Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.2, April 2026, pp. 1426~1437
e-ISSN: 2548-6861

Comparison of SVM and Random Forest for TikTok E10 Fuel Sentiment Analysis

Retno Eka Nurfirdaus [1], Mula Agung Barata [2], Ifnu Wisma Prastya [3]
[1,2,3] Informatics Engineering, Nahdlatul Ulama Sunan Giri University
retnoekanurfirdaus2804@gmail.com [1], mula.ab26@gmail.com [2], ifnuprastya@unugiri.ac.id[3]

Article Info

Article history:

Received 2026-02-01
Revised 2026-04-05
Accepted 2026-04-10

Keyword:

E10 Fuel,
Random Forest,
Sentiment Analysis,
Support Vector Machine,
TikTok.

ABSTRACT

One source of data that can be used to gauge public opinion on a specific public policy is social media. This study is to examine public opinion regarding the policy on the usage of a 10% bioethanol mix (E10) based on user comments on the TikTok platform. The sentiment analysis approach uses two classification algorithms Random Forest (RF) and Support Vector Machine (SVM). Pretreatment stages of data processing include tokenization, stemming, and lexicon-based techniques for identifying sentiment polarity. The Term Frequency–Inverse Document Frequency (TF-IDF) approach is used to extract features. The Synthetic Minority Over-sampling (SMOTE) technique was used to address class distribution imbalance in the data. Based on the test results, the accuracy achieved by the SVM and RF algorithms was 82.19% before applying SMOTE. Accuracy increased to 89.55% for SVM and 89.59% for RF after data balancing with SMOTE. Additionally, there was a more consistent improvement in precision, recall, and F1-score values. The findings of this study indicate that the use of SMOTE can improve the performance of classification models and reduce bias caused by class imbalance in sentiment analysis related to the E10 policy on social media.

I. INTRODUCTION

Social media is a digital platform that facilitates real-time communication, information sharing, and the expression of ideas. In the internet age, social media has emerged as a primary communication tool that shapes public opinion on various topics, including government regulations. One of the fastest-growing social media platforms in the world is TikTok. Short videos featuring music, visual effects, and user comments, ranging from 15 to 60 seconds in length, are one of its main attractions. Currently, Indonesia has the largest number of TikTok users in the world. TikTok can be used as a data source to analyze public opinion in real time thanks to its ability to disseminate information quickly and its high level of user interaction [1].

There has been a great deal of public discussion about government policies related to the development of renewable energy through the use of E10 fuel. Renewable energy is energy derived from natural resources that can be naturally replenished within a short period of time, such as the sun, wind, air, and biomass. In this situation, E10 fuel consists of 10% conventional gasoline and 90% bioethanol. Bioethanol is produced through the fermentation of biomass such as sugarcane, corn, or cassava [2]. In general, the purpose of E10 biofuel is to reduce dependence on fossil fuels while also reducing greenhouse gas emissions. This is made possible by its oxygen content, which allows the fuel to burn more efficiently and may even increase the octane rating. However, a decrease in energy content may also occur under certain conditions, and this must be taken into account. To enhance the country's energy security and increase the share of renewable energy in the energy mix, the development of E10 fuel is part of Indonesia's energy transition strategy [3]. Progress in Indonesia's biofuel-focused energy policy reflects a commitment to the transition toward sustainable energy, with plans to phase in E10 fuel starting in 2027. While this policy has the potential to yield benefits both environmentally and economically, its implementation has also sparked various reactions from the public, ranging from support for emission reductions to concerns regarding its impact on

vehicle engine performance, fuel efficiency, and the readiness of distribution infrastructure. Therefore, it is crucial to comprehensively understand public perceptions of this policy, including through sentiment analysis focused on social media data, to provide a more objective picture as a basis for consideration in future energy policy development.

Sentiment analysis is a method for evaluating public sentiments or opinions by classifying text into sentiment categories such as positive, negative, or neutral [4]. This approach is commonly used in social media-based research to examine public responses to certain digital laws and issues [5]. The process of sentiment analysis on social media data faces various challenges stemming from the linguistic characteristics of user-generated discourse. Abbreviations, slang, emoticons, and language mixing the combination of different languages within a single sentence are examples of non-standard language frequently encountered in social media comments. The interpretation of a text often heavily depends on context due to these linguistic variations, making it difficult for computer systems to understand. Natural Language Processing (NLP) techniques transform unstructured text data into a numerical format suitable for machine learning algorithms to address these challenges. Several text preprocessing techniques, including tokenization, case folding, stopword removal, and feature extraction using the Term Frequency Inverse Document Frequency (TF-IDF) method, are employed at this stage to reduce linguistic noise and enhance the relevance of text characteristics [6]. The quality of the final text representation is a significant factor influencing the model's ability to learn sentiment patterns more precisely.

Information regarding discussions about E10 fuel policy among TikTok users is complex because it is dynamic, multidimensional, and often involves a mix of languages. A classification method capable of handling high-dimensional feature spaces and nonlinear interactions among data points is required, as these conditions result in unstructured comment data with significant linguistic diversity. Support Vector Machines (SVMs) and Random Forests, two popular text analysis classification techniques, were used in this study. SVM is a supervised learning technique that successfully classifies high-dimensional data by identifying the optimal hyperplane to separate the data into different classes with the largest margin. By mapping the data to a higher-dimensional space using non-linear kernel functions such as the Radial Basis Function (RBF), this model is also capable of identifying patterns in sentiment distributions that cannot be linearly separated [7]. Meanwhile, Random Forest is an ensemble-based learning algorithm that produces more reliable and accurate predictions by combining multiple decision trees. This method can reduce the likelihood of overfitting and improve the model's generalization capacity, particularly on datasets with a wide variety of content such as social media comments [8]. Therefore, these two algorithms were selected because they demonstrate strong capabilities in classifying complex and unstructured text data.

Previous studies have examined various sentiment analysis algorithms and found varying results. The SVM algorithm showed the highest accuracy of 81.46% in a study of text reviews on the Google Play Store, while Naïve Bayes achieved 75.41% [9]. In addition, research on public opinion regarding COVID-19 booster vaccinations shows that the SVM algorithm has the highest AUC value (75.40%), while Naïve Bayes has the highest precision value (83.81%) [10]. Another study compared the TextBlob and Naïve Bayes public sentiment analysis algorithms. The results showed that TextBlob classified most tweets as positive, with an accuracy of 78.18%. Approximately 50.98% of comments from the TextBlob analysis showed positive sentiment, 16.01% negative, and 33.33% neutral [11]. However, studies using ID3 and KNN algorithms for classifying emotions in news articles found the highest accuracy to be 71.25% with a training and testing data ratio of 75–25% [12].

According to research, Random Forest (RF) and Support Vector Machine (SVM) methods are highly accurate in the classification process. Feature Selection Optimization in Bank Saqu Sentiment Analysis, test results show that SVM with Chi-Square achieved the highest accuracy of 93%, while Random Forest with Chi-Square obtained the best accuracy of 91% [13]. A policy study on the People's Housing Savings (Tapera) also showed similar results, with SVM obtaining 94.12%, while Random Forest (RF) obtained 91.76% [14]. In addition, BSI's analysis of BYOND app user sentiment shows SVM winning again with 93.16%, compared to Random Forest (RF) with 90.33% [15]. However, Random Forest is the most accurate medical classification for detecting early diabetes mellitus with an accuracy of 98.08% [16]. Overall, research indicates that SVM is generally better for text-based analysis than Random Forest for numerical data processing. However, the accuracy rates of both can vary depending on the data and research context, so both algorithms are still considered reliable and capable of producing accurate classifications.

Therefore, the purpose of this study is to analyze sentiment on the topic of E10 fuel using SVM and Random Forest algorithms. A comparison of the two algorithms works to classify TikTok users' sentiment towards the issue of E10 fuel. This study is expected to provide an overview of the public's perspective on bioethanol-based energy policies in Indonesia, as well as highlight the effectiveness and accuracy of each approach.

II. METHODS

To make the methodology utilized in this study more understandable, a research flowchart was produced that outlines each stage in a scientific and sequential manner [17].

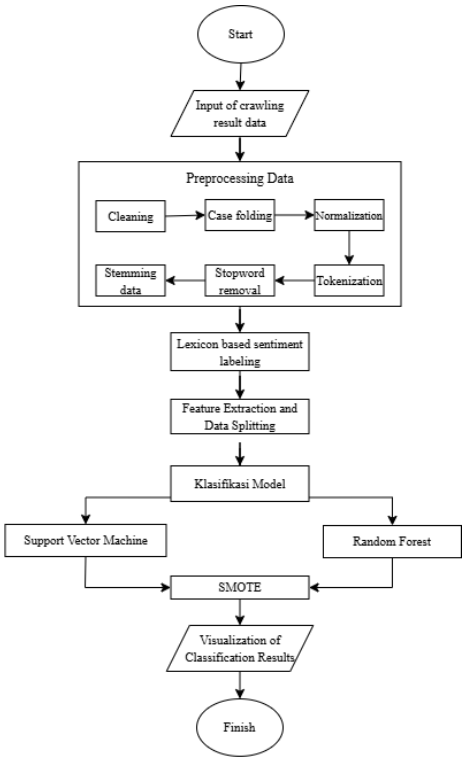

Figure 1. Research Stage

A. *Data Collection*

Every scientific study requires a very important initial stage known as data collection. This stage involves the initial gathering of relevant information to build an empirical foundation that can be used for further analysis. In this study, specific data collection was carried out using the Apify web scraping platform. Web scraping was used to obtain data from websites that could not be accessed via API, which facilitated a quick and automated data collection process [18].

The selection of TikTok videos for comment collection was conducted using purposive sampling to ensure data relevance and quality. The selected videos must explicitly discuss the E10 fuel policy whether from an informative, educational, or personal opinion perspective so videos that only briefly touch on the topic were excluded. Additionally, videos must have a minimum of 10,000 views to ensure broad reach and a diverse range of potential comments, as well as a minimum of 100 comments to ensure the data is sufficiently representative for sentiment analysis. Videos must also use Indonesian as the primary language in their narration and text. The video upload period is limited to October through November 2025, when discussions about E10 are active on Indonesian social media, to capture current public sentiment. Based on these considerations, a total of eight videos were successfully identified and selected as sources of comment data.

The data obtained from the web scraping process using the Apify platform produces a dataset in Excel format (.xlsx) consisting of 8,994 rows (comments) and 17 attribute columns. This dataset consists of 17 columns with key features such as videoWebUrl, submitVideoUrl, input, cid, createTime, createTimeIso, text, diggCount, likesByAuthor, pinnedByAuthor, repliesToId, replyCommentTotal, uid, UniqueId, avatarThumbnail, mentions, and detailMention. Of the 17 attributes available in the raw dataset, only three were used in this study: text, createTime, and createTimeISO. The text attribute is the primary attribute containing the content of user comments and serves as the primary data for sentiment analysis, as sentiment is extracted directly from the comment text. The createTime and createTimeISO attributes store information on the time the comment was created in Unix timestamp and ISO 8601 formats, which are used for temporal analysis or sentiment distribution based on time. Meanwhile, the other 14 attributes are not used because they are technical metadata or interaction information that does not directly contribute to text sentiment classification.

This data acquisition phase focuses on sourcing appropriate and relevant raw data, which is then imported into the Google Colab environment. In Colab, the data undergoes an initial cleaning process including the removal of duplicate entries to ensure that each observation is unique and to minimize the risk of bias. This enhances the accuracy of the baseline data and makes it available for further preprocessing and in-depth sentiment analysis.

B. *Data Pre-processing*

One important set of steps in Natural Language Processing (NLP) is the preprocessing of textual data [19]. At this stage, the goal is to make the raw data format more structured and informative so that it is ready for sentiment analysis [20]. Next, apply pre-processing steps designed to reduce noise, standardize word forms, and extract more relevant and meaningful text features. The following steps describe the pre-processing steps used in this study systematically:

a. Cleaning is an initial process aimed at removing non-textual elements or unnecessary characters, such as certain symbols, irrelevant punctuation, and other elements that can introduce noise into the data. This procedure is intended to ensure that the text used in the analysis contains only relevant information.
b. Case folding, converts all letters in the text to lowercase. This process is performed to standardize word forms so that words with different capitalization, such as “kata” and “kata,” can be treated as a single entity during analysis.
c. Normalization is the process of converting abbreviations, slang, and non-standard words commonly used on social media into more conventional words in accordance with language rules. This step is crucial for addressing variations in

informal language usage so that word meanings become more consistent.

d. Tokenization, is the process of breaking text into smaller units called tokens, which are typically words or phrases. This process allows the system to analyze each word individually during text processing.
e. Stopword removal, to reduce feature dimensions, improve signal-to-noise ratio, and direct the focus of analysis toward more substantive or sentiment-laden words.
f. **S**temming data, Words are converted to their base forms (root words) using the Sastrawi algorithm. For example, *"mencintai," "dicintai,"* and *"cintaan"* will be converted to *"cinta."* This step reduces word variation and allows the model to recognize the same concept even when different affixes are used.

In the next stage, the results of the preprocessing stage are used as input for the lexicon-based sentiment labeling process. Once preprocessing is complete, the dataset is thoroughly cleaned by removing duplicate data and data with missing or empty values [21]. To maintain data quality and consistency, this process involves removing rows containing NaN values and duplicates in the text and stemming_data columns. It is expected that by implementing systematic preprocessing steps, the text data will have a more structured and informative representation, which will improve the model's performance in sentiment analysis.

C. Lexicon-Based Sentiment Labeling

The labeling process is performed by analyzing each comment that has undergone stemming, specifically in the stemming_data column. Each word in the comment is compared with entries in the positive and negative lexicons. The difference between the number of words classified as positive and negative is then used to determine the sentiment score. The statement is categorized as positive sentiment if the majority of the words are positive; it is classified as negative sentiment if the majority of the words are negative. Meanwhile, if there is no significant difference or no matching words are found in the lexicon, the comment is classified as neutral. The results of this process are then stored in a DataFrame by adding a sentiment score column and a sentiment label column.

The lexicon-based and machine learning approaches in this study were gradually integrated into a single data processing workflow. The lexicon-based method is used to automatically generate initial sentiment labels (automatic labeling), which are then utilized as the target variable (y) in the machine learning model training process. Thus, classification models such as Support Vector Machine (SVM) and Random Forest not only utilize textual feature representations such as TF-IDF but also learn from sentiment patterns previously identified through the lexicon-based approach. This integration allows for the utilization of the strengths of both methods, where the lexicon approach provides a dictionary-based foundation for sentiment interpretation, while machine learning improves classification accuracy through the learning of more complex patterns from the data.

Before the model training process begins, the distribution of comments across each sentiment category is analyzed to identify potential class imbalance. This study is crucial since the classification model's performance may be impacted by an imbalanced data distribution. In order to employ data balancing approaches, like the Synthetic Minority Over-sampling Technique (SMOTE), in later phases, an initial evaluation of the class distribution is carried out.

D. Feature Extraction

Feature extraction is an important part of turning text input into a number format that machine learning algorithms can work with. The Term Frequency–Inverse Document Frequency (TF-IDF) approach was employed in this study to depict text as weighted vectors [22]. This method finds the term frequency of a word in a document and the inverse document frequency of that word throughout the whole corpus. This makes it easier to see words that are different and lessens the effect of words that are common.

TF-IDF is implemented using the *library* TfidfVectorizer with default parameters. The parameter ngram_range=(1,1) is used to represent unigrams, while the parameters min_df and max_df are not specifically limited. Additionally, default settings such as use_idf=True are retained to ensure the stability of the weighting. This process yields a vocabulary size of 7,585 unique words, which also represents the number of features in the TF-IDF matrix.

The information is then handled on data partitioning following the feature extraction procedure to obtain effective testing of the model. All the data has been categorized into two categories: training set and testing set [23]. This would generally be partitioned based on a predetermined proportion, which will be subdivided as 80 percent of the data used in the training and 20 percent used in the testing. The training set and testing set are used to train model parameters, and an objective yardstick is used to assess the predictive performance. The reason as to why such division is necessary is that the results of the evaluation are not the capacity of the model to memorise training cases but the capability of the model to generalise.

E. SMOTE

The Synthetic Minority Over-sampling Technique (SMOTE) is a data-level *resampling* method used to address class imbalance. This technique works by generating new synthetic examples through interpolation between neighboring *instances* in the feature space, specifically from

the minority class [24]. Instead of simply duplicating existing data, SMOTE generates synthetic data points based on the *k-nearest neighbor* principle for underrepresented classes, thereby effectively expanding the decision space for those classes.

The primary goal of using SMOTE is to achieve a more balanced class distribution in the training data, which in turn reduces the model's bias toward the majority class. In this implementation, SMOTE is applied to balance all classes, so that the number of instances for each class (Positive, Negative, Neutral) becomes equal to the number of instances from the original majority class (Neutral, i.e., 4033 instances). By increasing the representation of minority samples, SMOTE enables learning algorithms to capture class-specific patterns more effectively and achieve higher prediction accuracy, particularly in inherently imbalanced classification problems [25].

F. Classification Model

Machine learning is an important part of sentiment analysis since it allows us see how the model arranges data depending on what it learned during training. This study utilized two classification methodologies, Support Vector Machine (SVM) and Random Forest, to assess the efficacy of sentiment categorization on TikTok comment data [26].

A Support Vector Machine (SVM) is a supervised learning technique that identifies the optimal hyperplane for partitioning data into distinct classes while maximizing the inter-class distance [27]. The Radial Basis Function (RBF) kernel is a non-linear kernel function that moves data to a space with more dimensions. This makes it easier to see patterns that can't be separated by a straight line in their original context. The SVM implementation in this study uses the RBF kernel. We chose the RBF kernel because it can be used on text data that is difficult to read and unstructured. The input `class_weight='balanced'` is used to address class imbalance in the dataset. This ensures that all classes contribute equally during training. Note that the "C" (regularization) and "gamma" (kernel influence) parameters use the default values from the scikit-learn package and do not need to be changed at all during this process. The "C" option allows you to determine the balance between the widest margin and misclassification. The "gamma" parameter adjusts how much data points shift the decision boundary between classes. The default settings are considered sufficient for this query, so no parameters were changed. Starting with the default settings is a smart idea, but modifying them based on what is learned later can make the model significantly better [28].

Random Forest is a group learning algorithm that uses connections between various decision trees to make more accurate and robust final decisions. Each tree is created using random data and random features, so that the results obtained do not depend on a single model [29]. By performing this process, Random Forest becomes more stable and resistant to overfitting, a situation where the model fits the training data too well but does not work with new data. Random Forest has the advantage of handling large and complex data and determining which aspects most influence the prediction results. In addition to being accurate, this algorithm is also easy to interpret because the results can be explained through the importance of the variables used in the model. One of the best methods for prediction and classification tasks based on real data is prediction made through a combination of several trees that are more consistent, robust, and reliable [30].

G. Visualization of Classification Results

To better understand and compare model performance, visualization of classification results is very important [31]. The presentation of the confusion matrix starts before the SMOTE application. In simple machine learning tables, the confusion matrix is a tool that helps you see how well categorization models work. This tool looks at how the model's predictions compare to the original data. This distribution includes the number of true positives, true negatives, false positives, and false negatives [32]. Visualization of the accuracy level before SMOTE is also carried out to compare the performance between models before the data is balanced.

After applying SMOTE, label distribution visualization is used to show the effectiveness of this technique in balancing the proportion of each sentiment class. Then, the post-SMOTE confusion matrix displays changes in model performance on balanced data, which generally shows an improvement in the ability to recognize minority classes. The next step is to visualize the accuracy after SMOTE, which compares the classification results before and after data balancing.

The actual comparison chart of accuracy gives a detailed perspective on the extent to which oversampling enhances performance. Overall, this visualization framework enables the officers to directly compare the results of predictive models with the ground-truth sentiment labels, which enables gauging the ability of this model to learn and generalize the sentiment patterns. A comparison between model performance is considered with references to a number of measures, including accuracy, precision, recall, F1-score, and confusion matrix analysis [33]. To provide a more complete understanding of the performance model in multi-class sentiment classification, the following combination of metrics is required:

TABLE I
CONFUSION MATRIK

Prediction / Actual	Positive (Actual)	Negative (Actual)
Positive (Prediction)	TP (True Positive)	FP (False Positive)
Negative (Prediction)	FN (False Negative)	TN (True Negative)

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

In evaluating classification models, precision, recall, and F1-score are the main metrics used to assess model performance, especially on data with class imbalance. Precision measures the accuracy of the model's positive predictions, how many positive results are actually relevant. This metric is important when false positives must be minimized. Recall shows the model's ability to capture all actual positive data, and is very important when false negatives need to be avoided. Meanwhile, the F1-score is the harmonic mean of precision and recall, reflecting the balance between the two [34]. A high F1-score indicates that the model has consistent and balanced performance in recognizing both positive and negative classes.

To ensure the validity and generalizability of the research findings, additional validation was conducted using K-fold cross-validation. This cross-validation method is crucial for providing more robust and unbiased estimates of model performance, particularly for datasets with specific characteristics or when data size is limited. By dividing the dataset into several folds (parts) and iteratively training and testing the model on different combinations of folds, K-Fold Cross Validation effectively reduces the risk of overfitting to a specific data partition. This approach ensures that every data instance has the opportunity to be part of the test set, thereby yielding evaluation metrics that are more representative of the model's performance on unseen data in the real world. The consistency of accuracy, precision, recall, and F1-score results obtained from each fold confirms the model's stability and reliability, thereby significantly enhancing confidence in the validity of inferences drawn from these sentiment analysis results. Therefore, the application of K-Fold Cross Validation is an essential step in building a strong argument regarding the predictive capabilities of the developed model [35].

In this study, StratifiedKFold was used with the configuration n_splits=5, shuffle=True, and random_state=42. The selection of StratifiedKFold is crucial because it ensures that the class distribution proportions (Positive, Negative, Neutral) in each fold remain consistent with the class proportions in the original dataset, especially given the class imbalance. This process involves dividing the dataset into five equal folds, where in each iteration, one fold serves as the testing set and the remaining four folds as the training set. The classification model is trained and evaluated five times, and the final model accuracy is represented as the average of the accuracies obtained in each fold, providing a more comprehensive picture of the model's generalization ability on unseen data.

It is important to note that while K-Fold Cross Validation is applied to evaluate the model before SMOTE to obtain a robust performance estimate, for evaluating the model after applying SMOTE, the approach used is a single train-test split. In the post-SMOTE stage, the model is trained using the training set that has been oversampled by SMOTE, and then evaluated on the original testing set (which has not undergone SMOTE). This approach was chosen to analyze the direct impact of SMOTE on the model's ability to generalize to "real" data that has never been seen in its original proportions, rather than performing K-Fold cross-validation on an artificially balanced dataset.

III. RESULT AND DISCUSSION

A. Data Collection

The data preparation process begins by importing eight separate Excel files (from dataTiktok1.xlsx to dataTiktok8.xlsx). Using the Pandas library in Python, each file is loaded into a data structure called a DataFrame, which is essentially an individual data table for each file.

The next step is to integrate all this separate data. The pd concat function from Pandas is used to combine the eight DataFrames into one comprehensive main DataFrame. This process produces a single, neat dataset with a total of 8,994 entries, ensuring that there are no overlapping row numbers (indexes) from the original files.

Once all the data is consolidated, the focus shifts to data quality assurance. A thorough check is performed to identify two common problems: duplicate data and empty values. It is found that some columns, such as text, pinnedByAuthor, repliesToId, and replyCommentTotal, have missing values. Checking for empty values in the text column is considered crucial, as data rows without comment text will be useless for

sentiment analysis. This step is essential to ensure data integrity and cleanliness before further analysis.

Next, processing is performed on the createTimeISO column, which contains time information. After that, the DataFrame is filtered to retain only the three columns most relevant to sentiment analysis, namely date, time, and text. Other unnecessary columns are discarded to produce a lighter and more efficient DataFrame.

TABLE II
DATASET RESULTS

Tanggal	*Waktu*	*Text*
2025-10-14	08:39:48	*efek etanol*
2025-10-14	22:54:16	NaN
2025-10-10	07:55:53	*Itinil bigis*
2025-10-10	13:20:48	*setiap bahlil ngomong, kiamat maju 1jam*

B. Data Pre-processing

The text preprocessing stage involves a series of steps to clean and prepare text data before further analysis.

TABLE III
DATA PRE-PROCESSING RESULTS

Pre-processing	
Cleaning	*setiap bahlil ngomong kiamat maju jam*
Case Folding	*setiap bahlil ngomong kiamat maju jam*
Normalization	*setiap bahlil mengomong kiamat maju jam*
Tokenization	*[setiap, bahlil, mengomong, kiamat, maju, jam]*
Stopword Removal	*[bahlil, mengomong, kiamat, maju, jam]*
Stemming Data	*[bahlil, mengomong, kiamat, maju, jam]*

After that, a final cleanup was performed by removing empty and duplicate data to ensure uniqueness and data quality. As a result, the number of entries decreased from 8,994 to 8,277, indicating that the data was ready for the next stage of analysis.

C. Lexicon-Based Sentiment Labeling

This stage applies lexicon-based sentiment labeling to the previously processed text. The process begins with the use of a dictionary of positive and negative words from GitHub to calculate the frequency of each term's appearance in the stemming_data column. The difference between the number of positive and negative words produces a sentiment score, which is categorized as positive, negative, or neutral.

TABLE IV
LEXICON-BASED SENTIMENT LABELING RESULTS

stemming_data	**Score**	**label**
efek etanol	1	Positive
Negara maksud etanol normal	1	Positive
bahlil omong kiamat maju jam	-2	Negative
itinil bigis	0	Neutral
etanol bagus campur etanol ocktan mesin boros ...	1	Positive

D. Feature Extraction and Data Splitting

The pre-processed textual data, here the stemming data field, is changed into numerical feature vectors using the Term Frequency – Inverse Document Frequency (TF-IDF) Vectorizer technique. One will first run a TF-IDF vectorizer to compute the weight of frequencies of terms and inverse document frequencies, then perform a fit transform of the entire corpus. This simultaneously produces the vocabulary as well as generates TF-IDF images of individual documents. TF, IDF, and TF-IDF of the terms used by the researcher in the first three documents are performed to prove how the terms used by the researcher are applicable in this document and how the terms are distributed throughout the corpus. An increment in the values of the IDF will take the form of an increment in the terms of the number of unique terms in the corpus; however, an increment in the values of the TF-IDF will imply that the term in question is more significant in a specific document.

TABLE V
FEATURE EXTRACTION TF-IDF

Term	TF			IDF	TF-IDF		
	D1	D2	D3		D1	D2	D3
bigis	0	0	1	1.693147	0.000000	0.000000	0.707107
efek	1	0	0	1.693147	0.795961	0.000000	0.000000
etanol	1	1	0	1.287682	0.605349	0.402040	0.000000
itinil	0	0	1	1.693147	0.000000	0.000000	0.707107
maksud	0	1	0	1.693147	0.000000	0.528635	0.000000
negara	0	1	0	1.693147	0.000000	0.528635	0.000000
normal	0	1	0	1.693147	0.000000	0.528635	0.000000

Once features are extracted, they are separated into two parts so that the model can be trained and tested. The input features (X) and the target variable (y) are the TF-IDF vectors and sentiment labels of Positive, Negative, and Neutral respectively. To divide the data, the train-test split option of the scikit-learn library will be deployed, in which the sample of 80 percent of the information is utilized to conduct the training, and the remaining 20 percent is utilized to conduct the testing. A random state value is set at 42 to ensure that the experiment is reproducible. The result of such a procedure is 6,621 training cases and 1,656 test cases.

E. VISUALIZATION OF CLASSIFICATION RESULTS

The first classification step was performed with the original data set and no class balancing methods. In this case, the Neutral sentiment class showed obvious superiority over the Positive and Negative classes. This type of imbalanced allocation augments the chances of the classification models to concentrate on the majority class to a greater degree at the expense of the minority sentiment classification sensitivity and predictive accuracy.

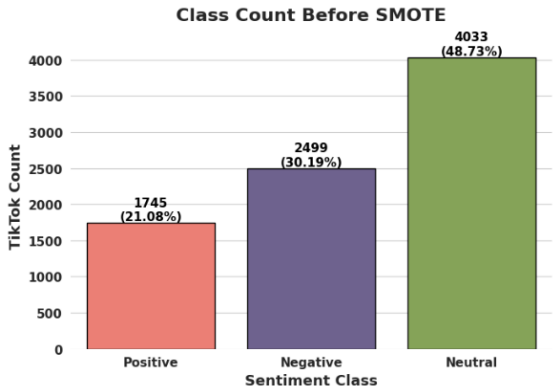

Figure 2. Classification Results Before SMOTE

In order to overcome this drawback, a data balancing method, the Synthetic Minority Over-sampling Technique (SMOTE), was proposed. The SMOTE will solve the problem of class imbalance by creating artificial samples of minor classes without altering the initial label of the classes. After its application, the sentiment label distribution was made homogenous, and 4,033 instances were assigned to each sentiment category- Positive, Negative, and Neutral. Such a balanced setup varied the majority class bias significantly, improved the capacity of the model to identify all sentiment classes, and increased stabilization along evaluation measures, such as precision, recall, and F1-score.

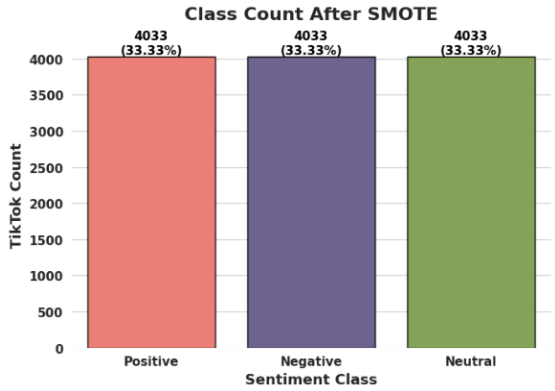

Figure 3. Classification Results After SMOTE

Next, feature representations based on TF-IDF and sentiment labels as the target variable were used in subsequent modeling, beginning with the partitioning of the data into training and testing sets. Before conducting an in-depth evaluation on a single test set, K-Fold Cross Validation was performed using StratifiedKFold (n_splits=5, shuffle=True, random_state=42) to obtain a more *robust* and reliable estimate of model performance on unbalanced data.

TABLE VI
AVERAGE ACCURACY OF K-FOLD CROSS-VALIDATION

Model	Accuracy Per Fold					Average Accuracy
SVM	0.7941	0.8194	0.7988	0.7994	0.8036	0.8031
Random Forest	0.7941	0.8146	0.7982	0.8060	0.8042	0.8034

The K-fold results show an average accuracy of approximately 0.8031 for SVM and 0.8034 for Random Forest, confirming the initial consistency of both models. Next, the Support Vector Machine (SVM) and Random Forest classification models were trained and evaluated using metrics such as accuracy, precision, *recall*, F1-score, and the *Confusion Matrix*. Although the overall accuracy on this unbalanced dataset is fairly moderate, a detailed analysis of the classification report and the *Confusion Matrix* reveals a strong bias toward the "Neutral" category (the majority class), while 'Positive' and "Negative" sentiments (the minority classes) are not optimally identified. This phenomenon, often evident in low *recall* values for minority classes, critically highlights the need for data balancing methods such as SMOTE to improve the fairness and classification power of the model across all sentiment categories.

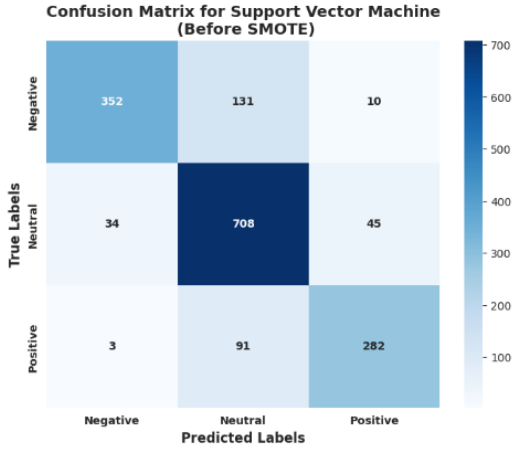

Figure 4. Confusion Matrix for Support Vector Machine Before SMOTE

The overall accuracy of the SVM for imbalanced data is approximately 81.04%, while the recall and accuracy of the model for the "negative" class are 0.71 and 0.90, respectively. The high precision (0.71) indicates that most "Negative" predictions are correct, but the lower recall means the model still misses many actual "Negative" comments, often misclassifying them into other classes (primarily "Neutral"). The "Neutral" class performs well and reflects the model's tendency to predict the most common class with a precision of 0.76 and a recall of 0.90. The "Positive" class has adequate performance with a precision of 0.84 and a recall of 0.75; however, the lower recall compared to the "Neutral" class indicates that the model struggles to identify all actual "Positive" instances. Due to inconsistent performance across classes, the F1 score of 0.79–0.82 reflects lower recall for minority classes.

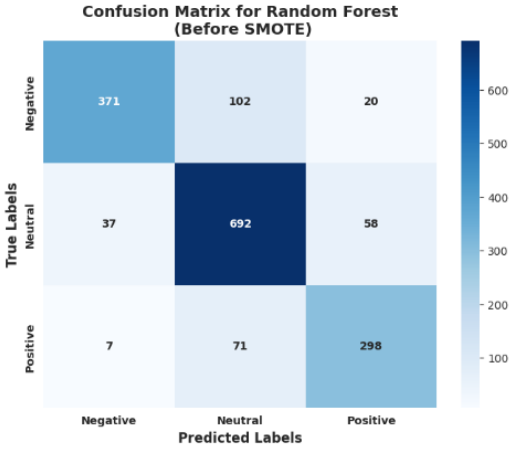

Figure 5. Confusion Matrix for Random Forest Before SMOTE

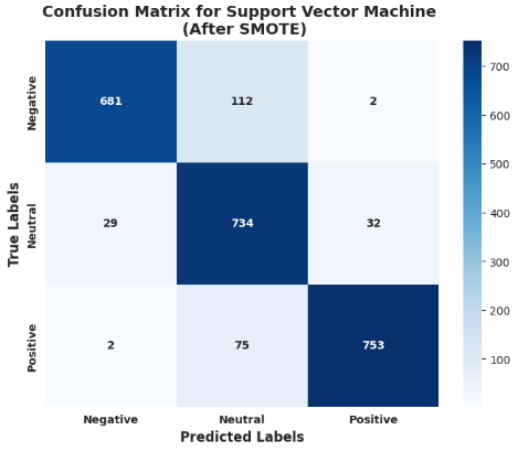

Figure 6. Confusion Matrix for Support Vector Machine After SMOTE

The Random Forest model also showed similar performance before SMOTE, with an accuracy of approximately 82.19%. For the "Negative" class, the model's precision and recall were 0.89 and 0.75, respectively. Just like the SVM, the high precision indicates that negative predictions are reliable, but the low recall highlights how difficult it is to detect every negative instance. With a precision of 0.80 and a recall of 0.88, the "Neutral" class once again demonstrates this model's exceptional ability to categorize the majority class. A precision of 0.79 and a recall of 0.79 for the "Positive" class indicate that there is still room for improvement in accurately identifying positive moods, even though the data for that class is balanced. With F1 scores ranging from 0.79 to 0.84, Random Forest is slightly superior to SVM overall, particularly for the "Negative" class.

TABLE VII
BEFORE SMOTE CLASSIFICATION RESULTS

Classifier	Label	Accuracy	Precision	Recall	F1-score
SVM	Negative	81.04%	0.90	0.71	0.80
	Neutral		0.76	0.90	0.82
	Positive		0.84	0.75	0.79
Random Forest	Negative	82.19%	0.89	0.75	0.82
	Neutral		0.80	0.88	0.84
	Positive		0.79	0.79	0.79

To address class imbalance in the training data, the synthetic sample over-sampling (SMOTE) technique adds synthetic samples to the minority class so that the distribution becomes balanced (4,033 samples per class). This method retrained the Random Forest and SVM models with the balanced data. The evaluation results showed that performance had improved significantly, with both models achieving an accuracy of around 0.90. SMOTE also improved the strength of the sentiment model by reducing model bias and improving classification capabilities for minority classes. This resulted in more consistent improvements in precision, recall, and F1 scores across all classes.

The SVM model performed well across all classes once the data became more balanced. With an accuracy of 0.96 and a recall of 0.86 for the negative sentiment class, the model demonstrated that its predictions for negative sentiment were highly accurate and capable of identifying the majority of actual negative data. For the Neutral class, the precision is 0.80 and the recall is 0.92, indicating that the model is highly effective at identifying neutral data, although there is a tendency for some data from other classes to be predicted as neutral. For the Positive class, the precision is 0.96 and the recall is 0.91, indicating that the SVM can predict positive sentiment with high accuracy during the process.

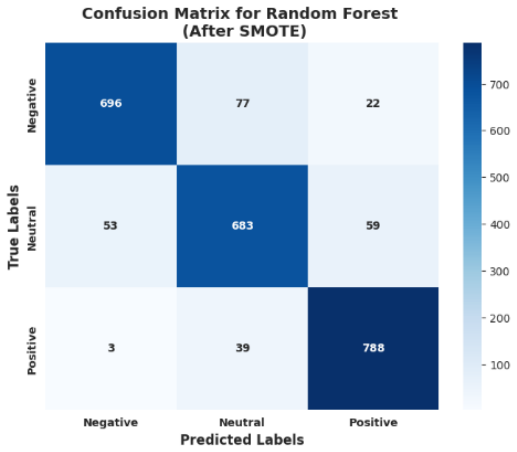

Figure 7. Confusion Matrix for Random Forest After SMOTE

The Random Forest model demonstrated exceptional classification accuracy for all classes following the application of SMOTE. For the negative class, the model's precision of 0.93 and recall of 0.88 demonstrate the model's ability to identify negative sentiment with a relatively low classification error rate. For the neutral class, the model's precision of 0.85 and recall of 0.86 indicate a good balance between prediction accuracy and the model's ability to consistently identify neutral data. For the positive class, the model's precision is 0.93 and the recall is Overall, Random Forest demonstrates excellent performance in sentiment

classification on the balanced dataset, with an F1-score consistently above 0.86 for each class.

TABLE VIII
AFTER SMOTE CLASSIFICATION RESULTS

Classifier	Label	Accuracy	Precision	Recall	F1-score
SVM	Negative	89.59%	0.96	0.86	0.90
	Neutral		0.80	0.92	0.86
	Positive		0.96	0.91	0.93
Random Forest	Negative	89.55%	0.93	0.88	0.90
	Neutral		0.85	0.86	0.86
	Positive		0.91	0.95	0.93

Based on the updated confusion matrix, the application of SMOTE to balance class distributions has improved the model's performance. The decrease in the number of False Negatives (FN) for both the positive and negative classes demonstrates this improvement. For example, the recall value for the negative class increased from 0.71 to 0.86 in the SVM model, while the recall value for the positive class increased from 0.75 to 0.91. This indicates that the data detection model has become more accurate. The number of True Positives (TP) for the minority class has also increased.

In addition, precision and F1 scores improved gradually across all classes, particularly in the minority classes. This indicates that the model not only improved detection capability (recall) but also made positive predictions more accurate.

However, error analysis shows that there are still many misclassifications between classes. As a result of the data balancing process, some increases in False Positives (FP) were observed in certain classes. Furthermore, the model continues to struggle to distinguish sentiments with ambiguous nuances, particularly in the neutral class, which is often confused with the positive or negative classes.

Overall, the confusion matrix results show that SMOTE successfully reduces bias toward the majority class and makes the model better able to identify minority classes. However, reducing classification errors caused by complex linguistic contexts in social media comment data remains challenging.

TABLE IX
COMMENTS AND MODEL CLASSIFICATION RESULTS

Original_Comment	True_Label	SVM_Predicted	RF_Predicted
situ nyuruh saya nonton supaya nambah pinter n...	Positive	Positive	Positive
ulah e goblin	Negative	Neutral	Neutral
mukanya aja ga meyakinkan jd mentri 😭🙏	Neutral	Neutral	Neutral

This table shows a comparison between the sentiment labels predicted by the Support Vector Machine (SVM) and Random Forest (RF) models on the test data, the original comments, and the actual sentiment labels obtained through a lexicon-based approach. This table enables a qualitative analysis to evaluate the performance of these models. This is done by comparing the alignment between the predicted results and the actual labels. Thus, we can immediately see how accurate the classification is for each comment. This table can also highlight prediction errors or classification errors, such as a model's tendency to classify negative sentiment as neutral or vice versa. Compared to quantitative metrics like accuracy, precision, recall, and F1 score, this provides a more contextual picture. Additionally, examining error patterns emerging from both analysis models or from a single model helps understand potential model biases. This can indicate how limited the model's capacity is in capturing specific linguistic attributes. Overall, this table helps bridge numerical evaluation with a deep understanding of the model's behavior in classifying sentiment at the individual comment level.

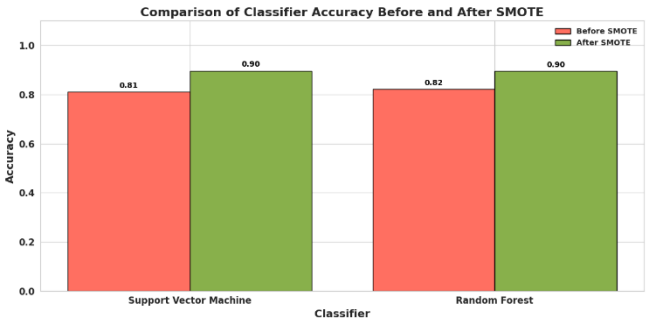

Figure 8. Comparison of Classifier Accuracy Before and After SMOTE

The test results show that the SMOTE method significantly improves accuracy. To demonstrate that the SMOTE method is effective in balancing class distributions, the Support Vector Machine (SVM) and Random Forest models each achieved an accuracy of approximately 0.90. This allows the models to balance each sentiment category and reduce bias toward the majority class. Additionally, a more proportional dataset distribution leads to more stable models during the prediction process.

Stability, complexity, and accuracy are key factors in evaluating model performance. Random Forest, a ensemble-based method that combines predictions from multiple decision trees, tends to be more stable and more robust against data noise. SVM with Radial Basis Function (RBF) kernels, on the other hand, excel at capturing complex nonlinear patterns. However, they require greater computational resources, especially during the training phase. This complexity stems from the process of finding the optimal hyperplane and transforming the data into a high-dimensional space, which is performed by the kernel function. Therefore, although both models have nearly identical accuracy levels, the improvement in performance is due to the algorithm's capabilities and the quality of the dataset distribution following the balancing process. Therefore, the factors that must be considered when selecting a model are not only accuracy but also computational efficiency and the model's stability in the face of data variations.

Nevertheless, the results of sentiment analysis must be interpreted with caution, taking into account the characteristics and quality of the dataset used. Datasets sourced from social media, such as TikTok, are potentially biased because they do not fully represent the population proportionally. User participation in online discussions is

generally influenced by factors such as age, interest in specific topics, and digital literacy levels. Additionally, comments on a post tend to come from users with a strong inclination to express opinions—whether in the form of support or criticism—which can introduce *self-selection bias* into the dataset. This situation risks causing the distribution of analyzed opinions to not fully reflect the general public's views.

Beyond demographic factors, the representativeness of the dataset is also influenced by the platform's own characteristics. TikTok's content distribution algorithm tends to prioritize content with high engagement levels, meaning the analyzed comments likely originate from groups of users with similar interests. Therefore, the sentiment analysis results in this study should be understood as a representation of opinions emerging on social media, not as an absolute picture of the entire population. Nevertheless, social media data retains strategic value as a source of information for quickly and dynamically identifying public opinion trends regarding a policy. Consequently, the interpretation of this study's results must be conducted contextually, taking into account the limitations of the dataset used.

IV. CONCLUSION

This work successfully created a sentiment analysis model for public debates on TikTok about the E10 fuel legislation by using systematic preprocessing, a lexicon-based labeling method, and a data balancing technique called Synthetic Minority Over-sampling (SMOTE). The results show that SMOTE, which used to have mostly neutral feelings, was able to fix the imbalance across classes. This improved the accuracy of the classification model. The Support Vector Machine (SVM) and Random Forest models become about 0.90% more accurate. Also, the F1 score, recall, and accuracy became more even across all sentiment categories. These results show that balancing the data is very important for improving a model's capacity to find sentiment patterns, especially for minority sentiment classes.

The findings of this sentiment study have substantial consequences for the development of energy policy. The government can use information about how people feel about the E10 policy, how much they accept it, and what their concerns are to make policies that are more flexible, responsive, and based on evidence. Also, the government can take steps to reduce policy risks by finding out about negative feelings and possible misunderstandings among the public. These steps could include improving the quality of fuel distribution, refining regulations, and coming up with better ways to communicate with the public.

Conversely, the results of this study also aid in the formulation of public education strategies. Sentiment analysis helps find gaps in information and misunderstandings about using E10 fuel, which makes it possible to create more focused and relevant instructional programs. Using social media sites like TikTok to communicate is also very important for getting information out to more people, faster, and in a way that fits the needs of the digital audience. So, it is easier to encourage the shift to sustainable energy by expanding people's understanding of energy.

However, this study has drawbacks, especially concerning the utilization of social media data, which is fluid, informal, and may not accurately reflect the full population. Consequently, subsequent research should utilize larger and more heterogeneous datasets, and explore the implementation of deep learning models, such as Transformer-based models (e.g., IndoBERT), to enhance accuracy and contextual comprehension of language. The advancement of aspect-based sentiment analysis may provide an additional method for acquiring more precise insights into policy components of popular interest.

REFERENCES

[1] D. Tewu, D. Destine, I. Gunawan, and I. M. A. Y. June, "Analysis of Social Media User Growth and Its Implications for Digital Marketing Strategies in Indonesia 2024," pp. 236–245, 2025.

[2] A. H. Sebayang, H. Ibrahim, S. Dharma, A. S. Silitonga, B. B. Ginting, and N. Damanik, "Pengaruh Campuran Bahan Bakar Pertalite-Bioetanol Biji Sorghum pada Mesin Bensin," *J. Teknosains*, vol. 9, no. 2, p. 91, Jul. 2020, doi: 10.22146/teknosains.40502.

[3] A. D. Nugroho, M. S. Alim, S. Sundari, and G. R. Soekarno, "Kebijakan Dekarbonisasi Sistem Energi Indonesia pada Sektor Energi Terbarukan," *CAKRAWALA*, vol. 17, no. 2, pp. 109–125, Dec. 2023, doi: 10.32781/cakrawala.v17i2.539.

[4] A. Yoga Pratama, G. Ananda Sanjaya, N. Khairunisa Lubis, and M. Rangga Aditya, "Analisis Sentimen Publik Terkait Danantara Menggunakan Algoritma IndoBERT pada Platform Media Sosial," vol. 9, p. 2025, doi: 10.47002/metik.v9i1.1055.

[5] A. Rustanta, S. Dwi Putranto, and P. Huang, "Maintaining the Digital Public Space: Communication Ethics and Regulatory Challenges in the TikTok Era," *J. Komun.*, vol. 17, no. 1, pp. 63–83, 2025, doi: 10.24912/jk.v17i1.32927.

[6] M. Joefitra Zaqy, L. Marlina, and R. F. Wijaya, "Analysis of Indonesian Netizen Sentiment on Platform X Regarding the Arrival of Refugees in Indonesia Using the Multinominal Naive Bayes Method," *sinkron*, vol. 8, no. 3, pp. 1945–1952, Jul. 2024, doi: 10.33395/sinkron.v8i3.13940.

[7] L. F. S. Minggow, A. V. Vitianingsih, S. Kacung, A. L. Maukar, and J. F. Rusdi, "Sentiment Analysis on Ajaib App Using the SVM Method," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 14, no. 4, pp. 551–556, Oct. 2025, doi: 10.32736/sisfokom.v14i4.2402.

[8] R. Yuranda, T. Sutabri, and D. Wahyuningsih, "Pendekatan Macine Learning dalam Evaluasi LabelBerita Berdasarkan Judul: Studi Kasus Media Online," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 12, no. 3, pp. 434–439, 2023.

[9] L. B. Ilmawan and M. A. Mude, "Perbandingan Metode Klasifikasi Support Vector Machine dan Naïve Bayes untuk Analisis Sentimen pada Ulasan Tekstual di Google Play Store," *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 154–161, Aug. 2020, doi: 10.33096/ilkom.v12i2.597.154-161.

[10] R. T. Aldisa and P. Maulana, "Analisis Sentimen Opini Masyarakat Terhadap Vaksinasi Booster COVID-19 Dengan Perbandingan Metode Naive Bayes, Decision Tree dan SVM," *Build. Informatics, Technol. Sci.*, vol. 4, no. 1, pp. 106–109, Jun. 2022, doi: 10.47065/bits.v4i1.1581.

[11] G. R. Putri, M. A. Maulana, and S. Bahri, "Perbandingan Algoritma Naïve Bayes dan TextBlob Untuk Mendapatkan Analisis Sentimen Masyarakat Pada Sosial Media," *Teknika*, vol. 13, no. 2, pp. 213–218, Jun. 2024, doi: 10.34148/teknika.v13i2.815.

[12] P. Insan and Kusrini, "Analisis Perbandingan Algoritma ID3 dan

KNN Pada Klasifikasi Emosi Teks Berita Berbahasa Indonesia," *METIK J.*, vol. 5, no. 1, pp. 36–41, Jun. 2021, doi: 10.47002/metik.v5i1.213.

[13] A. Tirta, P. Subandono, and D. Ariatmanto, "Optimalisasi Seleksi Fitur dalam Analisis Sentimen Bank Saqu: Studi Perbandingan SVM dan Random Forest Menggunakan Information Gain dan Chi-Square Optimizing Feature Selection in Sentiment Analysis of Bank Saqu: A Comparative Study of SVM and Random Fores," *Sist. J. Sist. Inf.*, vol. 14, pp. 1205–1219, 2025, [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[14] Y. Afandy, "Perbandingan SVM dan Random Forest Pada Analisis Sentimen Kebijakan Tabungan Perumahan Rakyat Berdasarkan Data Media Sosial X," vol. x, No.x, no. 2, pp. 28–36.

[15] F. Naifah Firzatullah, "Analisis Sentimen Pengguna Aplikasi Byond BSI Pada Google Play Store Menggunakan Algoritma SVM Dan Random Forest," vol. 9, p. 2025, doi: 10.47002/metik.v9i2.1089.

[16] C. Z. V. Junus, T. Tarno, and P. Kartikasari, "Klasifikasi Menggunakan Metode Support Vector Machine Dan Random Forest Untuk Deteksi Awal Risiko Diabetes Melitus," *J. Gaussian*, vol. 11, no. 3, pp. 386–396, Jan. 2023, doi: 10.14710/j.gauss.11.3.386-396.

[17] R. Astri, A. Kamal, and U. Dharma Andalas, "It-Dashboard Application To Determine the Type of Subsidized Assistance," *Res. Technol. Repub. Indones.*, vol. 17, no. 3, p. 2023, 2023, [Online]. Available: https://creativecommons.org/licenses/by/4.0/%0Ahttps://sinta3.kemdikbud.go.id/journals/profile/2143

[18] N. Adila, "Implementation of Web Scraping for Journal Data Collection on the SINTA Website," *Sinkron*, vol. 7, no. 4, pp. 2478–2485, 2022, doi: 10.33395/sinkron.v7i4.11576.

[19] S. Chaichulee, C. Promchai, T. Kaewkomon, C. Kongkamol, T. Ingviya, and P. Sangsupawanich, "Multi-label classification of symptom terms from free-text bilingual adverse drug reaction reports using natural language processing," *PLoS One*, vol. 17, no. 8 August, pp. 1–22, 2022, doi: 10.1371/journal.pone.0270595.

[20] A. N. Syafia, M. F. Hidayattullah, and W. Suteddy, "Studi Komparasi Algoritma SVM Dan Random Forest Pada Analisis Sentimen Komentar Youtube BTS," vol. 8, no. 3, 2023.

[21] T. Informatika and U. Dian, "Comparing Machine Learning Models for Sentiment Analysis of Tokopedia Reviews," vol. 9, no. 6, pp. 3642–3647, 2025.

[22] C. C. Sujadi, Y. Sibaroni, and A. F. Ihsan, "Analysis Content Type and Emotion of the Presidential Election Users Tweets using Agglomerative Hierarchical Clustering," *Sinkron*, vol. 8, no. 3, pp. 1230–1237, 2023, doi: 10.33395/sinkron.v8i3.12616.

[23] S. D. Amalia, M. A. Barata, and P. E. Yuwita, "Optimization of Random Forest Algorithm with Backward Elimination Method in Classification of Academic Stress Levels," *J. Appl. Informatics Comput.*, vol. 9, no. 3, pp. 633–641, 2025, doi: 10.30871/jaic.v9i3.9280.

[24] Bakti Putra Pamungkas, Muhammad Jauhar Vikri, and Ita Aristia Sa'ida, "Application of SMOTE-ENN Method in Data Balancing for Classification of Diabetes Health Indicators with C4.5 Algorithm," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 14, no. 2, pp. 183–188, 2025, doi: 10.32736/sisfokom.v14i2.2350.

[25] M. Mujahid *et al.*, "Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering," *J. Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00943-4.

[26] M. Fachrie, "Machine Learning for Data Classification in Indonesia Regional Elections Based on Political Parties Support," *J. Ilmu Komput. dan Inf.*, vol. 13, no. 2, pp. 89–96, 2020, doi: 10.21609/jiki.v13i2.860.

[27] E. F. Laili, Z. Alawi, R. Rohmah, and A. Barata, "Komparasi Algoritma Decision Tree Dan Support Vector Machine (SVM) Dalam Klasifikasi Serangan Jantung," *J. Sist. Inf. dan Inform.*, vol. 8, no. 1, 2025, [Online]. Available: https://www.kaggle.com/datasets/thxogg/heart-

[28] S. Dermawan, A. T. Ayunda, S. Informasi, F. Sains, and U. Pradita, "Sentiment Analysis of Coretax on Social Media X Using Naive Bayes , SVM , and LSTM for Service Improvement," vol. 9, no. 6, 2025.

[29] A. A. Yaqin, M. A. Barata, and N. Mahmudah, "Implementation of the Random Forest Algorithm with Optuna Optimization in Lung Cancer Classification," *Sistemasi*, vol. 14, no. 2, p. 561, 2025, doi: 10.32520/stmsi.v14i2.4877.

[30] W. A. Rayadhani and M. Rahardi, "Comparative Analysis of Random Forest , SVM , and Naive Bayes for Cardiovascular Disease Prediction," vol. 9, no. 6, pp. 3234–3243, 2025.

[31] R. A. Sitorus and I. Zufria, "Application of the Naïve Bayes Algorithm in Sentiment Analysis of Using the Shopee Application on the Play Store," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 15, no. 1, pp. 53–66, May 2024, doi: 10.31849/digitalzone.v15i1.19828.

[32] A. A. Ritonga, A. Amanda, and E. R. Hasibuan, "Predicting Prospective Student Interests Using the C4.5 Algorithm and Naive Bayes," *Sinkron*, vol. 9, no. 1, pp. 395–405, 2025, doi: 10.33395/sinkron.v9i1.14441.

[33] Darussalam and G. Arief, "Jurnal Resti," *Resti*, vol. 1, no. 1, pp. 19–25, 2018.

[34] Y. Aprianti, A. L. Hananto, and S. S. Hilabi, "Klasifikasi Sentimen Komentar Pengguna pada Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes Ruangguru menunjukkan inovasi dalam pendekatan Naïve Bayes menjadi alat klasifikasi teks yang populer . Penerapan Penelitian lain oleh Artanti Inez TF-IDF pen," pp. 101–110, 2025, doi: 10.47002/metik.v9i1.1023.

[35] K. Pal, "and Holdout Accuracy Estimation Methods with 5," no. Iccmc, pp. 83–87, 2020.