JURNAL ELTEK
Vol. 24, No. 2, October 2026, pp. 93~101
p-ISSN: 1693 – 4024 | e-ISSN: 2355-0740
DOI: 10.33795/eltek.v24i2.10852

Code-mixed sentiment analysis of Prabowo-Gibran on social media X using mBERT model

Alifia Puspaningrum[1], Nurul Fauziah[2], Muhamad Mustamiin[3], Meyer Mega Eklesia Silaban[4]
e-mail: alifia.puspaningrum@polindra.ac.id, nurul.fauziah@polindra.ac.id, mustamiin@polindra.ac.id, meyermega@polindra.ac.id
[1,2,3,4]Department of Informatics Engineering, Politeknik Negeri Indramayu, Indonesia

Article Information

Article History

Received August 2, 2026
Revised September 28, 2026
Published October 30, 2026

***Keywords*:**

Code-mixed,
Imbalanced data,
Opinion classification,
mBERT,
SMOTE

ABSTRACT

The sociopolitical dynamics following the 2024 Indonesian Presidential Election have significantly increased the intensity of public discourse on social media. Polarized opinions regarding various strategic policies are frequently expressed through Indonesian–English code-mixed text. Such texts are characterized by language switching, lexical variations, and semantic ambiguity, posing a major challenge for conventional Natural Language Processing (NLP) methods to accurately capture linguistic markers. This study evaluates Multilingual Bidirectional Encoder Representations from Transformers (mBERT) for opinion classification on Indonesian–English code-mixed text. mBERT was selected because it is a pre-trained model trained on monolingual corpora of 104 languages, granting it cross-lingual transfer capabilities without the need for explicit supervision or bilingual alignment. Additionally, to address the class imbalance issue in the dataset, this study applies the Synthetic Minority Over-sampling Technique (SMOTE). Experimental results demonstrate that mBERT achieved the best performance compared to other models, reaching an accuracy of 82.26% without SMOTE. Following the application of SMOTE, the accuracy increased to 83.00% (an increase of 0.74 percentage points). Although this overall numerical increase appears moderate, the value holds crucial significance: the model successfully mitigated the bias toward majority class predictions. This 0.74% increase represents a significant improvement in precisely recognizing minority classes, as evidenced by vastly more balanced class-wise evaluation metrics. In conclusion, mBERT is a highly promising approach for code-mixed opinion classification, while SMOTE plays an essential role in ensuring predictive fairness on imbalanced datasets.

Corresponding Author:
Alifia Puspaningrum,
Department of Informatics Engineering,
Politeknik Negeri Indramayu,
Jl. Lohbener Lama No. 8, Indramayu, West Java, Indonesia.
Email: alifia.puspaningrum@polindra.ac.id

1. INTRODUCTION

Social media has become an important channels for the public to express support, criticism, and evaluation of political actors and policies. Following the 2024 Indonesian Presidential Election, public discourse has not only centered on electoral contestation but has also expanded to include evaluations of the early phase of the Prabowo Subianto–Gibran Rakabuming Raka administration [1], [23]. One policy that has received wide attention is the Free Nutritious Meals Program (Makan Bergizi Gratis, MBG). Sianturi [2] analyzed 4,041 news reports and 138,301 social media posts collected on 6–12 January 2025. In the social media data, 13.1% of posts were classified as positive, 41.2% as negative, and 45.7% as neutral, which indicates the heterogeneity of public responses to the communication and early implementation of the policy. In addition, the dataset provided by Firdaus et al. [3] contains 30,000 tweets about three presidential candidates, illustrating the large volume of political opinion data that can be used to systematically identify tendencies in public perception through sentiment analysis.

Sentiment analysis of Indonesian-language user-generated content faces various challenges because social media text often contains non-standard language, abbreviations, slang, spelling variations, sarcasm, and Indonesian–English code-mixing. Language switching at the token, phrase, or clause level increases lexical and syntactic variation, creates semantic ambiguity, and limits the vocabulary coverage of models developed using monolingual corpora [4], [5]. Sparse representations such as Bag-of-Words and TF-IDF are relatively efficient as features for Support Vector Machine (SVM), Naïve Bayes, and Random Forest, but they do not directly represent word meaning based on context. Rizal and Stymne [6] showed that code-mixed embeddings trained on a synthetic corpus can achieve performance at least comparable to cross-lingual embeddings and better than monolingual embeddings. Although LSTM and BiLSTM are able to learn sequential dependencies, their performance is still affected by the limited availability of annotated data and by the high variation across languages and domains [4]. These conditions indicate the need for contextual representations that are more adaptive to the characteristics of multilingual and informal text [7].

Vaswani et al. [8] introduced the Transformer architecture, which uses a self-attention mechanism to model relationships between tokens without relying on recurrence or convolution. BERT subsequently produced deep bidirectional contextual representations through pre-training on a large-scale corpus and can be adapted to various downstream tasks through fine-tuning [9]. Multilingual BERT, or mBERT, extends this approach by pre-training on monolingual corpora of 104 languages and demonstrates zero-shot cross-lingual transfer ability without explicit cross-lingual supervision [10]. The GLUECoS benchmark shows that multilingual models are able to outperform several cross-lingual baselines on code-switched NLP tasks, including sentiment analysis, and generally obtain the best results after fine-tuning on code-switched data [11]. BERTologiCoMix also shows that fine-tuning on natural code-mixed data yields better performance gains and makes a number of mBERT attention heads more responsive to code-mixed input [12].

In the Indonesian context, IndoNLU [13] and IndoLEM [14] provide benchmarks, datasets, and pre-trained language models for various Natural Language Understanding tasks. IndoBERT was developed using an Indonesian-language corpus, whereas IndoBERTweet was specifically designed for the language characteristics of Twitter through domain-specific vocabulary initialization [15]. However, good performance on monolingual data does not always guarantee robustness when a model is confronted with code-mixed text. IndoRobusta evaluated model robustness against various combinations of Indonesian and local languages and showed that changes in language combinations remain a challenge for pre-trained models [16]. Meanwhile, the IndoJavE model, which was specifically pre-trained on Indonesian–Javanese–English text, achieved better performance than a number of general monolingual and multilingual models on various downstream tasks [17]. These findings underscore the importance of alignment between pre-training data, language combinations, and the target domain in processing code-mixed text.

A number of studies have evaluated sentiment analysis on Indonesian–English code-mixed text using Transformers, mBERT, IndoBERTweet, back-translation, and DistilBERT [19]–[22]. In the political domain, Hashmi et al. [18] evaluated multilingual Transformers on election tweets in Roman Urdu–English, while Sidauruk and Herowati [23] applied IndoBERT to analyze public discourse on the early phase of the Prabowo–Gibran administration. Research on the MBG policy has also combined TF-IDF features, IndoBERT embeddings, lexicon-based features, machine learning, and SMOTE, but used Indonesian-language tweets and did not specifically focus on Indonesian–English code-mixed text [24].

Based on these conditions, this study aims to evaluate mBERT for binary sentiment classification, namely positive and negative sentiment, on Indonesian–English code-mixed political text. The performance of mBERT is compared with BERT, IndoBERT, DistilBERT, LSTM, BiLSTM, SVM, Naïve Bayes, and Random Forest using the same experimental setting. This study also evaluates the application of the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance in the training data [25].

2. RESEARCH METHOD

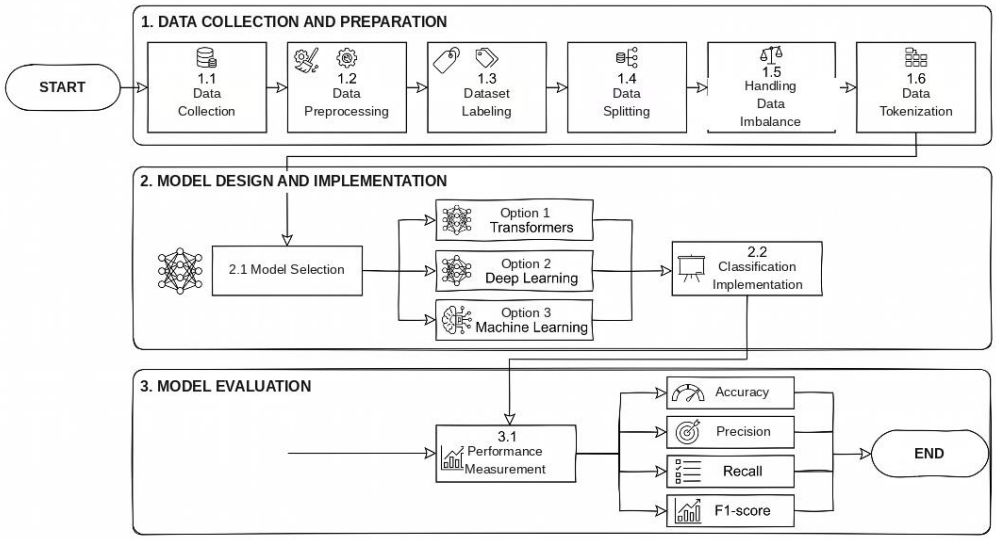


Figure 1. Code-Mixed Text Classification Flowchart

This section describes the research method used to build and evaluate a sentiment classification model for public opinion on the Prabowo–Gibran administration on platform X, as shown in Figure 1. Based on Figure 1, the following is an explanation of each stage:

1. Data Collection and Preparation

   This stage focuses on obtaining and cleaning the text data before it is processed by the model.

   a. Data Collection

      The dataset was collected through crawling using the X application platform. Data were selected based on predetermined keywords, focusing on the term of office of President and Vice President Prabowo-Gibran, from their inauguration in October 2024 to April 2025, covering six months of their administration, and yielding 2,166 code-mixed text records. To comply with the terms of service of platform X, all user names (*username*) and identifying information were strictly masked (*anonymized*). The keywords used in the data collection process are as follows:

      - Prabowo Subianto
      - Gibran Rakabuming Raka
      - Kabinet Merah Putih (Red and White Cabinet)
      - Lapor Mas Wapres! (Report to the Vice President!)
      - Efisiensi Anggaran (Budget Efficiency)
      - Danantara
      - Bank Emas (Gold Bank)
      - Kenaikan UMP 6,5% (6.5% Minimum Wage Increase)
      - Kenaikan PPH 12% (12% Income Tax Increase)
      - Makan Bergizi Gratis (Free Nutritious Meals)
      - LPG Gas

   b. Data Preprocessing

      In the preprocessing stage, the data was first processed to meet the input standard required by the model. The first step was dataset cleaning by removing unused elements such as URLs, emojis, punctuation, numbers, and excess whitespace, after which the text was converted to lowercase, and then word normalization was carried out using a slang dictionary taken from a Kaggle dataset https://www.kaggle.com/datasets/fornigulo/kamus-slag.

   c. Dataset Labeling:

      After the data was collected, the next step was the dataset labeling process. For the dataset containing code-mixed text, labeling was done manually with the help of annotators. To ensure labeling quality, the annotation protocol involved two independent annotators with a linguistics background in language mixing (code-mixing), so that local colloquialisms could be interpreted accurately. Meanwhile, Indonesian-language text was labeled using a lexicon-based approach with the InSet lexicon, the Indonesia Sentiment Lexicon, available on GitHub https://github.com/fajri91/InSet. For text containing English, labeling was carried out using TextBlob.

d. Data Splitting:
The dataset was split into 80% training data and 20% testing data using a stratified split to maintain proportional balance.

e. Handling Data Imbalance:
The class distribution in the dataset shows an imbalance in the number of positive and negative sentiment labels across the three categories. To address this problem, the researchers applied the SMOTE (Synthetic Minority Oversampling Technique) method, which works by generating new synthetic data for the minority class until its number equals that of the majority class. In this study, SMOTE was implemented on the numerical feature representation to balance the sentiment class distribution before model training. The data after SMOTE totaled 2,562 records.

f. Data Tokenization
The final stage of preprocessing before the data is used by the model is tokenization, i.e., the process of converting clean text into numerical sequences compatible with the model architecture. When processing code-mixed text, which often combines English root words with Indonesian affixes and informal slang, the tokenizer segments words in different ways. Subword-based tokenization prevents out-of-vocabulary (OOV) errors by breaking rarely occurring code-mixed terms into meaningful sub-units, which ultimately directly affects the comparative performance between models. Because this study is comparative, each Transformer model uses a specific tokenizer suited to its own architecture:
- mBERT: bert-base-multilingual-uncased
- IndoBERT: indobenchmark/indobert-base-p1
- BERT & DistilBERT: bert-base-uncase and distilbert-base-uncased

2. Model Design and Implementation
This stage discusses model selection, architecture design, and the training configuration applied in the study. This study aims to conduct a comprehensive comparative analysis, with the primary focus on evaluating the effectiveness of the Transformer architecture and its variants. The performance of Transformers is then compared with alternative approaches, namely conventional deep learning models such as RNNs (BiLSTM and LSTM) and classical machine learning algorithms, in order to obtain comprehensive insights into the performance of each method in text classification tasks.
   a. Model Selection: The study tested several approaches, namely Transformer architectures (mBERT, IndoBERT, BERT, DistilBERT), Deep Learning models (BiLSTM, LSTM), and classical Machine Learning methods (Naïve Bayes, SVM, Random Forest).
   b. Classification Implementation: In Transformer-based models, numerical tokens are processed through encoder layers to produce contextual vectors. These vectors are then passed to a classification layer (classifier head), where the Softmax function is used to produce a probability prediction of whether the sentiment is positive or negative.

3. Model Evaluation
This final stage aims to measure the success and reliability of the trained model. Performance Measurement: Evaluating the classification performance of the model using standard evaluation metrics, namely Accuracy, Precision, Recall, and F1-score, as shown in Equations 1, 2, 3, and 4.

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

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

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

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

3. RESULTS AND DISCUSSION

In this study, the dataset used is divided into two classes, namely positive and negative. However, during data collection, the amount of data was found to be imbalanced. Therefore, SMOTE was applied to balance the data. The distributions are shown in Table 1 and Table 2.

Table 1. Data Distribution Before and After SMOTE

Dataset	Before SMOTE		After SMOTE	
	Positive	Negative	Positive	Negative
	885	1281	1281	1281
Total	2166		2562	

Table 2. Training and Testing Data Distribution Before and After SMOTE

Dataset	Before SMOTE		After SMOTE	
	Training	Testing	Training	Testing
	1732	434	2048	434
Total	2166		2482	

Sample data for each category are shown in Table 3. Two labels are used in the classification, namely positive (1) and negative (0).

Table 3. Examples of Comments per Label

Id	Data	Label
1	Support full buat danantara demi investasi yang jelas dan bersih [Full support for Danantara for clear and clean investment]	1
2	Sudah tau tapi Cuma nato not action talk only omon omon [Already knew, but it is just NATO (no action, talk only), empty talk]	0

In addition, to assess the reliability of the model, several comparisons with classification algorithms were carried out. To maintain objectivity in the comparison and ensure consistent results, a standardized configuration adapted to the characteristics of each architecture was used. Given the limited time and available computing capacity, this study applied a set of commonly used hyperparameter configurations. For *Transformer-based* and hybrid models, optimization was performed using AdamW, which was chosen because it improves the handling of *weight decay* that is often coupled with the *gradient update* and is more effective in regularization and helps the model generalize better, which is very important to prevent *overfitting* in complex architectures. The *learning rates* used were 2e-5 and 1e-5, and the *batch sizes* were 16 and 32. For RNN models such as BiLSTM and LSTM, a *learning rate* of 1e-4 was used with the same optimizer and batch size configuration. All training processes were run for a maximum of 25 *epochs*. The parameters used in the experiments are shown in Table 4.

Table 4. Parameters Used

Parameter	Value
Batch	16, 32
Random state	40, 42
Epochs	5, 10, 25
Learning Rate	0.00001, 0.00002

Based on these parameters, experiments were then carried out for each model. The overall results for each model are shown in Table 5.

Table 5. Experimental Results with the Best Parameters

Model	Best Parameters	Accuracy Results	
		Training	Testing
mBERT	Batch size: 16 Learning rate: 0.00001 Epochs: 10	97.81	82.26
IndoBERT	Batch size: 32 Learning rate: 0.00002 Epochs: 25	98.61	81.11
BERT	Batch size: 16 Learning rate: 0.00002 Epochs: 5	95.84	80.18
DistilBERT	Batch size: 32 Learning rate: 0.00002	98.67	82.26

Model	Best Parameters	Accuracy Results	
		Training	Testing
	Epochs: 25		
BiLSTM	Batch size: 32 Learning rate: 0.00001 Epochs: 50	94.40	78.34
LSTM	Batch size: 32 Learning rate: 0.00001 Epochs: 50	86.95	79.95

Based on Table 5, Transformer-based models consistently outperformed the RNN/LSTM architectures in this experiment. mBERT and DistilBERT recorded the same highest testing accuracy of 82.26%. For mBERT, this result was achieved efficiently in only 10 epochs with a batch size of 16 and a small learning rate of 0.00001, which allows weight updates to occur precisely without overshooting the optimum. The superior performance of mBERT is supported by its ability to handle code-mixed text naturally as well as by its very fast loss convergence. Meanwhile, DistilBERT required a batch size of 32, a learning rate of 0.00002, and 25 epochs to reach a similar testing accuracy. This good performance was followed by IndoBERT (81.11%) and BERT (80.18%). In contrast, the non-Transformer models produced lower testing accuracy, namely 79.95% for LSTM and 78.34% for BiLSTM, which was the lowest. Although both were optimized using a batch size of 32, a learning rate of 0.00001, and the longest training duration of up to 50 epochs (the entire dataset repeated up to 50 times), their learning capacity remained limited compared to the Transformer architectures. Overall, although the parameter combinations for BERT-based models produced very high training accuracy (up to 98.67% for DistilBERT), there is a gap of around 13%–17% relative to the testing data, indicating moderate overfitting over the epoch iterations.

Table 6. Experimental Results With and Without SMOTE

Model	Without SMOTE				With SMOTE			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
mBERT	82%	81%	82%	81%	**83%**	**83%**	**83%**	**83%**
IndoBERT	80%	80%	80%	80%	83%	83%	83%	83%
BERT	**82%**	**82%**	**82%**	**82%**	80%	80%	80%	80%
DistilBERT	78%	78%	78%	78%	81%	81%	81%	81%
BiLSTM	81%	81%	81%	81%	76%	76%	76%	76%
LSTM	80%	80%	80%	80%	59%	35%	59%	44%
Naïve Bayes	76%	76%	76%	75%	77%	77%	77%	77%
SVM	75%	76%	75%	73%	76%	76%	76%	74%
Random Forest	76%	76%	76%	74%	76%	75%	76%	75%

Based on Table 6, which compares the experimental results with and without the SMOTE method, the highest performance was achieved by the Transformer architectures when combined with SMOTE. Specifically, mBERT and IndoBERT recorded the best and most balanced performance after the application of SMOTE, reaching 83% on all evaluation metrics, namely *Accuracy*, *Precision*, *Recall*, and *F1-score*. Before SMOTE was applied, mBERT and BERT dominated with the highest accuracy at the 82% level, whereas traditional models such as Naïve Bayes, SVM, and Random Forest recorded only moderate performance in the 75%–77% range. Interestingly, the addition of synthetic data through the SMOTE technique had a very negative impact on sequential *Deep Learning* models; the accuracy of BiLSTM dropped from 81% to 76%, while the performance of LSTM plummeted from 80% accuracy to 59%, with precision falling to 35%. This may be caused by fundamental differences in how the two architectures process the representation of text data. In addition, this indicates that although SMOTE is effective in addressing data imbalance for Transformer models, it actually disrupts and damages the sequential patterns that LSTM-based architectures try to learn.

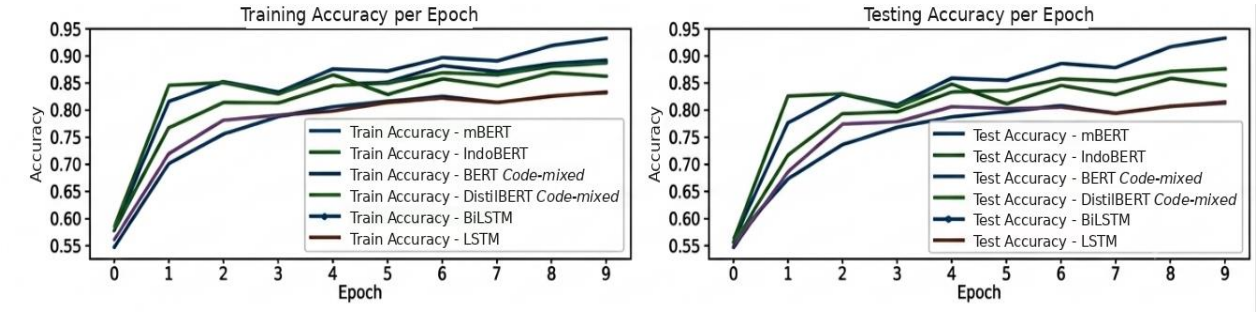


Figure 2. Comparison of Training Accuracy (a) and Testing Accuracy (b) Charts

Based on Figure 2, the accuracy of all models shows a sharp upward trend from *epoch* 0 to *epoch* 2, and then continues to rise gradually until it peaks at *epoch* 9. The *testing accuracy* graph shows a growth pattern consistent with the *training accuracy*, in which Transformer-based models (such as mBERT, IndoBERT, and BERT *Code-mixed*) consistently occupy the top positions with the highest performance, while the LSTM and BiLSTM architectures are at the bottom. This steady improvement indicates that the models are able to learn effectively from the training data and generalize well to the test data as the number of *epoch* iterations increases.

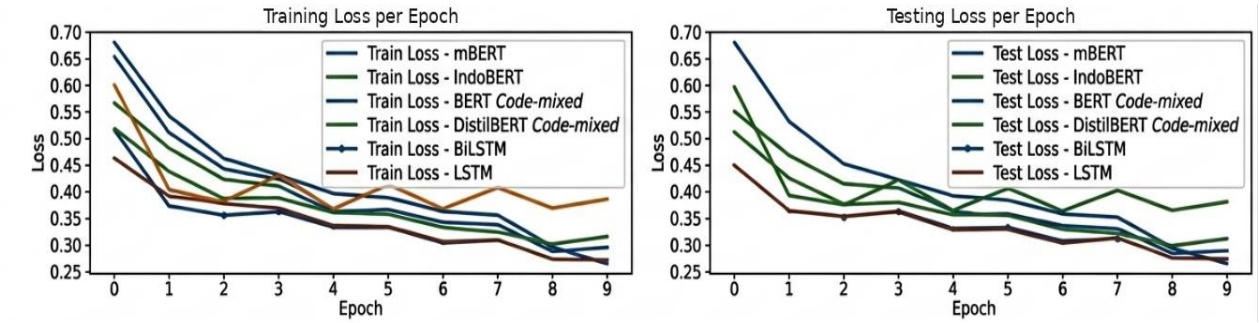


Figure 3. Comparison of Training Loss (a) and Testing Loss (b) Charts

Based on Figure 3, both graphs generally show a consistent decline from the initial *epoch* to *epoch* 9, indicating that all models successfully learned the data patterns well. The *testing loss* graph also shows a declining pattern consistent with the *training loss*, indicating that the models are able to maintain a low error rate on the test data without severe *overfitting*. Nevertheless, models such as LSTM start training with a lower *training loss* at the beginning, but their *testing loss* graphs show fluctuations and less optimal stability compared to Transformer-based models in several later *epochs*.

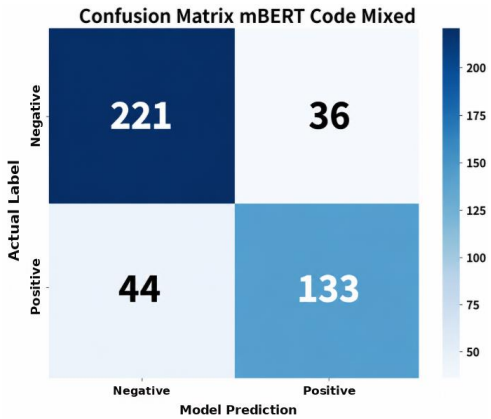


Figure 4. mBERT Confusion Matrix

Figure 4 shows the confusion matrix of the mBERT model. Overall, the confusion matrix indicates that the model has good classification ability with a relatively low error rate, although a number of positive samples have not yet been successfully identified, so improving recall could be the focus of further model development.

Table 7. Model Evaluation on Real Data

Sentence	System Prediction	Expert Validation
Saya suka prabowo sangat humble ke masyakarat	Positive	Positive
Baru dilantik, bukan kerja yang terasa,tapi tekanan yang makin menggila.Kebijakan condong ke konglomerat,rakyat kecil makin sesak dan terjerat. Janji tinggal slogan kampanye,isi negara malah buat elite berpesta.Yang lapar disuruh sabar,yang kaya makin lebar tawa dan harta.Katanya demi rakyat,nyatanya demi korporat.Selamat datang di negeri sultan,rakyat cuma jadi latar tontonan.	Positive	Negative

Table 7 shows several test data samples used to evaluate the model. Based on the table, some data were classified correctly and some sample sentences were misclassified. One factor causing misclassification is that some sentences could not recognized as sarcastic sentences, so the resulting sentiment is still not accurate enough.

4. CONCLUSION

This study concludes that mBERT is the most optimal architecture for sentiment analysis of code-mixed text, achieving an accuracy of 82.26% on its own and increasing to 83% when combined with the SMOTE technique. Scientifically, this study demonstrates that the cross-lingual representation capability of mBERT is highly effective in capturing the semantic meaning of language mixing, making it a robust solution for processing non-standard social media text data. For future work, this model has great potential to be applied to datasets from other domains (such as e-commerce or healthcare) in order to test its generalization more comprehensively and to explore alternative balancing techniques such as ADASYN or Borderline-SMOTE, which may avoid the performance degradation seen in LSTM and BiLSTM.

5. ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to P3M Politeknik Negeri Indramayu for the financial support and funding provided through the PUKTI SIPASTI Research Scheme, 2026, under Contract Number 005/DST/PL42/AL.04.03/SIPASTI/2026.

6. REFERENCES

[1] M. H. Susanti and S. Khu, “The 2024 Indonesian presidential election in the accounts of millennials: A case study of Prabowo Subianto and Gibran Rakabuming Raka voters in Semarang, Central Java,” Social Sciences & Humanities Open, vol. 11, Art. no. 101629, 2025, doi: 10.1016/j.ssaho.2025.101629.

[2] H. R. P. Sianturi, “Politics on a plate: Equivocal communication in Indonesian presidential nutrition policy,” Frontiers in Communication, vol. 10, Art. no. 1612652, 2025, doi: 10.3389/fcomm.2025.1612652.

[3] A. A. Firdaus, A. Yudhana, I. Riadi, and Mahsun, “Indonesian presidential election sentiment: Dataset of response public before 2024,” Data in Brief, vol. 52, Art. no. 109993, 2024, doi: 10.1016/j.dib.2023.109993.

[4] M. K. Nazir, M. Bilal, and S. C. Shongwe, “Sentiment analysis for code-mixed low-resource languages: A systematic review of approaches, techniques, applications, challenges, and future directions,” Social Network Analysis and Mining, vol. 16, Art. no. 47, 2026, doi: 10.1007/s13278-026-01588-2.

[5] A. M. Barik, R. Mahendra, and M. Adriani, “Normalization of Indonesian-English code-mixed Twitter data,” in Proc. 5th Workshop on Noisy User-Generated Text (W-NUT), Hong Kong, 2019, pp. 417–424, doi: 10.18653/v1/D19-5554.

[6] A. N. Rizal and S. Stymne, “Evaluating word embeddings for Indonesian–English code-mixed text based on synthetic data,” in Proc. 4th Workshop on Computational Approaches to Code Switching, Marseille, France: European Language Resources Association, 2020, pp. 26–35.

[7] G. Manias, A. Mavrogiorgou, A. Kiourtis, C. Symvoulidis, and D. Kyriazis, “Multilingual text categorization and sentiment analysis: A comparative analysis of the utilization of multilingual approaches for classifying Twitter data,” Neural Computing and Applications, vol. 35, pp. 21415–21431, 2023, doi: 10.1007/s00521-023-08629-3.

[8] A. Vaswani et al., “Attention is all you need,” in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 5998–6008.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional Transformers for language understanding,” in Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[10] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" in Proc. 57th Annu. Meeting of the Association for Computational Linguistics, Florence, Italy, 2019, pp. 4996–5001, doi: 10.18653/v1/P19-1493.
[11] S. Khanuja, S. Dandapat, A. Srinivasan, S. Sitaram, and M. Choudhury, "GLUECoS: An evaluation benchmark for code-switched NLP," in Proc. 58th Annu. Meeting of the Association for Computational Linguistics, 2020, pp. 3575–3585, doi: 10.18653/v1/2020.acl-main.329.
[12] S. Santy, A. Srinivasan, and M. Choudhury, "BERTologiCoMix: How does code-mixing interact with multilingual BERT?" in Proc. 2nd Workshop on Domain Adaptation for NLP, Kyiv, Ukraine: Association for Computational Linguistics, 2021, pp. 111–121.
[13] B. Wilie et al., "IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding," in Proc. 1st Conf. Asia-Pacific Chapter of the Association for Computational Linguistics and 10th Int. Joint Conf. Natural Language Processing, Suzhou, China, 2020, pp. 843–857, doi: 10.18653/v1/2020.aacl-main.85.
[14] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP," in Proc. 28th Int. Conf. Computational Linguistics, Barcelona, Spain, 2020, pp. 757–770, doi: 10.18653/v1/2020.coling-main.66.
[15] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A pretrained language model for Indonesian Twitter with effective domain-specific vocabulary initialization," in Proc. 2021 Conf. Empirical Methods in Natural Language Processing, 2021, pp. 10660–10668, doi: 10.18653/v1/2021.emnlp-main.833.
[16] M. F. Adilazuarda, S. Cahyawijaya, G. I. Winata, P. Fung, and A. Purwarianti, "IndoRobusta: Towards robustness against diverse code-mixed Indonesian local languages," in Proc. 1st Workshop on Scaling Up Multilingual Evaluation, 2022, pp. 25–34, doi: 10.18653/v1/2022.sumeval-1.5.
[17] A. F. Hidayatullah, R. A. Apong, D. T. C. Lai, and A. Qazi, "Pre-trained language model for code-mixed text in Indonesian, Javanese, and English using Transformer," Social Network Analysis and Mining, vol. 15, Art. no. 30, 2025, doi: 10.1007/s13278-025-01444-9.
[18] E. Hashmi, S. Y. Yayilgan, and S. Shaikh, "Augmenting sentiment prediction capabilities for code-mixed tweets with multilingual Transformers," Social Network Analysis and Mining, vol. 14, Art. no. 86, 2024, doi: 10.1007/s13278-024-01245-6.
[19] C. Tho, Y. Heryadi, I. H. Kartowisastro, and W. Budiharto, "Code-mixed sentiment analysis Indonesian-English using Transformer model," ICIC Express Letters, vol. 17, no. 11, pp. 1295–1302, 2023, doi: 10.24507/icicel.17.11.1295.
[20] L. W. Astuti, Y. Sari, and Suprapto, "Code-mixed sentiment analysis using Transformer for Twitter social media data," International Journal of Advanced Computer Science and Applications, vol. 14, no. 10, pp. 498–504, 2023, doi: 10.14569/IJACSA.2023.0141053.
[21] N. H. Setiono and Y. Sari, "Exploring the impact of back-translation on BERT's performance in sentiment analysis of code-mixed language data," IJCCS (Indonesian Journal of Computing and Cybernetics Systems), vol. 19, no. 2, pp. 223–234, 2025, doi: 10.22146/ijccs.104757.
[22] S. A. Nurbayani, D. S. Maylawati, and A. R. Atmadja, "Klasifikasi sentimen teks code-mixed Indonesia–Inggris non-formal pada X menggunakan model fine-tuned DistilBERT," Explorer, vol. 6, no. 2, 2026, doi: 10.47065/explorer.v6i2.2856.
[23] V. E. Sidauruk and W. Herowati, "IndoBERT-based sentiment analysis of political discourse on Platform X: The case of Prabowo-Gibran administration," Journal of Applied Informatics and Computing, vol. 10, no. 1, pp. 673–683, 2026, doi: 10.30871/jaic.v10i1.11586.
[24] D. A. Sulistyo and E. Setiadi, "Analisis sentimen kebijakan Makan Bergizi Gratis menggunakan IndoBERT dan machine learning," Jurnal FASILKOM, vol. 15, no. 3, pp. 507–513, 2025, doi: 10.37859/jf.v15i3.10546.
[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.