Jurnal ELTIKOM :
Jurnal Teknik Elektro, Teknologi Informasi dan Komputer

Vol. 9, No. 1, June 2025, page. 35-46
ISSN 2598-3245 (Print), ISSN 2598-3288 (Online)
DOI: http://doi.org/10.31961/eltikom.v9i1.1432
Available online at http://eltikom.poliban.ac.id

MULTI-LABEL CLASSIFICATION FOR OPINION MINING IN THE PRESIDENTIAL ELECTION USING TF-IDF WITH NB AND SVM

Ricy Ardiansyah[1], Herman Yuliansyah[1*], Anton Yudhana[2]
[1)] Department of Informatics, University Ahmad Dahlan, Yogyakarta, Indonesia
[2)] Department of Electrical Engineering, University Ahmad Dahlan, Yogyakarta, Indonesia
e-mail: ardianriki199@gmail.com, herman.yuliansyah@tif.uad.ac.id, eyudhana@ee.uad.ac.id

Received: 12 January 2025 – Revised: 20 February 2025 – Accepted: 24 February 2025

ABSTRACT

Public opinion plays a crucial role in presidential elections, shaping voter choices and influencing outcomes. Most sentiment analysis studies focus on binary (positive vs. negative) or multiclass (positive, negative, neutral) classification, which limits their ability to capture opinions that express multiple sentiments simultaneously. In presidential elections, a single opinion may support one candidate while criticizing another. This study proposes a MultiLabelBinarizer model to classify candidate and sentiment labels simultaneously—an approach that remains underexplored. The model combines Naïve Bayes (NB) and Support Vector Machine (SVM) for opinion mining using public data and TF-IDF for feature extraction, applying Multinomial and Linear kernels. Performance is evaluated using Accuracy, Precision, Recall, and F1-score. The study is conducted in two stages: developing a multi-label analysis model for presidential candidates and testing the effectiveness of cross-validation. Results show that multi-label classification is effective for both candidate and sentiment categories. Cross-validation with NB and SVM yields high accuracy. NB achieves 0.89 for candidate labels and 0.86 for sentiment labels. SVM performs better, with 0.93 for candidate labels and 0.94 for sentiment labels. While SVM provides higher accuracy, NB offers faster implementation with still competitive results.

Keywords*: multi-label classification, naïve bayes classifier, SVM, opinion mining, presidential election 2024, TF-IDF.*

I. INTRODUCTION

In recent years, opinion mining and sentiment analysis in the political sphere have attracted significant attention [1]. The rapid growth of social media platforms and online forums has created a rich data source for analyzing public opinion [2], [3]. Opinion mining, also known as sentiment analysis, involves extracting and analyzing subjective information from text data [4]. In politics, it helps reveal public perceptions of candidates and their policies. Traditional sentiment analysis (single-label) typically classifies text as either positive or negative [5]. However, real-world opinions are often more complex. They may contain multiple sentiments toward different candidates [6]. This complexity calls for a more advanced method—multi-label classification—where each text can be associated with several candidate labels.

The 2024 Indonesian presidential election offers a compelling case due to its dynamic and polarized political climate [7]. Analyzing public opinion in this context requires addressing diverse and sometimes conflicting sentiments within the same text. For example, a comment may support a candidate's education policy while criticizing their economic stance. This complexity highlights the relevance of multi-label classification [8] for this study.

Machine learning algorithms are commonly used for text classification and can be combined to improve performance. Naïve Bayes and Support Vector Machine (SVM) are two such algorithms [9], chosen for this study due to their efficiency and ability to handle high-dimensional data—common in political texts that span multiple categories. While most previous studies have emphasized binary or

multiclass classification, they often struggle to capture complex opinions containing multiple sentiment labels. In political discourse, a single statement may support one candidate while criticizing another. Traditional approaches face challenges in detecting such multi-label patterns, making it necessary to evaluate the performance of Naïve Bayes and SVM in multi-label contexts.

The purpose of this study is to propose a MultiLabelBinarizer classification model that enables the simultaneous classification of sentiment and candidate labels—an approach that remains underexplored in current research, especially regarding the diverse discussions surrounding presidential candidates. The model applies Naïve Bayes and SVM algorithms for opinion mining by combining candidate and sentiment labels. This study focuses specifically on the context of presidential elections. It builds upon the work of Asno Azzawagama [7], [10] by refining sentiment analysis into a multi-label classification task. The dataset used consists of reviews expressing various sentiments toward presidential candidates. Therefore, this study aims to develop a multi-label classification model using the Naïve Bayes and SVM algorithms [11].

The contributions of this study are threefold. First, it presents a multi-label classification model developed using data from the Indonesian presidential election, offering a more nuanced analysis of public sentiment. Second, it enhances the 2024 Indonesian presidential election sentiment dataset by Asno, transitioning it from a binary classification to a multi-label format. This transformation involves the use of Scikit-Learn's MultiLabelBinarizer to generate two distinct label categories, converting textual labels into binary vectors through one-hot encoding. The approach enables the identification of all unique labels within the candidate category column, addressing the scarcity of research that analyzes sentiment toward individual candidates in comparison to others using both label types. Third, the model's performance is assessed using appropriate multi-label classification metrics, with the experimental results shedding light on the dynamics of public opinion during the election. This study applies multi-label classification to explore political opinions in greater depth and demonstrates the effectiveness of Naïve Bayes and SVM in aspect-based sentiment analysis. It contributes to the development of text classification methods for analyzing public opinion. The structure of this study includes research methods, experiments, results, discussions, and conclusions.

Previous studies on multi-label classification and aspect-based sentiment analysis (ABSA) have explored various domains. Norlaila [1] conducted Twitter sentiment analysis using K-Means and Naïve Bayes, achieving 99% accuracy. Novresia [6] used a lexicon-based Naïve Bayes approach, reaching 92% accuracy with an F1 score of 87%. Asno Azzawagama [7] compared Naïve Bayes and SVM with TF-IDF for political opinion analysis, reporting 79% and 86% accuracy, respectively. Hisyam [12] ABSA with Naïve Bayes, achieving an average accuracy of 79% and a weighted accuracy of 93%. Alhakiem [13] also used ABSA with Naïve Bayes, recording a similar average accuracy of 79% and a maximum of 93%. A study analyzing Telkomsel tweets reported an F1 score improvement to 96.48% through feature expansion. Theo Ari [14] used CNN with Word2Vec to analyze reviews of Indonesian online stores, achieving 85.54% accuracy and a 92.02% F1 score. Kevin Tanoto [15] compared BERT, RF, NB, and SVM in analyzing Indonesian political sentiment, yielding balanced results with a 50:50 aspect ratio. Jakob Fehle [16] tested ABSA on German hotel reviews and obtained a micro F1 score of 0.91. Omar and Siyam [17] developed a hybrid ABSA model that improved accuracy by 5% over lexicon-based methods and outperformed SVM models.

These studies demonstrate that multi-label classification can be applied to sentiment analysis through the specific lens of ABSA. Therefore, this study explores the use of multi-label classification in the context of the presidential election, examining opinions across two categories—sentiment and candidate—as part of an aspect-based sentiment analysis approach.

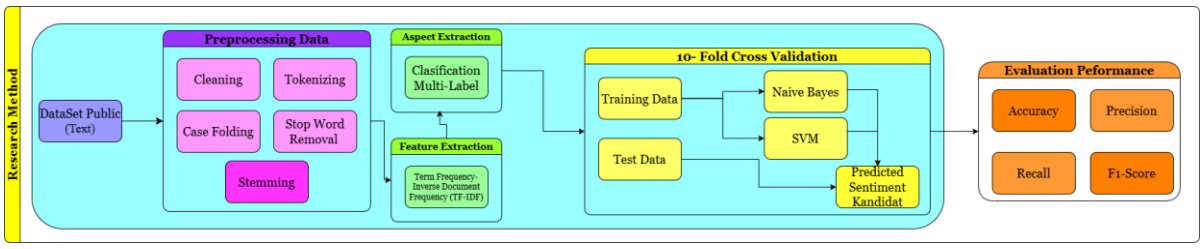

Figure 1. Research methods

II. RESEARCH METHOD

This study employs the Naive Bayes algorithm and SVM for multi-label classification to analyze public opinion on presidential candidates. The objective is to classify each review into sentiment labels (positive and negative) and candidate labels. The research process is illustrated in Figure 1.

The research process for multi-label classification using Naive Bayes and SVM is outlined in Figure 1. This study uses a public dataset curated by Asno Azzawagama (https://data.mendeley.com/datasets/7w5zvr8jgp/5). In the original study, Asno analyzed sentiment related only to presidential candidates. In this study, three Excel datasets are used: the first contains 10,001 entries for Anies Baswedan, the second contains 10,002 entries for Prabowo Subianto, and the third contains 10,002 entries for Ganjar Pranowo. The combined dataset of these three candidates is shown in Figure 2. Each candidate's dataset includes two sentiment labels: positive and negative. Anies Baswedan has 6,455 positive and 3,546 negative labels; Prabowo Subianto has 7,369 positive and 2,633 negative labels; Ganjar Pranowo has 7,831 positive and 2,171 negative labels. The sentiment data used to evaluate the model is presented in Table 1. This dataset was selected based on criteria including the number of category labels [18], the number of classes and candidates, and the fact that it is open-access and had not previously been used for multi-label classification.

The next stage is preprocessing. This step is essential to ensure data uniformity, reduce noise, and improve the accuracy of multi-label sentiment analysis. Although the data has been previously cleaned, preprocessing is still required to address text variability. This study applies preprocessing to a dataset consisting of three presidential candidates, totaling 30,005 entries, each classified by category.

A. Data Preprocessing

1) Cleaning

The cleaning phase (see Figure 3) involves removing unnecessary characters from the dataset, such as punctuation, numbers, and hashtags [19].

TABLE 1
PUBLIC DATASET

Dataset	Number of Data	Positive Label	Negative Label
Anies Baswedan	10001	6455	3546
Prabowo Subianto	10002	7369	2633
Ganjar Pranowo	10002	7831	2171

```
df.candidate.value_counts()
```

```
Index(['Text', 'label', 'candidate'], dtype='object')
```

candidate	count
Prabowo Subianto	10002
Ganjar Pranowo	10002
Anies Baswedan	10001

dtype: int64

Figure 2. Dataset of three candidates

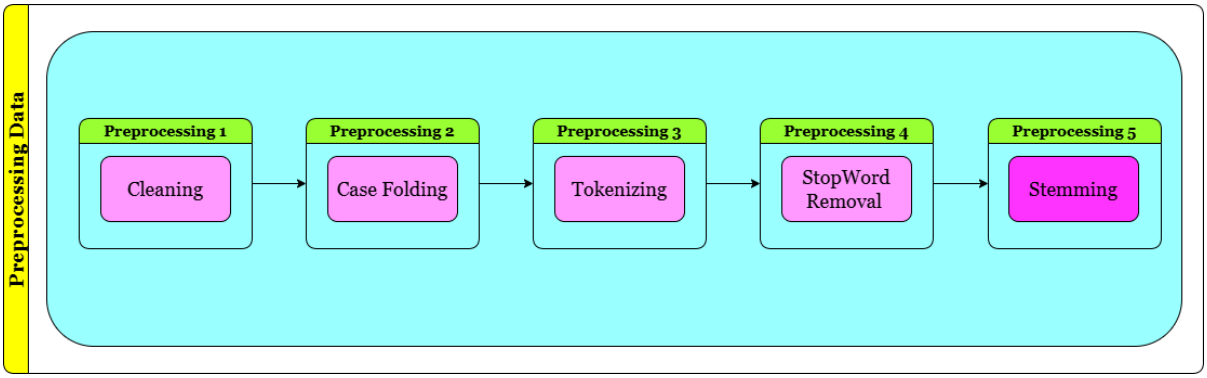

Figure 3. Data preprocessing

$$TF(t,d) = \frac{Number\ of\ occurrences\ of\ word\ t\ in\ document\ d}{Total\ number\ of\ words\ in\ document\ d} \quad (1)$$

$$IDF(t) = log\ \frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ the\ word\ t} \quad (2)$$

$$TF - IDF(t,d) = TF(t,d)\ x\ IDF(t) \quad (3)$$

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (4)$$

2) Case Folding

This step standardizes the text by converting all characters to lowercase, reducing redundancy and inconsistencies.

3) Tokenizing

This process splits sentences into individual words (tokens) to facilitate text weighting and further analysis. Filtered tokens are retained after removing irrelevant words.

4) Stopword Removal

In this step, common and insignificant words (stopwords) are removed from the text. After tokenization, the words are checked against a stopword list to retain only meaningful content.

5) Stemming

Stemming reduces words to their root form by removing affixes, helping to minimize variations of the same word and simplify analysis.

B. Feature Extraction

Term Frequency–Inverse Document Frequency (TF-IDF) is a widely used weighting scheme in information retrieval and text mining. TF-IDF indicates the importance of a word within documents in a corpus [20]. Equation (1) defines term frequency (TF), Equation (2) presents the inverse document frequency (IDF), and Equation (3) calculates the TF-IDF score based on a word's appearance in a document. In (1), a higher TF value suggests greater importance of a word within a specific document. In (2), IDF represents the inverse of document frequency (DF) and is used to determine how unique or rare a word is across a set of documents. A high IDF value indicates that the word appears in fewer documents and is therefore considered more significant [21]. In this study, the preprocessed text is used to generate features for machine learning models by applying the TF-IDF method shown in (3). For a word *t* in document *d*, the TF-IDF value is computed in (1)-(3).

Abstracts should be explained at the beginning of the manuscript. The abstract section must clearly state the research's background, problems, objectives, results, and conclusions. The Introduction section must explicitly state the problem, update, and research objectives. The introduction must also be equipped with state-of-the-art research accompanied by the latest primary library sources.

C. Naïve Bayes and SVM for Aspect-based Sentiment Analysis

Naïve Bayes is a fast, simple, and effective algorithm widely used in data mining for classification tasks. It predicts class membership probabilities using Bayes' theorem. Equation (4) presents the foundational concept of Bayes' theorem. Naïve Bayes classification relies on this theorem, offering high accuracy and computational efficiency when applied to large datasets. The theorem estimates class probabilities based on observed data. Here, *X* represents the data to be classified or analyzed, whose class or hypothesis is unknown. *H* denotes a hypothesis or assumption that *X* belongs to a particular class. The posterior probability *P(H|X)* reflects the likelihood that the hypothesis *H* is true given the observed data *X*, enabling informed decision-making. *P(H)* is the prior probability of the hypothesis before considering the data. *P(X|H)* is the likelihood, showing how probable the data *X* is, assuming the hypothesis *H* is true. Finally, *P(X)* is the overall probability of observing the data *X* [22].

ABSA effectively classifies sentiment related to specific aspects within a sentence. The process involves several operations, typically including Opinion Target Extraction (OTE), Aspect Category Detection (ACD), and Sentiment Polarity (SP). OTE extracts aspect terms (e.g., entities or attributes), ACD identifies associated entities and attributes, and SP determines the sentiment polarity of each aspect [23].

$$TF(t,d) = \frac{Number\ of\ occurrences\ of\ word\ t\ in\ document\ d}{Total\ number\ of\ words\ in\ document\ d} \quad (4)$$

$$IDF(t) = log\ \frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ the\ word\ t} \quad (5)$$

```
from sklearn.preprocessing import MultiLabelBinarizer

    # Binarize the multilabel data
    mlb = MultiLabelBinarizer(classes=['Anies Baswedan', 'Prabowo Subianto', 'Ganjar Pranowo'])
    y = mlb.fit_transform(df['candidate'])
```

Figure 4. Multi-label classification for candidates

```
    # Clean text column
    df['Text'] = df['Text'].apply(clean_text)

    # Binarize the multilabel data
    mlb = MultiLabelBinarizer()
    y = mlb.fit_transform(df['label'])

    # TF-IDF Vectorization
    tfidf = TfidfVectorizer(max_features=30000, stop_words='english', ngram_range=(1, 3), min_df=3
    X = tfidf.fit_transform(df['Text'])
```

Figure 5. Multi-label classification for sentiment

In this study, opinion targets refer to aspects of presidential candidates mentioned in tweets on the Twitter platform. Aspect categories define the specific features of an entity that are the focus of sentiment analysis. The Naïve Bayes and SVM algorithms are employed to identify aspects at the sentence level. Extracted topic keywords are mapped to relevant entity aspects to perform aspect-based sentiment analysis on candidate-related tweets. In this context, Naïve Bayes and SVM are used to detect and analyze sentiment toward particular features or aspects of a target entity within a given text. The program code for these two categories is shown in Figures 4 and 5, representing the multi-label classification model. This code utilizes the *MultiLabelBinarizer* class from the scikit-learn library to binarize multi-label data into two label categories. It creates a *MultiLabelBinarizer* object to convert the original multi-label format into a one-hot encoded format. The *fit_transform* function serves two purposes: *fit* identifies all unique labels in the dataset, and *transform* converts the original data into binary format based on these labels.

Sentiment classification is the stage where machine learning algorithms are used to group text or documents into defined sentiment classes. commonly used algorithms for text classification is Support Vector Machine (SVM). SVM can be applied for both classification and regression tasks. It works by constructing a hyperplane that separates data points from different classes [27]. The formulation for SVM classification is shown in (5). In this equation, X_i represents a tuple of input values, α_i is the Lagrange multiplier constant, y_i denotes the sentiment class label, $K(x, x_i)$ is the kernel function for the test data, and b is the bias term.

D. Cross-Validation

Evaluation of multi-label classification cannot rely solely on accuracy, as is common in single-label classification. In this study, the multi-label classification involves three presidential candidates, each with distinct sentiment labels. The dataset is partitioned into K equally sized segments. When using 10-Fold Cross-Validation, the dataset is divided into 10 subsets for 10 iterations of training and testing [24]. In each iteration, one subset is used as the test set (validation set), while the remaining $K–1$ subsets are used to train the model. This process is repeated until every subset has been used once as the test set. The program code implementing 10-Fold Cross-Validation for the Naïve Bayes and SVM algorithms is shown in Figures 6 and 7. The code in Figures 6 and 7 defines a function that splits the dataset into training and test sets using the *scikit-learn* library. The evaluation process applies *StratifiedKFold*, which divides the dataset into multiple folds while ensuring that each fold maintains a similar distribution of labels. This method provides a more balanced and reliable evaluation of multi-label model performance.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (6)$$

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP + TN}{TP + FN} \quad (8)$$

$$F1 - Score = \frac{2\ X\ Recall\ x\ Precision}{Recall + Precision} \quad (9)$$

```
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)
```

```
def cross_validation_evaluation(X, y, mlb):
    model = OneVsRestClassifier(MultinomialNB(alpha=0.5))

    accuracies, precisions, recalls, f1_scores, hamming_losses = [], [], [], [], []

    for fold, (train_index, test_index) in enumerate(skf.split(X, y.argmax(axis=1)), start=1):
        X_train, X_test = X[train_index], X[test_index]
        y_train, y_test = y[train_index], y[test_index]

        # Train the model
        model.fit(X_train, y_train)

        # Predict on test set
        y_pred = model.predict(X_test)

        # Calculate metrics for the fold
        accuracy = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred, average='macro', zero_division=0)
        recall = recall_score(y_test, y_pred, average='macro', zero_division=0)
        f1 = f1_score(y_test, y_pred, average='macro', zero_division=0)
        hamming = hamming_loss(y_test, y_pred)
```

Figure 6. 10-fold cross-validation using Naive Bayes

```
def cross_validation_evaluation(X, y, mlb):
    model = OneVsRestClassifier(SVC(kernel='linear', C=1.0, random_state=42))

    accuracies, precisions, recalls, f1_scores, hamming_losses = [], [], [], [], []

    for fold, (train_index, test_index) in enumerate(skf.split(X, y.argmax(axis=1)), start=1):
        X_train, X_test = X[train_index], X[test_index]
        y_train, y_test = y[train_index], y[test_index]

        # Train the model
        model.fit(X_train, y_train)

        # Predict on test set
        y_pred = model.predict(X_test)

        # Calculate metrics for the fold
        accuracy = accuracy_score(y_test, y_pred)
        precision = precision_score(y_test, y_pred, average='macro', zero_division=0)
        recall = recall_score(y_test, y_pred, average='macro', zero_division=0)
        f1 = f1_score(y_test, y_pred, average='macro', zero_division=0)
        hamming = hamming_loss(y_test, y_pred)
```

Figure 7. 10-fold cross-validation using SVM

E. Confusion Matrix Performance

The Confusion Matrix is a commonly used method to evaluate the performance of a classification model. It provides information by comparing the predicted classification results with the actual results [25]. The key evaluation metrics considered in this study include Accuracy, Precision, Recall, and F1-Score. The final stage of this study involves evaluating the performance of the developed system using these metrics. As described in (6), Accuracy is calculated as the proportion of correctly predicted instances. The F1-Score, which is the harmonic mean of Precision and Recall, offers a more balanced evaluation, especially when the data is imbalanced.

Accuracy, as defined in (6), refers to the percentage of instances correctly predicted by the model. In this context, *P* represents all actual data in the positive class, and *N* represents all actual data in the negative class. The accuracy value is calculated by summing the number of correctly predicted positive and negative instances, then dividing that by the total number of instances in the dataset [26]. To calculate the F1-Score, the values of Precision and Recall must first be determined. These values are obtained

using the confusion matrix, which includes True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). A True Positive (TP) occurs when the model correctly predicts a positive class. A False Positive (FP) occurs when the model predicts a positive class, but the actual class is negative. A True Negative (TN) is when the model correctly predicts a negative class, while a False Negative (FN) occurs when the model predicts a negative class, but the actual class is positive.

Precision measures how many of the model's positive predictions are correct. Equation (7) is used to calculate the Precision value as the ratio of True Positives (TP) to the sum of True Positives (TP) and False Positives (FP).

Recall measures how many actual positive cases are correctly predicted by the classifier. Equation (8) calculates Recall as the ratio of True Positives (TP) to the sum of True Positives (TP) and False Negatives (FN).

F1-Score, or F1-measure, is one of the most widely used evaluation metrics in machine learning tasks. Equation (9) is used to compute the F1-Score, which combines Precision and Recall into a single metric. Defined as the weighted harmonic mean of Precision and Recall, the F1-Score provides a balanced evaluation when both Precision and Recall are equally important.

III. Result And Discussions

This study analyzes multi-label classification using test data from the 2024 presidential candidates. The analysis involves a more complex opinion classification process; therefore, two categories are used—sentiment labels and candidate labels. The results show a K-Fold evaluation using both Naïve Bayes and SVM algorithms. These results include the performance of both sentiment and candidate labels, and the final output highlights the highest average metric across the two categories. The findings indicate improved performance when addressing multi-label classification involving two categories.

A. K-Fold Evaluation

The results of K-Fold Cross-Validation provide the average performance metrics of the model evaluated across multiple subsets of the data. Using 10-fold cross-validation, evaluation is carried out over 10 iterations with 10 different data splits. The average accuracy across all folds reflects the model's overall performance on the dataset. Tables 2 and 3 present the evaluation results for the training and testing datasets. The average performance metrics are reported for the model using cross-validation combined with the SVM algorithm. Evaluation results from 10 iterations are shown using 10 different data subsets. Tables 4 and 5 present the evaluation for sentiment and candidate labels, respectively.

TABLE 2
10-Fold Naïve Bayes – Sentiment Labels

K-fold	Accuracy	Precision	Recall	F1-Score
1	82	80	73	75
2	81	79	71	74
3	82	80	72	74
4	84	82	74	77
5	82	80	72	75
6	83	81	73	76
7	82	80	73	75
8	81	78	71	73
9	81	79	71	73
10	83	81	74	76

TABLE 3
10-Fold Naïve Bayes – Candidate Labels

K-fold	Accuracy	Precision	Recall	F1-Score
1	85	90	87	88
2	85	91	86	88
3	86	92	88	90
4	86	91	87	89
5	87	92	88	90
6	85	90	87	88
7	85	91	87	89
8	86	91	88	89
9	86	91	87	89
10	86	91	88	89

TABLE 4
10-FOLD SVM – SENTIMENT LABELS

K-fold	Accuracy	Precision	Recall	F1-score
1	89	86	85	86
2	89	86	85	86
3	89	87	85	86
4	89	87	85	86
5	89	86	85	86
6	90	88	86	87
7	90	88	86	87
8	89	86	85	86
9	89	87	85	86
10	90	87	86	87

TABLE 5
10-FOLD SVM – CANDIDATE LABELS

K-fold	Accuracy	Precision	Recall	F1-score
1	88	93	89	91
2	87	92	88	90
3	90	94	90	92
4	89	93	90	91
5	90	94	90	92
6	88	92	89	90
7	88	93	89	91
8	88	93	89	91
9	88	93	89	91
10	88	92	89	91

TABLE 6
CANDIDATE LABELS AND SENTIMENT – NAÏVE BAYES

Category	Candidate/Sentiment	Precision	Recall	F1-score	Accuracy
Candidates	Anies Baswedan	0.92	0.94	0.93	0.89
	Prabowo Subianto	0.94	0.86	0.90	
	Ganjar Pranowo	0.95	0.91	0.93	
Sentiment	Negative	0.84	0.62	0.72	0.86
	Positive	0.87	0.96	0.91	

TABLE 7
CANDIDATE LABELS AND SENTIMENT – SVM

Category	Candidate/Sentiment	Precision	Recall	F1-score	Accuracy
Candidates	Anies Baswedan	0.97	0.95	0.96	0.93
	Prabowo Subianto	0.97	0.92	0.95	
	Ganjar Pranowo	0.96	0.94	0.95	
Sentiment	Negative	0.92	0.90	0.91	0.94
	Positive	0.96	0.97	0.97	

Tables 2 and 3 show the model evaluation using 10-fold cross-validation based on Accuracy, Precision, Recall, and F1-Score. The accuracy for sentiment labels ranges from 81% to 83%, while for candidate labels it ranges from 85% to 87%, indicating consistent model performance. Precision values for sentiment labels range from 78% to 82%, showing more variability than candidate labels, which are more stable between 90% and 92%. Recall scores are relatively stable, ranging from 71% to 74% for sentiment labels and 86% to 88% for candidate labels. The F1-Score ranges from 73% to 76% for sentiment labels and from 88% to 90% for candidate labels. These results indicate strong multi-label classification performance for both sentiment and candidate categories.

Based on Tables 2 and 3, the model demonstrates consistent performance across all folds in terms of Precision, Recall, and F1-Score. High Precision values suggest accurate positive predictions. Fold 4 shows the best performance, possibly due to a more balanced distribution of data. The multi-label model shows stability across both categories over 10 folds and yields more accurate predictions for candidate classes, highlighting its effectiveness in handling complex opinion data.

Tables 4 and 5 present the results of model evaluation using a combination of SVM and 10-fold cross-validation. Each fold is assessed using four main metrics—Accuracy, Precision, Recall, and F1-Score—to evaluate overall model performance. Based on the results, Accuracy for sentiment labels ranges from 89% to 90%, while for candidate labels it ranges from 88% to 90%, indicating consistent model performance across all folds. Precision for sentiment labels remains stable between 86% and 88%, while for candidate labels it increases to a range of 92% to 94%. Recall scores are also consistent, ranging from

85% to 86% for sentiment labels and from 88% to 90% for candidate labels. These results suggest that the model achieves both high Precision and stable Recall in predicting sentiment labels. The F1-Score, as the harmonic mean of Precision and Recall, ranges from 86% to 87% for sentiment labels and from 91% to 92% for candidate labels. Overall, the model using SVM demonstrates more consistent performance across all folds and proves effective in handling complex multi-label classification tasks.

Based on the results in Tables 4 and 5, the model consistently performs well in each fold. The high Precision scores reflect the model's ability to make accurate predictions for sentiment classes, while the relatively stable Recall indicates that SVM performs reliably in this context. This contributes to F1-Score values ranging from 86% to 87% for sentiment labels. For candidate labels, the Precision scores are stable and consistently high, while Recall values range from 88% to 90%, showing the model's accuracy in predicting candidate-related sentiment. These findings demonstrate that the multi-label model is effective in managing complex opinion data. In conclusion, the combination of the SVM algorithm with multi-label classification produces consistent and optimal performance.

B. Method Performance

Table 6 presents the performance evaluation of the Naïve Bayes algorithm, while Table 7 shows the results of the SVM method using Precision, Recall, and F1-Score metrics for two labels—candidate and sentiment—from the multi-label classification. In the Naïve Bayes model, the negative sentiment label achieved a Precision of 0.84, Recall of 0.62, and F1-Score of 0.72. The positive label obtained a Precision of 0.87, Recall of 0.96, and F1-Score of 0.91. For candidate labels, Anies Baswedan scored 0.92 in Precision, 0.94 in Recall, and 0.93 in F1-Score; Prabowo Subianto scored 0.94, 0.86, and 0.90, respectively; and Ganjar Pranowo achieved 0.95, 0.91, and 0.93. The SVM model produced higher results. For the negative sentiment label, it achieved a Precision of 0.92, Recall of 0.90, and F1-Score of 0.91. The positive label obtained Precision, Recall, and F1-Score values of 0.96, 0.97, and 0.97, respectively. In candidate labels, Anies Baswedan recorded Precision 0.97, Recall 0.95, and F1-Score 0.96; Prabowo Subianto scored 0.97, 0.92, and 0.95; and Ganjar Pranowo scored 0.96, 0.94, and 0.95. In the Naïve Bayes method, the lower Recall value for the negative sentiment label indicates that while the model performs reasonably well, there is still room for improvement in identifying negative examples. However, it performs strongly in predicting positive examples, with high accuracy across sentiment labels. The model also demonstrates good stability and effectiveness in handling complex opinion data across all three candidates.

The SVM results show high Precision, suggesting that the model minimizes false positive errors. High Recall indicates its ability to identify most relevant examples correctly. The high F1-Score reflects a strong balance between Precision and Recall. For candidate labels, the SVM model maintains consistently high performance across all three candidates, indicating its effectiveness in detecting public opinion with high accuracy. These findings suggest that the model classifies public opinion toward each candidate reliably, balancing both precision and sensitivity. Overall, both models handle multi-label classification effectively, but SVM demonstrates greater stability and higher performance across both sentiment and candidate categories.

Based on the results obtained from the multi-label classification involving two categories—candidate labels and sentiment labels—using the Naïve Bayes algorithm and SVM, it is evident that SVM demonstrates superior performance in both. SVM achieves a higher accuracy of 0.94 for both negative and positive sentiment labels, indicating its effectiveness in capturing sentiment patterns, particularly in predicting negative sentiment with a better balance between Precision and Recall. With an accuracy of 0.93 across all three candidate labels, SVM proves to be more reliable in handling multi-label classification and consistent in predicting candidate-related categories.

Although SVM outperforms, the Naïve Bayes algorithm still delivers solid performance in multi-label classification. The application of K-Fold Cross-Validation contributes to reliable model evaluation and helps reduce the risk of overfitting. Naïve Bayes records high Precision and Recall values of 0.86 for positive sentiment labels, and reasonable performance for negative labels. For candidate labels, the model achieves a high overall accuracy of 0.89, demonstrating that the integration of a two-category multi-label classification with Naïve Bayes is effective.

There are several justifications for choosing Naïve Bayes in this context. First, it offers time efficiency, operating faster than SVM. Second, its performance remains competitive, with accuracy levels

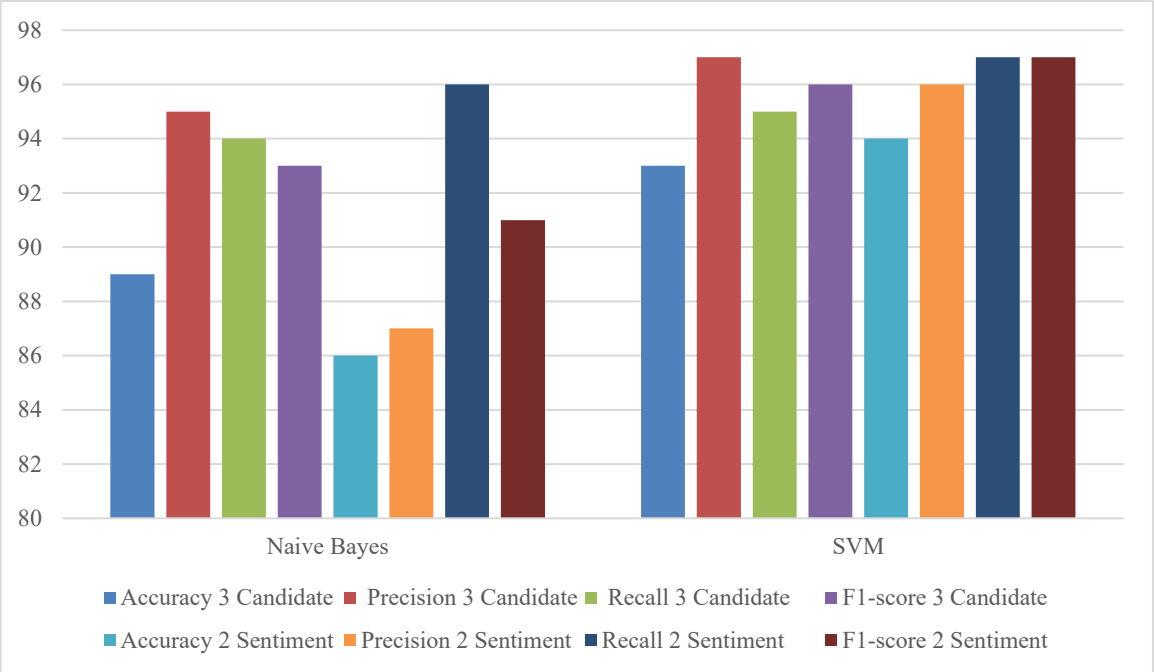

Figure 8. Highest Matrix for Two Labels Using Naïve Bayes and SVM

ranging from 0.86 to 0.89, which are still comparable to other classification models. Third, its simplicity makes it easy to implement in various multi-label classification scenarios.

These findings represent an improvement over Asno's previous study, which reported an accuracy of only 78% [10]. This comparison highlights that binary sentiment analysis alone may not provide balanced results across a single category, whereas the use of multi-label classification allows for more comprehensive and nuanced analysis of each category.

One limitation of the algorithms used in this study—particularly Naïve Bayes—is its assumption of feature independence, which affects its ability to handle feature correlation. This limitation led to several misclassifications of negative sentiment labels. This is evident in the lower Recall value for negative sentiments (0.62) compared to that for positive sentiments (0.96). The model appears to struggle in correctly identifying negative sentiments, likely because words expressing criticism toward one candidate may also imply support for another. This overlap introduces feature correlation, which Naïve Bayes cannot effectively capture due to its independence assumption. In the multi-label candidate classification, although Naïve Bayes achieved reasonably strong results with an accuracy of 0.89, the model may misclassify opinions that are affiliated with more than one candidate. This issue arises when opinions contain phrases or terms commonly associated with multiple candidates simultaneously, which the model fails to interpret correctly due to its inability to account for inter-feature relationships.

Figure 8 presents the evaluation of the highest performance results from both algorithmic models. In Naïve Bayes, the sentiment classification achieved high Precision (0.87), Recall (0.96), and F1-Score (0.91). Candidate classification also yielded strong results, with Precision of 0.95, Recall of 0.94, and F1-Score of 0.93 across the three presidential candidates. The performance of the SVM model shows a significant improvement in both sentiment and candidate categories. For candidate classification, SVM achieved a Precision of 0.97, Recall of 0.95, and F1-Score of 0.96. In sentiment classification, the model demonstrated even stronger performance, with Precision of 0.96, Recall of 0.97, and F1-Score of 0.97. The highest sentiment label accuracy reached 0.94, and the overall accuracy for candidate labels was 0.93. These results demonstrate that the SVM model, in comparison to Naïve Bayes, provides superior performance and highlights the effectiveness of the multi-label classification approach over traditional binary classification.

IV. Conclusion

Based on the classification analysis of the public dataset, the multi-label classification approach successfully produced two categories—sentiment labels and candidate labels—for identifying presidential candidates, with both the Naïve Bayes and SVM algorithms performing effectively. The use of cross-

validation and confusion matrix evaluation yielded high prediction accuracy, particularly when applied to the dataset of the three candidates. The confusion matrix results indicate that both sentiment and candidate labels achieved accurate and meaningful test outcomes, confirming that the study met its intended objectives. The SVM method demonstrated superior accuracy across both label categories compared to Naïve Bayes. However, Naïve Bayes still offers advantages in terms of time efficiency and competitive performance, making it a viable option for multi-label classification in opinion mining related to presidential elections. The findings show that the model can accurately handle two multi-label categories—positive and negative sentiment labels, as well as candidate labels for Anies Baswedan, Prabowo Subianto, and Ganjar Pranowo. The application of Naïve Bayes and SVM demonstrates competitive accuracy in classifying opinions across various sentiment categories, supporting their potential as efficient algorithms for multi-label classification of opinion data.

ACKNOWLEDGMENT

This research was supported by Universitas Ahmad Dahlan under the Penelitian Tesis Magister (Master's Thesis Research) scheme with grant number: PTM-025/SP3/LPPM-UAD/XII/2024 (09-12-2024).

REFERENCES

[1] W. W. Norlaila, Winarno dan E. T. Luthfi, "Analisis Sentimen Masyarakat Tentang Tambang Di Indonesia Pada Twitter Menggunakan Data Mining," *J. Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 3, hal. 1091–1099, 2024.

[2] Elinda, H. Yuliansyah, M. Iqbal, dan A. Latiffi, "Sentiment Analysis of the Sheikh Zayed Grand Mosque ' s Visitor Reviews on Google Maps Using the VADER Method," *Int. J. Adv. Data Inf. Syst.*, vol. 5, no. 1, 2024, doi: 10.59395/ijadis.v5i1.1320.

[3] H. Yuliansyah, S. A. Mulasari, dan F. A. Ghozali, "Sentiment Analysis of the Waste Problem based on YouTube comments using VADER and Deep Translator," *J. MEDIA Inform. BUDIDARMA*, vol. 8, hal. 663–673, 2024, doi: 10.30865/mib.v8i1.6918.

[4] T. Tukino dan F. Fifi, "Penerapan Support Vector Machine Untuk Analisis Sentimen Pada Layanan Ojek Online," *J. Desain Dan Anal. Teknol.*, vol. 3, no. 2, hal. 104–113, 2024, doi: 10.58520/jddat.v3i2.59.

[5] J. Tao dan X. Fang, "Toward Multi - Label Sentiment Analysis : A Transfer Learning Based Approach," *J. Big Data*, hal. 1–26, 2020, doi: 10.1186/s40537-019-0278-0.

[6] N. Wijaya dan E. S. Panjaitan, "Analisis Sentimen Ulasan Aplikasi Instagram di Google Play Store : Pendekatan Multinomial Naive Bayes dan Berbasis Leksikon," *Build. Informatics, Technol. Sci.*, vol. 6, no. 2, hal. 921–929, 2024, doi: 10.47065/bits.v6i2.5615.

[7] A. A. Firdaus, A. Yudhana, dan I. Riadi, "Prediction of Presidential Election Results using Sentiment Analysis with Pre and Post Candidate Registration Data," *J. Ilmu Komput. dan Inform.*, vol. 10, no. 1, hal. 36–46, 2024.

[8] I. S. Dara Tursina, Sherly Rosa, Chastine Fatichah, "Metode Hibrida Oversampling Untuk Menangani Imbalanced Multi-Label," *J. Ilm. Teknol. Inf.*, vol. 22, no. 1, hal. 32–44, 2024.

[9] H. I. Ramanda M, Reza Dwi Restiyan, "Analisis Sentimen Masyarakat terhadap Perilaku Lawan Arah yang diunggah pada Media Sosial Youtube Menggunakan Naïve Bayes," *J. Informatics Comput. Eng.*, vol. 02, no. 02, hal. 75–83, 2024.

[10] A. Azzawagama, A. Yudhana, dan I. Riadi, "Indonesian presidential election sentiment : Dataset of response public before 2024," *Data Br.*, vol. 52, hal. 109993, 2024, doi: 10.1016/j.dib.2023.109993.

[11] M. I. Nia Mardiah, Leni Marlina, Khairul, Zulham Sitorus, "Analysis Of Indonesian People ' s Sentiment Towards 2024 Presidential Candidates On Social Media Using Naïve Bayes Classifier and Support Vector Machine," *Build. Informatics, Technol. Sci.*, vol. 6, no. 2, hal. 950–960, 2024, doi: 10.47065/bits.v6i2.5766.

[12] H. Y. Hisyam Agus Setiawan, "Analisis Sentimen Berbasis Aspek Terhadap Ulasan Pengguna pada Game Honkai: Star Rail Menggunakan Naïve Bayes Classifier," *J. Sist. Inf.*, vol. 13, no. 5, hal. 1956–1971, 2024.

[13] H. R. Alhakiem dan E. B. Setiawan, "Aspect-Based Sentiment Analysis on Twitter Using Logistic Regression with FastText Feature Expansion," *J. Rekayasa Sist. dan Teknol. Inf.*, vol. 5, no. 158, hal. 840–846, 2022, [Daring]. Tersedia pada: doi: https://doi.org/10.29207/resti.v6i5.4429

[14] M. Theo, A. Bangsa, S. Priyanta, dan Y. Suyanto, "Aspect-Based Sentiment Analysis of Online Marketplace Reviews Using Convolutional Neural Network," *Indones. J. Comput. Cybern. Syst.*, vol. 14, no. 2, hal. 123–134, 2020, doi: 10.22146/ijccs.51646.

[15] K. Tanoto, A. A. S. Gunawan, D. Suhartono, T. N. Mursitama, A. Rahayu, dan M. I. M. Ariff, "Investigation of Challenges in Aspect-Based Sentiment Analysis Enhanced Using Softmax Function on Twitter During the 2024 Indonesian Presidential Election," *Procedia Comput. Sci.*, vol. 245, no. 2022, hal. 989–997, 2024, doi: 10.1016/j.procs.2024.10.327.

[16] J. Fehle, T. Schmidt, L. Münster, dan C. Wolff, "Aspect-Based Sentiment Analysis as a Multi-Label Classification Task on the Domain of German Hotel Reviews," *Assoc. Comput. Linguist.*, hal. 202–218, 2023.

[17] O. Alqaryouti, N. Siyam, A. A. Monem, dan K. Shaalan, "Aspect-Based Sentiment Analysis Using Smart Government Review Data," *Appl. Comput. Informatics*, vol. 20, no. 1/2, hal. 142–161, 2019, doi: 10.1016/j.aci.2019.11.003.

[18] K. A. Pradani dan L. H. Suadaa, "Automated Essay Scoring Menggunakan Semantic Textual Berbasis Transformer Untuk Penilaian Ujian Esai," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 6, hal. 1177–1184, 2023, doi: 10.25126/jtiik.2023107338.

[19] A. Syam, G. Hardy, A. Salim, D. Fatmarani, dan M. Fajar, "Analisis Teknik Preprocessing pada Sentimen Masyarakat Terkait Konflik Israel-Palestina Menggunakan Support Vector Machine," *J. Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 3, hal. 1464–1472, 2024.

[20] R. Saputra dan M. G. Pradana, "Implementasi Algoritma Cosine Similarity dan TF- IDF dalam Menentukan Rumpun Jabatan," *J. Tek. Inform.*, vol. 12, no. 1, hal. 1–11, 2024, doi: 10.32832/kreatif.v12i1.15470.

[21] E. Pratiwi, "Analisa Sentimen Penghapusan Tilang Manual Menjadi Tilang Elektronik Menggunakan Text Mining Dan TermFrequency Inverse Document Frequency (Tf-Idf)," *Konf. Nas. Teknol. Inf. dan Komput.*, vol. 7, no. 1, hal. 89–97, 2024, doi: 10.30865/komik.v6i1.8043.

[22] Y. Pratama, A. Roberto Tampubolon, L. Diantri Sianturi, R. Diana Manalu, dan D. Frietz Pangaribuan, "Implementation of Sentiment Analysis on Twitter Using Naïve Bayes Algorithm to Know the People Responses to Debate of DKI Jakarta Governor Election," *J. Phys. Conf. Ser.*, vol. 1175, hal. 012102, Mar 2019, doi: 10.1088/1742-6596/1175/1/012102.

[23] L. Š. and D. B.-Š. Damir Krstinić, Maja Braović, "Multi-Label Classifier Performance Evaluation With Confusion Matrix," *Comput. Sci. Inf. Technol.*, 2020, doi: 10.5121/csit.2020.100801.

[24] A. W. Ishlah, S. Sudarno, dan P. Kartikasari, "Implementasi Gridsearchcv Pada Support Vector Regression (SVR) Untuk Peramalan

Harga Saham," *J. Gaussian*, vol. 12, no. 2, hal. 276–286, Jul 2023, doi: 10.14710/j.gauss.12.2.276-286.

[25] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, "Multi-label Classifier Performance Evaluation with Confusion Matrix," in *Computer Science & Information Technology*, Jun. 2020, vol. 10, no. 08, pp. 01–14. doi: 10.5121/csit.2020.100801.

[26] A. W. Ishlah, S. Sudarno, and P. Kartikasari, "Implementasi Gridsearchcv Pada Support Vector Regression (SVR) Untuk Peramalan Harga Saham," *J. Gaussian*, vol. 12, no. 2, pp. 276–286, Jul. 2023, doi: 10.14710/j.gauss.12.2.276-286.

[27] A. Zaiem dan N. Charibaldi, "Komparasi Fungsi Kernel Metode Support Vector Machine untuk Analisis Sentimen Instagram dan Twitter (Studi Kasus : Komisi Pemberantasan Korupsi)," *J. Sarj. Tek. Inform.*, vol. 9, no. 2, hal. 33–42, 2021, [Daring]. Tersedia pada: https://doi.org/10.12928/jstie.v8i3.xxx.