Journal of System & Technology
ISSN: 3110-0171
Vol. 2, No. 1, June 2026

PERFORMANCE EVALUATION OF FIVE MACHINE LEARNING ALGORITHMS WITH SMOTE FOR SENTIMENT CLASSIFICATION OF YOUTUBE COMMENTS ON PURBAYA'S POLICY

M. Darma Alfero[1]*
[1]Digital Business, Department of Business Administration, Padang State Polytechnic, Indonesia
mdalfero@gmail.com
*Corresponding author: mdalfero@gmail.com

Abstract: The evolution of public policy communication in the digital age has encouraged social media to become a deliberative space that openly reflects the dynamics of public perception and sentiment. This paper aims to conduct a comparative analysis of the performance of several machine learning algorithms in classifying the sentiment of YouTube user comments as a representation of public response to policy communication delivered by Minister Purbaya. The research was conducted using web scraping techniques with the YouTube Data API v3 on three public policy videos, with a total of 6,466 comments. The text data were processed using preprocessing, lexicon-based sentiment labeling, TF-IDF feature weighting, and class imbalance handling via the Synthetic Minority Over-sampling Technique (SMOTE). Five machine learning algorithms were applied, comprising SVM, KNN, RF, NB, and LR. Model performance was evaluated using accuracy, precision, recall, and F1-score metrics. The results showed that SVM produced the most optimal performance with an accuracy value of 0.78 and a consistent F1-score balance across all sentiment classes. These findings highlight the importance of algorithm suitability, feature representation, and data balancing strategies in developing reliable sentiment analysis models for evaluating data-driven public policy communication.

Keywords—Machine Learning Algorithms, SMOTE, Sentiment Classification, YouTube Comments, Public Policy Analysis

Article Information:
Received: 11-Feb-2026; Revised: 20-Jun-2026; Accepted: 28-Jun-2026; Published: 28-Jun-2026

This work is licensed under a **CC BY-SA**. Copyright ©2026 by Author. Published by Universitas Riau.

I. INTRODUCTION

Digitalization has brought fundamental changes to the pattern of communication between the government and the public [1]. Public policy communication, which was previously one-way through conventional media, has now evolved into interactive two-way communication through social media platforms. Social media serves not only as a means of disseminating information but also as a participatory space that allows the public to respond to, interpret, and evaluate public policy openly and in real time. Open and participatory social media serve as a strategic medium for facilitating interactions among the government, the public, and the private sector in inclusive public communication [2].

In this context, YouTube, as one of the world's largest video-sharing platforms, plays a strategic role by integrating visual elements, verbal narratives, and public interaction through its comment feature [3]. Through the comment section, the public can express their support, criticism, dissatisfaction, and expectations regarding policies and the government actors who deliver them [4]. Therefore, YouTube comments have strategic value as a source of data for empirically understanding public attitudes toward government policy communication [3].

In public policy communication, the role of state officials is crucial, as they not only act as policy implementers but also as key communicators who shape public perception [5]. Minister Purbaya is a public official who actively conveys his views, explanations, and policy narratives through various communication channels, including digital platforms such as YouTube. As a policymaker and communicator, his statements and communication style often attract public attention and trigger various responses from the community. This makes the policy communication delivered by Minister Purbaya relevant for further study from the perspective of public response and sentiment.

The public's response to policy communication is reflected in unstructured comments that use informal language, abbreviations, sarcasm, irony, and complex emotional expressions [4]. In addition, the large volume of comments makes manual analysis inefficient, subjective, and difficult to replicate. Therefore, a computational approach is needed that systematically, objectively, and consistently processes and analyzes large-scale text data to produce reliable, scientifically replicable findings. In this case, machine learning is a relevant and effective approach for classifying sentiment in text. Selecting the right algorithm is very important for determining data processing speed and the accuracy of the results [6]. In addition to algorithm selection, data labeling quality is a crucial factor in building a reliable sentiment classification model. This study uses a lexicon-based labeling approach as an objective and consistent alternative, utilizing a

sentiment dictionary to automatically determine the sentiment polarity of comments [7]. This approach enables large-scale data processing while reducing the potential for subjective bias introduced by human annotators.

Another common problem in sentiment analysis is class imbalance (also called imbalanced data) [8],[9], where one sentiment class typically dominates the others. Class imbalance can degrade classification model performance, leading machine learning algorithms to achieve high predictive accuracy in the majority class but low accuracy in the minority class. To mitigate this problem, this study adopts the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic data for the minority class, thereby making the data distribution more proportional and enabling the model to learn more representatively, without biasing toward a particular class [10].

Based on the conceptual and empirical description presented above, this study aims to conduct a comprehensive and systematic comparative analysis of the performance of several machine learning algorithms in classifying the sentiment of YouTube user comments as a proxy for public response to public policy communications delivered by Minister Purbaya. The results of this study are expected to make a substantive contribution to the development of a theoretical and methodological framework for sentiment analysis and machine learning.

II. METHODS

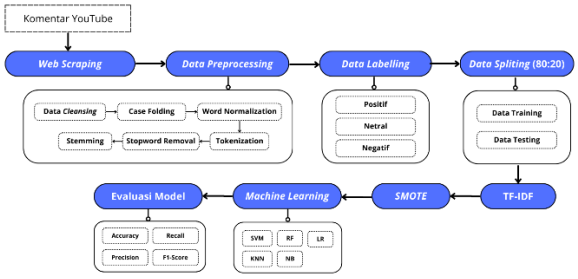

Figure 1. Research flow

In general, this research flow was designed and implemented through a series of systematic and structured methodological stages as follows:

A. Web Scraping

The data collection technique in this study used web scraping, leveraging the YouTube Data API v3 [11]. The data collection process used an API Key, the Google API Client library, and the Python programming language to access and extract comment data based on Video ID. The following YouTube videos were used as research material, as shown in Table 1.

Table 1. Research data source

Sources	Title	Upload Date	Total Data
Kompas TV	[FULL] Rapat Perdana! Menkeu Purbaya Dicecar Komisi XI DPR soal Janji Pertumbuhan Ekonomi	Sept. 10, 2025	1.225
Nusantara TV	[FULL] Dicecar habis, Purbaya Skakmat Balik DPR: Kenapa Rapat Sama Menkeu 'Dulu' Gak Disinggung?	Sept. 11, 2025	2.988
Kaltim Post	[FULL] Menkeu Purbaya Kasih Paham 100 Ekonom Indonesia Arah Kebijakan Ekonomi Nasional	Oct. 31, 2025	2.255

B. Data Preprocessing

After the data was successfully collected, the next step was data preprocessing, which consisted of several important processes [9],[12]:

1) Cleansing, aimed at removing irrelevant elements.
2) Case folding was performed to convert all letters to lowercase to ensure consistent analysis.
3) Word Normalization aims to perform lexical normalization by converting non-standard words into standard forms in accordance.
4) Tokenization, this process aims to segment text by breaking sentences into smaller lexical units, which are then represented as tokens.
5) Stopword Removal is a selection process for tokenization results with the aim of retaining words that are essential in effectively representing document content.
6) Stemming: In this stage, each word is replaced with its base form. In this study, the Stemming process uses StemmerFactory from the Sastrawi library.

C. Data Labelling

The cleaned data was then labeled using a lexicon-based approach. The lexicon-based approach identifies sentiment polarity by using a pre-compiled lexicon as a reference for matching words found in the text [13]. Through this stage, the data was classified into three sentiment categories, namely positive, neutral, and negative, which represent the polarity of users' attitudes towards the research object.

D. Data Splitting

At this stage, the dataset is split into 80:20, with 80% for training and 20% for testing [14].

E. TF-IDF Weighted

Term Frequency-Inverse Document Frequency (TF-IDF) is a feature weighting technique used to represent text data in numerical form, taking into account the frequency of a word's appearance in a document relative to its appearance across all documents [11],[15]. This method is the result of combining the Term Frequency (TF) and Ineverse Document Frequency (IDF) [14].

TF is a measure that indicates the frequency of a word's occurrence in a document [16]. Weighting using TF only takes into account the frequency of a term's accurrence in a document, so it is not yet capable of distinguishing the level of importance of each tem, therefore, IDF is applied to refine the weighting process by

assigning a lower weight to terms that appear frequently in many documenta and a higher weight to terms that appear rarely [17].

F. Aplication of SMOTE

The distribution of sentiment in the collected data shows an imbalance between classes, so to overcome this problem, the Synthetic Minority Over-sampling Technique (SMOTE) method was applied.

SMOTE is a resampling approach used to address class imbalance in a dataset, especially when the number of minority classes is much smaller than the majority classes [18]. This method generates synthetic samples from the minority class to balance the class distribution and minimize the model's bias towards the majority class [10].

G. Algorithm Implementation

The machine learning algorithms used for sentiment analysis in this study are Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), Naive Bayes (NB), and Logistic Regression (LR).

Support Vector Machine

SVM is a supervised machine learning method used for classification and regression tasks. SVM builds a predictive model that separates data points using a hyperplane [19], with the aim of maximizing the distance or margin between the hyperplane and the closest data points [9].

K-Nearest Neighbor

The KNN algorithm is a supervised machine learning algorithm typically used for classification. This algorithm has a parameter k that specifies the number of nearest neighbors. KNN works by finding the k nearest data points in the training dataset for each test data point, based on the shortest distance. After finding the k nearest neighbors, this algorithm uses the majority voting rule to determine the most frequent class, which is used as the final classification result for the test data [11].

Random Forest

RF is an ensemble method that combines multiple decision trees. This algorithm works by building multiple trees from different data samples, then determining the classification result based on the majority vote across all trees. The main advantage of RF is its ability to achieve high accuracy while minimizing the risk of overfitting [8],[12].

Naive Bayes

This method is a classification approach based on the principles of probability and statistics proposed by Thomas Bayes. This approach estimates the probability of future events using historical information or data and is therefore known as Bayes's theorem. In its application, this theorem assumes that each attribute is independent of the others [18].

Logistic Regression

LR is a machine learning method for classification, where the model predicts the class or category of data rather than continuous values, as in linear regression. LR works by calculating the probability that a sample belongs to a particular class [15].

H. Confusion Matrix

A confusion matrix is a table used to assess the performance of a classification model in machine learning. This table compares the predictions generated by the model with the actual labels (ground truth), providing a clear picture of the distribution of correct and incorrect predictions for each class. This information makes the confusion matrix an important basis for calculating evaluation metrics [20].

I. Algorithm Model Evaluation

After the data was tested using machine learning algorithms, the model was then evaluated. Model evaluation is a crucial stage in this research, assessing the classification model's ability to accurately and reliably predict classes. The evaluation process used accuracy, precision, recall, and F1-score as quantitative metrics to assess prediction accuracy, sensitivity to specific classes, and the overall balance of model performance.

Accuracy is the proportion of data correctly classified by the model [3], where the highest accuracy value is 1, and the lowest value is 0 [21]. The equation used to calculate the Accuracy value in the model is shown in Equation 1.

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

Where TP (True Positive) is the number of correct predictions for positive class, TN (True Negative) is the number of correct predictions for the negative class, FP (False Positive) indicates the number of data that are actually positive but predicted to be negative, and FN (False Negative) indicates the number of data that are actually negative but predicted to be positive. The Accuracy value reflects the extent to which the model correctly predicts the target.

Precision is used to assess the proportion of correct positive predictions compared to all positive predictions made by the model [21],[22]. The higher the Precision value obtained, the fewer errors there are in predicting the positive class [24]. Equation 2 shows how to find the Precision value in the model.

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

Recall assesses the proportion of correct positive predictions from all data that actually belong to a particular class [19],[23]. The higher the Recall value, the fewer positive data points are missed. The equation used to calculate Recall in the model is shown in Equation 3.

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

The final evaluation metric used is the F1-Score, a combination of Precision and Recall that is effective at

handling data imbalance. The higher the F1-Score value, the better the balance between Precision and Recall [25]. The equation for calculating the F1-Score value is shown in Equation 4.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

III. RESULTS

A. Data Collection

Using the web scraping method with the YouTube Data API v3, 6.466 comments were successfully collected, including Date (publishedAt), User Name (authorDisplayName), and Comment (textDisplay). All data obtained was then stored in XLSX format to facilitate further processing and analysis.

Table 2. Data collection results

Date	Username	Comments
2025-12-25 T04:59:58Z	@rediflute7421	Para Dpr gak tau dosa pak purbaya wkwkwkwk😊
2025-12-25 T04:53:59Z	@rediflute7421	baru kali ini raker langsung live media publik. mantap pak purbaya lanjutkan
2025-12-25 T03:12:04Z	@srirejeki2160	Mantap BPK Purbaya rakyat bersama bpk
2025-12-24 T06:03:44Z	@UsmanSupandi-m2z	"DPR di bikin malu di depan mata oleh pa Purbaya. Malu gak tuh DPR Ribuan pejabat di bikin meriang sma Purbaya. Purbaya tidak bisa di takut2tin"
2025-12-22 T15:39:04Z	@asdakur2338	Kalau anggota DPR hanya mikirnya Tunjangan dan Gaji sendiri dulu yang kaya dulu masyarakat yang lain Taro dulu yang penting tuk pemilu akan datang itu Benerkan kan Bapak Ibu DPR yah
...	.....	.....

B. Data Preprocessing

After Data Preprocessing, the number of comments decreased from 6.466 to 6.213, indicating the removal of redundant and irrelevant data [11]. Table 3 shows the results of the entire Data Preprocessing series.

Table 3. Data preprocessing results

Process	Comments
Data collection	Mantap BPK Purbaya rakyat bersama bpk
Cleansing	Mantap BPK Purbaya rakyat bersama bpk
Case folding	mantap bpk purbaya rakyat bersama bpk
Word normalization	mantap bapak purbaya rakyat bersama bapak
Tokenization	[mantap, bapak, purbaya, rakyat, bersama, bapak]
Stopword removal	mantap purbaya rakyat
Stemming	mantap purbaya rakyat

C. Data Labeling

In the weighting mechanism, data with a weight value greater than 1 is classified as positive sentiment, data with a weight less than 1 is categorized as negative sentiment, while data with a weight equal to 0 is designated as neutral sentiment [26]. The results of data labeling and weighting are shown in Table 4.

Table 4. Data labeling and weighting results

No	Comments	Score	Sentiment
1	dpr tau dosa purbaya wkwkwkwk	-8	Negative
2	kali raker langsung live media publik mantap purbaya lanjut	8	Positive
3	mantap purbaya rakyat	5	Positive
4	dpr bikin malu mata pa purbaya malu tuh dpr ribu jabat bikin meriang sma purbaya purbaya takuttin	-18	Negative
...	.....	...	.....
6213	langsung sks gratis ekonom manggutmanggut moga hasil sesuai ucap senang jabat optimis	10	Positive

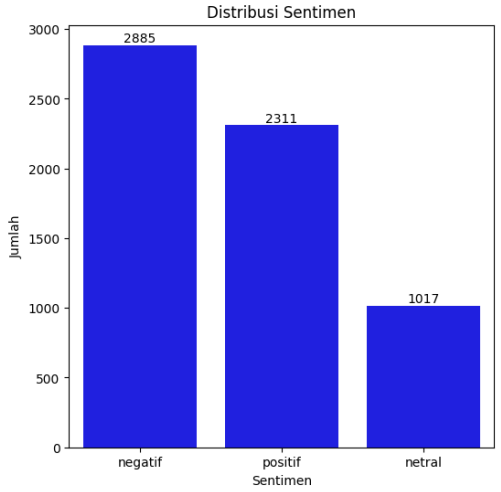

Figure 2. Sentiment distribution diagram

At this stage, three sentiment categories were produced: positive, neutral, and negative. In the processed comment data, negative sentiment dominated with 2.885 comments, followed by positive sentiment with 2.311 comments, and neutral sentiment with 1.017 comments.

D. Application of SMOTE in Algorithms

Table 5 shows the amount of data before applying SMOTE in the machine learning algorithm.

Table 5. Quantity of data before SMOTE implementation

Data Training	Data Testing	Data Label			Total
		Positive	Negative	Neutral	
80%	20%	2.308	1.849	813	4.970

Before applying SMOTE, the amount of data in each class showed significant imbalance, with the negative class dominating with 2.308 data points, followed by the positive class with 1.849 data points, while the neutral class consisted of only 813 data points. This condition can introduce bias into the model's learning process because algorithms tend to prioritize the majority class. After applying SMOTE, the amount of data became balanced, as shown in Table 6.

Table 6. Amount of data after SMOTE application

Data Training	Data Testing	Data Label			Total
		Positive	Negative	Neutral	
80%	20%	2.308	2.308	2.308	6.924

After applying SMOTE, the number of data points per class was standardized to 2.308, resulting in a proportional class distribution. This balance is expected to better reflect each class's characteristics and improve classification performance, especially for detecting minority classes.

E. Confusion Matrix

Support Vector Machine

Based on the confusion matrix model SVM with SMOTE shown in Figure 3, the model correctly classified 471 data points as negative. However, 64 data points were incorrectly predicted as neutral, and 42 were misclassified as positive. In the neutral class, the model's performance is fairly moderate, with 142 data points correctly classified. The classification errors in this class consist of 36 data points predicted as negative and 26 as positive. Meanwhile, in the positive class, the model performed well, with 353 data points correctly classified as positive. However, 69 data points were misclassified as neutral and 40 as negative.

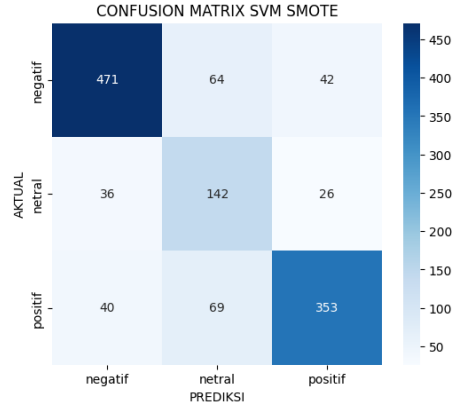

Figure 3. Confusion matrix SVM

K-Nearest Neighbor

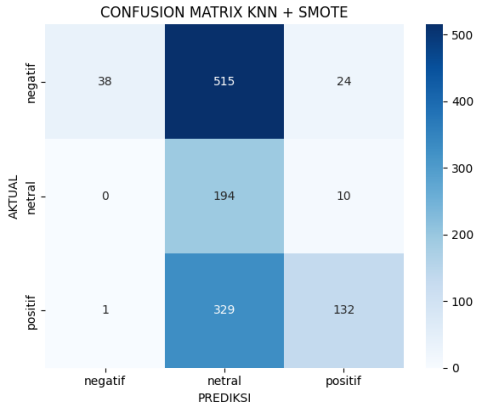

Figure 4. Confusion matrix KNN

Based on Figure 4, the model correctly classified only 38 data points as negative. Most of the data in this class (515 points) was incorrectly predicted as neutral, and 24 points were incorrectly classified as positive. In the neutral class, the model performed relatively well, with 194 data points classified correctly. No neutral data points were incorrectly predicted as negative, but 10 data points were incorrectly predicted as positive. Meanwhile, in the positive class, the model correctly identifies only 132 data points. Most of the positive data (329) were incorrectly classified as neutral, and 1 data point was incorrectly predicted as negative.

Random Forest

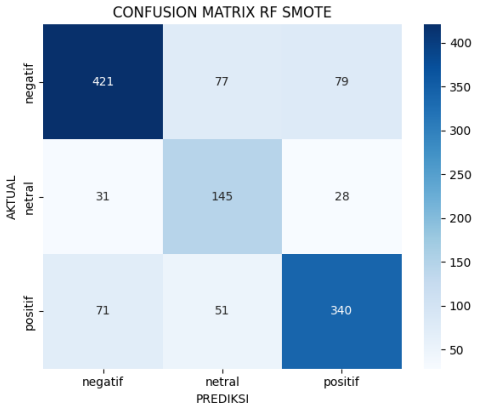

Figure 5. Confusion matrix RF

Figure 5 above shows that the model correctly classified 421 data points as the negative class. However, there were still a number of classification errors: 77 negative data points were misclassified as neutral, and 79 were misclassified as positive. In the neutral class, the model correctly identified 145 data points. However, 31 neutral data points are misclassified as negative, and 28 are predicted as positive. Meanwhile, in the positive class, the model shows relatively good performance with 340 data points correctly classified as positive. However, there are still 71 positive data points incorrectly predicted as negative and 51 misclassified as neutral.

Naive Bayes

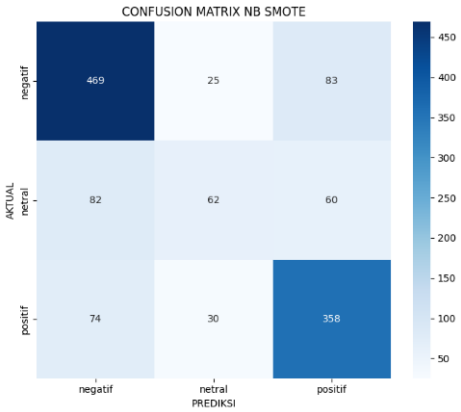

Figure 6. Confusion matrix Naive Bayes

Figure 6 is a confusion matrix generated by the Naive Bayes model with SMOTE, which shows that in the negative class, the model performs quite well, with 469 data correctly classified as negative. However, 25 data points are still incorrectly predicted as neutral, and 83 are misclassified as positive. In the neutral class, the model's performance tends to be lower than in other classes. Of all the neutral data, only 62 data points were correctly classified. Meanwhile, 82 data points were incorrectly predicted as negative, and 60 as positive. This condition shows that the neutral class is the most difficult for the Naive Bayes model to recognize, as it has relatively

ambiguous patterns and overlaps with the other two classes. Meanwhile, for the positive class, the model correctly classified 358 data points. However, 74 data points were incorrectly predicted as negative, and 30 as neutral.

Logistic Regression

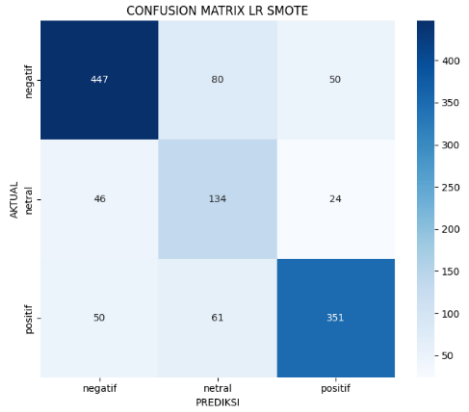

Figure 7. Confusion matrix LR

Based on the confusion matrix for Logistic Regression with SMOTE shown in Figure 7, the model correctly classified 447 data points as negative. However, 80 data points are incorrectly predicted as neutral, and 50 are incorrectly classified as positive. This shows that although the performance in the negative class is relatively good, there is still ambiguity between the negative class and the neutral and positive classes. In the neutral class, the model shows fairly balanced performance, with 134 data points successfully classified correctly. The incorrect classifications comprised 46 data points predicted as negative and 24 as positive. As for the positive class, the model showed good performance with 351 data points correctly classified as positive. However, 61 data points were still incorrectly predicted as neutral, and 50 were incorrectly classified as negative.

F. Algorithm Model Evaluation

In this study, five machine learning algorithms were implemented and compared to determine the algorithm with the highest accuracy. The performance results for each machine learning model are shown in Table 7.

Table 7. Machine learning model performance results

Models	Accuracy	Precision	Recall	F1-Score
SVM	0,78	0,80	0,78	0,78
KNN	0,29	0,78	0,29	0,26
RF	0,73	0,74	0,73	0,73
NB	0,72	0,70	0,73	0,70
LR	0,75	0,77	0,75	0,76

IV. DISCUSSION

This study began with a data collection phase using web scraping techniques via the YouTube Data API v3. This process yielded a total of 6.466 comments obtained from the three videos designated as the subjects of the study. Next, the data underwent preprocessing, reducing the number of comments to 6.213. This reduction in data volume indicates that the preprocessing successfully eleminated redundent, irrelevant, or potentially disruptive data that could compromise the quality of the analysis. The subsequent stage was data labeling, conducted usig a lexicon-based approach. This approach was chosen because the lexicon used includes thousands of words with explicitly defined sentiment polarities, along with weights representing the intensity of sentiment in each sentence. The labeling results show that negative sentiment was the most dominant category in the analyzed set of comments.

Before the modeling process began, the dataset first underwent data splitting, with 80% allocated as the training set and 20% as the testing set. The results of data splitting revealed an imbalance in class distribution, as shown in Table 5. This imbalance has the potential to reduce the model's ability to learn patterns from the minority class and cause bias toward the majority class. Therefore, this study applies the SMOTE to balance the class distribution in the training data. After the oversampling process, each class has the same number of data points, namely 2.308 samples. The balanced dataset was then used as training data for each machine learning algorithm; Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), Naive Bayes (NB) and Logistic Regression (LR) to proceed with the training process and evaluate model performance.

Based on the results of implementation into machine learning models, each model generated a confusion matrix corresponding to its analysis performance. Based on confusion matrix analysis, each machine learning algorithms exhibited distinct classification performance characteristics in identifying the three sentiment classes. SVM provided the most consistent classification performance with high prediction rates for both the negative and positive classes, as well as a relatively better ability to recognize the neutral class compared to other algorithms. RF and LR also showed fairly good performance, although a number of classification errors were still found across all classes. In contrast, KNN produced the lowest performance, as indicated by the prevalence of misclassifications into the neutral class for both the negative and positive classes. Meanwhile, NB was able to classify the negative and positive classes well but experienced a significant drop in performance for the neutral class.

Overall, the confusion matrix indicates that the neutral class is the most difficult to classify consistently across all models, even after applying SMOTE [27]. This is due to the ambiguous nature of neutral sentiment, which tends to lack strong polar keywords and often overlaps with both positive and negative expressions. On the other hand, the negative and positive classes show clearer classification patterns because they have more explicit linguistic characteristics. Thus, the interpretation of the confusion matrix not only serves as a technical evaluation tool but also provides a deeper conceptual understanding of the model's behaviour in classifying public sentiment towards policy communication.

Based on the evaluation results summarized in Table 7, Support Vector Machine showed the best overall performance with an Accuracy value of 0,78 and relatively

balanced Precision, Recall, and F1-score values. The high F1-score in the SVM model indicates that this algorithm maintains a balance between prediction accuracy (Precision) and the ability to detect all data in each class (Recall). This condition shows that SVM is not only superior in classifying the majority classes, especially the neutral class, which is semantically more ambiguous. This advantage aligns with the characteristics of SVMs, which are designed to maximize the separation margin between classes in a high-dimensional feature space, making them more robust against noise and the complexity of text data.

The Logistic Regression model ranks second with an Accuracy value of 0,75 and an F1-score of 0,76. The performance of LR, which is relatively close to that of SVM, shows that the linear approach remains quite effective at capturing the relationship patterns among features in text data. However, the Recall and F1-score values for the neutral class remain lower than those for the positive and negative classes, indicating the limitations of LR in modeling implicit sentiment expressions with weak polarity. This shows that although LR is efficient and stable, this model still has limitations in handling high semantic complexity.

Meanwhile, K-Nearest Neighbor showed the lowest performance among all algorithms tested, with an Accuracy value of 0,29 and an F1-score of 0,26. The poor performance of KNN indicates that this algorithm is less capable of handling high-dimensional text data, even after class balancing using SMOTE. KNN's reliance on distance calculations makes it highly sensitive to feature distributions and noise, leading to many data points being misclassified as the neutral class. These findings confirm that KNN is not suitable for text-based sentiment classification on large, complex datasets.

The differences in performance between models indicate that an algorithm's effectiveness depends heavily on its suitability to the data's characteristics and the feature representation used. In the context of this study, Support Vector Machine proved to be the mostoptimal algorithm for classifying YouTube comment sentiment. This finding reinforces the argument that selecting the right algorithm, combined with data balancing techniques and appropriate feature representation, is a key factor in producing a reliable sentiment analysis model that can serve as a basis for data-driven evaluation of public perception.

V. CONCLUSION

The results of a comparative analysis of five machine learning algorithms show that Support Vector Machine consistently produces the best performance with the most optimal Accuracy and F1-score, as well as a strong balance between Precision and Recall across all sentiment classes. Furthermore, the findings of this study confirm that the neutral sentiment class is the most difficult category to classify by all models tested. This phenomenon reflects the characteristics of neutral sentiment: it is latent, implicit, and often intersects with both positive and negative expressions. The difficulty in classifying neutral sentiment does not reflect solely algorithmic limitations but also the linguistic and pragmatic complexity of public communication in the digital space.

This study confirms that social media does not merely function as a channel for disseminating policy information but also as a space for public articulation that reflects the legitimacy, trust, and level of public acceptance of state actors and policy narratives.

REFERENCES

[1] R. R. E. Nainggolan, "Analisis Penggunaan Website dan Media Sosial Pemerintah untuk Pelayanan Publik," *JTKP*, vol. 6, no. 1, pp. 1–21, Jun. 2024, doi: 10.33701/jtkp.v6i1.4221.

[2] I. R. Agustina, "Implementasi Open Government Indonesia melalui Saluran Youtube Resmi Sekretariat Presiden," *JKP*, vol. 7, no. 1, pp. 134–151, Mar. 2023, doi: 10.25139/jkp.v7i1.5674.

[3] M. F. A. Hakim, S. Subhan, and P. Listiaji, "Social Media Comments For Government Institution Video Classification Using Machine Learning," *jitk*, vol. 10, no. 2, pp. 433–440, Nov. 2024, doi: 10.33480/jitk.v10i2.5187.

[4] A. S. W. Darmawan, A. Degaf, and N. F. Anggrisia, "Expressive Speech Acts and Public Sentiments in Netizen Responses to Political Posts on X," *JOLLT*, vol. 13, no. 2, p. 960, Apr. 2025, doi: 10.33394/jollt.v13i2.13167.

[5] S. Kondolele, M. I. Sultan, M. Akbar, and B. A. Putra, "The nexus between public communication and policy implementation revisited: insights from the Population and Civil Registration Service of South Sulawesi, Indonesia," *Front. Commun.*, vol. 10, p. 1556747, May 2025, doi: 10.3389/fcomm.2025.1556747.

[6] E. S. Negara *et al.*, "Sentiment Analysis Of Public Opinion On Presidential Advisory Appointments Using Naive Bayes Classification," *Jur. Ilmh. Ilm. Ter. Un. Ja*, vol. 8, no. 2, pp. 452–466, Nov. 2024, doi: 10.22437/jiituj.v8i2.35254.

[7] R. H. Muhammadi, T. G. Laksana, and A. B. Arifa, "Combination of Support Vector Machine and Lexicon-Based Algorithm in Twitter Sentiment Analysis," *khif*, vol. 8, no. 1, pp. 59–71, Mar. 2022, doi: 10.23917/khif.v8i1.15213.

[8] E. Erlin, Y. Desnelita, N. Nasution, L. Suryati, and F. Zoromi, "Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang," *matrik*, vol. 21, no. 3, pp. 677–690, Jul. 2022, doi: 10.30812/matrik.v21i3.1726.

[9] L. L. Van Fc, M. K. Anam, M. B. Firdaus, Y. Yunefri, and N. A. Rahmi, "Enhancing Machine Learning Model Performance in Addressing Class Imbalance," *CogITo Smart Journal*, vol. 10, no. 1, pp. 478–490, Jun. 2024, doi: 10.31154/cogito.v10i1.626.478-490.

[10] M. K. Anam, T. A. Fitri, A. Agustin, L. Lusiana, M. B. Firdaus, and A. T. Nurhuda, "Sentiment Analysis for Online Learning using The Lexicon-Based Method and The Support Vector Machine Algorithm," *Ilk. J. Ilm.*, vol. 15, no. 2, pp. 290–302, Aug. 2023, doi: 10.33096/ilkom.v15i2.1590.290-302.

[11] E. Susanti, M. Maimunah, and S. Nugroho, "Sentiment Analysis of YouTube Comments Using Machine Learning Models," *PIKSEL*, vol. 13, no. 1, pp. 103–114, Mar. 2025, doi: 10.33558/piksel.v13i1.10743.

[12] A. F. Anjani, D. Anggraeni, and I. M. Tirta, "Implementasi Random Forest Menggunakan SMOTE untuk Analisis Sentimen Ulasan Aplikasi Sister for Students UNEJ," *TEKNOSI*, vol. 9, no. 2, pp. 163–172, Sep. 2023, doi: 10.25077/TEKNOSI.v9i2.2023.163-172.

[13] M. A. Al-Shabi, "Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining," *International Journal of Computer Science and Network Security*, vol. 20, no. 1, 2020.

[14] R. Chandra and E. M. Sipayung, "Analisis Sentimen Ulasan Aplikasi Samsat Digital Nasional Menggunakan Algoritma Naive Bayes Classifier," *TEKNOSI*, vol. 10, no. 3, pp. 156–164, Jan. 2025, doi: 10.25077/TEKNOSI.v10i3.2024.156-164.

[15] H. Leidiyana, T. Misriati, and R. Aryanti, "Klasifikasi Sentimen Terhadap Kebijakan Tapera Menggunakan Komparasi Machine Learning dan SMOTE," *JKKI*, vol. 8, no. 2, pp. 125–135, Nov. 2024, doi: 10.31603/komtika.v8i2.12595.

[16] W. Priatna and J. S. Hidayat, “Implementasi Term Frequency – Inverse Document Frequency (Tf-Idf) Dan Vector Space Model (Vsm) Untuk Pencarian Berita Bahasa Indonesia,” *Jurnal Ilmiah Informatika, Arsitektur dan Lingkungan*, vol. 14, no. 2, pp. 119–133, 2019.
[17] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” *IJCA*, vol. 181, no. 1, pp. 25–29, Jul. 2018, doi: 10.5120/ijca2018917395.
[18] D. N. Agustia and R. R. Suryono, “Comparison of Naïve Bayes, Random Forest, and Logistic Regression Algorithms for Sentiment Analysis Online Gambling,” *ISI*, vol. 10, no. 1, pp. 284–295, Jan. 2025, doi: 10.35314/prk93630.
[19] A. Yaman and M. A. Cengiz, “The Effects of Kernel Functions and Optimal Hyperparameter Selection on Support Vector Machines,” *Journal of New Theory*, vol. 34, pp. 64–71, 2021.
[20] M. Muttakin, N. R. Rusmana, and D. Ramadhani, “Analisis Perbandingan Algoritma Decision Tree, Random Forest, Knn, Dan Svm Dalam Prediksi Penyakit Jantung,” *Journal of System & Technology*, vol. 1, no. 2, 2025.
[21] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood, and G. S. Choi, “A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis,” *PLoS ONE*, vol. 16, no. 2, p. e0245909, Feb. 2021, doi: 10.1371/journal.pone.0245909.
[22] Vincent, G. Darian, and N. Surantha, “Performance Evaluation of Convolutional Neural Network (CNN) for Skin Cancer Detection on Edge Computing Devices,” *Applied Sciences*, vol. 15, no. 6, p. 3077, Mar. 2025, doi: 10.3390/app15063077.
[23] B. Shannaq, O. Ali, S. A. Maqbali, and A. Al-Zeidi, “Advancing user classification models: A comparative analysis of machine learning approaches to enhance faculty password policies at the University of Buraimi,” *JIPD*, vol. 8, no. 13, p. 9311, Nov. 2024, doi: 10.24294/jipd9311.
[24] M. Fairuzabadi, *Mahir Data Science dengan Python Analisis Data & Machine Learning Untuk Pemula*. Yogyakarta: Yash Media, 2025.
[25] N. Neshat, S. Gupta, N. Unadkat, K. Jain, and A. Rathwa, “AI-Driven Stroke Classification and Detection: A Retrospective Study Using MRI Imaging,” *J Neonatal Surg*, vol. 14, no. 15S, pp. 2294–2300, Apr. 2025, doi: 10.63682/jns.v14i15S.4928.
[26] A. M. Van Der Veen and E. Bleich, “The advantages of lexicon-based sentiment analysis in an age of machine learning,” *PLoS ONE*, vol. 20, no. 1, p. e0313092, Jan. 2025, doi: 10.1371/journal.pone.0313092.
[27] N. S. Sediatmoko, Y. Nataliani, and I. Suryady, “Sentiment Analysis of Customer Review Using Classification Algorithms and SMOTE for Handling Imbalanced Class,” *Indonesian J. of Inf. Syst.*, vol. 7, no. 1, pp. 38–52, Aug. 2024, doi: 10.24002/ijis.v7i1.8879.