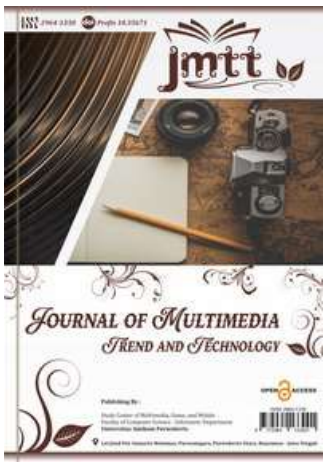
Journal of Multimedia Trend and Technology - JMTT
Volume 4, Issue 3, December 2025, ISSN 2964-1330
https://journal.educollabs.org/index.php/jmtt/

Classification of Hate Speech in TikTok Social Media Comments Using Naive Bayes Algorithm and TF-IDF Weighting

Putri Febi Utami [1*], Dwi Krisbiantoro [2], Irfan Santiko [3], Andi Dwi Riyanto [4]

[1,3] Departement Of Informatic, Faculty of Computer Science, Universitas Amikom Purwokerto, Indonesia.
[2,4] Departement Of Information System University, Faculty of Computer Science, Universitas Amikom Purwokerto, Indonesia.

ARTICLE INFO



History :

Submit on 27 September 2025
Review on 10 October 2025
Accepted on 12 November 2025

Keyword :

Hate speech;
TikTok;
Multinomial Naive Bayes;
TF-IDF;
Text classification

ABSTRACT

This research focuses on the classification of hate speech in Indonesian Tik Tok comments. Tik Tok, as a social media platform with high interaction intensity, generates a large volume of comments with diverse linguistic characteristics, including the use of formal and informal language. This linguistic variation poses challenges in the content moderation process, particularly in automatically identifying hate speech. The research dataset is secondary data obtained by combining public datasets and scraped Tik Tok comments, with an initial total of 5,698 comments. The collected data represent general user comments with variations in formal and informal language. To improve data quality, pre-processing stages were carried out including text cleaning, tokenization, normalization, stop-word removal, and stemming. After pre-processing, 4,542 comments were obtained that were suitable for use in the modeling process. Experimental results show that the Multinomial Naïve Bayes model with TF-IDF weighting is able to classify hate speech with high performance. Model accuracy reached 93% before parameter optimization and increased to 95% after hyperparameter tuning with an alpha value of 0.5. The confusion matrix results show a relatively low misclassification rate, although the class distribution in the dataset still shows imbalance. The findings of this study indicate that the Multinomial Naïve Bayes approach is effective in recognizing linguistic patterns of hate speech in Indonesian TikTok comments, including text with informal language characteristics.

***Corresponding Author:**

Putri Febi Utami
Departement Of Informatic, Faculty of Computer Science, Universitas Amikom Purwokerto, Letjend. Pol. Sumarto, Purwanegara, Banyumas, Indonesia.
Email: febi.utami@gmail.com

INTRODUCTION

The development of social media has shaped digital communication patterns with a very high level of interaction. TikTok is one of the platforms with the highest adoption rate in Indonesia. The We Are Social (2024) report shows that Indonesia is among the countries with the largest number of TikTok users in the world [1]. This high user activity has a direct impact on the volume of textual content produced, particularly in the comments section [1][2].

The comments section represents an open interaction space that allows for the free exchange of opinions. This phenomenon increases the potential for the emergence of negative content, including hate speech [3]. Hate speech is defined as a form of expression that attacks individuals or groups based on specific attributes, such as ethnicity, religion, race, or other social characteristics [4]. The existence of this type of content has the potential to cause social, psychological, and legal impacts in the digital ecosystem.

The problem of hate speech is becoming increasingly complex due to the characteristics of social media, which allows for the rapid and widespread dissemination of information. Manual detection faces scalability limitations, making automated, computationally based approaches urgently needed. In this context, the field of Natural Language Processing (NLP) plays a crucial role in the processing and analysis of textual data [5][6]. Content distribution on TikTok is controlled by the dynamic and personalized algorithmic mechanism of the For You Page (FYP). This characteristic results in highly heterogeneous variations in topics, contexts, and interaction patterns in the comments section [7][8]. In this study, the context of observation is TikTok content on the theme of infidelity, chosen as a representation of relational discourse with a high intensity of user interaction and expression. The implication of this characteristic is that the hate speech that appears is not always tied to a specific issue or theme, but can appear in various types of content [9]. This situation positions hate speech detection on TikTok as a general text classification problem that is independent of a specific topic domain.

The linguistic characteristics of comments on TikTok show significant differences compared to other social media platforms. The language used tends to be non-standard, dominated by abbreviations, spelling variations, and dynamic slang terms [10]. The use of slang and word symbolism is often used to disguise specific meanings [11]. This variation has the potential to degrade the performance of classification models if not addressed through systematic preprocessing stages, particularly text normalization [12].

Machine learning-based text classification is a commonly used approach in digital content analysis. The Multinomial Naive Bayes algorithm is known for its high computational efficiency and stable performance on large-dimensional text data [13]. At the feature representation stage, the Term Frequency–Inverse Document Frequency (TF-IDF) method is widely used because it can improve the quality of numeric text representation by emphasizing informative words [5][14].

Several previous studies have examined negative content detection using a combination of Naive Bayes and TF-IDF, particularly on the Twitter platform and online news corpuses [15][16]. The characteristics of TikTok data show significant differences in language structure and user interaction patterns. These differences indicate a research gap that requires testing classification models in the context of Indonesian-language TikTok comments.

Classification model performance is influenced by the parameter configuration used. Using default parameters does not always produce optimal performance [17]. The parameter optimization process using GridSearchCV allows for systematic identification of the best parameter combination, allowing for objective and measurable model evaluation [18]. Hate speech, which is a comment containing linguistic expressions containing insults, harassment, or verbal attacks targeting individuals or groups [19]. This phenomenon is a serious concern in the digital ecosystem due to the characteristics of social media which allows for the rapid and massive dissemination of information. Previous research has shown that public interaction on social media contributes to the increased potential for the spread of negative content, including hate speech [20].

Based on the description, this study focuses on the classification of hate speech in Indonesian TikTok comments using the Multinomial Naive Bayes algorithm with TF-IDF weighting and parameter optimization through GridSearchCV.

METHOD

A. Propose Research

This research uses a quantitative approach with an experimental method in a computing laboratory environment. The quantitative approach was chosen to measure the algorithm's performance objectively through statistical evaluation metrics. The experimental method was applied to test the performance of the text classification model using the Multinomial Naïve Bayes algorithm. This research is a study of science and technology in the field of Informatics Engineering with a focus on computational experiments and data analysis. The research is planned to be carried out for 5 (five) months, starting from September 2025 to January 2026.

B. System Flowchart

This research concept is designed as a systematic framework to ensure the entire research process is structured, logical, and can be scientifically evaluated. The research framework emphasizes the stages of textual data analysis, modeling, and quantitative evaluation of classification model performance. All stages of the process, from data collection to model performance evaluation, are illustrated in Figure 1 below:

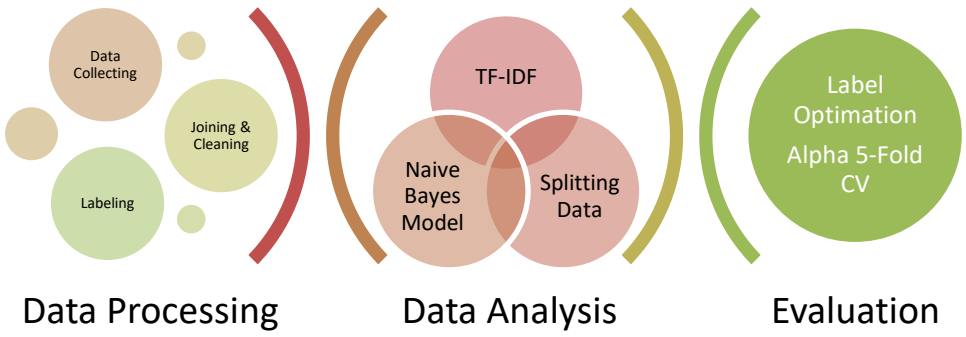


Figure 1. System Flowchart

Figure 1 illustrates the systematic research flow, starting with the TikTok comment data acquisition process, followed by text preprocessing, feature extraction and representation using TF-IDF, the division of training and test data, the classification process using the Multinomial Naïve Bayes algorithm, parameter optimization using GridSearchCV, and model performance evaluation. This flow is designed to ensure that each stage of data processing and modeling is carried out in a structured and consistent manner. The research stages are as follows:

1. Research Problem Identification

This stage aims to define the main research problem, namely the high frequency of hate speech in TikTok's comment section, which has the potential to trigger social conflict in the digital space. This phenomenon shows that social media, especially TikTok, functions not only as a means of entertainment, but also as a public interaction space that is vulnerable to the spread of negative content. The high intensity of user interaction on TikTok increases the potential for aggressive and provocative comments [21]. Therefore, this research is directed at building an automatic detection model based on Naïve Bayes to support the content moderation process more efficiently and objectively compared to manual approaches.

2. Data Characteristic Analysis

The data analysis phase aims to obtain an initial overview of the characteristics of the TikTok comment dataset. This phase explores all TikTok comment data, which is categorized as secondary data, whether obtained through scraping or public datasets. Using data from digital platforms with massive user bases, such as social media, allows researchers to obtain a more objective representation of language phenomena and user interaction patterns in the digital space [22].

Data exploration is conducted to identify data structure, language variation, and potential problems within the dataset, such as the use of informal language, abbreviations, slang, and other text noise. Furthermore, this phase also includes analysis of data quantity, class distribution, and attribute consistency that will be used in the classification process.

3. Data Pre-Processing and Preparation

This stage is crucial in text classification research, as it transforms the raw data into a format ready for processing by machine learning algorithms. Given that the data used comes from unstructured social media comments, a series of systematic processes are required to improve the dataset's quality. The processes carried out at this stage are:

a. Labeling Criteria (Keyword-based)
b. Data Preprocessing
c. Feature Extraction (TF-IDF)
d. Data Splitting

4. Classification Model Development

In the modeling phase, this study applied the Multinomial Naïve Bayes algorithm as the primary classification method. The selection of this variant was based on the characteristics of the text data, which were represented in numerical features through TF-IDF weighting. Multinomial Naïve Bayes is designed to handle discrete features that represent the distribution of words within a document, making it suitable for text classification tasks.

To achieve optimal model performance, a hyperparameter optimization process was performed using the GridSearchCV method. This process aims to determine the optimal parameter values that can improve model stability and accuracy [23].

5. Model Performance Evaluation

The evaluation phase aims to measure the model's effectiveness in recognizing hate speech by comparing the model's predictions with the actual labels (gold standard). Model performance is measured using a Confusion Matrix, which summarizes classification results into four main categories:

a. TP (True Positive): The number of hate speech data items correctly predicted as hate speech by the model.
b. TN (True Negative): The number of non-hate speech data items correctly predicted as non-hate speech by the model.
c. FP (False Positive): The number of non-hate speech data items incorrectly predicted as hate speech (Type I Error).
d. FN (False Negative): The number of hate speech data items incorrectly predicted as non-hate speech (Type II Error).

RESULT AND DISCUSSION

A. Data Collecting

The data collection phase of this study utilized secondary data from the TikTok digital platform. Secondary data was chosen because this study did not involve direct interaction with respondents but instead utilized digital trace data in the form of user comments already available on the platform's system.

1. TikTok Scraping Data

Part of the dataset was obtained through web scraping of the comments section of TikTok videos with high engagement rates. Although the data was collected independently by the researchers, it is

still methodologically categorized as secondary data, as the data source was content already publicly available on the TikTok platform.

The data acquisition process was carried out using the Instant Data Scraper tool in the Google Chrome browser. From the scraping results, attribute selection was performed to retain only comment text relevant to the research objectives. The data was then saved in .csv format with UTF-8 encoding to maintain text consistency. Through this process, 4,187 comments were obtained.

2. Public Dataset

An additional dataset was obtained from the GitHub repository in a study titled "Indonesian TikTok Cyberbullying Comments Dataset." This dataset consists of 1,511 comments that have been labeled with the cyberbullying category [24]. This dataset was used to increase the variety of language patterns and enrich the distribution of utterances in the research corpus.

Table 1. Data Source Details

Data Resource	Collecting Model	Count of Comment
TikTok	*Scraping*	4.187
Public Agent	GitHub [24]	1.511

The use of these two secondary data sources aims to increase vocabulary diversity, enrich the characteristics of informal language, and reduce the potential bias of a single dataset.

B. Data Preprocessing

Data preprocessing is a fundamental stage in Natural Language Processing (NLP) that aims to transform raw text data into a more structured representation ready for analysis. In this study, the preprocessing stage was conducted to address the characteristics of TikTok comments, which tend to be unstructured (noisy text), including the use of slang, abbreviations, spelling variations, emojis, and other non-linguistic elements. This stage plays a crucial role in improving the quality of input data for the Multinomial Naïve Bayes algorithm, as probabilistic-based classification models are highly sensitive to inconsistencies and noise in text. Therefore, text cleaning and normalization are systematically performed to ensure that the features extracted through TF-IDF weighting represent relevant linguistic information. The first step is to import the library. This is necessary because different libraries will be used at different stages.

```
!pip install Sastrawi

import numpy as np
import pandas as pd
import re
import string
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab')
stop_words = set(stopwords.words('indonesian'))
factory = StemmerFactory()
stemmer = factory.create_stemmer()
```

Figure 2. Import Library Program Code

After importing various libraries, the next step is to load the dataset that was prepared previously.

```
df_scraping = pd.read_csv('/content/drive/MyDrive/SKRIPSI FEBI/Dataset Mentah/Dataset Scrapping Mentah.csv', sep=';', encoding='latin-1')
df_github = pd.read_csv('/content/drive/MyDrive/SKRIPSI FEBI/Dataset Mentah/Dataset-Research-GitHub.csv', sep=';', encoding='latin-1')

print("Jumlah data scraping:", len(df_scraping))
print("Jumlah data github:", len(df_github))
```

Figure 3. Load Data Program Code

This code snippet is used to load raw datasets from two different sources: web scraping of social media comments and a public dataset obtained from an online repository. Both datasets are read using the read_csv() function from the Pandas library, with semicolons as the column separator and Latin-1 encoding to ensure proper processing of Indonesian text characters without errors.

C. Data Distribution

This code separates the dataset into features and targets for supervised learning purposes, where X contains the comment text as features and y contains the comment sentiment as the target variable. Next, the dataset is split into a training set and a testing set using the train_test_split function, with 80% of the data for training and 20% for testing, with stratify=y to maintain a proportional class distribution across both subsets. This split uses random_state to produce a consistent split each time the code is run. This practice is scientifically important because it maintains model generalizability, prevents overfitting, and ensures that model evaluations on new data provide an accurate picture of performance.

After data preprocessing and labeling are complete, text visualization using a word cloud method depicts the most dominant words in comments categorized as hate speech. This visualization aims to provide an exploratory representation of vocabulary patterns frequently used in negative comments.

Figure 4. Interpretation of Hate Speech Word Cloud Visualization

Based on the provided word cloud image, several words stand out with larger size, indicating their relatively high frequency of occurrence in the dataset of comments categorized as hate speech. The most dominant words include:

1. "fefek", "fefeknya", "fe" – Repeated occurrences indicate the use of these words as specific terms or insults in negative comments.
2. "dog", "pig", "tolol", "stupid" – Common swear words that indicate the negative emotional

intensity of the user.

3. "run", "busy", "promo", "kali" – While not always vulgar, these words appear quite frequently and can represent sarcasm or situational insults.
4. "najis", "korban", "anak" – Words that carry connotations of insult or attack on a specific individual.

Overall, this word cloud displays a vocabulary pattern that tends to be emotional and offensive, with several common swear words and specific terms that may be related to the context of specific online communities. Larger word sizes indicate high frequency, making them important indicators for linguistic analysis and hate speech detection.

D. Data Distribution

The program code above is used to evaluate and visualize the distribution of the dataset after it has been split into training and testing sets in the context of supervised learning. First, the number of entries in each subset is obtained using the .shape[0] attribute of the dataframe, which represents the number of observations in the training set (X_train) and testing set (X_test). Next, the data is visualized using a pie chart through the matplotlib.pyplot library, with the labels Training Data and Test Data and percentages automatically displayed using the autopct parameter. This pie chart provides a proportional picture between the training and testing sets, making it easier for readers to understand the data distribution intuitively. The plt.axis('equal') function is used to ensure the graph is a perfect circle, thus ensuring accurate visual interpretation of the proportions. Finally, the program displays the number of data points numerically for both subsets, strengthening quantitative information that supports the analysis of the data distribution before the model training process.

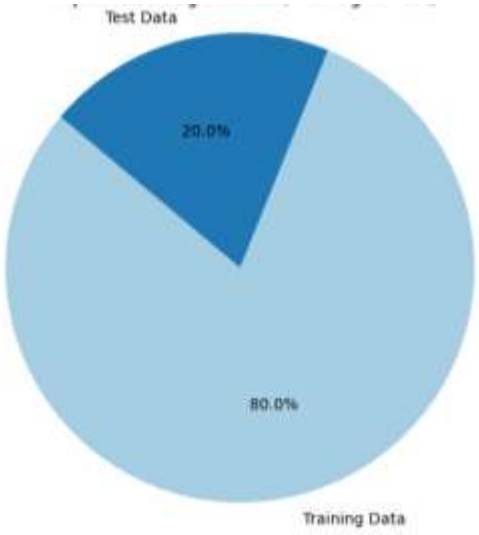


Figure 5. Data Splitting Proportion

The pie chart shows the data distribution across the training and testing sets, with 80% of the data used for training and 20% for testing. This proportion is consistent with the test_size=0.2 parameter in the train_test_split function, which is standard practice in machine learning to ensure the model has enough data to learn while still being validly evaluated. The 80:20 distribution was also chosen to ensure the training set is large enough to build a generalizable model, while the testing set is representative enough to measure the model's performance on unseen data. The use of a pie chart makes it easier for the thesis reader to visually understand the data proportions, strengthening the quantitative explanation of the dataset division before the model training process.

E. Word Weighting and Data Characteristics Analysis

Feature extraction from comment text using the TF-IDF (Term Frequency-Inverse Document Frequency) method is a common technique in Natural Language Processing (NLP) to represent text in numerical form so it can be used by machine learning algorithms.

First, the TfidfVectorizer() object from the scikit-learn library is used to build a vocabulary and calculate the TF-IDF weights of each word in the training set. The fit_transform function is applied to X_train, which transforms the comment text into a sparse matrix with dimensions (number of documents × number of unique features), where each value represents the TF-IDF weight of a word in a given document. For X_test, transform alone is used without fit to ensure that the test set representation uses the same vocabulary as the training set, maintaining model consistency and preventing data leakage. The use of apply(lambda x: ' '.join(x)) ensures that each comment entry is a single string, as TfidfVectorizer requires text input, not a list of words.

After obtaining a baseline for model performance, hyperparameter tuning was performed to improve classification performance. In the Multinomial Naive Bayes algorithm, one of the critical parameters is α (alpha), which functions as a smoothing parameter. Smoothing is used to address the problem of zero probabilities for words that do not appear in the training data, thereby improving the model's generalizability to the test data. The tuning process was performed using Grid Search Cross-Validation (GridSearchCV) with five folds (5-fold cross-validation), which evaluated combinations of alpha values: 0.1, 0.5, and 1.0. Each combination was tested against the training data to measure accuracy, and the model with the best alpha value was selected as the best estimator.

Tuning results showed that the best α value was 0.5, so the final model used this alpha to build predictions on the test data. This optimal alpha selection is expected to improve the model's performance, particularly in recognizing minority classes that previously had low recall in the initial evaluation.

F. Modeling

The initial stage of the research involved training a model using Multinomial Naive Bayes (MultinomialNB). This model was chosen for its effectiveness in frequency-based text classification, specifically with the TF-IDF (Term Frequency–Inverse Document Frequency) representation, which extracts the importance of each word in a document relative to the entire corpus.

The training process was carried out using training data, where X_train_vec acts as the feature and y_train as the label. The model learns to predict class labels based on the probability of word occurrence, assuming independence between features. After training, the model is tested on the test data X_test_vec, resulting in a prediction y_pred_before.

Table 2. Training Results Before Tuning

Class	Precision	Recall	F1-Score	Support
0	0.93	1.00	0.97	839
1	1.00	0.22	0.36	77
Accuracy	-	-	0.93	916
Macro Avg	0.97	0.61	0.66	916
Weighted Avg	0.94	0.93	0.91	916

Model performance evaluation using accuracy and classification reports showed the following results:

a. Total accuracy: 0.9345
b. Class 0 (majority): precision 0.93, recall 1.00, F1-score 0.97
c. Class 1 (minority): precision 1.00, recall 0.22, F1-score 0.36

While the overall accuracy is high (93.45%), other metrics indicate a performance imbalance between the majority and minority classes. Low recall in class 1 indicates the model tends to fail to identify most instances of the minority class. This suggests the need for hyperparameter tuning or additional strategies, such as data balancing or threshold adjustment, to improve the model's ability to detect the minority class more accurately.

Table 3. Training Results After Tuning

Class	Precision	Recall	F1-Score	Support
0	0.94	1.00	0.97	839
1	1.00	0.35	0.52	77
Accuracy	-	-	0.95	916
Macro Avg	0.97	0.68	0.75	916
Weighted Avg	0.95	0.95	0.93	916

These results show several important improvements:

a. Overall accuracy increased from 0.9345 to 0.9454, indicating a generally better model classification ability.
b. Recall for the minority class (class 1) increased from 0.22 to 0.35, making the model better able to recognize minority class instances.
c. The improvement in the minority class F1-score from 0.36 to 0.52 indicates a better balance between precision and recall after tuning.

This comparison confirms that optimizing the α hyperparameter significantly contributes to model performance, particularly in improving the detection of previously poorly predicted minority classes. This stage produces a final model ready for further prediction and analysis.

After tuning the model, a further evaluation was performed using a confusion matrix to examine the prediction distribution in more detail. The confusion matrix visualization was created using the seaborn and matplotlib libraries. To evaluate model performance in more detail, a confusion matrix analysis was performed on the test data. The confusion matrix visualizes the number of correct and incorrect predictions for each class, making it easier to identify the model's strengths and weaknesses in each category.

Based on the test results after tuning, the following values were obtained:

a. True Negative (TN) = 839
b. False Positive (FP) = 0
c. False Negative (FN) = 50
d. True Positive (TP) = 27

These values indicate that out of a total of 916 test data sets, the model was able to classify most of the data very well.

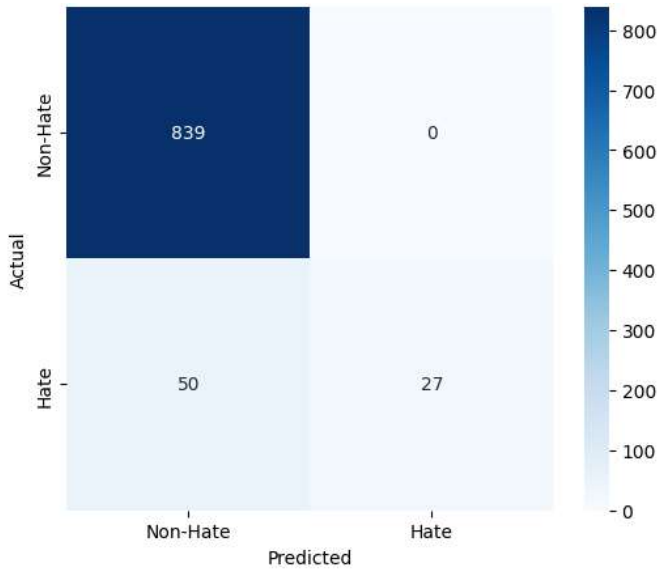


Figure 6. Confusion Matrix

Based on the Confusion Matrix visualization, the classification results can be described as follows:

a. True Negative (TN) - 839 comments, meaning non-hate speech data that were correctly predicted as non-hate speech. This indicates that the model is very effective in recognizing normal comments.
b. True Positive (TP) - 27 comments, meaning hate speech data that were correctly predicted according to their original labels.
c. False Positive (FP) - 0 comments, meaning no normal comments were incorrectly classified as hate speech. This indicates that the model did not make any false accusation errors.
d. False Negative (FN) - 50 comments, meaning hate speech comments that were still incorrectly predicted as non-hate speech. This number indicates that despite the model's improved performance, challenges remain in detecting all hate speech.

G. Evaluation

At this stage, testing was conducted on the Multinomial Naive Bayes classification model optimized using GridSearchCV with an alpha value of 0.5. This testing was not conducted on a finished system prototype, but rather through simulation of the inference function to validate the model's ability to classify new textual data in real time.

The testing process was carried out by integrating all components of the developed pipeline, including:

a. Pre-processing Stage: Input sentences were processed through text cleaning, slang normalization, tokenization, stemming using the Sastrawi library, and stopword removal.
b. Feature Transformation: Converting the cleaned text into a numeric vector using the TF-IDF (Term Frequency-Inverse Document Frequency) weighting method.
c. Model Prediction: The best model performed probability calculations to determine the final label, namely "Hate Speech (1)" or "Non-Hate Speech (0)".

Based on the series of experiments and evaluations outlined in Chapter IV, this research successfully addressed the proposed research questions.

The first research question, concerning the implementation of the Multinomial Naïve Bayes algorithm with TF-IDF weighting for hate speech classification, was addressed through the development of a systematic text processing pipeline. The dataset used in this research was entirely secondary data, obtained from online sources, without involving primary data collection.

The data labeling process used a keyword-based labeling approach, where comments were categorized based on the presence of words associated with hate speech. This approach was chosen to maintain consistency in data annotation and to adapt to the characteristics of informal language on social media.

All data then underwent text preprocessing, feature extraction using TF-IDF, and modeling using Multinomial Naïve Bayes, resulting in a binary classification system capable of classifying comments into hate speech and non-hate speech categories.

The second research question, concerning model performance, was addressed through an evaluation process on the test dataset. After parameter optimization using GridSearchCV, the Multinomial Naïve Bayes model with optimal alpha value produces the following performance:

Table 4. Model Performance Evaluation After Tuning

Class	*Precision*	*Recall*	*F1-Score*	*Support*
Non-Hate Speech (0)	0.94	1.00	0.97	839
Hate Speech (1)	1.00	0.35	0.52	77

This value is consistent with the confusion matrix, which shows:

a. True Negative (TN) = 839
b. False Positive (FP) = 0
c. False Negative (FN) = 50
d. True Positive (TP) = 27

The evaluation results indicate that the model has excellent ability to identify non-hate speech comments, as indicated by a recall value of 1.00. Furthermore, the absence of false positives indicates

that the model did not misclassify normal comments. However, the limited recall value for the hate speech class indicates that some hate speech was not detected. This condition may be influenced by language variations, semantic context, and the limitations of the keyword-based labeling approach. Through the implementation of the deployment function, the best model was proven to be able to classify new data consistently with the quantitative evaluation results. Thus, this study successfully developed and evaluated a text classification system based on TF-IDF and Multinomial Naïve Bayes for detecting hate speech in social media comments.

Based on the experimental results and analysis presented in Chapter IV, this study successfully addressed the research questions established in Chapter I. First, the application of the Multinomial Naive Bayes algorithm with Term Frequency–Inverse Document Frequency (TF-IDF) weighting proved effective in classifying Indonesian TikTok comments into hate speech and non-hate speech categories. The implementation of a pipeline encompassing text preprocessing, slang normalization, tokenization, stemming, TF-IDF feature extraction, and keyword-based automatic labeling enabled the model to capture linguistic patterns in comments characterized by complex and varied informal language.

Second, the model performance evaluation demonstrated satisfactory results. After parameter optimization using Grid-Search CV, the final model achieved an accuracy of 94.5%, with improved recall and F1-score for the previously lower-performing hate speech class. These results demonstrate that parameter optimization significantly improved the model's ability to recognize minority classes, resulting in a more balanced and reliable classification system.

Furthermore, this study provides insights into the socio-economic implications of the spread of hate speech on social media. Massive and rapidly spreading negative comments can influence public perception of products, services, and government policies. This impact can impact consumer behavior, market demand, and the prices of goods and services, potentially exacerbating inflationary pressures in the digital and e-commerce sectors. Therefore, automated detection and moderation of hate speech are not only crucial for maintaining the quality of social interactions but also relevant for economic stability and controlling inflation in society.

Overall, the results demonstrate that this study successfully developed and evaluated a text classification system based on TF-IDF and Multinomial Naive Bayes that addresses the research problem formulation and provides an understanding of the broad impact of hate speech on socio-economic dynamics. This research confirms that a systematic machine learning approach can be an effective tool in supporting content moderation and mitigating the socio-economic risks of hate speech on digital platforms.

CONCLUTIONS

Based on the research results and discussion regarding the classification of hate speech in Indonesian TikTok comments using the Multinomial Naïve Bayes algorithm with TF-IDF weighting and parameter optimization through GridSearchCV, the following conclusions were obtained:

The Multinomial Naïve Bayes algorithm with TF-IDF feature representation was successfully applied to classify TikTok comments into two categories: Non-Hate Speech (0) and Hate Speech (1). The model is capable of systematically processing unstructured text data that has undergone preprocessing.

The parameter optimization process using GridSearchCV improved model performance. After tuning, the model produced an accuracy value of 0.95, indicating an improvement compared to the model before optimization.

The model performed very well in the Non-Hate Speech class, with a precision of 0.94, a recall of 1.00, and an F1-score of 0.97. In the Hate Speech class, the model achieved a precision of 1.00, a recall of 0.35, and an F1-score of 0.52. These results indicate that the model has highly precise hate speech identification capabilities, although detection sensitivity is still limited.

Differences in performance between classes indicate the influence of data imbalance. The model tends to be more optimal in recognizing the majority class, while detection of the minority class remains challenging.

Overall, the combination of Multinomial Naïve Bayes, TF-IDF, and parameter optimization proved effective as a text classification approach for detecting hate speech in Indonesian TikTok comments.

Acknowlegdement

TikTok has become the fastest-growing social media platform in Indonesia, but behind its popularity lies a disturbing phenomenon of hate speech. Comments containing slut-shaming, body shaming, and moral judgments have gone viral. This study uses the Multinomial Naïve Bayes + TF-IDF approach with GridSearchCV hyperparameter tuning to classify TikTok comments from a dataset resulting from a combination of primary data collected by the author through scraping and secondary data sourced from GitHub. This research was compiled to fulfill one of the graduation requirements for the Informatics study program at Amikom University Purwokerto, as an effort to contribute academically in the fields of Natural Language Processing (NLP) and Data Science. The research process includes TikTok web scraping, normalization of Indonesian slang (the main challenge), hyperparameter tuning alpha smoothing, and model validation with F1-Score. The writing of this thesis cannot be separated from the guidance, support, and blessings of various parties.

Author Contributions

P.F.U, Project Initiation, Data Analysis, Writting, D.K., A.D.R Promotor, Design Model. I.S., Data Science Modeling.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

Not applicable.

REFERENCE

[1] We Are Social, “Digital 2024 Indonesia,” 2024.
[2] A. W. Utami and I. D. Arianto, “Perilaku Cyberbullying pada Media Sosial TikTok (Analisis Isi Kualitatif Perilaku Cyberbullying di Kolom Komentar dalam Akun TikTok @ofp24),” Ilmu Komun., vol. VII, no. 2, 2024.
[3] R. M. Yazid, F. R. Umbara, and P. N. Sabrina, “Deteksi Ujaran Kebencian dengan Metode Klasifikasi Naïve Bayes dan Metode N-Gram pada Dataset Multi-Label Twitter Berbahasa Indonesia,” vol. 2, pp. 46–52, 2022.
[4] A. Ariska and M. Kamayani, “Indonesian Journal of Computer Science,” vol. 13, no. 1, pp. 4825–4836, 2024.
[5] P. M. S. Ardinata, A. A. J. Permana, and I. N. S. W. Wijaya, “Identifikasi Dan Normalisasi Teks Slang Dengan Fasttext Pada Twitter Dalam Bahasa Indonesia,” J. Pendidik. Teknol. dan Kejuru., vol. 21, no. 1, pp. 34–44, 2024.
[6] S. Nabila, Kharisma Agustya Zahra Salsabilla, Nathania Trixie Aryanti, Vira Adhelia Andjani, Alfina Zahrah Umardi, and Eni Nurhayati, “Analisis Ujaran Kebencian dalam Kolom Komentar pada Media Sosial X, Tik Tok, dan Instagram,” SOSMANIORA J. Ilmu Sos. dan Hum., vol. 2, no. 4, pp. 645–651, 2023, doi: 10.55123/sosmaniora.v2i4.2997.
[7] J. Amalia, “Membangun Slang Dictionary Untuk Normalisasi Teks Menggunakan Pre-Trained Fasttext Model,” JSR Jar. Sist. Inf. Robot., vol. 6, no. 2, pp. 250–256, 2022, doi: 10.58486/jsr.v6i2.184.

[8] N. E. Febriyanty, M. A. Hariyadi, and C. Crysdian, "Hoax Detection News Using Naïve Bayes and Support Vector Machine Algorithm," Int. J. Adv. Data Inf. Syst., vol. 4, no. 2, pp. 191–200, 2023, doi: 10.25008/ijadis.v4i2.1306.

[9] M. R. Ningsih, "Sentiment Analysis on SocialMedia Using TF-IDF Vectorization and H2O Gradient Boosting for Student Anxiety Detection," vol. 11, no. 4, pp. 1137–1144, 2024, doi: 10.15294/sji.v11i4.20582.

[10] M. Karo Karo, R. Romia, S. Dewi, and P. M. Fadilah, "Hoax Detection on Indonesian Tweets using Naïve Bayes Classifier with TF-IDF," J. Inf. Syst. Res., vol. 4, no. 3, pp. 914–919, 2023, doi: 10.47065/josh.v4i3.3317.

[11] A. Gerliandeva, Y. Chrisnanto, and H. Ashaury, "Optimasi Klasifikasi Sentimen pada Komentar Online menggunakan Multinomial Naïve Bayes dan Ekstraksi Fitur TF-IDF serta N-grams," J. Pekommas, vol. 9, no. 2, pp. 260–272, 2024, doi: 10.56873/jpkm.v9i2.5585.

[12] Zaenal, Y. Salim, and L. Budi Ilmawan, "Buletin Sistem Informasi dan Teknologi Islam Analisis Sentimen terhadap Komentar Negatif di Media Sosial Facebook dengan Metode Klasifikasi Naïve Bayes INFORMASI ARTIKEL ABSTRAK," Bul. Sist. Inf. dan Teknol. Islam, vol. 1, no. 4, pp. 259–265, 2020.

[13] W. A. Luqyana, I. Cholissodin, and R. S. Perdana, "Analisis Sentimen Cyberbullying Pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine dengan mengimplementasikan algoritma Lexicon Based Features. Berdasarkan," J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya, vol. 2, no. 11, pp. 4704–4713, 2018.

[14] R. M. Yazid, F. R. Umbara, and P. N. Sabrina, "Deteksi Ujaran Kebencian dengan Metode Klasifikasi Naïve Bayes dan Metode N-Gram pada Dataset Multi-Label Twitter Berbahasa Indonesia," Informatics Digit. Expert, vol. 4, no. 2, pp. 46–52, 2024, doi: 10.36423/index.v4i2.894.

[15] F. T. Tinanda, H. Sujaini, and H. Nasution, "Comparison Analysis of Naive Bayes and K-Nearest Neighbor Algorithms in Classifying Language Styles in Indonesian Texts," J. Syst. Comput. Eng., vol. 6, no. 4, pp. 318–328, 2025, doi: 10.61628/jsce.v6i4.2158.

[16] K. S. Chong and N. Shah, "Comparison of Naive Bayes and SVM Classification in Grid-Search Hyperparameter Tuned and Non-Hyperparameter Tuned Healthcare Stock Market Sentiment Analysis," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 12, pp. 90–94, 2022, doi: 10.14569/IJACSA.2022.0131213.

[17] A. Prameswari, H. S. Oktaviani, T. R. Wicaksono, B. P. Leonard, S. Achmad, and R. Sutoyo, "Indonesian TikTok Cyberbullying Comments Dataset (Dataset)," IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 1–7., 2023.

[18] M. Firdaus and P. Nur Miftahur Rizki, "BIJAKAWEB: Platform Berbasis Web Untuk Deteksi Hate Speech Pada Komentar Berita Bahasa Indonesia," J. Teknol. Inf. dan Ilmu Komput., vol. 11, no. 4, pp. 939–948, 2024, doi: 10.25126/jtiik.1148719.

[19] N. T. Sri, K. Geethika, N. Kotha, and H. Kandoori, "Modified TF - IDF with Machine Learning Classifier for Hate Speech Detection on Twitter," vol. 14, no. 03, pp. 978–984, 2023.

[20] K. Andana, M. Othman, and R. Ibrahim, "Comparative analysis of text classification using naive bayes and support vector machine in detecting negative content in Indonesian twitter," Int. J. Adv. Trends Comput. Sci. Eng., vol. 8, no. 1.3 S1, pp. 356–362, 2019, doi: 10.30534/ijatcse/2019/6481.32019.

[21] Khairati and H. Putra, "Prediksi Kuantitas Penggunaan Obat pada Layanan Kesehatan Menggunakan Algoritma Backpropagation Neural Network," J. Sistim Inf. dan Teknol., vol. 4, pp. 128–135, 2022, doi: 10.37034/jsisfotek.v4i3.158.

[22] Hendera, R. Mulyani, and A. A. Iftikhar, "Evaluasi Implementasi Pelayanan Farmasi Klinis Di Puskesmas: Studi Kasus Di Kecamatan Banjarmasin Utara," J. Insa. Farm. Indones., vol. 7, no. 2, pp. 77–86, 2024, doi: 10.36387/jifi.v7i2.2104.

[23] P. Sari, Efan, and R. Syahri, "Algoritma K-Means Clustering: Sebuah Studi Literatur," J. Inform., vol. 1, pp. 1--7, 2024.

[24] P. Apriyani, A. R. Dikananda, and I. Ali, "Penerapan Algoritma K-Means dalam Klasterisasi Kasus Stunting Balita Desa Tegalwangi," Hello World J. Ilmu Komput., vol. 2, no. 1, pp. 20–33, 2023, doi: 10.56211/helloworld.v2i1.230.