Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2026, 2026:2
https://doi.org/10.28919/cmbn/9682
ISSN: 2052-2541

# TOPIC-BASED SENTIMENT ANALYSIS OF JAMSOSTEK MOBILE APPLICATION REVIEWS USING THE BERTopic METHOD

NISA AKBARILAH PUTRI, ANINDYA APRILIYANTI PRAVITASARI, SRI WINARNI*

Department of Statistics, Universitas Padjadjaran, Bandung, Indonesia



**Abstract:** Rapid developments in digital technology have changed the way public services are delivered, shifting from conventional systems to mobile-based platforms. BPJS Ketenagakerjaan has adopted this transformation through the development of the Jamsostek Mobile (JMO) application. The application can be accessed on the Google Play Store, and users can provide ratings and reviews on the Google Play Store page. To improve the quality of the JMO application, it is important to pay attention to these reviews. Therefore, sentiment analysis is needed to identify and analyse positive and negative sentiments related to the use of the JMO application. The method used combines the IndoBERT and BERTopic models to understand user opinions in greater depth. This study shows that 75% of 2,846 comments contain negative sentiments, and the IndoBERT model achieves 100% accuracy in sentiment classification. In topic modelling, five clusters were formed, with the most discussed topics being requests for app updates and users' difficulties in accessing the JMO app due to constant errors and slowness.

**Keywords:** BERTopic; IndoBERT; JMO; topic modelling; sentiment analysis.

**2020 AMS Subject Classification:** 68T50, 68T10.

*Corresponding author
E-mail address: sri.winarni@unpad.ac.id
Received November 05, 2025

## 1. INTRODUCTION

The rapid development of digital technologies has reshaped how public services are delivered, moving from conventional systems to mobile-based platforms. BPJS Ketenagakerjaan, Indonesia's national social security agency, has embraced this transformation through the creation of the Jamsostek Mobile (JMO) application, enabling users to access social security services anywhere and anytime. Despite its wide adoption, user reviews indicate that application performance and usability issues persist. User feedback, often expressed in textual form, provides valuable insights for evaluating system quality, identifying pain points, and optimizing digital services. Therefore, sentiment analysis of user reviews is essential to understanding user satisfaction levels and guiding system improvement initiatives.

Sentiment analysis, or opinion mining, aims to determine the polarity (positive, negative, or neutral) of user opinions from text data [1][2]. Machine learning approaches, such as bidirectional, can help capture dependencies in both directions and model them for better text analysis and understanding, such as sentiment analysis. However, there are issues related to information decay that occur in RNN, LSTM, and GRU methods. These methods tend to forget information in the initial text sequence, causing the semantic information contained in previous words to be quickly forgotten. Thus, these methods have limitations in capturing the semantic structure contained in text data [3]. As a solution to overcome these weaknesses, this study uses IndoBERT, a pre-trained model based on BERT (Bidirectional Encoder Representations from Transformers) that is specifically designed for the Indonesian language. IndoBERT has the ability to understand the context of words in both directions, allowing it to better capture the nuances and variations of the Indonesian language, making it suitable for sentiment classification in Indonesian [4].

However, sentiment alone is not enough to capture the thematic structure of user feedback. Topic modeling methods such as Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF) have been traditionally used but often fail to capture semantic meaning in short, informal texts like app reviews [5][6]. BERTopic, an advanced topic modeling technique, combines contextual embeddings, dimensionality reduction, and density-based clustering to produce coherent and meaningful topics [6][7]. This research integrates IndoBERT for sentiment classification with BERTopic for topic extraction to analyze the collective opinions of JMO application users.

## 2. MATERIAL AND METHOD

### 2.1 Data

In this study, the data used were reviews of the Jamsostek Mobile (JMO) application obtained by scraping the Google Play Store platform. The data collected were text reviews/comments from JMO application users between 1 January 2025 and 6 July 2025. A total of 2,846 data points were collected for this study.

### 2.2 Text Mining

Text mining is a field that focuses on extracting information from texts stored in various forms such as news articles, books, emails, blogs, and other text sources [8]. This concept is part of data mining, which is the process of analysing large amounts of data to identify patterns, relationships, or useful information. The difference between text mining and data mining can be seen in the data patterns used. In text mining analysis, the data originates from textual language, which has unstructured data properties, whereas the data patterns used in data mining analysis originate from structured database facts. As a result, text mining presents additional challenges not encountered in data mining, such as complex and incomplete text data structures, unclear and non-standard meanings, and different languages coupled with inaccurate translations [9].

### 2.3 Transformer

Transformers are a deep learning-based architecture was initially developed for machine translation and later became the foundation for other transformer-based models, particularly in the field of NLP. This architecture consists of an encoder and decoder structure. The encoder structure consists of a collection of identical layers, where each layer consists of several sub-layers. The first sub-layer is multi-head self-attention and the next sub-layer is a position-wise fully connected feed-forward network. In addition, there are normalisation and residual connection layers in each sub-layer. The decoder structure has components similar to the encoder, but there is an addition of a third sub-layer, namely a multi-head attention sub-layer to the output of the encoder structure. Figure 1 shows the architecture of the transformer [10].

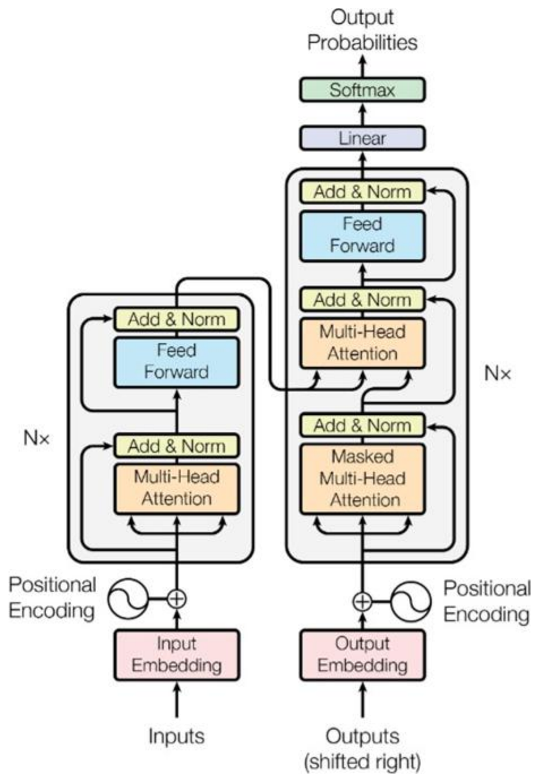


**Figure 1**. Transformer Architecture

### 2.4 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representation from Transformers (BERT) is a deep learning model introduced by Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanov from Google AI Language. BERT is part of Transformers, where the BERT model processes words in a sentence based on whether or not there is a connection between those words and the sentence as a whole. BERT works differently from other algorithms in that it processes words by learning the context based on existing words. BERT was trained using 800 million words from BooksCorpus and 2.5 billion words from English Wikipedia [11]. BERT has several layers in its architecture, such as the Multi-Head Attention, Add and Norm, and Feed Forward layers. The multi-head attention layer plays a role in generating representations from the input, the feed-forward layer plays a role in the nonlinear transformation of the input, and in the add and norm layer, the output is added to the input element-wise and layer normalisation occurs.

**2.5 Indonesian Bidirectional Representations from Transformers (IndoBERT)**

IndoBERT is a BERT architecture trained using an Indonesian language corpus. IndoBERT can understand context by processing all words in a sentence in parallel and using a multi-head attention mechanism. With parallel input processing and the use of multi-head attention, IndoBERT can handle long sentence inputs. When there are long sentences, the IndoBERT model will still be able to capture the relationship between words in the sentence from beginning to end, as well as capture the meaning of the sentence by weighting the words. This is different from other models that process words in sentences one by one, making it more difficult to capture the overall context of the sentence.

There are three types of embeddings used in IndoBERT that also help in understanding the context of a sentence, namely Token Embeddings, Position Embeddings, and Segment Embeddings, which are not found in other models. In addition to its ability to understand context, IndoBERT's computational process is also simpler. Figure 2 has show the example of input BERT [12].

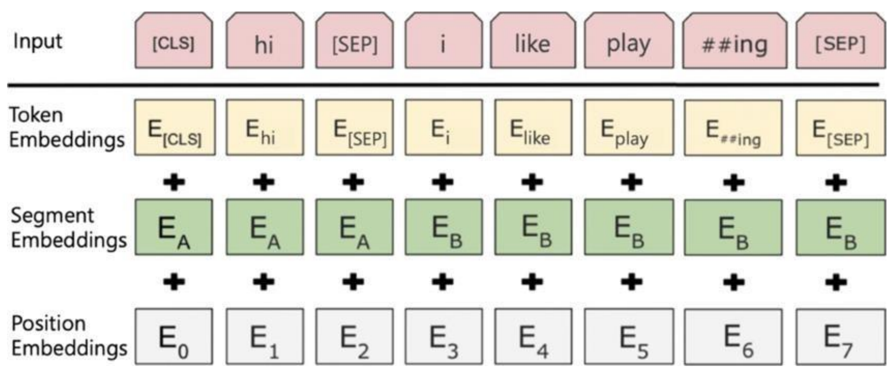


**Figure 2.** Example of BERT Input

IndoBERT is an Indonesian version of the BERT model and has several layers in its architecture, namely the Multi-Head Attention layer, the Position-wise Feed-Forward Networks layer, and the Add and Norm layer.

**a. Multi-Head Attention layer**

The Multi-Head Attention layer in the Transformer architecture consists of several self-attention layers that work in parallel, allowing the model to process various aspects of the relationships between words in a sentence. In this mechanism, the query, which is a vector representation of the word being analysed, is compared with the key, which is a vector representation of other words in the sentence, to determine their relevance through a compatibility function. Based on this relevance, the values associated with the key are weighted in a weighted sum calculation, which is then used as the output. In this way, self-attention helps the model capture the context of the entire sentence, enabling a better and deeper understanding of the relationships between words.

The output of this layer is as follows [10].

$$Attention\ (\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = softmax\left(\frac{1}{\sqrt{d_k}}\boldsymbol{QK^T}\right)\boldsymbol{V} \tag{1}$$

where **Q** is the query matrix, **K** is key matrix, **V** is value matrix, and $d_k$ is key dimension. If self-attention layers are stacked, they produce Multi-Head Attention layers.

$$MultiHead\ (\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = Concat(head_1,\dots,head_h)\boldsymbol{W^0} \tag{2}$$

$$head_i = Attention\left(\boldsymbol{QW_i^Q},\boldsymbol{KW_i^K},\boldsymbol{VW_i^V}\right) \tag{3}$$

Where:

$$\boldsymbol{W_i^Q} \in R^{d_{model}\times d_q}, \boldsymbol{W_i^K} \in R^{d_{model}\times d_k}, \boldsymbol{W_i^V} \in R^{d_{model}\times d_v}, dan\ \boldsymbol{W^0} \in R^{hd_v\times d_{model}}$$

Where $h = 12$ parallel attention layer and $d_k = d_v = \frac{d_{model}}{h} = \frac{768}{12} = 64$

$\boldsymbol{W_i^Q}$ : weight matrix for queries on the i-th head

$\boldsymbol{W_i^K}$ : weight matrix for key on the i-th head

$\boldsymbol{W_i^V}$ : weight matrix for value on the i-th head

$\boldsymbol{W^0}$ : weight matrix for combining the outputs of all heads

$h$ : number of heads in multi-head attention

**b. Position-wise Feed Forward Network Layer**

The feed forward layer plays a role in adding nonlinear transformations to the output of the multi-head attention layer. In this layer, there are two linear transformations and a relu activation between them. The relu activation function plays a role in introducing the concept of nonlinearity, which enables the network to model more complex patterns [13].

$$\mathrm{FFN}(\mathbf{x}) = \mathrm{ReLU}(\mathbf{xW}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{4}$$

Where:

$\boldsymbol{W}_1 \in R^{d_{model} \times d_{ff}}, \boldsymbol{b}_1 \in R^{d_{ff}}, \boldsymbol{W}_2 \in R^{d_{ff} \times d_{model}}, and\ \boldsymbol{b}_2 \in R^{d_{model}}$

$d_{ff} = 4d_{model} = 4(768) = 3072\ and\ \mathbf{x} \in R^{d_{model}}$

$\mathbf{W}_1$ : first layer weight matrix

$\mathbf{b}_1$ : first layer bias vector

$\mathbf{W}_2$ : second layer weight matrix

$\mathbf{b}_2$ : second layer bias vector

**c. Add and Norm Layer**

After passing through the self-attention layer and feed forward network, the output from these layers is added to the original input on an element-wise basis. Adding the input helps prevent the vanishing gradient problem. After that, the layers are normalised. Thus, the output from each sub-layer (multi-head attention and feed forward) is as follows:

$$\text{Output} = \text{LayerNorm}(\mathbf{x} + \text{Sublayer}(\mathbf{x})) \tag{5}$$

where Sublayer($\mathbf{x}$) is a function implemented in each sublayer.

### 2.6 Evaluation Model

One method that can be used to measure the performance of a classification model is the confusion matrix [14]. Table 1 is the confusion matrix in this study, which is a binary classification with only two class outputs. The True Negative ($TN$) value is the number of negative data correctly detected by the system, while False Positive ($FP$) is the number of negative data detected as positive data by the system. Similarly, True Positive ($TP$) is the number of positive data correctly detected by the system, and False Negative ($FN$) is the number of positive data detected as negative data by the system [15].

Table 1. Confussion Matrix 2x2

| | | **Predict** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **Actual** | **Positive** | TP *(True Positive)* | TN *(False Negative)* |
| | **Negatife** | FP *(False Positive)* | FN *(False Negative)* |

Precision describes the ratio of correct positive predictions compared to the total number of predicted positive results. The formula for calculating precision is shown in equation 6.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \times 100\% \quad (6)$$

Recall or sensitivity describes the ratio of true positive predictions compared to the total true positive data. The recall formula is described in equation 7.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \times 100\% \quad (7)$$

The F-1 Score describes the weighted average comparison of precision and recall. Accuracy is appropriate to use as a benchmark for algorithm performance if the dataset has a balanced amount of data. However, if the amounts are not close, it is better to use the F-1 Score as a benchmark (Sani et al., 2022). Equation 8 shows how the F-1 Score is calculated.

$$\text{F} - 1\ \text{Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall+ Precision}} \quad (8)$$

Accuracy describes how accurately the model classifies the entire data correctly. Equation 9 is the formula for calculating accuracy:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100\% \quad (9)$$

### 2.7 BERTopic

One effective method for identifying and modelling topics from a collection of texts is BERTopic, a deep learning-based approach that combines transformer-based embeddings (such as BERT), dimensionality reduction, and clustering to generate more coherent and contextual topics. BERTopic has proven capable of overcoming the limitations of traditional topic methods such as LDA, particularly in capturing the semantic meaning of short and unstructured texts such as application reviews [5].

BERTopic combines the power of text representation from transformer models such as BERT, dimensionality reduction using UMAP (Uniform Manifold Approximation and Projection), and clustering using HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with

Noise). With this approach, BERTopic is able to generate topics that are more coherent and semantically relevant.

Based on Figure 3, The BERTopic method has several main algorithms, namely sentence embedding (SBERT), dimension reduction using Uniform Manifold Approximation and Projection (UMAP), clustering using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), and Class-based Term Frequency-Inverse Document Frequency (c-TF-IDF) [7].

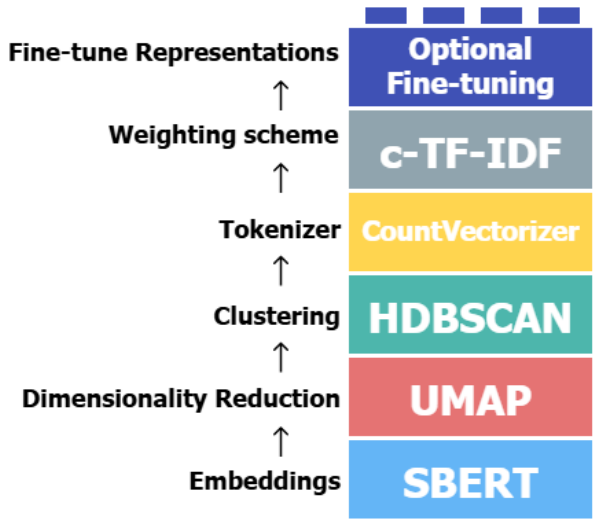


**Figure 3.** Algorithm BERTopic

**a. SBERT**

The initial stage of the BERTopic method is to convert the embedding for each piece of text data to be analysed. The embedding is a numerical vector of a certain dimension that will represent each piece of data. BERTopic uses Sentence BERT (SBERT) embedding [16]. Essentially, SBERT is a BERT model specifically designed to generate embeddings of sentences or paragraphs in clustering, sentence similarity, and information retrieval tasks. SBERT is trained using sentence pairs to assess the similarity between sentences.

SBERT will produce a 768-dimensional vector embedding representation for the base version. Unlike the IndoBERT model, the embedding produced by SBERT will represent the entire text, not just each token. This is due to the mean pooling algorithm, which averages the embedding for each word, so that the average value can represent the entire text. Furthermore, texts with similar meanings will have vector embedding representations with similar values. Several texts with similar meanings, when mapped, can be centred in an area with high density. Thus, these texts can be grouped into one cluster based on this density using the BERTopic algorithm (UMAP and HDBSCAN).

**b. UMAP**

Uniform Manifold Approximation and Projection (UMAP) is a method for performing dimensionality reduction using manifold learning techniques. Manifold learning is a non-linear dimensionality reduction technique [17]. High-dimensional embeddings are reduced to lower dimensions using the UMAP (Uniform Manifold Approximation and Projection) algorithm. This process enables the modelling of complex data structures in low-dimensional space, thereby simplifying the clustering process.

The first step in the UMAP method is to create a k-neighbour graph (KNN). The creation of a k-neighbour graph is intended to reveal local relationships in the data. The value of k as the closest neighbour in (KNN) is determined by the researcher. After that, the distance metric probability used to measure the proximity between data uses Euclidean distance. There are two vector embeddings $\boldsymbol{x_i} = (x_{i_1}, x_{i_2}, \dots, x_{i_{768}})$ and $\boldsymbol{x}_{\mathrm{j}} = (x_{\mathrm{j}_1}, x_{\mathrm{j}_2}, \dots, x_{\mathrm{j}_{768}})$, with $\boldsymbol{x_i}$, $\boldsymbol{x_j} \in \mathbb{R}^{768}$. Therefore, the Euclidean distance calculation for vectors $\boldsymbol{x_i}$ dan $\boldsymbol{x}_{\mathrm{j}}$ is as follow equation 10.

$$euclidean(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt{\sum_{g=1}^{768} \left(x_{i_g} - x_{j_g}\right)^2} \tag{10}$$

Where, $g = 1,2, \dots, 768$ is dimension from data.

UMAP uses exponential probability distribution to calculate the similarity between data vectors in high dimensions, with the following formula [18].

$$p_{i|j} = \exp\left(-\frac{euclidean(\boldsymbol{x_i}, \boldsymbol{x_j}) - \rho_i}{\sigma_i}\right) \tag{11}$$

Where:

$euclidean(\boldsymbol{x_i}, \boldsymbol{x_j})$ : Euclidean distance metric between data i and j

$\rho_i$: the distance between data point $i$ and its nearest neighbour

Value of $p_{i|j}$ will not be the same as the value $p_{j|i}$. UMAP uses symmetrisation on high dimensionality probabilities with the following formula.

$$p_{ij} = p_{i|j} + p_{j|i} - p_{i|j}p_{j|i} \tag{12}$$

Value of $\sigma_i$ determined using equation 13.

$$\sum_{j=1}^{k} \exp\left(-\frac{euclidean(\boldsymbol{x_i}, \boldsymbol{x_j})}{\sigma_i}\right) = \log(k) \tag{13}$$

In forming distances in low dimensions, UMAP uses similar probability values in the Student's t-distribution, which can be formulated as in equation 14.

$$q_{ij} = \left(1 + a\left(\boldsymbol{x}_i' - \boldsymbol{x}_j'\right)^{2b}\right)^{-1} \tag{14}$$

Where, $\boldsymbol{x}_i'$ and $\boldsymbol{x}_j'$ is a vector embedding that has been reduced to a low-dimensional space and for the value $a \approx 1{,}93$ and $b \approx 0{,}79$.

Furthermore, UMAP uses binary cross-entropy (CE) as the loss function. The binary cross-entropy formula can be formulated in equation 15.

$$CE(P, Q) = \sum_i \sum_j \left[p_{ij} log\left(\frac{p_{ij}}{q_{ij}}\right) + (1 - p_{ij}) log\left(\frac{1 - p_{ij}}{1 - q_{ij}}\right)\right] \tag{15}$$

Where, $P$ is the similarity probability in high-dimensional data and $Q$ is the similarity probability in low-dimensional data.

The cross-entropy is used to update the data in low-dimensional space for projection space optimisation until convergence is achieved. Stochastic Gradient Descent (SGD) is used by UMAP in performing this optimisation by calculating gradients for subsets in the data set.

### c. HDBSCAN

The reduced-dimension documents were then clustered using the HDBSCAN (Hierarchical

Density-Based Spatial Clustering of Applications with Noise) algorithm. This algorithm does not require a predetermined number of clusters and is capable of handling noise and outliers in the data.

HDBSCAN has several stages in performing the clustering process [19]. First, HDBSCAN will calculate the core distance for each data point with its closest neighbours using the Euclidean distance metric. Thus, the core distance calculation can be formulated as follows.

$$Core\ Distance\ (\boldsymbol{x}_i') = euclidean(\boldsymbol{x}_i', \boldsymbol{x}_{\boldsymbol{mpts}}) \tag{16}$$

Where $\boldsymbol{x}_i'$ is the embedding of the i-th data point that has been reduced in dimension using UMAP and $\boldsymbol{x}_{\boldsymbol{mpts}}$ is the closest neighbour to $\boldsymbol{x}_i'$ to the $mpts$, with $mpts = 1,2,\ldots,n_{mpts}$.

After this, mutual reachability distance (MRD) will be calculated for each pair of vectors, namely $\boldsymbol{x}_i'$ and $\boldsymbol{x}_j'$. MRD value for $\boldsymbol{x}_i'$ and $\boldsymbol{x}_j'$ vector obtained based on the maximum core distance value $\boldsymbol{x}_i'$, $\boldsymbol{x}_j'$ and distance bertween $\boldsymbol{x}_i'$ and $\boldsymbol{x}_j'$.

$$MRD(\boldsymbol{x}_i', \boldsymbol{x}_j') = max\left(Core\ Distance\ (\boldsymbol{x}_i'), Core\ Distance\ (\boldsymbol{x}_j'), euclidean(\boldsymbol{x}_i', \boldsymbol{x}_j')\right) \tag{17}$$

After obtaining these values for each pair of vectors, HDBSCAN will create a minimum spanning tree (MST), which is a series of nodes that connect all data with the minimum mutual reachability distance value without any cycles. Then, each vector will be assigned a weight with itself (self edge) in the form of the core distance value of that vector. The addition of the self edge weight will then be referred to as the extended minimum spanning tree (MSText).

After that, HDBSCAN will extract the hierarchy as a dendrogram from MSText. As the basis of the dendrogram, all data will be set as one cluster. Next, the iteration process will be carried out by removing nodes based on the largest MRD value in sequence. Nodes with the same MRD value will be removed simultaneously. After each node is removed, the data connected by the new node connection will be set as one cluster, while data that is not connected to other data will be considered noise.

**d. c-TF-IDF**

After the clusters are formed, BERTopic extracts keywords from each cluster using frequency-based and TF-IDF methods. These keywords are used to represent and name each topic that is formed. Data that has been clustered using the HDBSCAN algorithm will be assigned a topic. Each cluster that has been formed has its own characteristics, so these characteristics will be represented by specific keywords. The keywords in each cluster are determined using the class-based Term Frequency-Inverse Document Frequency (c-TF-IDF) algorithm. The c-TF-IDF formula can be written as follows [7].

$$W_{t,c} = tf_{t,c} log\left(1 + \frac{A}{tf_t}\right) \tag{18}$$

Where, $W$t,c is the weight formed in the c-TF-IDF algorithm for the tth word in the cth sentiment, $tf$t,c is the frequency of the tth word in the cth sentiment, $tf$t is the frequency of the nth word for the entire dataset, and $A$ is the total number of words for the entire dataset.

## 3. MAIN RESULTS

### 3.1 Data

In the dataset used, four variable columns were selected that provided information related to the username, rating, time of review submission, and content of the review written by the user. The reviews were then labelled by three judges. Before modelling, the data was divided into three types: training set, validation set, and testing set [20]. The proportion of data division is subjective and depends on the researcher. In this study, data division was carried out with a ratio of 81:9:10. Thus, 81% was training data used to train the model, 9% was validation data to evaluate the model during the training process to determine the best parameters, and 10% was testing data used to test the selected model.

### 3.2 Balancing Data

In this research data, positive sentiment has a proportion of 25% and negative sentiment has a proportion of 75%. Therefore, this research data is categorised as imbalanced data with a mild degree of imbalance. To overcome imbalanced data in this study, the text augmentation method was used. Data balancing was only carried out on training and validation data. The following is the proportion of sentiment after augmentation.


**Figure 4**. Proportion of sentiment after augmentation.

### 3.3 Sentiment Analysis with IndoBERT

In sentiment classification modelling using IndoBERT, hyperparameter tuning was performed on several hyperparameters, namely the number of neurons, batch size, epoch, and learning rate. Based on these hyperparameter adjustments, it was found that the optimal parameters for the IndoBERT model are 128 neurons, a batch size of 8, 12 epochs, and a learning rate of 10^-5. Figure 6 shows a graph of the training and validation process using the IndoBERT method with optimal parameters.

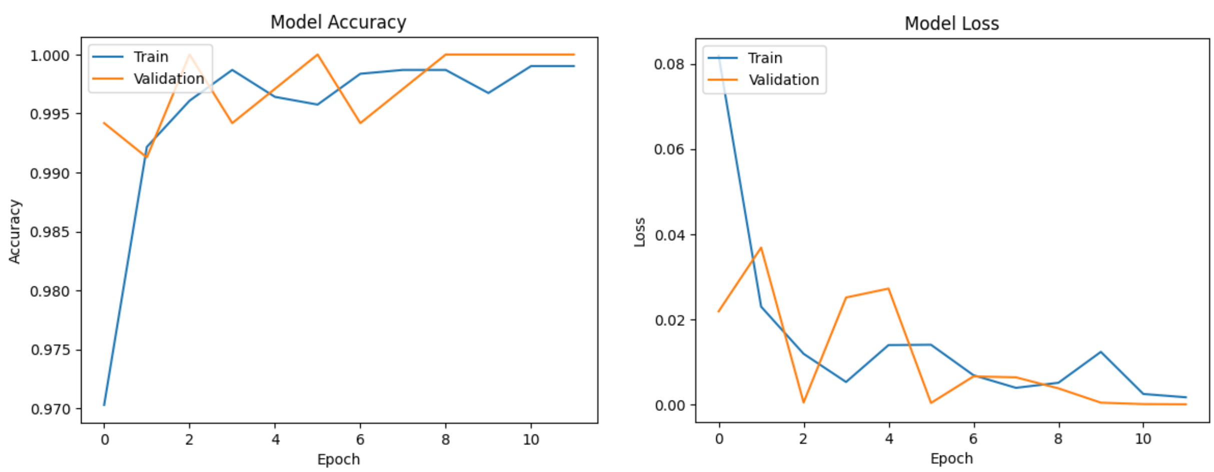


**Figure 5.** Training and Validation Accuracy and Loss Graphs

According to Figure 5, as the number of iterations (epochs) in the model increases, the accuracy graph shows an upward trend to the right. Meanwhile, the loss graph shows a downward trend. This indicates that the model has been trained well and is increasingly accurate in making predictions. In addition, the accuracy and loss values between the training data and the validation data are not that far apart. This indicates that there is no indication of overfitting in the model. This is a positive indication that the model has undergone effective training and is ready to be tested on test data. This study involves evaluating the performance of models that have been trained using test data with a confusion matrix, which provides additional detail in understanding the performance of models on test data. The process of evaluating models against test data is described in detail in Figure 6.

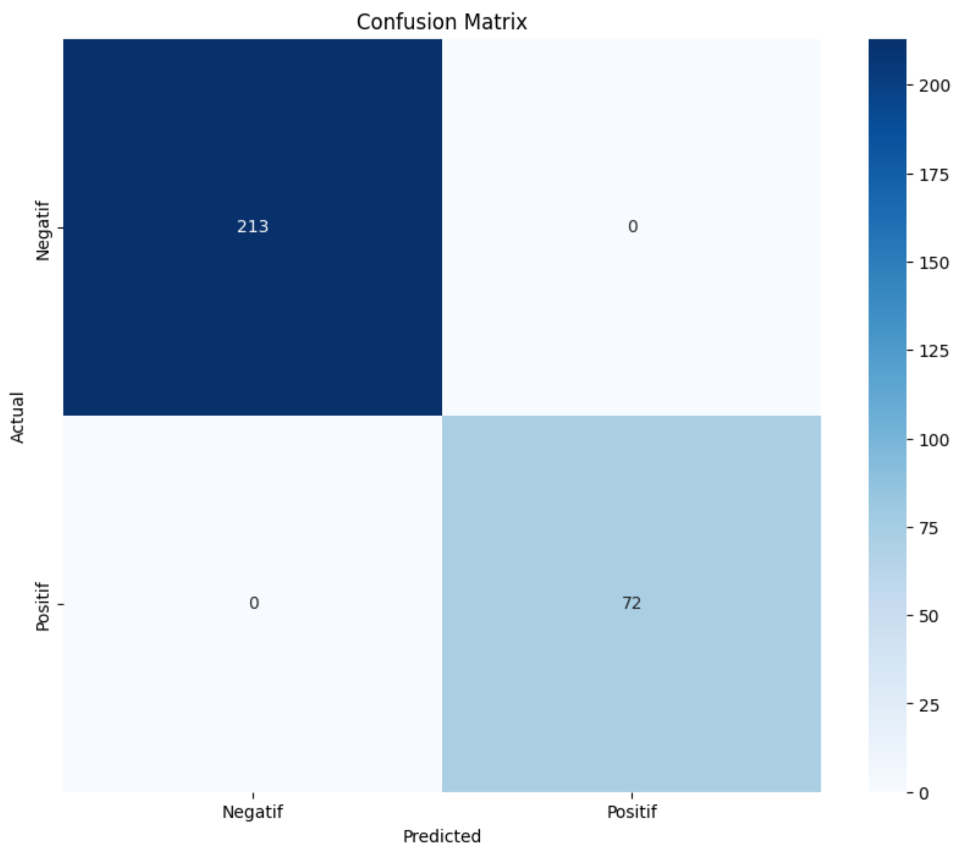


**Figure 6.** Confusion Matrix

According to Figure 6, the IndoBERT model demonstrates excellent predictive capabilities in classifying sentiments related to Jamsostek Mobile review. The results of this study show a satisfactory accuracy score of 100%, an F1 score of 100%, a recall of 100%, and a precision of 100%. Overall, the evaluation results show that this model has a perfect of accuracy, with a good balance between precision and recall. This model also demonstrates good ability in recognising positive examples and providing accurate positive predictions for sentiment analysis of the jamsostek mobile application.

### 3.4 Topic Modelling with BERTopic

Topic modelling analysis was conducted on the negative sentiment group because this research focused on improving the mobile social security application. The data used in the topic modelling analysis underwent pre-processing and was labelled. This allowed us to understand which topics were most frequently discussed in the use of the mobile social security application for negative sentiment by analysing the frequency of recurring words.

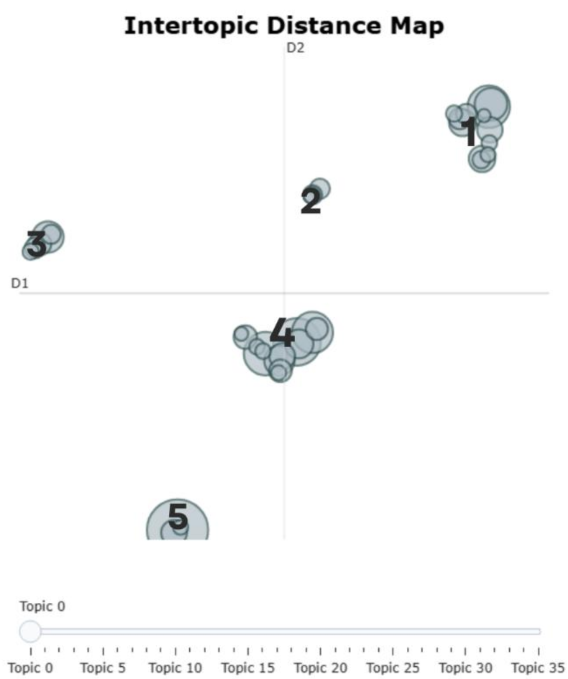


**Figure 7.** Clustering Negative Sentiment

Based on topic modelling analysis of negative sentiment in reviews of the Jamsostek Mobile (JMO) application, as shown in Figure 7, five clusters were formed that reflect negative sentiment from application users:

In the first cluster in quadrant I, several topics were formed. These topics contain users complaining that the JMO app always asks for updates every time the app is opened. However, even after updating, the JMO app still cannot be opened even though the internet connection used is stable.

In the second cluster, which is near the quadrant I line, there are 3 topics related to the active status of users and balances related to workers' monthly wages, which are used as monthly contributions. In the third cluster, located in quadrant II, several topics were formed discussing data updates that always fail. Additionally, biometric verification performed by users also consistently fails. Users also complained about the difficulty of making claims on the JMO application.

Then, in the fourth cluster, which is in the middle of the intertopic distance map, is the cluster with the most topics. The most discussed topic in this fourth cluster is related to application logins that are difficult and always error, even though the email and password entered are correct according to the account used. Then, users try to update the application and log in again, but it still fails even though the application has been updated, so users feel that logging into the JMO application is difficult. Furthermore, regarding user data such as bank accounts and tax identification numbers, it is also mentioned here that users find it difficult to fill in the data and claim their security deposits even though the required data has been provided.

Finally, in the fifth cluster, located in the bottom right quadrant, there are three topics discussing applications that are always slow when used. Then, in the warranty claim process, users also find it difficult because the application always crashes and the warranty money is difficult to withdraw.

## 4. CONLUSIONS

In this study, data collection was carried out using scraping to extract user reviews of the Jamsostek Mobile (JMO) application, resulting in a dataset of 2,846 reviews. A pre-processing step was then applied to improve the suitability of the data for algorithm implementation. The review data was then labelled to determine the sentiment polarity related to the JMO application. The research data showed an imbalance, with 75% of the data being negative sentiment and 25% being positive sentiment. To address this imbalance, data balancing was performed using augmentation. Sentiment analysis was then performed using IndoBERT, showing strong classification performance with an accuracy score of 100%, an F1 score of 100%, a recall of 100%, and a precision of 100%. The BERTopic model was applied to negative sentiment data because this study focused on improving the JMO application.

In topic modelling using BERTopic, five clusters were formed. In these clusters, the most discussed topic was about applications that always experience errors and are slow when used. In addition, every application always requires updates. Users also find it difficult to log in to the application and claim their security deposits.

Based on the topics discussed, the developers of the Jamsostek Mobile application are expected to focus on improving the application so that it can be accessed by many users without experiencing errors or slow performance. The issue of the application constantly requesting updates must also be addressed promptly, as many users are unable to access the application due to this issue.

Furthermore, technical issues related to the insurance claim process and user data updates must also be addressed as they affect user contributions and the claim process.

## ACKNOWLEDGMENT

This research was funded by an Internal Matching Fund (IMF) grant from Padjadjaran University, Indonesia, as part of a project entitled *‘Classification Model of Rice Plant Diseases Using Deep Learning and Gaussian Copula to Promote Sustainable Precision Agriculture’* (Contract Number 4356/UN6.D/PT.00/2025).

## CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

## REFERENCES

[1] Y. Zhang, W. Ren, T. Zhu, E. Faith, MoSa: A Modeling and Sentiment Analysis System for Mobile Application Big Data, Symmetry 11 (2019), 115. https://doi.org/10.3390/sym11010115.

[2] F.M. Matulatuwa, E. Sediyono, A. Iriani, Text Mining Dengan Metode Lexicon Based Untuk Sentiment Analysis Pelayanan PT. Pos Indonesia Melalui Media Sosial Twitter, Thesis, Universitas Kristen Satya Wacana, Indonesia, 2017.

[3] Y. Su, C.J. Kuo, On Extended Long Short-Term Memory and Dependent Bidirectional Recurrent Neural Network, Neurocomputing 356 (2019), 151-161. https://doi.org/10.1016/j.neucom.2019.04.044.

[4] B. Wilie, K. Vincentio, G.I. Winata, S. Cahyawijaya, X. Li, et al., IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding, in: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 2020, pp. 843-857. https://doi.org/10.18653/v1/2020.aacl-main.85.

[5] Y. An, H. Oh, J. Lee, Marketing Insights from Reviews Using Topic Modeling with BERTopic and Deep Clustering Network, Appl. Sci. 13 (2023), 9443. https://doi.org/10.3390/app13169443.

[6] R. Egger, J. Yu, A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts, Front. Sociol. 7 (2022), 886498. https://doi.org/10.3389/fsoc.2022.886498.

[7] M. Grootendorst, BERTopic: Neural Topic Modeling with a Class-Based TF-IDF Procedure, arXiv:2203.05794 (2022). https://doi.org/10.48550/arXiv.2203.05794.

[8] M. Al-Maitah, Text Analytics for Big Data Using Rough-Fuzzy Soft Computing Techniques, Expert Syst. 36 (2019), e12463. https://doi.org/10.1111/exsy.12463.

[9] T. Siddiqui, A.Y.A. Amer, N.A. Khan, Criminal Activity Detection in Social Network by Text Mining: Comprehensive Analysis, in: 2019 4th International Conference on Information Systems and Computer Networks (ISCON), IEEE, 2019, pp. 224-229. https://doi.org/10.1109/ISCON47742.2019.9036157.

[10] A. Vaswani, N. Shazeer, N. Parmar, et al., Attention Is All You Need, in: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.

[11] J. Devlin, M.W. Chang, K. Lee, K.T. Google, A.I. Language, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of NAACL-HLT 2019, Minneapolis, Minnesota, pp. 4171-4186, 2019.

[12] Y. Kim, T. Lee, Korean Clinical Entity Recognition from Diagnosis Text Using BERT, BMC Med. Inform. Decis. Mak. 20 (2020), 242. https://doi.org/10.1186/s12911-020-01241-8.

[13] M.S. Islam, L. Zhang, A Review on BERT: Language Understanding for Different Types of NLP Task, Preprint, (2024). https://doi.org/10.20944/preprints202401.1857.v1.

[14] S. Susanti, A.A. Sari, M.K. Anam, M. Jamaris, H. Hamdani, Sistem Prediksi Keuntungan Influencer Pengguna E-Commerce Shopee Affiliates Menggunakan Metode Naïve Bayes, Edumatic: J. Pendidik. Inform. 6 (2022), 394-403. https://doi.org/10.29408/edumatic.v6i2.6787.

[15] A. Baratloo, M. Hosseini, A. Negida, G. El Ashal, Part 1: Simple Definition and Calculation of Accuracy, Sensitivity and Specificity, Emerg. (Tehran), 3 (2015), 48-49.

[16] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Stroudsburg, PA, USA, 2019, pp. 3980-3990. https://doi.org/10.18653/v1/d19-1410.

[17] M. Meilă, H. Zhang, Manifold Learning: What, How, and Why, Annu. Rev. Stat. Appl. 11 (2024), 393-417. https://doi.org/10.1146/annurev-statistics-040522-115238.

[18] M. Allaoui, M.L. Kherfi, A. Cheriet, Considerably Improving Clustering Algorithms Using UMAP Dimensionality Reduction Technique: A Comparative Study, in: A. El Moataz, D. Mammass, A. Mansouri, F. Nouboud, (eds) Image and Signal Processing, ICISP 2020, Lecture Notes in Computer Science, vol. 12119. Springer, Cham, (2020). https://doi.org/10.1007/978-3-030-51935-3_34.

[19] R.J.G.B. Campello, D. Moulavi, J. Sander, Density-Based Clustering Based on Hierarchical Density Estimates, in: J. Pei, V.S. Tseng, L. Cao, H. Motoda, G. Xu (Eds.), Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013: pp. 160–172. https://doi.org/10.1007/978-3-642-37456-2_14.

[20] V.C. Raykar, A. Saha, Data Split Strategiesfor Evolving Predictive Models, in: A. Appice, P.P. Rodrigues, V. Santos Costa, C. Soares, J. Gama, A. Jorge (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, 2015: pp. 3–19. https://doi.org/10.1007/978-3-319-23528-8_1.