Digital Zone
Jurnal Teknologi Informasi dan Komunikasi

Volume 17 Issue 1 Year 2026 | Page 80-94 | e-ISSN: 2477-3255 | ISSN: 2086-4884
Received: 27-04-2026 | Revised: 03-05-2026 | Accepted: 20-05-2026

Hyperparameter Optimization of IndoBERT for DeepSeek User Sentiment Analysis

Andro Nicus Saragih[1], Lisnawita*[2]
[1,2]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Lancang Kuning
E-mail: andronicus018@gmail.com[1], Lisnawita@unilak.ac.id[2]
*Corespondence: lisnawita@unilak.ac.id

Abstract: *This study aims to evaluate the performance of the IndoBERT model in sentiment classification of user reviews for the DeepSeek application on the Google Play Store. The reviews were categorized into three sentiment classes: positive, neutral, and negative. The dataset was collected through web scraping of Indonesian-language reviews and processed using several preprocessing stages, including cleaning, stopword removal, and stemming. This study contributes by systematically comparing hyperparameter optimization methods using Grid Search and Random Search under two data split schemes (60:20:20 and 80:20). In addition, oversampling and Focal Loss techniques were implemented to address class imbalance and improve neutral class classification. Experimental results show that the best performance was achieved using Grid Search with an 80:20 data split, resulting in a testing accuracy of 80.40% and a macro F1-score of 70.85%. This configuration also produced a lower GAP value, indicating better model generalization and reduced overfitting. The findings demonstrate that appropriate hyperparameter optimization significantly improves IndoBERT performance for Indonesian sentiment analysis tasks.*

Keywords: *IndoBERT, Sentiment Analysis, Hyperparameter Optimization, Grid Search, DeepSeek*

1. Introduction

The rapid advancement of Artificial Intelligence (AI) has transformed how users interact with digital applications. One of the most prominent developments is the emergence of Large Language Model (LLM)-based chatbot systems capable of generating human-like responses and assisting users in various activities, including learning, information retrieval, and content generation. DeepSeek is one of the recently developed AI applications that has attracted significant attention due to its advanced reasoning capability and accessibility through mobile platforms. As the number of users continues to grow, reviews submitted through the Google Play Store provide valuable information regarding user satisfaction, application performance, and areas requiring improvement.

User reviews contain opinions, experiences, and perceptions that can be leveraged to evaluate the quality of digital services. However, manually analyzing large volumes of reviews is inefficient and time-consuming. Therefore, sentiment analysis has become an important Natural Language Processing (NLP) technique for automatically identifying opinions and categorizing them into positive, neutral, and negative sentiments. Previous studies have successfully applied sentiment analysis in various domains, including political issues [1], e-commerce reviews [2], and halal certification discussions [3]. These studies demonstrate the applicability of sentiment analysis for analyzing large-scale textual data.

Recent advances in NLP have significantly improved sentiment classification performance through Transformer-based language models. Among these models, IndoBERT has become one of the most widely used pre-trained language models for Indonesian-language processing tasks. IndoBERT has been successfully implemented in various sentiment analysis studies, including

Indonesian sentiment toward Rohingya refugees [4], Ajaib Kripto application reviews [5], Mobile JKN reviews [6], public opinions on the Whoosh high-speed railway [7], work-from-home discussions during the COVID-19 pandemic [8], electric vehicle discussions on social media [9], and COVID-19 news sentiment analysis [10]. These studies demonstrate that IndoBERT achieves competitive performance across various Indonesian NLP tasks.

The development of IndoBERT is supported by benchmark resources such as IndoNLU and IndoLEM, which provide datasets and pre-trained models specifically designed for Indonesian NLP tasks [13],[14]. In addition, several studies have explored IndoBERT-based approaches for sentiment analysis of product reviews and user-generated content. Aras et al. [12] demonstrated the effectiveness of IndoBERT for Shopee product review classification, while Jayadianti et al. [20] showed that fine-tuned IndoBERT achieved competitive performance compared to hybrid deep learning approaches. These findings suggest that IndoBERT is a promising approach for Indonesian sentiment classification.

Besides IndoBERT, researchers have also developed IndoBERTweet, a Transformer-based language model specifically trained on Indonesian social media data [16]. IndoBERTweet has been applied to hate speech detection [17] and sentiment analysis of TikTok reviews [19], demonstrating its suitability for processing Indonesian social media text. These studies indicate the potential of transformer-based architectures for Indonesian-language text classification.

Despite the promising performance achieved by IndoBERT-based models, several challenges remain. Previous studies reported challenges related to model overfitting and variations in classification performance across sentiment categories, particularly for neutral sentiments [7]. Such challenges highlight the importance of optimization strategies to improve classification performance and model robustness.

Hyperparameter optimization is an important approach for improving machine learning and deep learning model performance. Parameters such as learning rate, batch size, and training epochs influence model convergence and classification results. Grid Search has been widely used as a hyperparameter optimization technique to systematically identify suitable parameter combinations [11]. Iskoko et al. [15] showed that Grid Search, Random Search, and Bayesian Optimization affected IndoBERT performance in sentiment analysis of e-government application reviews. Similarly, Simanjuntak et al. [18] reported that hyperparameter tuning improved IndoBERT performance in fake news detection. These studies suggest that appropriate hyperparameter selection can contribute to better classification performance.

Although previous studies have investigated IndoBERT-based sentiment analysis and hyperparameter optimization separately, limited research has integrated hyperparameter optimization, oversampling, and Focal Loss within a unified framework for Indonesian sentiment analysis.

Therefore, this study aims to evaluate the performance of the IndoBERT model in classifying DeepSeek user reviews collected from the Google Play Store into positive, neutral, and negative sentiment categories. Unlike previous studies that primarily focused on applying IndoBERT for sentiment classification [4]-[10],[12],[20] or investigating hyperparameter optimization in different application domains [15], [18], this study integrates hyperparameter optimization, oversampling, and Focal Loss within a unified framework for Indonesian sentiment analysis. The proposed approach aims to address class imbalance problems, particularly in the neutral sentiment class, while evaluating its impact on classification performance and model generalization. The findings are expected to contribute to the development of more robust and reliable sentiment classification systems for Indonesian user reviews of AI-based applications

2. Research Method

The following figure illustrates the stages of the research, beginning with data collection and preprocessing, followed by IndoBERT model training, hyperparameter optimization, and model evaluation. If the resulting parameter combination is not optimal, the process returns to the optimization stage. Once the best combination is obtained, visualization and interpretation of the model results are performed.

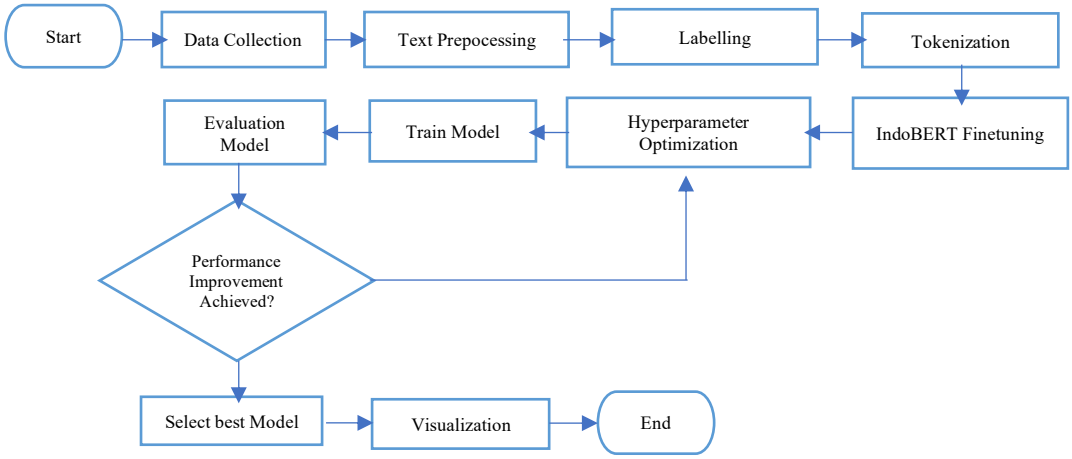

Figure 1. Flowchart of Hyperparameter Optimization for IndoBERT

The following figure illustrates the stages of the research, beginning with data collection and preprocessing, followed by IndoBERT model training, hyperparameter optimization, and model evaluation. If the resulting parameter combination is not optimal, the process returns to the optimization stage. Once the best combination is obtained, visualization and interpretation of the model results are performed.

a) Data Collection

The data used in this study were derived from user reviews of the DeepSeek application on the Google Play Store, collected using the google-play-scraper library.

b) Text Preprocessing

The collected reviews underwent several preprocessing steps, including punctuation removal, elimination of non-alphabetic characters, stopword removal, and stemming using the Sastrawi library. These steps ensured that the model received only essential linguistic information, free from noise that may hinder the learning process.

c) Sentiment Labeling

Prior to model training, a sentiment labeling process was conducted based on user review scores obtained from the Google Play Store. Each review score was categorized into three main sentiment classes: positive, neutral, and negative. This labeling process transforms numerical rating data into classification labels that can be processed by the model. The labeling criteria are described in the following table1.

Table 1. Class Labeling Based on Scores

No.	Class	Score	Labeling
1	Positif	5-4	2
2	Netral	3	1
3	Negatif	1 - 2	0

d) Tokenization

Before being fed into the model, the preprocessed text was tokenized using the tokenizer associated with the indoBERT model. This process converts raw text into token sequences that can be interpreted by the IndoBERT architecture.

e) IndoBERT Fine-Tuning

This study employs a supervised learning approach, where the IndoBERT model was fine-tuned using the labeled dataset. The model was trained to learn the relationship between user reviews and their corresponding sentiment categories.

f) Hyperparameter Optimization
Hyperparameters such as learning rate, batch size, and the number of epochs significantly influence model performance. Therefore, Grid Search and Random Search methods are employed to identify the best combination of the following parameters:
a) Learning rate: [2e-5, 3e-5, 5e-5]
b) Batch size: [8, 16]
c) Epochs: [5, 10, 20]
If the results are not optimal, the process is repeated to evaluate other parameter combinations.

g) Model Training
Each hyperparameter combination was used to train the model on the labeled dataset. This process applied a stratified split technique to divide the dataset into training, validation, and testing sets. Additionally, oversampling was implemented to address class imbalance across sentiment categories.

h) Model Evaluation
Model performance was evaluated using accuracy, precision, recall, F1-score (macro average), and GAP defined as the difference between training and testing accuracy to assess the degree of model generalization. The best model was selected based on the highest performance on the test dataset and the smallest GAP value. Furthermore, Focal Loss was utilized to balance class weights, particularly to improve classification accuracy in the neutral class.

i) Select Best Model
Upon completing the evaluation, the hyperparameter configuration yielding the highest test performance and lowest GAP was selected as the final model. This selection process ensures that the chosen model demonstrates both strong predictive accuracy and robust generalization to unseen data.

j) Visualization & Interpretation
The final stage involved interpreting and visualizing the model results. The classification outcomes were presented using Word Clouds for each sentiment label and a confusion matrix to assess the distribution of predictions across classes. These visualizations provide insight into how effectively the model distinguishes between positive, neutral, and negative sentiments in user reviews of the DeepSeek application.

3. Results and Discussion

3.1. Data Collection

The initial stage of this study begins with the data collection process in the form of user reviews of the DeepSeek application. The data were obtained from the Google Play Store using a web scraping technique implemented through the *google-play-scraper* library in Python. This technique enables automatic data extraction from the application page, including information related to users, review content, ratings, and review dates. The following is an example of the data obtained from scraping the Google Play Store for the DeepSeek application.

Table 2. Sample of Scraped DeepSeek User Review Data

User Name	score	time	content
Google Play Store Users	4	29/04/2025 07.09	Coba dulu..

User Name	score	time	content
Google Play Store Users	1	29/04/2025 07.06	aplikasi tidak berguna busuk ini aplikasi macam apah gratis tapi chatgpt ga gini juga masih lancar ini memang unggul di terjemahan tapi minusnya lola sering error apk pembodohan ini. najisssss. jauh chatgpt kemana mana. mending balik ke chatgpt. aplikasi JELEKKKK
Google Play Store Users	4	29/04/2025 06.29	bagus tapi menurut saya kalau chat nya suka banyak bug jadi harus di ulang lagi agar bisa dapat jawaban nya
Google Play Store Users	1	29/04/2025 06.02	Lebih bagus daripada Chat Bot yg lain, hahah bercanda. Masih kalah sama yang pendahulunya, apalagi soal loading lama banget.
Google Play Store Users	5	29/04/2025 05.33	I Like DeepSeek - Asisten AI by DeepSeek!
Google Play Store Users	1	29/04/2025 02.47	server oror mulu

3.2. Data Preprocessing Stages

3.2.1 Data Labeling

The initial dataset obtained from scraping the Google Play Store for the DeepSeek application consists of several attributes, one of which is the *Score* (rating) assigned by users, ranging from 1 to 5.

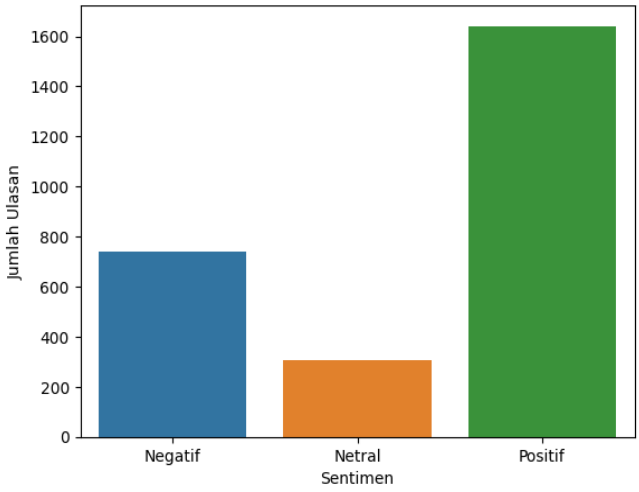

Figure 2. Sentiment Distribution of DeepSeek User Reviews

Figure 2 above presents the visualization of the sentiment distribution of user reviews for the DeepSeek application. Negative sentiment (blue) accounts for approximately 750 user reviews. Neutral sentiment (orange) includes around 350 reviews, while positive sentiment (green) represents the majority, with approximately 1,600 reviews.

Based on the analysis of user review data obtained from the Google Play Store, the sentiment distribution is dominated by positive sentiment, followed by negative and neutral sentiments. This indicates that the majority of users have had a favorable experience using the DeepSeek application.

https://doi.org/10.31849/digitalzone.v17i1. 33024

3.2.2 Text Preprocessing

After sentiment labeling, text preprocessing is conducted to improve the classification performance of the IndoBERT model. The process begins with lowercasing to reduce variations in text representation, followed by the removal of punctuation, emojis, numbers, mentions, and hashtags to eliminate meaningless or rare tokens. Stemming is applied to ensure that derived words are recognized under a common root meaning, while stopwords are removed to simplify the input and enhance the focus on emotionally significant words. Overall, this process aims to clean and restructure the data to make it more suitable for training the IndoBERT model.

3.2.3 Data Splitting

Based on the dataset consisting of 2,688 data points, the distribution for each scenario is as follows:

a) Scenario 1 (80% Training – 20% Testing)

Table 3. Training and Testing Split Data

No.	Data Type	Count
1	Training Data	2.150
2	Testing Data	538
Total		2.688

b) Scenario 2 (60% Training – 20% Validation – 20% Testing)

Table 4. Training, Validation, and Testing Split Data

No	Data Type	Count
1	Training Data	1.612
2	Validation Data	538
3	Testing Data	538
Total		2.688

3.2.4. Hyperparameter Tuning Experiment

This experiment aimed to evaluate the effect of hyperparameter tuning on the performance of the IndoBERT model in classifying sentiment from DeepSeek user reviews. The study is conducted using three approaches: without optimization (baseline), Grid Search, and Random Search, with two data split schemes: 60:20:20 and 80:20.

3.2.5 Tokenization with IndoBERT

After the text data have been cleaned and labeled, the next stage is tokenization, which is the process of converting text into numerical representations (tokens) that can be understood by transformer-based models such as IndoBERT. Tokenization is performed using the IndoBERT tokenizer from Hugging Face, which automatically splits sentences into sub-word tokens according to the predefined format of the model. An example of the tokenization output for the first five data entries is shown in the following table 5

Table 5. Tokenization Output Results

No.	Original Text	Token
1	lebih sering server busy moga baik	[CLS] lebih sering server bus ##y moga baik [SEP] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] ....]

No.	Original Text	Token
2	ngeleg	[CLS] ngel ##eg [SEP] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] ....]
3	this ai is very good for asking questions because there is a deep thinking feature but there is one problem the server is too busy hopefully in the future it will be more efficient good job deepseek	[CLS] this ai is very good for ask ##ing question ##s bec ##ause there is a deep thinking feature but there is one problem the server is too bus ##y hope ##full ##y in the future it will be more eff ##ici ##ent good job deep ##se ##ek [SEP]
4	server sering sibuk	[CLS] server sering sibuk [SEP] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] ....]
5	server sibuk mulu	[CLS] server sibuk mulu [SEP] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] ....]

3.2.6 IndoBERT (Without Optimization)

The initial step involves training the IndoBERT model using default parameters without any optimization process. The configuration used includes a learning rate of 5e-5, a batch size of 16, and 5 epochs. This training process resulted in a training accuracy of 90.17%, while the testing accuracy reached only 74.54%. The macro F1-score was 64.29%, with an accuracy GAP of 0.1563.

Table 6. Accuracy Results of the IndoBERT Model Without Optimization

No	Metric	Value
1	Training Accuracy	90.17%
2	Testing Accuracy	74.54%
3	GAP (*Training - Testing*)	15.63%
4	*Precision* (*Macro Avg*)	66 %
5	*Recall (Macro Avg*)	65%
6	*F1-Score (Macro Avg)*	64.29%

Based on Table 6, the model without optimization demonstrates strong performance on the training data but fails to maintain its performance on the testing data. Although the accuracy GAP is not excessively large (15.63%), it still indicates the need for hyperparameter tuning. The F1-score of only 64.29% further suggests that the neutral class is particularly difficult to predict. Therefore, hyperparameter optimization strategies and techniques to address data imbalance are required to improve the model's performance.

1) Grid Search (60% Train : 20% Validation : 20% Test) & (80% Train – 20% Test)
Grid Search was employed to identify the best combination of hyperparameters by exhaustively exploring all possible combinations of the following parameters:
a) learning_rate: [2e-5, 3e-5, 5e-5]
b) batch_size: [8, 16]
c) epochs: [5, 10, 20]

Table 7. Grid Search Results (60% Training – 20% Validation – 20% Testing)

No	Parameter *(LR, BS, Epoch)*	*Training Accuracy*	*Validation Accuracy*	*Macro F1*	*GAP*
1	3e-5, 16, 10	98.52%	80.04%	62.06%	18.49%
2	2e-5, 8, 5	96.68%	79.12%	62.86%	17.56%
3	3e-5, 16, 20	98.34%	79.12%	63.25%	19.22%
4	5e-5, 8, 10	98.59%	79.12%	62.54%	19.47%
5	5e-5, 8, 5	98.22%	79.12%	63.17%	19.10%
6	2e-5, 16, 10	98.52%	78.57%	62.54%	19.95%
7	3e-5, 8, 5	97.85%	78.57%	62.80%	19.28%
8	5e-5, 8, 20	98.40%	78.39%	63.49%	20.01%
9	3e-5, 8, 10	98.59%	78.39%	61.97%	20.20%
10	2e-5, 16, 20	98.59%	78.39%	62.67%	20.20%
11	5e-5, 16, 20	98.59%	78.39%	62.56%	20.20%
12	5e-5, 16, 5	97.48%	78.02%	61.66%	19.46%
13	5e-5, 16, 10	98.59%	78.02%	61.63%	20.56%
14	2e-5, 8, 10	98.59%	77.66%	60.06%	20.93%
15	3e-5, 16, 5	96.50%	77.66%	61.01%	18.84%
16	2e-5, 8, 20	98.59%	77.11%	61.25%	21.48%
17	3e-5, 8, 20	98.59%	76.92%	61.56%	21.66%
18	2e-5, 16, 5	93.05%	76.92%	61.80%	16.13%

Based on Table 7, a total of 18 hyperparameter combinations were comprehensively evaluated. The training process was conducted using 60% of the data, with 20% used for validation. The best combination identified was a learning rate of 3e-5, batch size of 16, and 10 epochs. The results show a training accuracy of 98.52%, validation accuracy of 80.04%, and testing accuracy of 77.29%, with a macro F1-score of 62.06% and an accuracy GAP of 0.2123. The relatively high GAP indicates a tendency toward overfitting. Next, the results for Grid Search with an 80% training and 20% testing split are presented as follows:

Table 8. Grid Search Results (80% Training – 20% Testing)

No	Parameter *(LR, BS, Epoch)*	*Training Accuracy*	*Testing Accuracy*	*Macro F1*	*GAP*
1	5e-5, 8, 5	97.02%	80.40%	70.85%	16.62%
2	2e-5, 8, 20	98.85%	78.94%	66.70%	19.92%
3	5e-5, 8, 20	98.85%	78.94%	67.02%	19.92%
4	5e-5, 8, 10	98.81%	78.39%	65.57%	20.42%
5	2e-5, 8, 10	98.12%	78.21%	67.84%	19.92%
6	5e-5, 16, 10	98.76%	78.02%	66.01%	20.74%
7	2e-5, 16, 20	98.85%	78.02%	65.72%	20.83%
8	3e-5, 8, 5	96.10%	77.66%	67.45%	18.45%

No	Parameter (LR, BS, Epoch)	Training Accuracy	Testing Accuracy	Macro F1	GAP
9	3e-5, 8, 20	98.85%	77.47%	64.00%	21.38%
10	3e-5, 8, 10	98.72%	77.47%	65.28%	21.24%
11	5e-5, 16, 5	97.48%	77.29%	65.92%	20.19%
12	5e-5, 16, 20	98.85%	77.11%	63.11%	21.75%
13	3e-5, 16, 20	98.72%	76.92%	64.68%	21.79%
14	2e-5, 8, 5	92.80%	76.74%	67.26%	16.06%
15	2e-5, 16, 10	98.40%	76.74%	63.25%	21.66%
16	3e-5, 16, 10	98.58%	75.64%	62.32%	22.94%
17	3e-5, 16, 5	91.43%	74.54%	65.19%	16.89%
18	2e-5, 16, 5	83.18%	68.32%	58.65%	14.87%

In the scheme shown in the table above, the subsequent training process uses a data split of 80% for training and 20% for testing. Grid Search was applied using the same parameter combinations. The best configuration is obtained with a learning rate of 5e-5, batch size of 8, and 5 epochs. The results show a training accuracy of 97.02%, testing accuracy of 80.40%, a macro F1-score of 70.85%, and an accuracy GAP of 0.1662. The relatively low GAP indicates only a slight tendency toward overfitting. Overall, this configuration provides the most optimal and balanced performance compared to other combinations.

2) Random Search (60% Train : 20% Validation : 20% Test) & (80% Train – 20% Test)
The Random Search method was employed to compare the effectiveness of hyperparameter optimization with Grid Search. In this approach, five hyperparameter combinations are randomly selected from the total of 18 possible combinations. The experiments are conducted using two data split schemes: 60% training, 20% validation, and 20% testing, as well as 80% training and 20% testing. The following table presents a summary of the best results from Random Search:

Table 9. Random Search Results (60% Training – 20% Validation – 20% Testing)

No	Parameter (LR, BS, Epoch)	Training Accuracy	Validation Accuracy	MacroF1 (Testing)	GAP
1	3e-5, 8, 10	98.59%	78.39%	61.97%	20.20%
2	2e-5, 16, 20	98.59%	78.39%	62.67%	20.20%
3	2e-5, 16, 10	98.46%	78.02%	62.33%	20.44%
4	2e-5, 16, 5	93.05%	77.29%	63.11%	15.77%
5	2e-5, 8, 5	95.02%	76.19%	61.02%	18.83%

In the table above, five hyperparameter combinations were randomly selected from the same parameter space as Grid Search. The objective was to evaluate whether a random approach could identify competitive or even superior combinations. The best configuration obtained was a learning rate of 3e-5, batch size of 8, and 10 epochs. The evaluation results show a training accuracy of 98.59%, validation accuracy of 78.39%, and testing accuracy of 81.32%, with a macro

F1-score of 61.97% and an accuracy GAP of 0.1727. Although the GAP is relatively high, the testing performance is the highest among the 60:20:20 data split scheme. Next, the results of the Random Search method with an 80% training and 20% testing split are presented as follows:

Table 10. Random Search Results (80% Training – 20% Testing)

No	Parameter (LR, BS, Epoch)	*Training Accuracy*	*Testing Accuracy*	Macro F1	GAP
1	2e-5, 16, 20	98.85%	77.66%	63.63%	21.20%
2	3e-5, 8, 10	98.72%	77.47%	65.28%	21.24%
3	2e-5, 8, 5	94.09%	76.56%	66.71%	17.53%
4	2e-5, 16, 10	98.40%	76.56%	64.05%	21.84%
5	2e-5, 16, 5	83.18%	68.32%	58.65%	14.87%

In the scheme shown in Table 10, the subsequent training process uses a data split of 80% for training and 20% for testing. Grid Search is again applied using the same parameter combinations. The best configuration is obtained with a learning rate of 2e-5, batch size of 16, and 20 epochs. The results show a training accuracy of 98.85%, testing accuracy of 77.66%, a macro F1-score of 63.63%, and an accuracy GAP of 0.2120. The high GAP indicates a tendency toward overfitting, likely caused by longer training duration and the larger size of the training data.

3) Comparison of IndoBERT and Hyperparameter Optimization Results

Table 11. Hyperparameter Optimization Comparison

No	Methode and data split	*Training Accuracy*	*Testing Accuracy*	GAP	Precision	Recall	F1 Score
1	Without Optimization (60:20:20)	90.17%	74.54%	15.63%	66.00%	65.00%	64.00%
2	*Grid Search* (60:20:20)	98.52%	77.29%	21.23%	64.85%	64.35%	64.49%
3	*Random Search* (60:20:20)	98.59%	81.32%	17.27%	71.07%	67.19%	67.94%
4	*Grid Search* (80:20)	97.02%	80.40%	16.62%	70.84%	71.32%	70.85%
5	*Random Search* (80:20)	98.85%	77.66%	21.20%	64.09%	63.30%	63.63%

Based on the table above, which presents five IndoBERT modeling scenarios, it can be concluded that the Grid Search approach with an 80:20 data split provides the most optimal and

balanced results. This model achieves:
a. The highest testing accuracy: 80.40%
b. A competitive macro F1-score: 70.85%
c. A relatively small GAP: 16.62%, indicating good model generalization
d. Reasonably balanced precision and recall across sentiment classes, although the neutral class remains the most challenging to classify.

The following diagram illustrates the comparison of hyperparameter optimization results used in this study.

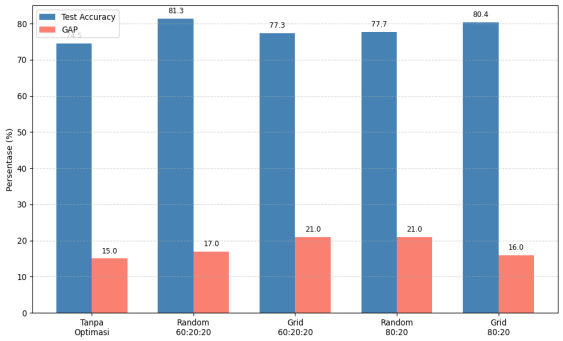

Figure 5. Hyperparameter Optimization Comparison Diagram

As shown in Figure 5, the model without optimization using a 60:20:20 split produces the smallest GAP (only 15%), indicating that the model appears relatively stable and shows a lower tendency toward overfitting compared to the optimized models. However, this approach is not selected as the best model for several reasons:
a. Its accuracy and F1-score are still lower compared to those achieved by Grid Search (both 60:20:20 and 80:20 splits).
b. A small GAP indicates stability, but the overall classification performance is not yet optimal, particularly for the neutral class.
c. The primary objective of this study is to obtain the best hyperparameter combination with a balance between high accuracy and strong generalization, not merely to prevent overfitting.
d. Therefore, Grid Search with an 80:20 split is superior, as it improves testing accuracy without sacrificing model stability, making it the best-performing model in this study.

In conclusion, hyperparameter optimization has a positive impact on model performance, especially when conducted using a comprehensive approach such as Grid Search. Although Random Search can also identify good configurations, its effectiveness is more dependent on the randomness of parameter selection. Therefore, in the context of this study, Grid Search is recommended as the preferred hyperparameter optimization method to enhance the performance of the IndoBERT model in sentiment analysis of DeepSeek user reviews.

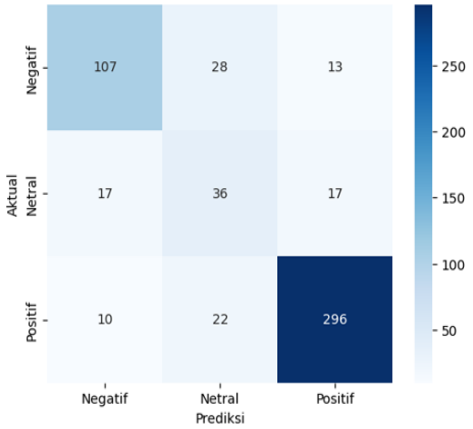

Figure 6. Confusion Matrix Diagram Results

Based on Figure 6, the confusion matrix evaluation shows that the model optimized using Grid Search is able to classify positive and negative sentiments relatively well; however, it still struggles with the neutral class. For the positive class, a total of 296 instances are correctly classified, although there are 32 false negatives and 30 false positives. In the negative class, the model correctly classifies 107 instances, but still produces 41 false negatives and 27 false positives.

Meanwhile, the performance on the neutral class is the weakest, with only 36 true positives and a relatively high number of misclassifications, including 34 false negatives and 50 false positives. This indicates that the model is not yet able to accurately distinguish neutral reviews, possibly due to data imbalance or insufficient feature representation for this class.

3.3. Visual Analysis of DeepSeek User Review Word Cloud

As part of the initial data exploration, a Word Cloud visualization is generated to identify the most frequently occurring words in user reviews of the DeepSeek application on the Google Play Store. The Word Cloud is created by combining all text from the *content* column that has undergone preprocessing, and then visualizing it based on word frequency. The figure below presents the resulting Word Cloud.

Figure 8. Word Cloud Visualization of DeepSeek User Reviews

As shown in Figure 8, the Word Cloud effectively visualizes the most frequently occurring words in user reviews. Words such as *"bagus"* (good), *"jawab"* (answer), *"server"*, and *"AI"* appear as dominant terms, reflecting the main topics of concern among users when providing reviews of the DeepSeek application.

3.4 Discussion

This study aimed to evaluate the performance of the IndoBERT model in classifying sentiment from user reviews of the DeepSeek application on the Google Play Store into three classes: positive, neutral, and negative. The evaluation was conducted through initial testing without optimization, followed by hyperparameter optimization using Grid Search and Random Search, with two data split schemes: 60:20:20 and 80:20.

Without optimization, the model achieved a testing accuracy of 74.54%; however, it exhibited a GAP of 15.63% and an F1-score of 64.00%, indicating a slight tendency toward overfitting and class imbalance issues. Grid Search with a 60:20:20 split improved testing accuracy to 77.29%; however, the GAP increased to 21.23%, and the F1-score showed only a marginal improvement. In contrast, Random Search produced better results, with a testing accuracy of 81.32%, a GAP of 17.27%, and an F1-score of 67.94%.

The 80:20 data split yielded the best overall results when using Grid Search, achieving a testing accuracy of 80.40%, the highest F1-score of 70.85%, and a GAP of 16.62%. Meanwhile, Random Search under the 80:20 scheme showed lower performance compared to Grid Search. These results suggest that a larger proportion of training data, combined with appropriate hyperparameter tuning, contributes positively to model generalization.

The superior performance achieved by the Grid Search configuration with a learning rate of 5e-5, batch size of 8, and 5 epochs indicates that moderate training duration and smaller batch

sizes contribute positively to model generalization. Smaller batch sizes allow the model to perform more frequent weight updates, enabling better adaptation to complex data patterns. Meanwhile, limiting the number of epochs helps reduce overfitting tendencies, as excessive training iterations may cause the model to memorize training data rather than learning generalized representations.

The neutral sentiment class remained the most difficult category to classify accurately. Confusion matrix analysis indicated that the model still struggled to distinguish the neutral class, as many neutral reviews were incorrectly predicted as either positive or negative. This misclassification occurred because neutral reviews often contain ambiguous expressions and mixed opinions whose semantic content overlaps with other sentiment classes. Furthermore, the distribution of neutral data was significantly smaller compared to positive sentiment data, limiting the model's ability to learn representative neutral features effectively. Although oversampling and Focal Loss improved classification performance, the results suggest that additional techniques—such as data augmentation or semantic-based balancing methods may still be required in future studies.

The combination of effective preprocessing, hyperparameter optimization, and appropriate data splitting significantly enhanced the performance of IndoBERT in sentiment classification. Nevertheless, challenges in accurately classifying the neutral class remain and warrant further investigation

4. Conclusions

This study demonstrates that the IndoBERT model can be effectively utilized for sentiment classification of user reviews of the DeepSeek application on the Google Play Store. Through the application of text preprocessing and hyperparameter optimization, the model's performance can be significantly improved. The best results were achieved using the Grid Search method with an 80:20 data split, yielding a testing accuracy of 80.40% and a macro F1-score of 70.85%. These results indicate that the model has good generalization capability and is able to recognize the three sentiment classes in a relatively balanced manner.

The use of Focal Loss and oversampling also contributed to addressing class imbalance, particularly in improving model performance on neutral reviews. Compared to previous studies, the approach proposed in this research provides improvements in both accuracy and model stability. This study also demonstrates that the combination of hyperparameter optimization, oversampling, and Focal Loss can improve both classification performance and model generalization capability in Indonesian sentiment analysis tasks involving imbalanced datasets.

For future work, model development can be directed toward exploring other transformer-based architectures, such as IndoBERTweet or XLM-RoBERTa, as well as applying data augmentation techniques to achieve a more balanced class distribution.

References

[1] L. Septian, T. Aljauza, and Juliane, "Analisis Sentimen Putusan Mahkamah Konstitusi Terhadap Batas Usia Capres dan Cawapres Menggunakan IndoBERT," The Indonesian Journal of Computer Science, vol. 12, no. 6, 2023. https://doi.org/10.33022/ijcs.v12i6.3614

[2] N. Agustina, D. H. Citra, W. Purnama, C. Nisa, and Kurnia, "Implementasi algoritma Naive Bayes untuk analisis sentimen ulasan Shopee pada Google Play Store," MALCOM: Indonesian Journal of Machine Learning and Computer Science, vol. 2, no. 1, pp. 47–54, 2022, https://doi.org/10.57152/malcom.v2i1.195

[3] A. S. Rusydiana and Marlina, "Analisis sentimen terkait sertifikasi halal," Journal of Economics and Business Aseanomics, vol. 5, no. 1, pp. 69–85, 2020, https://doi.org/10.33476/j.e.b.a.v5i1.1405

[4] G. A. B. Baliputra, S. Kacung, and B. Santoso, "Sentiment Analysis of Rohingya Refugees in Aceh Using Support Vector Machine (SVM) and Multinomial Logistic

Regression," Sistemasi: Jurnal Sistem Informasi, vol. 14, no. 3, 2025, https://doi.org/10.32520/stmsi.v14i3.5159.

[5] A. L. Basuki, B. Rahayudi, and D. Pramono, "Analisis Sentimen Ulasan Aplikasi Ajaib Kripto menggunakan IndoBERT dan Metode Root Cause Analysis," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 9, no. 4, 2025. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14709.

[6] N. Azka, W. Sayudha, R. Graha, and R. Nugroho, "Analisis Sentimen Ulasan Aplikasi Mobile JKN di Google PlayStore Menggunakan IndoBERT," Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), vol. 9, no. 2, pp. 495–505, 2025, https://doi.org/10.35870/jtik.v9i2.3340

[7] G. Hakim, T. N. Fatyanosa, and A. W. Widodo, "Analisis Sentimen Masyarakat terhadap Kereta Cepat Whoosh pada Platform X menggunakan IndoBERT," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 8, no. 10, 2024. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14291.

[8] M. S. Adhantoro, F. A. Haq, D. Hartanto, and A. P. Sudaryanto, "IndoBERT-Based Sentiment Analysis of Electric Motorcycle Policy in Indonesia Using Instagram Data," *Jurnal Penelitian Sains Teknologi*, vol. 2, no. 2, pp. 165–181, 2026, https://doi.org/10.23917/saintek.v2i2.17021

[9] R. Merdiansah and Ridha, "Analisis Sentimen Pengguna X Indonesia Terkait Kendaraan Listrik Menggunakan IndoBERT," Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI), vol. 7, no. 1, pp. 221–228, 2024, https://doi.org/10.55338/jikomsi.v7i1.2895.

[10] W. Nurfitri and A. Chowanda, "Analisis Sentimen Pada Kasus Positif Covid-19 Berdasarkan Pemberitaan Media di Indonesia Menggunakan IndoBERT," Progresif: Jurnal Ilmiah Komputer, vol. 20, no. 1, pp. 580–593, 2024, https://doi.org/10.35889/progresif.v20i1.1897

[11] I. M. Gananta, I. N. Purnama, and I. N. Fredlina, "Optimasi Prediksi Harga Emas Dengan Metode Support Vector Regression (SVR) Menggunakan Algoritma Grid Search," JATI (Jurnal Mahasiswa Teknik Informatika), vol. 7, no. 6, pp. 3160–3165, 2023, https://doi.org/10.36040/jati.v7i6.8000

[12] M. Yusuf, R. Y. Ruimassa, E. A. B. Wambrauw, E. B. Pala'langan, and S. Aras, "Sentiment Analysis on Shopee Product Reviews Using IndoBERT," Journal of Information Systems and Informatics, vol. 6, no. 3, pp. 1616–1627, 2024, https://doi.org/10.51519/journalisi.v6i3.814

[13] B. Wilie et al., "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," arXiv preprint arXiv:2009.05387, 2020. [Online]. Available: https://doi.org/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," arXiv preprint arXiv:2011.00677, 2020 https://doi.org/10.18653/v1/2020.coling-main.66

[15] A. Iskoko, I. Tahyudin, and P. Purwadi, "Hyperparameter Optimization of IndoBERT Using Grid Search, Random Search, and Bayesian Optimization in Sentiment Analysis of E-Government Application Reviews," Jurnal Teknik Informatika (JUTIF), vol. 6, no. 5, pp. 3430–3444, 2025, https://doi.org/10.52436/1.jutif.2025.6.5.4897.

[16] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," in Proc. EMNLP 2021, 2021, pp. 10660–10668, https://doi.org/10.18653/v1/2021.emnlp-main.833

[17] J. F. Kusuma and A. Chowanda, "Indonesian Hate Speech Detection Using IndoBERTweet and BiLSTM on Twitter," JOIV: International Journal on Informatics Visualization, 2023. http://dx.doi.org/10.30630/joiv.7.3.1035

[18] A. Simanjuntak et al., "Research and Analysis of IndoBERT Hyperparameter Tuning in Fake News Detection," Jurnal Nasional Teknik Elektro dan Teknologi Informasi, vol. 13, no. 1, pp. 60–67, 2024, https://doi.org/10.22146/jnteti.v13i1.8532.

[19] J. C. Setiawan, K. M. Lhaksmana, and B. Bunyamin, "Sentiment Analysis of Indonesian TikTok Review Using LSTM and IndoBERTweet Algorithm," JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), vol. 8, no. 3, pp. 774–780, 2023, https://doi.org/10.29100/jipi.v8i3.3911.

[20] H. Jayadianti, W. Kaswidjanti, A. T. Utomo, S. Saifullah, F. A. Dwiyanto, and R. Drezewski, "Sentiment Analysis of Indonesian Reviews Using Fine-Tuning IndoBERT and R-CNN," ILKOM Jurnal Ilmiah, vol. 14, no. 3, pp. 348–354, 2022, https://doi.org/10.33096/ilkom.v14i3.1505.348-354.