ComTech: Computer, Mathematics and Engineering Applications, 17(1), June 2026, 23−37 P-ISSN: 2087-1244
DOI: 10.21512/comtech.v17i1.14132 E-ISSN: 2476-907X

Balinese Language Classification on Social Media using Multinomial Naive Bayes Method with TF-IDF

Putu Widyantara Artanta Wibawa[1]*; Cokorda Pramartha[2]; I Gusti Ngurah Anom Cahyadi Putra[3]; Luh Gede Astuti[4]

[1-4]Informatics, Faculty of Mathematics and Natural Sciences, Udayana University, Denpasar, Indonesia 80361
[1]putuwaw973@gmail.com; [2]cokorda@unud.ac.id;
[3]anom.cp@unud.ac.id; [4]lg.astuti@unud.ac.id

Received: 2nd August 2025/ **Revised:** 5th October 2025/ **Accepted:** 11th November 2025

How to Cite: Wibawa, P. W. A., Pramartha, C., Putra, I. G. N. A. C., & Astuti, L. G. (2026). Balinese language classification on social media using Multinomial Naive Bayes method with TF-IDF. *ComTech: Computer, Mathematics and Engineering Applications, 17*(1), 23−37. https://doi.org/10.21512/comtech.v17i1.14132

Abstract **-** Balinese is a local language that is widely used and spoken by Balinese people, including on social media platforms. However, the nuances of its politeness levels are often lost in informal digital communication, and there is a significant lack of computational models that automatically classify these levels, particularly for low-resource languages such as Balinese. The primary objective of this study is to evaluate the performance of the Multinomial Naive Bayes method combined with Term Frequency–Inverse Document Frequency (TF-IDF) feature extraction, Chi-square feature selection, and the Synthetic Minority Oversampling Technique (SMOTE) in classifying Balinese language levels. The dataset used in this study consists of 1,314 annotated social media posts and comments, primarily sourced from Instagram. A Balinese language expert performs the annotation, categorizing the texts into six levels that represent varying degrees of politeness and formality. These levels include *alus singgih* (polite, used for respecting others), *alus sor* (polite, used for self-humbling), *alus mider* (polite, used for both respecting others and self-humbling), *alus madia* (an intermediate level of politeness), *basa andap* (casual, commonly used in everyday life), and *basa kasar* (impolite, often used during arguments or toward animals). The experimental results show that the model achieves 96.53% accuracy on the training data and 61.45% accuracy on the test data. In addition, hyperparameter tuning reveals that the Multinomial Naive Bayes model with 2,720 selected features and SMOTE oversampling achieves 91.78% accuracy, significantly outperforming the baseline model without feature selection or oversampling, which achieves only 64.93% accuracy.

Keywords**:** classification, Balinese language, TF-IDF, Chi-square, SMOTE, Multinomial Naive Bayes

I. INTRODUCTION

Balinese is a local language widely spoken by the Balinese people. Rapid technological development currently changes people's lifestyles, including how they communicate through social media. According to Mastini et al. (2021), the presence of social media provides a new space for people to post content that reflects the ongoing use of the Balinese language. However, when using social media, Balinese speakers often do not apply the appropriate language level (*anggah ungguhing basa*). This aspect is crucial because *anggah ungguhing* basa functions as a speech-level system based on degrees of politeness and formality, which reflects a person's ethics and influences politeness in communication (Suardiana, 2019). Some language levels are used to show respect, while others are intended for casual daily interaction, including expressions that may be impolite and must be used carefully because they can be offensive in certain contexts. For example, in Balinese, more than four words exist to express the meaning of "death," and the appropriate choice depends on the social status of the person being referred to. Using an inappropriate term, such as a casual or low-level expression for an elder, is considered disrespectful and socially unacceptable within the community.

Given the importance of proper Balinese language-level usage in communication, especially on social media, there is a need for an automated system to classify these speech levels. Such a system

*Corresponding Author

can be applied to social media content moderation to detect inappropriate or offensive language, or to digital writing assistants that suggest more suitable speech levels to ensure respectful communication. One classification method used for this purpose is Multinomial Naive Bayes. This method is chosen because it is recognized as a state-of-the-art baseline for text classification (Raza et al., 2025) and is suitable for low-resource languages such as Balinese, where available datasets are relatively small. Multinomial Naive Bayes operates based on Bayes' theorem and term frequency. However, terms that appear too frequently in a document can reduce semantic significance. To address this issue, Term Frequency–Inverse Document Frequency (TF-IDF) feature extraction is employed to measure term importance. In addition, since excessively large feature sets can reduce classification performance (Hamzah, 2021), a feature selection process is required, such as Chi-square feature selection.

In text classification, several studies use the Multinomial Naive Bayes method because it is simple, fast, and easy to apply. One study that applies Multinomial Naive Bayes is Ardhana's (2018) classification of language levels in Javanese articles, which uses Multinomial Naive Bayes with N-Gram and TF-IDF feature extraction as well as the Synthetic Minority Over-Sampling Technique (SMOTE) resampling method. This study reports precision, recall, and accuracy of 72.67%, 75.00%, and 74.99%, respectively, when using TF-IDF and SMOTE, and finds that TF-IDF feature extraction yields better results than N-Gram.

Another study by Angeline et al. (2022) uses the Multinomial Naive Bayes method to classify dialects in Java. This study employs Term Frequency (TF) feature extraction and the SMOTE method to examine the effect of oversampling techniques on algorithm performance. The results show strong performance, achieving 96.97% accuracy, 97.53% precision, and 96.83% recall. In addition, Dewi and Putra (2021) apply the Naive Bayes method combined with rule-based and N-Gram stemming to detect hate speech and achieve an accuracy of 85%. Research conducted by Azad et al. (2021) on the Kurdish language also implements TF-IDF as the feature selection method.

This study adds novelty by applying the Multinomial Naive Bayes method with TF-IDF feature extraction, the SMOTE oversampling technique, and Chi-square feature selection to classify Balinese language levels. The stages carried out in this study include dataset collection, text preprocessing, feature extraction using TF-IDF, and feature selection using Chi-square. Furthermore, the study applies SMOTE oversampling, conducts hyperparameter tuning to identify optimal parameters, trains the Multinomial Naive Bayes model with the selected configuration, and evaluates the model using a confusion matrix and accuracy score.

II. METHODS

This study is conducted in several stages, as shown in Figure 1 (see Appendices). The first stage is data collection. The data used in this study consist of primary data collected through scraping posts and comments from social media platforms, particularly Instagram. The resulting dataset is stored in JavaScript Object Notation (JSON) format, as JSON facilitates data storage, especially for data with multiple nested attributes. The attributes stored from the scraping process include post text and the post Uniform Resource Locator (URL) or source.

The dataset is then filtered to include data from the last five years, exclude entries dominated by or containing more than half of their content in Balinese, and retain entries with a minimum length of five words. Next, the dataset is provided to Balinese language experts, namely *Penyuluh Bahasa Bali* and their team, who annotate or categorize the data based on appropriate speech levels. *Penyuluh Bahasa Bali* is a group of professionals appointed by the Cultural Department of Bali Province to preserve, foster, and develop the Balinese language, script, and literature. Finally, the annotated dataset is refined to ensure label consistency. All stages of the data collection process are illustrated in Figure 2 (see Appendices).

Since this study focuses on classification, labels are required for model training and evaluation. The labels used in this study take the form of Balinese language levels. According to Suwija (2019), the Balinese language is classified into six levels based on the vocabulary used in a sentence. Accordingly, this study uses six Balinese language levels: *alus singgih* (polite, used to show respect), *alus sor* (polite, used for self-humbling), *alus mider* (polite, used for both respecting others and self-humbling), *alus madya* (an intermediate level of politeness), *basa andap* (casual, commonly used in daily communication), and *basa kasar* (impolite, often used in arguments or when referring to animals).

Basa alus singgih is a Balinese language level used to show respect when addressing people of higher rank. *Basa alus sor* is a polite form of Balinese that is used to humble oneself (Suwija, 2019). *Basa alus mider* is a polite Balinese language form that serves both to show respect to others and to humble oneself. *Basa madia* is an intermediate level of the Balinese language, with a linguistic value that lies between *andap* and *alus*. *Basa andap* is a level of the Balinese language commonly used in everyday communication (Sosiawan et al., 2021). Finally, *basa kasar* is a Balinese language level characterized by harsh or offensive expressions, is considered very impolite, and is often used during arguments or when referring to animals

The next stage is text preprocessing, which converts the original text data into structured data and identifies the most significant text features for distinguishing between text categories (Hickman et al., 2022). Before the data are processed for the

training stage, they are prepared during the text preprocessing stage. The preprocessing procedures include case folding, removal of non-alphabetic characters, stop-word removal, stemming, and tokenization. The stemming technique used in this study adopts a rule-based approach grounded in the structural characteristics of the Balinese language. A study by Nugraha and Wardani (2020) employs a similar rule-based approach, while Agus et al. (2019) apply a comparable method by combining rule-based stemming with an N-gram approach.

The next step is feature extraction using TF-IDF. In this stage, a matrix is generated in which the rows represent the number of documents in the dataset and the columns represent the number of distinct tokens or features. The underlying principle assumes that a word or phrase that appears frequently in a document but occurs rarely in other documents possesses strong discriminative power and is suitable for classification (Gifari et al., 2022). The mathematical representation of term weighting in a document using TF-IDF is shown in Equation (1):

$$W(d,t) = tf(d,t) \times log\left(\frac{N}{df(t)}\right) \quad (1)$$

Here, d represents document d, t denotes term t, $tf(d, t)$ refers to the frequency of term t in document d, N indicates the total number of documents, and $df(t)$ represents the number of documents containing term t in the corpus (Zhou, 2022). After feature extraction is completed, the next step is feature selection using the Chi-square test. In this stage, features with high relevance or importance are selected to improve classification performance. Since the total number of extracted features is not predetermined, a percentage-based selection of all features is applied. Compared with other feature selection methods, such as Information Gain (IG), Mutual Information (MI), and Gini Coefficient (GI), the Chi-square method is recognized as one of the most effective algorithms (Bahassine et al., 2020). The Chi-square statistic measures the association between a feature or word t and class c. The distribution between word t and class c is presented in Table 1.

Table 1 Feature and Category

	c	**¬c**	**Total**
t	*A*	*B*	*A+B*
¬t	*C*	*D*	*C+D*
Total	*A+C*	*B+D*	*N*

It is assumed that feature t and class c follow a Chi-square distribution with one degree of freedom. A higher Chi-square score for class c indicates that feature t carries more category-related information and has greater relevance to class c. The formula of Chi-square for feature c with class c is shown in Equation (2) (Bahassine et al., 2020).

$$\chi^2(t,c) = \frac{N(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)} \quad (2)$$

After the important features are successfully selected, the next step is to perform SMOTE oversampling. Oversampling uses a non-majority approach, meaning that all minority or non-dominant classes are oversampled until their sample sizes match that of the majority class, so no minority classes remain underrepresented. In other words, the oversampling stage ensures that all classes have the same distribution or number of instances (Chen et al., 2024). This process helps reduce class imbalance and supports fairer learning during model training.

After oversampling, the next step is to perform hyperparameter tuning to identify the best hyperparameters for training the Multinomial Naive Bayes model. One common approach to hyperparameter tuning is grid search, which evaluates combinations of predetermined hyperparameter values to determine the optimal configuration (Hossain & Timmer, 2021). The hyperparameters explored in this study include the alpha parameter in Multinomial Naive Bayes, which functions as Laplacian smoothing, the number of features selected using the Chi-square method, and the application of SMOTE oversampling, as summarized in Table 2.

Table 2 Hyperparameter Options

No.	**Parameter**	**Options**
1.	Alpha	0.25, 0.5, 0.75, 1
2.	% Features	40%, 50%, 60%, 70%, 80%, 90%, 100%
3.	SMOTE	True, False

The hyperparameter tuning scheme uses K-fold cross-validation with K = 5 (Nti et al., 2021). This scheme works by dividing the dataset into training data and test data to ensure robust performance evaluation. In this study, the dataset is split into 80% training data and 20% testing data. After the best hyperparameters are obtained, the model is retrained using these optimal values. The stages of the hyperparameter tuning process are illustrated in Figure 3(see Appendices).

In the training stage, the model employed is Multinomial Naive Bayes, which is a probabilistic method commonly used for text classification (Zulfikar et al., 2023). In principle, Multinomial Naive Bayes calculates the probability that each word in a sentence belongs to each class by using prior and likelihood probabilities based on Bayes' theorem. When Multinomial Naive Bayes is combined with TF-IDF weighting, the formulation follows the equation shown in Equation (3) (Damanik & Setyohadi, 2021).

$$P(X_n|C) = \frac{\sum tf(X_n, d \in C) + \alpha}{\sum N_{d \in C} + V} \quad (3)$$

Where $\sum tf(X_n, d \in C)$ is the weighted sum of the words X_n across all training documents belonging to class C. Meanwhile, $\sum N_{d \in C}$ denotes the weighted sum of all terms appearing in the training documents within class C. Here, α is the Laplace smoothing parameter used to handle zero probabilities, and V refers to the total number of unique words or tokens in the training data.

To ensure end users can effectively use the system, an interface is developed as a simple website. The development process uses a prototyping model, which is an enhancement of the Software Development Life Cycle (SDLC) method. Unlike the traditional waterfall model, which is often criticized for its lengthy stages and high costs (Made et al., 2022), prototyping offers a more iterative and flexible approach that supports continuous refinement.

As shown in Figure 4 (see Appendices), the prototyping process begins with requirements gathering, which includes identifying both functional and non-functional system requirements. The next phase involves building a prototype or mid-fidelity prototype that serves as the initial interface design (Pramartha et al., 2023). This prototype is then evaluated to assess whether it meets the system requirements. If the evaluation indicates shortcomings, the process loops back either to the prototype design phase for refinement or to the requirements-gathering phase for additional input. Once the prototype satisfies the requirements, the system proceeds to the development phase, where it is implemented in code.

In the final evaluation stage, system performance is assessed using a confusion matrix, which provides a detailed view of prediction results in terms of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) (Valero-Carreras et al., 2023). In addition to accuracy metrics, the evaluation examines the influence of specific parameters, such as the alpha value, the SMOTE oversampling technique, and Chi-square feature selection, on the performance of the classification model. This study particularly analyzes how different combinations of these parameters affect the accuracy of the Multinomial Naive Bayes classifier and contribute to overall model effectiveness.

III. RESULTS AND DISCUSSIONS

The data in this study are obtained through web scraping techniques, specifically by extracting text from captions and comments on selected Instagram posts. The data are collected from several Instagram accounts, including @lanmalajah.id, @basabali.id, @basabaline, and @melajahbahasabali, between October 29 and November 7, 2024. These accounts are selected because they frequently post content in Balinese and have substantial followings, making them likely sources of rich and relevant linguistic data.

A targeted, account-based scraping approach is adopted to overcome the limitations of keyword-based searches, which often yield inaccurate or irrelevant results on Instagram. The scraping results remain raw data and are stored in JSON format. The collected information includes the post URL, caption, and comments. The combined caption and comment data total 18,224 entries, and an example of the raw dataset is shown in Figure 5 (see Appendices).

Table 3 Sample Data with Annotation

No.	Text (Balinese)	English Translation	Label (Language Level)
1	*Sekadi patut pisan nike kakak cantik…milet pisan titiang wikan sekadi kakak*	That's absolutely right, beautiful sister. I really want to be as smart as you.	*Alus Singgih*
2	*Becik pisan satua ne gek*	The story she told is very good.	*Alus Sor*
3	*Yening krame Banjar Boye je semeton Jero. sepatutne Ida dane sane patut.ampura*	If speaking with community members (*krama banjar*), do not use "*semeton*" or "*jero*"; instead, use "*ida dane*".	*Alus Mider*
4	*Becik pisan tatanan wacananne. Untuk memelihara kauningane puniki becikne lanturang ngripta pustaka wangun gancaran utawi irisan/ puisi.*	The structure of the text is very good. To retain your knowledge, you should continue creating other forms, such as prose (*gancaran*) or poetry.	*Alus Madia*
5	*Suud mebalih ne, Mare tiang nawang ternyata basa bali nika keweh*	After watching this, I realized Balinese is really hard.	*Basa Andap*
6	*Lengeh ci,ngoyong lengeh oyongan iban ci dikubu*	You're stupid, just stay in your little house.	*Basa Kasar*

The next stage is data filtering. From the initial 18,224 data points, entries that predominantly contain Balinese are selected, where dominance is defined as Balinese words accounting for more than half of the total words. Non-Balinese words in these entries are retained as long as the dominant Balinese threshold is met, resulting in 2,790 data points. The data are then filtered to include only entries with more than five words, producing a final dataset of 1,350 data points. This filtered dataset is annotated by four experts, including a Balinese language expert, according to Balinese language levels, and a sample of the annotated data is shown in Table 3.

Before text preprocessing, label standardization is performed to ensure uniform language level labels. This step is necessary because several labels provided by the annotators require normalization, such as differences in capitalization (e.g., *alus mider* and *Alus Mider*) and spelling errors (e.g., *alis singgih*, which should be *alus singgih*). After normalization into six classes—*alus singgih*, *alus sor*, *alus mider*, *alus madia*, *basa andap*, and *basa kasar*—the final dataset consists of 1,314 data points, with the class distribution shown in Figure 6 (see Appendices).

From the distribution shown in Figure 6 (see Appendices), the dataset appears to be highly imbalanced, as the *basa kasar* class contains only 12 data points, representing 0.9% of the total dataset. In contrast, the *basa andap* label dominates the dataset with 549 records, accounting for approximately 41.8% of all data. The small number of *basa kasar* instances reflects its limited usage, particularly on social media, where such language is considered inappropriate in public discourse and may convey rudeness, especially in non-verbal contexts. Next, the dataset of 1,314 data points is divided into 80% training data and 20% test data, resulting in 1,052 training instances and 262 test instances. The distribution of the training data is illustrated in Figure 7 (see Appendices).

Meanwhile, the distribution of test data classes is shown in Figure 8 (see Appendices). Both the training and test datasets exhibit an imbalanced class distribution, which may affect classification performance. To address this issue, the training data are resampled using SMOTE, and the impact of this resampling is evaluated during the hyperparameter tuning process. The test data remain unaltered, as they are intended to represent real-world conditions.

The existing training data are then used for cross-validation and hyperparameter tuning to determine parameter values that yield the best performance. The parameters used in this process follow the configuration presented in Table 2, including alpha values of 0.25, 0.5, 0.75, and 1. The number of features is selected using the Chi-square feature selection method, with options of 40%, 50%, 60%, 70%, 80%, 90%, and 100% of the total features, corresponding to 1,360, 1,700, 2,040, 2,380, 2,720, 3,060, and 3,400 features, respectively. In addition, the use of SMOTE oversampling is evaluated, where the applied scheme is non-majority oversampling, meaning that all classes other than the majority class are oversampled..

With these parameters, a total of 56 parameter combinations are obtained. The process of identifying the optimal parameters involves training the Multinomial Naive Bayes model on the training data using cross-validation, with performance evaluated based on accuracy. After hyperparameter tuning, the comparison results are presented in Figure 9 (see Appendices), which shows boxplots of accuracy for datasets with and without oversampling. The results indicate that the oversampled dataset achieves a higher average accuracy.

Figure 10 (see Appendices) illustrates the comparison of average accuracy as a function of the number of selected features and the alpha value during hyperparameter tuning with SMOTE oversampling. The results show that using a small feature subset (40%) is insufficient to achieve optimal model performance. However, incorporating all available features also does not guarantee the best performance, indicating that an intermediate number of features yields more effective classification results.

Figure 11 (see Appendices) shows the relationship between the number of features, the alpha value, and the average accuracy during hyperparameter tuning on a dataset without SMOTE oversampling. The graph demonstrates that selecting appropriate alpha values and feature counts affects the model's performance. However, without oversampling, the overall performance remains significantly lower than that achieved with SMOTE. Based on the results in Figures 10 and 11, it is necessary to select an appropriate combination of feature numbers and

Table 4 Best Parameter from Hyperparameter Tuning

Parameter	Value
Alpha	0.25
Features	2720
% Features	80%
SMOTE	True

hyperparameter values to obtain the highest average accuracy. Consequently, the model achieves the best accuracy of 91.78% as shown in Figure 10 (see Appendices), and it is trained using the parameter configuration presented in Table 4.

From Table 4, the best parameters consist of using 80% of the total features (2,720 features), selecting features with the highest Chi-square values, setting alpha to 0.25, and applying SMOTE oversampling. Once SMOTE oversampling is identified as the optimal setting, the training data are oversampled, producing the distribution shown in Figure 12 (see Appendices). The not-majority oversampling strategy resamples all classes that are not the majority class, which in this case excludes the *basa andap* class. After this step, the Multinomial Naive Bayes model is retrained using the selected optimal parameters.

Based on Table 5, the model trained with the best parameters achieves an accuracy of 96.53%. Overall, the model performs well across all classes, particularly for the *basa kasar* class, which achieves perfect recall (1.0) and the highest F1-score of 0.9955. These results indicate that the *basa kasar* class has distinct features and is easily recognized by the model. Meanwhile, the *alus mider* class records the lowest scores, indicating that the model experiences greater difficulty in classifying this class. This finding is consistent with the fact that many words in *alus mider* are also commonly used in *alus singgih* and *alus sor*, such as *semeton*, *bali*, *wenten*, *sampun*, *ring*, *puniki*, and *sane*, which increases ambiguity among these classes.

The confusion matrix in Figure 13 (see Appendices) indicates that the model effectively learns the training data, as reflected by the strong concentration of values along the main diagonal. This pattern suggests that most instances are correctly classified across the six Balinese language levels. However, the matrix also reveals the model's difficulty in classifying the *alus mider* class, with 19 instances misclassified as *alus madia* and 13 instances misclassified as *alus sor*.

Based on the testing results presented in Table 6, the model achieves an accuracy of 61.45%. The precision and recall values indicate that the model performs well in classifying the *basa andap* class. However, it shows considerable difficulty in accurately categorizing the *alus sor* and *alus singgih* classes.

The confusion matrix for the test data in Figure 14 (see Appendices) shows that a significant number of *alus singgih* instances are misclassified as *alus sor* and *alus mider*. Additionally, many instances from the *alus mider* class are incorrectly classified as *alus madia*. These results indicate that the model continues to struggle to distinguish between closely related Balinese language levels.

Based on the existing results, the model still cannot generalize effectively on the test data, even though it performs well on the training data augmented with SMOTE. This behavior indicates overfitting, as evidenced by the large gap between the training and test accuracies. One reason is that the test dataset is unbalanced, and oversampling cannot be applied to it because it must remain unseen and reflect the actual data distribution. Furthermore, because the number of instances in some minority classes in the test data is very small, the model has difficulty making accurate predictions when new words appear that are not present in the training data. In addition, most common words, such as *sampun*, *bali*, *semeton*, *puniki*, *sane*, *wenten*, and *ring*, frequently appear across *alus singgih*, *alus sor*, *alus mider*, and *alus madia*. This pattern suggests

Table 5 Evaluation Results on Training Data

Class	Precision	Recall	F1-Score	Accuracy
Alus Singgih	0.9664	0.9751	0.9707	0.9653
Alus Sor	0.9564	0.9932	0.9745	
Alus Mider	0.9416	0.9118	0.9264	
Alus Madia	0.9416	0.9842	0.9624	
Basa Andap	0.9976	0.9276	0.9613	
Basa Kasar	0.9910	1.0000	0.9955	

Table 6 Evaluation Results on Testing Data

Class	Precision	Recall	F1-Score	Accuracy
Alus Singgih	0.4118	0.4118	0.4118	0.6145
Alus Sor	0.3103	0.6000	0.4091	
Alus Mider	0.5968	0.5606	0.5781	
Alus Madia	0.3696	0.4595	0.4096	
Basa Andap	0.9432	0.7757	0.8513	
Basa Kasar	0.3333	0.3333	0.3333	

that the distinction between these levels does not rely on shared vocabulary but instead depends on the subtle use of specific, less frequent discriminative words, which the model struggles to differentiate across classes.

After model evaluation, the model is integrated into the end-user system, allowing users to classify Balinese language levels easily. The prototype of the end-user system is shown in Figure 15 (see Appendices). After entering a sentence, the user can view the predicted level of that sentence along with detailed prediction results, including the probability for each class.

IV. CONCLUSIONS

This study aims to address the challenge of automatically classifying the levels of Balinese text on social media. Using the Multinomial Naive Bayes method to classify 1,314 data points, the results show an accuracy of 96.53% on the training data and 61.45% on the test data. The optimal number of features, selected using Chi-square and SMOTE oversampling during hyperparameter tuning, yields the best performance during training. A significant limitation identified during the experiment is overfitting, as indicated by the large gap between the training and test accuracies. Despite these limitations, this study provides a machine learning approach for a low-resource language classification task. It produces a dataset and a baseline model that serve as a foundation for future research.

Based on the study and its results, future work should prioritize expanding the dataset to achieve a more balanced class distribution, thereby improving model performance. In addition, future studies should explore more diverse data collection methods, as this study relies on data from only a limited number of accounts. This approach allows the resulting dataset to be more varied and comprehensive, covering a broader range of topics. To further improve performance and better capture contextual information, future research should explore more advanced computational methods, such as deep learning or Transformer-based models.

AUTHOR CONTRIBUTIONS

Conceived and designed the analysis, P. W. A. W. and C. P.; Collected the data, P. W. A. W.; Contributed data or analysis tools, P. W. A. W. and C. P.; Performed the analysis, P. W. A. W.; Wrote the paper, P. W. A. W. and C. P.; Other contribution, I. G. N. A. C. P. and L. G. A.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Zenodo at https://doi.org/10.5281/zenodo.17582015, reference number 17582015.

REFERENCES

Agus, M., Subali, P., & Fatichah, C. (2019). Kombinasi metode Rule-Based dan N-Gram Stemming untuk mengenali stemmer bahasa Bali. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *6*(2), 219–228. https://doi.org/10.25126/JTIIK.2019621105

Angeline, G., Wibawa, A. P., & Pujianto, U. (2022). Klasifikasi dialek bahasa Jawa menggunakan metode Naives Bayes. *Jurnal Mnemonic*, *5*(2), 103–110. https://doi.org/10.36040/mnemonic.v5i2.4748

Ardhana, A. P. (2018). *Klasifikasi Tingkatan Bahasa pada Artikel Berbahasa Jawa dengan Metode Multinomial Naïve Bayes*.[Under Graduate Thesis, Universitas Sebelas Maret]. UNS Institutional Repository.

Azad, R., Mohammed, B., Mahmud, R., Zrar, L., & Sdiq, S. (2021). Fake News Detection in low-resourced languages "Kurdish language" using Machine learning algorithms. *Turkish Journal of Computer and Mathematics Education*, *12*(6), 4219–4225.

Bahassine, S., Madani, A., Al-Sarem, M., & Kissi, M. (2020). Feature selection using an improved Chi-square for Arabic text classification. *Journal of King Saud University - Computer and Information Sciences*, *32*(2), 225–231. https://doi.org/10.1016/J.JKSUCI.2018.05.010

Chen, W., Yang, K., Yu, Z., Shi, Y., & Chen, C. L. P. (2024). A survey on imbalanced learning: Latest research, applications and future directions. *Artificial Intelligence Review*, *57*(6), 1–51. https://doi.org/10.1007/S10462-024-10759-6/FIGURES/11

Damanik, F. J., & Setyohadi, D. B. (2021). Analysis of public sentiment about Covid-19 in Indonesia on twitter using multinomial Naive Bayes and Support Vector Machine. *IOP Conference Series: Earth and Environmental Science*, *704*(1), 012027. https://doi.org/10.1088/1755-1315/704/1/012027

Dewi, D. A. E. R., & Putra, A. A. N. M. A. (2021). Kebencian pada bahasa Bali dengan metode Naive Bayes. *Jurnal Teknologi Informasi Dan Komputer*, *7*(2).

Gifari, O. I., Adha, M., Rifky Hendrawan, I., Freddy, F., & Durrand, S. (2022). Film review sentiment analysis using TF-IDF and Support Vector Machine. *Journal of Information Technology*, *2*(1), 36–40. https://doi.org/10.46229/JIFOTECH.V2I1.330

Hamzah, M. B. (2021). Classification of movie review sentiment analysis using Chi-Square and Multinomial Naive Bayes with Adaptive Boosting. *Journal of Advances in Information Systems and Technology, 3*(1), 67–74. https://doi.org/10.15294/JAIST.V3I1.49098

Hickman, L., Thapa, S., Tay, L., Cao, M., & Srinivasan, P. (2022). Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations. *Organizational Research Methods*, *25*(1), 114–146. https://doi.org/10.1177/1094428120971683

Hossain, R., & Timmer, D. (2021). Machine learning model optimization with hyper parameter Tuning Approach. *Glob. J. Comput. Sci. Technol. D Neural Artif. Intell*, *21*(2).

Made, I., Wirawan, W., & Pramartha, C. (2022). Pengembangan sistem informasi penanganan penderita gangguan jiwa dengan pendekatan Enteprise Systems. *Sintech (Science and Information Technology) Journal*, *5*(1), 31–41. https://doi.org/10.31598/SINTECHJOURNAL.V5I1.1070

Mastini, G. N., Kantriani, N. K., & Arini, N. W. (2021). Peran media sosial instagram dalam upaya menjaga eksistensi bahasa Bali. *Ganaya : Jurnal Ilmu Sosial Dan Humaniora*, *4*(2), 686–695. https://doi.org/10.37329/ganaya.v4i2.1414

Nti, I. K., Nyarko-Boateng, O., & Aning, J. (2021). Performance of machine learning algorithms with different K Values in K-fold CrossValidation. *International Journal of Information Technology and Computer Science*, *13*(6), 61–71. https://doi.org/10.5815/IJITCS.2021.06.05

Nugraha, P. G. S. C., & Wardani, N. W. (2020). Stemming dokumen teks bahasa bali dengan metode rule base approach. *JATISI*, *7*(3), 510–521. https://doi.org/10.35957/JATISI.V7I3.538

Pramartha, C., Made, I., Mahendra, Y., Primahadi, G., Rajeg, W., & Arka, W. (2023). The development of semantic dictionary prototype for the Balinese Language. *International Journal of Cyber and IT Service Management (IJCITSM)*, *3*(2), 96–106. https://doi.org/10.34306/IJCITSM.V3I2.132

Sosiawan, P., Martha, I. N., & Artika, I. W. (2021). Penggunaan bahasa bali pada keluarga muda di kota Singaraja. *Jurnal Pendidikan Dan Pembelajaran Bahasa Indonesia*, *10*(1), 40–54. https://doi.org/10.23887/JURNAL_BAHASA.V10I1.403

Raza, M. O., Mahoto, N. A., Shaikh, A., Pathan, N., Alshahrani, H., & Elmagzoub, M. A. (2025). A Machine Learning Approach of text classification for high-and low-resource languages. *Computational Intelligence*, *41*(4), e70114. https://doi.org/10.1111/COIN.70114

Suardiana, I. W. (2019). Bahasa Bali dan pemertahanan kearifan Lokal. *Linguistika*, *19*(1), 1–7.

Suwija, I. (2019). Tingkat-tingkatan bicara bahasa bali (dampak anggah-ungguh kruna). *Sosiohumaniora*, *21*(1), 90. https://doi.org/10.24198/sosiohumaniora.v21i1.19507

Valero-Carreras, D., Alcaraz, J., & Landete, M. (2023). Comparing two SVM models through different metrics based on the confusion matrix. *Computers & Operations Research*, *152*, 106131. https://doi.org/10.1016/J.COR.2022.106131

Zhou, H. (2022). Research of text classification based on TF-IDF and CNN-LSTM. *Journal of Physics: Conference Series*, *2171*(1), 012021. https://doi.org/10.1088/1742-6596/2171/1/012021

Zulfikar, W. B., Atmadja, A. R., & Pratama, S. F. (2023). Sentiment analysis on social media against public policy using multinomial naive bayes. *Scientific Journal of Informatics*, *10*(1), 25–34. https://doi.org/10.15294/SJI.V10I1.39952

APPENDICES

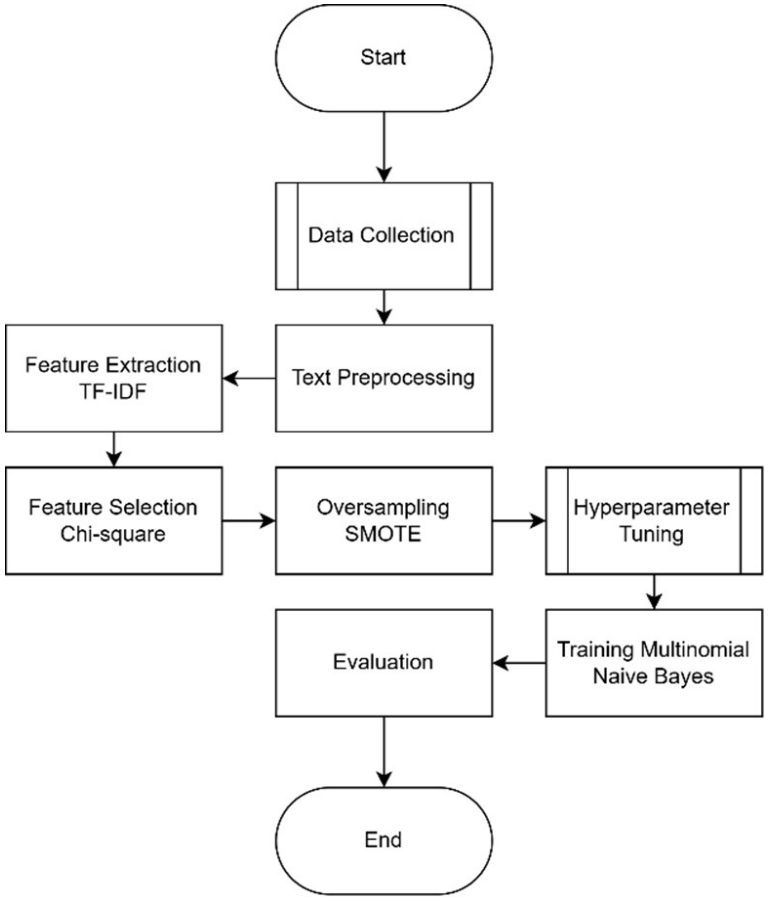

Figure 1 Research Flow

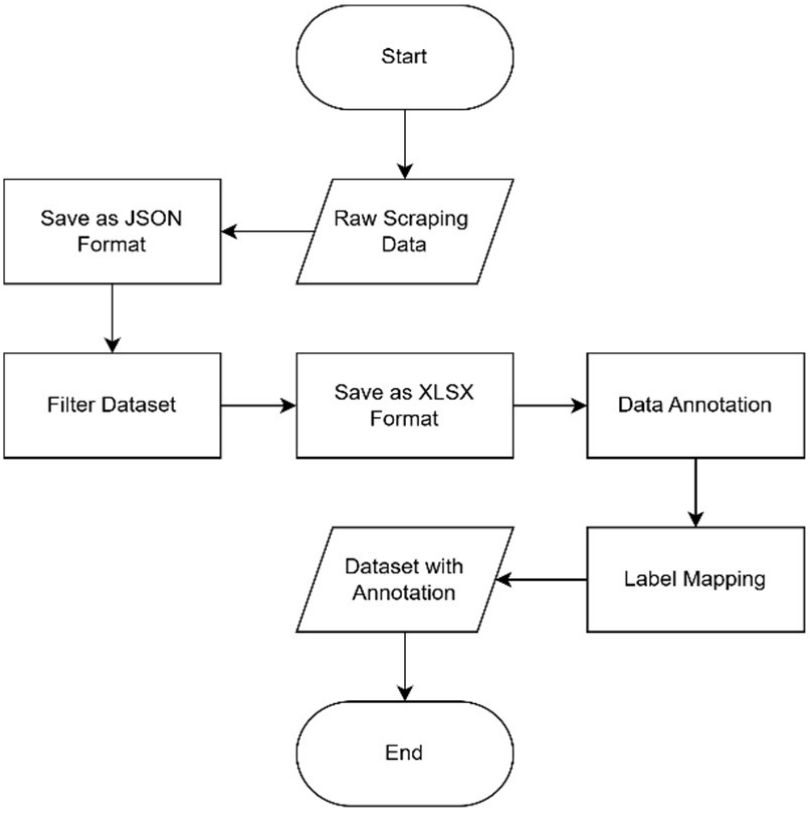

Figure 2 Data Collection Flow

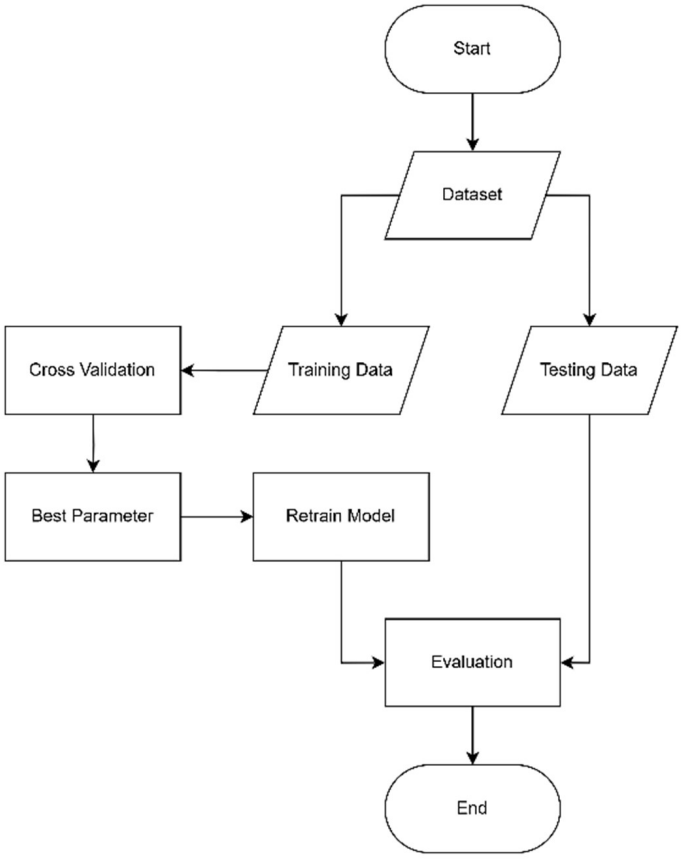

Figure 3 Hyperparameter Tuning Flow

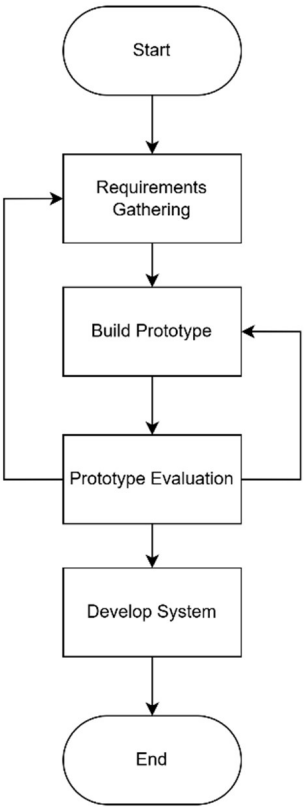

Figure 4 Prototyping Flow

```
{
  "data": {
    "https://www.instagram.com/p/CRfRXL5pfwP/": {
      "caption": "#basabali #basabaline #bahasabali #belajarbahasabali",
      "comments": [
        "Sukseme , tiang melajah mebase bali",
        "@petuah_petuah Ngiring sareng-sareng malajah \ud83d\ude4f",
        "@basabali.id sawire timpal timpal dini uli Bali\ud83d\ude4f",
        "Anadap & alus lebih sopan mana min?",
        "Bli, apakah ada les bahasa bali? Karna saya pengen belajar",
        "Cokor dalam bahasa Sunda malah bahasa kasar untuk kaki",
        "Jawa (Silit:Anus)",
        "@herdi_ryan Halo kak.. Sama dengan bahasa Bali ya kak \ud83d\ude0a"
      ]
  },
  "author": "Putu Widyantara Artanta Wibawa",
  "updated_at": "2024-11-06T12:22:22.204537"
}
```


Figure 5 Sample Dataset

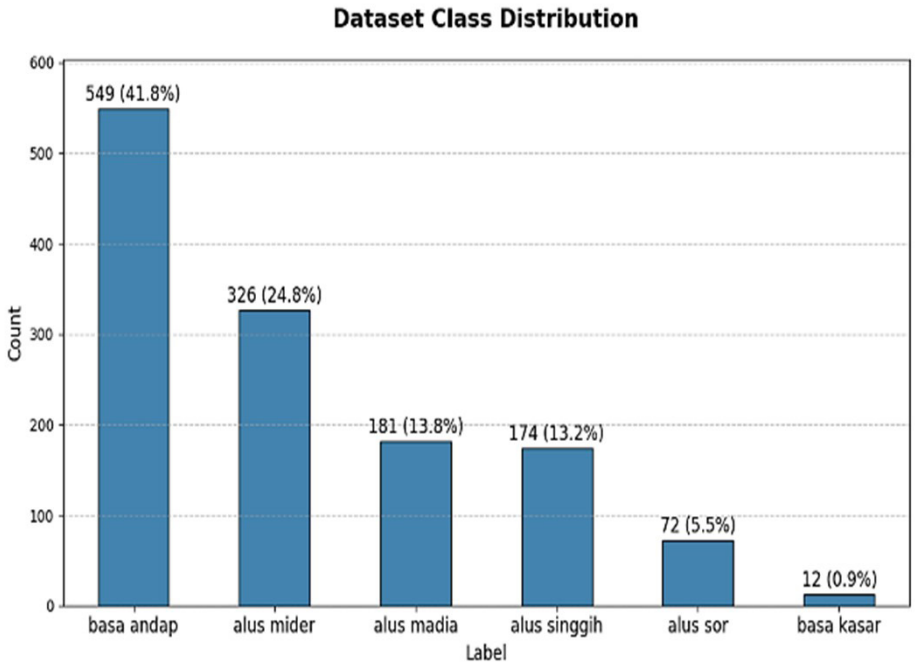

Figure 6 Dataset Class Distribution

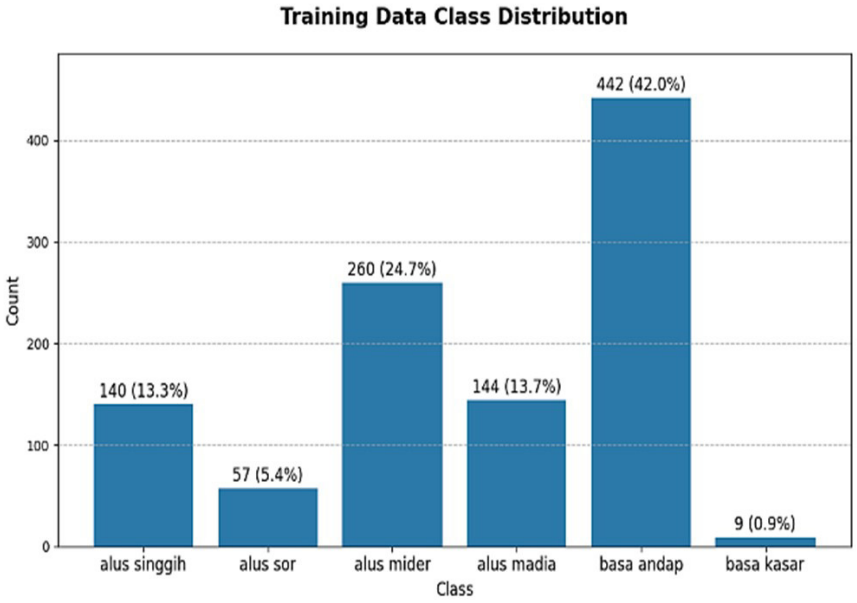

Figure 7 Training Data Class Distribution

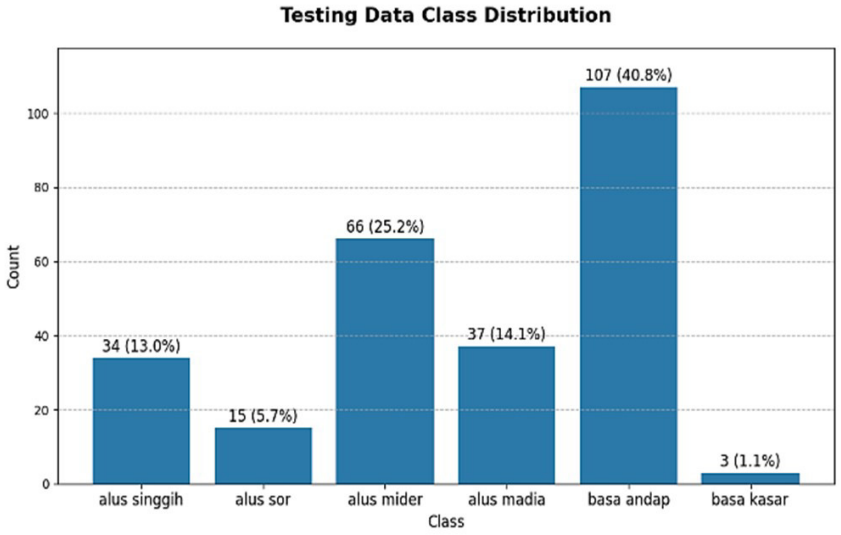

Figure 8 Testing Data Class Distribution

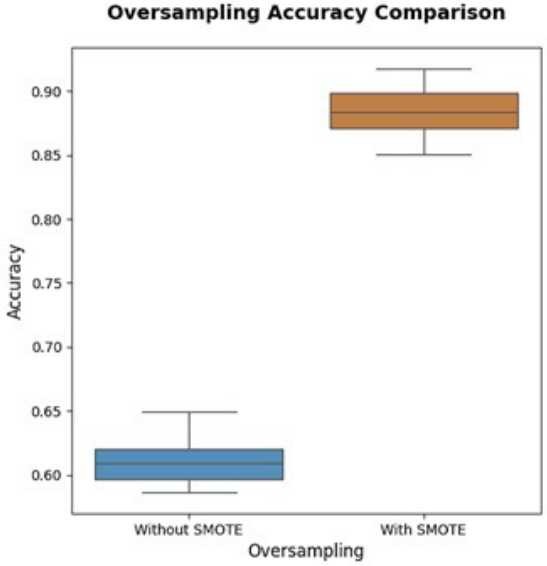

Figure 9 Oversampling Accuracy Comparison

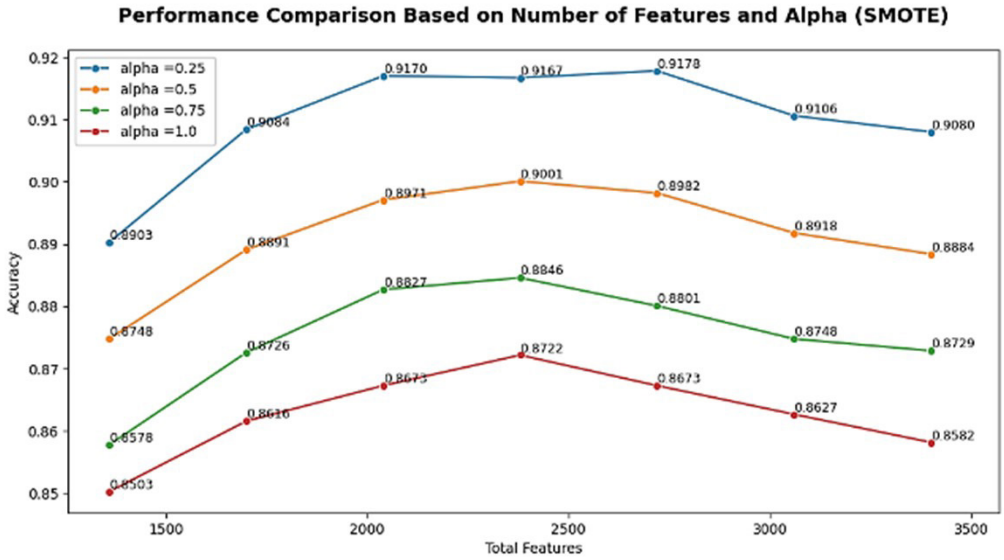

Figure 10 Performance Comparison Based on Number of Features and Alpha (SMOTE)

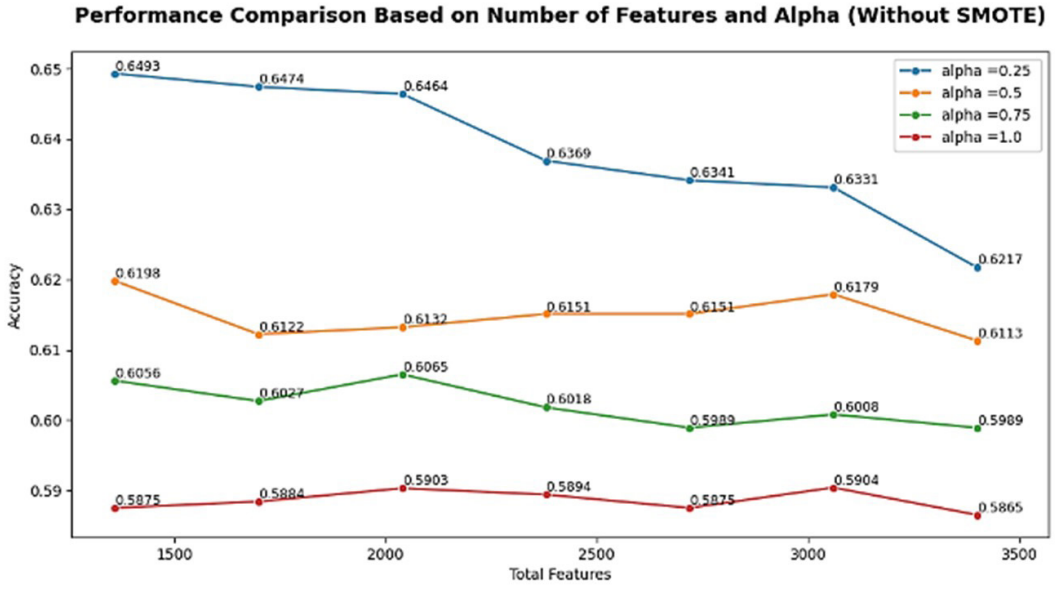

Figure 11 Performance Comparison Based on Number of Features and Alpha (Without SMOTE)

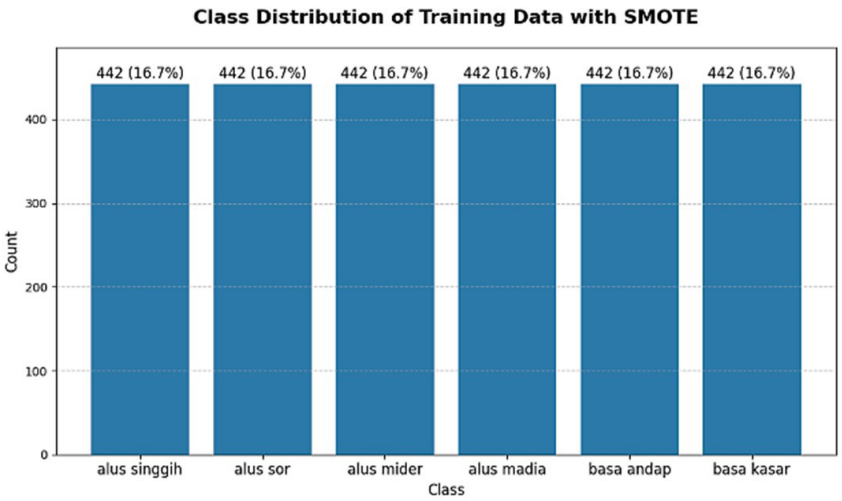

Figure 12 Class Distribution of Training Data with SMOTE

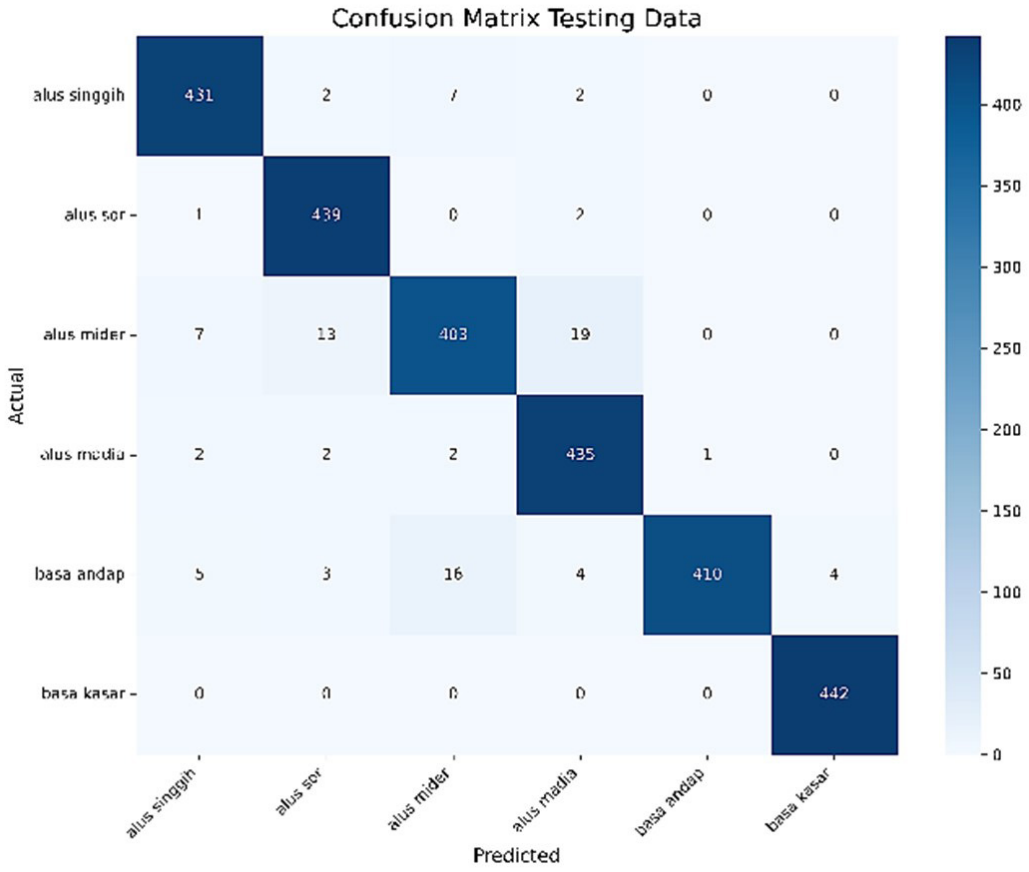

Figure 13 Confusion Matrix Training Data

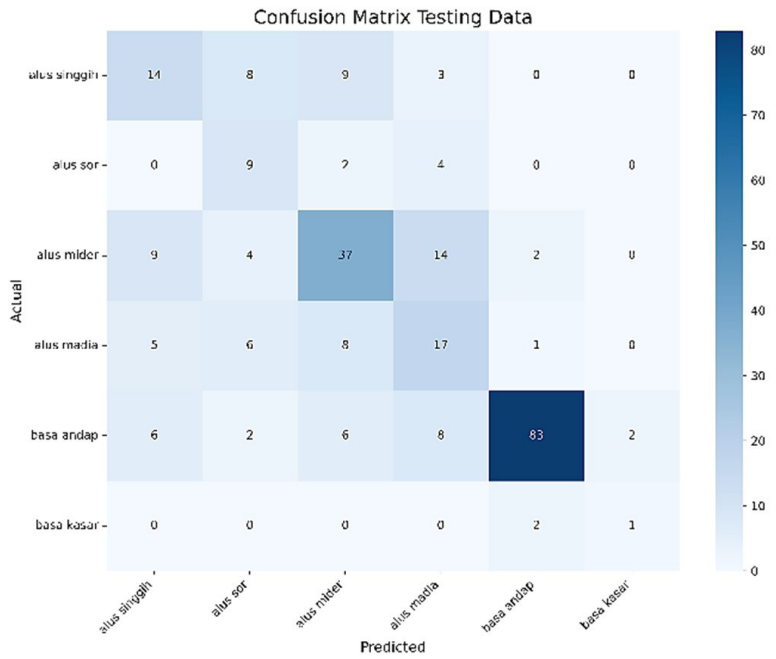

Figure 14 Confusion Matrix Testing Data

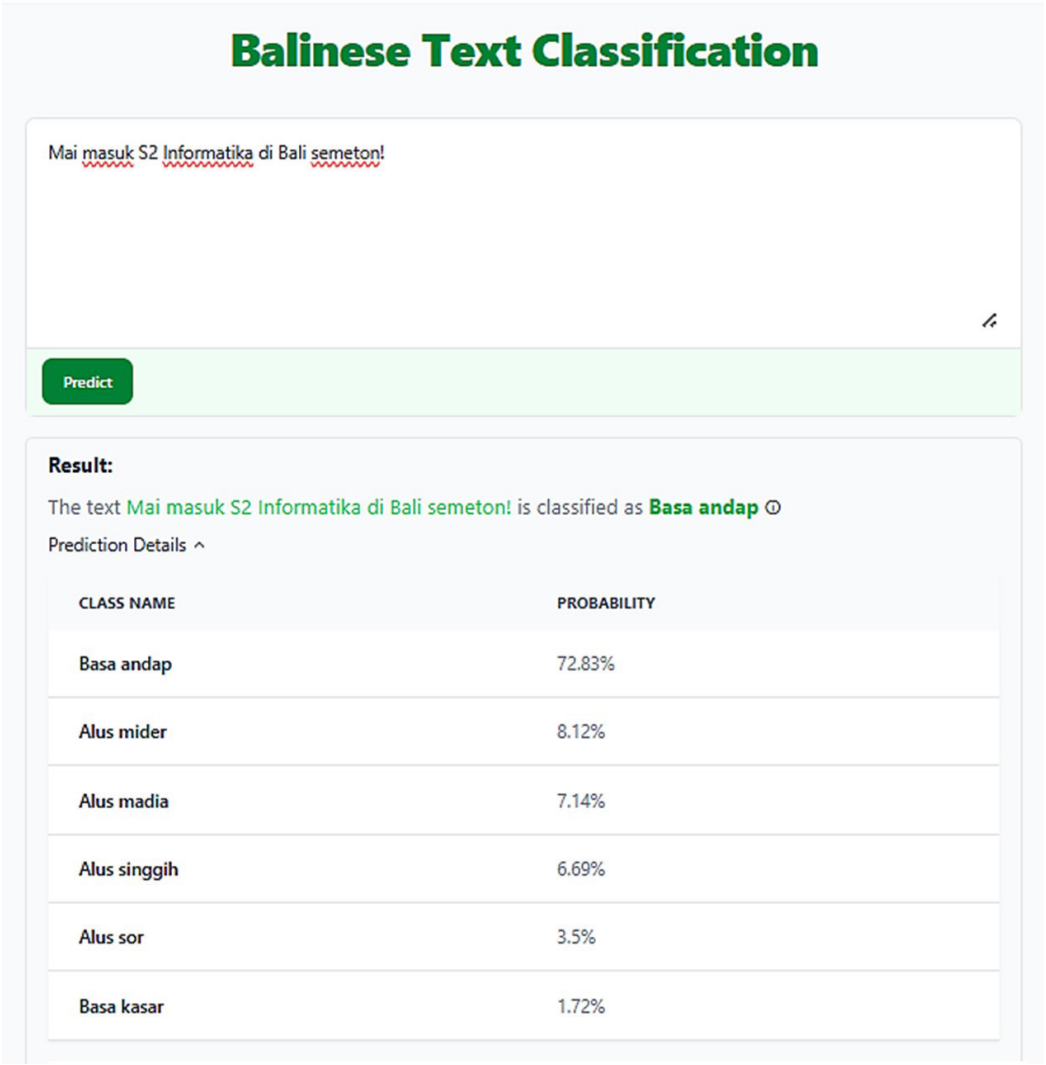

Figure 15 System Interface