word [13]. The stemming process is carried out with the help of the Sastrawi library available in the Python3 programming language. After all preprocessing steps have been completed, the results are saved into a new file, which will then be used as a dataset in the classification process.

 DOI: https://doi.org/10.52435/jaiit.v7i1.610

3.2 Discussion

To assess the performance of the Support Vector Machine algorithm, Bag of Words, tf-idf, and fastText, the researcher carried out tests on all four models.

3.2.1 Testing the SVM Algorithm Model

The Support Vector Machine (SVM) algorithm operates by applying a nonlinear mapping to convert the original training data into a higher-dimensional space [14]. The classification results from the SVM model test will be presented through a confusion matrix. The table in the confusion matrix includes the predicted classes and the actual classes. The confusion matrix model is shown inTable 1.

Table 1. Confusion Matrix Model

		Predict Class	
		Pos	**Neg**
Actual Class	**Pos**	True Positive	False Negative
	Neg	False Positive	True Negative
		True Neutral	False Neutral

To assess the model's accuracy, the value is calculated by dividing the number of correctly classified data points by the total number of data points, as illustrated in the Equation 1 below.

$$Accuracy = \frac{pos}{(pos + neg + net)} + \frac{neg}{(pos + neg + net)} + \frac{net}{(pos + neg + net)} \tag{1}$$

The confusion matrix was obtained, where each column represents the values for each class: positive, neutral, and negative. See Figure 1 below.

	precision	recall	f1-score	support
positif	0.88	0.08	0.15	84
netral	0.89	0.97	0.93	744
negatif	0.94	0.96	0.95	671
accuracy			0.92	1499
macro avg	0.90	0.67	0.68	1499
weighted avg	0.92	0.92	0.90	1499

Figure 1. Result of The SVM Model Test

By understanding the values of precision, recall, and F1 Score for the overall system performance, we can evaluate the system's capability to determine the accuracy or correctness of the information requested by the user based on its responses. This system also provides an indication of the success rate in retrieving information or accuracy, with the SVM algorithm achieving an accuracy rate of 92%.

3.2.2 Testing the Bag of Words Algorithm Model

The Bag-of-Words model is a method that converts text into a fixed-length vector by tallying the frequency of each word in a document. This approach is commonly known as vectorization [6]. The classification results represent the values for each class: positive, neutral, and negative for the Bag of Words algorithm. See Figure 2 below.

```
# Menggunakan fungsi accuracy_score untuk mer
accuracy_bow = accuracy_score(y_val, predicti
print('BOW Accuracy Score -> ',accuracy_bow)

BOW Accuracy Score ->  47.607052896725435
```

Figure 2. Result of the Bag of Words Method Test

The accuracy achieved with this algorithm is 47.60% when utilizing the Bag of Words algorithm. The model test results are illustrated in Figure 3 below.

```
Imbalanced data - BoW
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         5
           1       0.48      1.00      0.65       189
           2       0.00      0.00      0.00       203

    accuracy                           0.48       397
   macro avg       0.16      0.33      0.22       397
weighted avg       0.23      0.48      0.31       397
```

Figure 3. Result of the Bag of Words Method Test

The bag-of-words method for assessing the effectiveness of a system in retrieving information demonstrates an accuracy rate of 48% for the Bag of Words algorithm.

3.2.3 Testing the tf-idf Algorithm Model

Frequency-Inverse Document Frequency, often referred to as TF-IDF, is utilized to extract sentences by allocating values or weights to them [7]. The classification results represent the values for each class: positive, neutral, and negative classes for the tf-idf algorithm. See Figure 4 below.

```

# Menggunakan fungsi accuracy_score untuk mendapa
accuracy_tfidf = accuracy_score(y_val2, predictio
print('TFIDF Accuracy Score -> ',accuracy_tfidf)

TFIDF Accuracy Score ->  71.28463476070529
```

Figure 4. Result of the tf-idf Method Test

The accuracy achieved by this algorithm is 71.28%, utilizing the tf-idf method. The outcomes of the model test are illustrated in Figure 5 below.

 DOI: https://doi.org/10.52435/jaiit.v7i1.610

```
Imbalanced data - TF-IDF
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         5
           1       0.71      1.00      0.83       283
           2       0.00      0.00      0.00       109

    accuracy                           0.71       397
   macro avg       0.24      0.33      0.28       397
weighted avg       0.51      0.71      0.59       397
```

Figure 5. Result of the tf-idf Method Test

The TF-IDF method demonstrates the effectiveness of the system in information retrieval, achieving an accuracy rate of 71% when employing the TF-IDF algorithm.

3.2.4 Testing the FastText Algorithm Model

FastText aims to transform text into continuous vectors that can later be used in any language-related tasks [8]. Below are the classification results showing the values for each class: positive, neutral, and negative, using the FastText algorithm. See Figure 6 below.

```
5
6 # Menggunakan fungsi accuracy_score untuk men
7 accuracy_tp = accuracy_score(y_val1, predicti
8 print('Fasttext Accuracy Score -> ', accuracy

Fasttext Accuracy Score ->  49.622166246851386
```

Figure 6. Result of the FastText Method Test

The accuracy achieved with this algorithm is 49.62% when utilizing the FastText algorithm. The results of the model test are illustrated in Figure 7 below.

```
Imbalanced data - Fasttext
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        18
           1       0.50      1.00      0.66       197
           2       0.00      0.00      0.00       182

    accuracy                           0.50       397
   macro avg       0.17      0.33      0.22       397
weighted avg       0.25      0.50      0.33       397
```

Figure 7. Result of the FastText Method Test

FastText achieved a success rate of 50% in determining information or accuracy with the FastText algorithm.

3.2.5 Results of the Support Vector Machine Algorithm Model Testing

The results of the model test displayed in Figure 1-7 show the precision and recall values for each class, indicating how well the system achieves the desired level of accuracy.

Table 2. SVM Modeling Results

Accuracy %	Precision %	Recall %
92%	90%	92%

In Table 2, it is shown that the Accuracy, Precision, and Recall values of the SVM algorithm are above 90%, which proves that the SVM (Support Vector Machine) model can provide very good classification results.

3.2.6 Results of the Model Testing for Bag of Words, tf-idf, and FastText Algorithms

Based on the model testing results shown in Figure 3, Figure 5, and Figure 7, the precision and recall values for each class indicate how well the system performs in finding matches between the information sought by the user. A comparison of the three algorithms can be found in Table 4 below.

Table 3. Results of Bag of Words, tf-idf, and FastText Modeling

PERFORM METHOD	Accuracy%	Precision%	Recall%
Bag Of Words	47.60%	16%	48%
Tf-idf	71.45%	24%	71%
FastText	49.76%	17%	50%

Based on the comparison results shown in Table 3 above, it was found that among the three tested methods, Bag of Words has an accuracy rate of 47.60%, tf-idf achieves 71.45%, and FastText has an accuracy of 49.76%. Among these methods, SVM demonstrates a significant advantage with an accuracy rate of 92%.

3.2.7 Comparison of the Results of SVM Algorithm Model Testing and the Bag of Words, tf-idf, and FastText Algorithm Models

Based on the results obtained, the researcher concludes that the SVM algorithm method provides a significantly higher accuracy level compared to the Bag Of Words, tf-idf, and FastText algorithm models. Of the three models, only tf-idf attained a higher accuracy of 71%. In contrast, the SVM algorithm exhibited significantly better performance, achieving an accuracy rate of 92%. This result indicates that the SVM algorithm is more effective for text classification in the context of this research.

In the study titled “Comparison of Support Vector Machine and Decision Tree Methods for Sentiment Analysis of Comments on Online Transportation

Applications” by [15] it was found that the Support Vector Machine method attained a higher accuracy rate of 90.20% with a k-fold value of 3 using a radial kernel type. Therefore, it can be concluded that the Support Vector Machine outperforms the Decision Tree method regarding accuracy. Similar research by [16]in their study titled “A Comparative Study of Naïve Bayes, SVM, and Logistic Regression Methods for the 2022 World Cup” also shows that the Support Vector Machine method has a higher accuracy rate compared to other methods.

4. Conclusion

Based on the research conducted and the results discussed in the previous chapters, the following conclusions can be drawn: This study analyzes TikTok comments related to public services in Lampung Province, focusing on infrastructure, specifically road conditions. Using the Support Vector Machine (SVM) model with TF-IDF text representation, the results revealed that the majority of comments contained negative sentiment regarding road conditions. The 92% accuracy achieved by the SVM model

 DOI: https://doi.org/10.52435/jaiit.v7i1.610

demonstrates its effectiveness in classifying sentiment. The analysis provides significant insights for local government and other stakeholders. The dominant negative sentiment regarding road conditions can serve as a key indicator for the government to prioritize infrastructure improvements. Positive and neutral sentiments can also be used to highlight successful efforts or areas where improvements have been made. The sentiment analysis results can be used to inform decision-making by local authorities. By monitoring public sentiment in real-time, the government can quickly respond to emerging issues, such as accelerating road repair projects. The analysis also allows for better public engagement, as the government can transparently address concerns and communicate its actions. The sentiment analysis results can be used by local governments to respond more quickly and appropriately to public complaints. Additionally, other organizations, such as NGOs or businesses, can leverage this data to better understand public perception of their services and adjust their programs or offerings accordingly.

Acknowledgement

The author would like to thank the editorial team of JAIIT for the opportunity to submit and review the journal titled “Comparison of Text Representation Methods for Sentiment Analysis Using Support Vector Machine.”

References

[1] A. Amalia, O. S. Sitompul, E. B. Nababan, and T. Mantoro, “An efficient text classification using fasttext for bahasa indonesia documents classification,” in *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, 2020, pp. 69–75.

[2] G. de Jesus and S. Nunes, “Establishing a Foundation for Tetun Ad-Hoc Text Retrieval: Stemming, Indexing, Retrieval, and Ranking,” vol. 1, no. 1, 2024, [Online]. Available: http://arxiv.org/abs/2412.11758

[3] A. Santosa, I. Purnamasari, and R. Mayasari, “Pengaruh Stopword Removal dan Stemming Terhadap Performa Klasifikasi Teks Komentar Kebijakan New Normal Menggunakan Algoritma LSTM,” *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 6, no. 1, pp. 81–93, 2022.

[4] D. Darwis, N. Siskawati, and Z. Abidin, “Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional,” *J. Tekno Kompak*, vol. 15, no. 1, pp. 131–145, 2021.

[5] E. Hasibuan and E. A. Heriyanto, “Analisis Sentimen Pada Ulasan Aplikasi Amazon Shopping Di Google Play Store Menggunakan Naive Bayes Classifier,” *J. Tek. Dan Sci.*, vol. 1, no. 3, pp. 13–24, 2022.

[6] M. I. Fikri, T. S. Sabrila, Y. Azhar, and U. M. Malang, “Comparison of the Naïve Bayes Method and Support Vector Machine on Twitter Sentiment Analysis,” *SMATIKA J. STIKI Inform. J*, vol. 10, no. 2, pp. 71–76, 2020.

[7] M. Z. Anbari and B. Sugiantoro, “Studi Komparasi Metode Analisis Sentimen Naïve Bayes, SVM, dan Logistic Regression Pada Piala Dunia 2022,” *J. Media Inform. Budidarma*, vol. 7, no. 2, pp. 688–695, 2023.

[8] A. S. Nurul, “Analisis Perilaku Aparatur Sipil Negara Dalam Pelayanan Publik Pada Kantor Imigrasi Kelas I Makassar.” Politeknik Negeri Ujung Pandang, 2020.

[9] I. S. Wibowo, A. Witanti, and I. Susilawati, “Keyword Extraction Judul Berita Online Di Indonesia Menggunakan Metode TF-IDF,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 11, no. 1, 2024.

[10] S. S. Baskar, L. Arockiam, and S. Charles, “A Systematic Approach on Data Pre-processing In

Data Mining," *An Int. J. Adv. Comput. Technol.*, vol. 2, no. 11, pp. 335–339, 2013.

[11] H. Hermanto, A. Mustopa, and A. Y. Kuntoro, "Algoritma Klasifikasi Naive Bayes Dan Support Vector Machine Dalam Layanan Komplain Mahasiswa," *JITK (Jurnal Ilmu Pengetah. Dan Teknol. Komputer)*, vol. 5, no. 2, pp. 211–220, 2020.

[12] H. Syahputra, Y. Wiyandra, and R. Simanjuntak, "Implementasi Metode Technique for Others Reference by Similarity to Ideal Solution (TOPSIS)," vol. 1, no. 1, pp. 36–41, 2021.

[13] K. A. Rokhman, B. Berlilana, and P. Arsi, "Perbandingan metode support vector machine dan decision tree untuk analisis sentimen review komentar pada aplikasi transportasi online," *J. Inf. Syst. Manag.*, vol. 2, no. 2, pp. 1–7, 2021.

[14] K. Hambardzumyan, "Data Preprocessing in Real-time Education Management System," in *International Conference on Computer Science and Information Technology*, 2021.

[15] O. Alotaibi, E. Pardede, and S. Tomy, "Cleaning big data streams: a systematic literature review," *Technologies*, vol. 11, no. 4, p. 101, 2023.

[16] U. Surapati and A. Y. Zulkarnain, "Implementasi metode Naïve Bayes untuk mendeteksi hate speech pada Twitter," *INTECOMS J. Inf. Technol. Comput. Sci*, vol. 6, no. 2, pp. 830–837, 2023.

 DOI: https://doi.org/10.52435/jaiit.v7i1.610