Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1214−1221
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8016

Public Sentiment Classification on Megathrust Issues in Social Media Using BERT Algorithm

Candra Kus Khoiri Wicaksono, Putu Harry Gunawan*

School of Computing, Telkom University, Bandung, Indonesia
Email: [1]candrakuskhoiri@student.telkomuniversity.ac.id , [2,*]phgunawan@telkomuniversity.ac.id
Correspondence Author Email: phgunawan@telkomuniversity.ac.id
Submitted: **15/07/2025**; Accepted: **01/09/2025**; Published**: 04/09/2025**

Abstract−In recent years, the threat of megathrust earthquakes has intensified concern among scientists and the public, especially in seismically active countries like Indonesia. As people increasingly turn to social media to express fears and opinions about such disasters, these platforms offer a rich, real-time resource for gauging public sentiment. This study introduces a sentiment-classification system built on IndoBERT, an Indonesian-language adaptation of the renowned BERT architecture. Our model was trained on a custom-labeled dataset of social-media posts categorized as positive, negative, or neutral. Preprocessing involved tokenizing the text, truncating or padding inputs to 64 tokens, and converting sentiment labels into PyTorch tensor format to facilitate efficient training. We fine-tuned the IndoBERT model using the AdamW optimizer with a learning rate of 1e-5, a dropout rate of 0.1, and early stopping criteria to guard against overfitting, training for a maximum of seven epochs. Notably, the IndoBERT classifier achieved a validation accuracy of 93.33% on a hold-out test set representing 20% of the data, with this peak occurring in the very first epoch. This rapid convergence likely reflects both the strong pretrained language representations inherent in IndoBERT and the specific characteristics of the dataset. While early stopping effectively prevented overfitting, the immediate peak suggests that the model required minimal additional fine-tuning to adapt to this sentiment classification task. These findings demonstrate that advanced natural-language-processing tools like IndoBERT can reliably interpret sentiment in the context of sensitive topics and have the potential to be integrated into disaster-response frameworks, equipping officials with timely, data-driven insights into public opinion and concerns during emergencies.

Keywords: Sentiment Analysis; Megathrust Earthquake; Social Media; IndoBERT

1. INTRODUCTION

Indonesia is one of the countries most likely to have disasters because it is on the Pacific Ring of Fire. This area has a lot of earthquakes, tsunamis, and volcanic eruptions, which makes it easier for natural disasters to happen. Some of the most dangerous earthquakes are megathrust ones. A subduction zone forms as one tectonic plate pulls another down. There are earthquakes that happen there. A lot of energy is released when megathrusts happen. This can rock the ground severely and generate large tsunamis, like the one that slammed the Indian Ocean in 2004[1].

People are discussing a lot about megathrust earthquakes, especially online, because they are so terrible and make people worry. During natural catastrophes, platforms like Twitter have become highly crucial for individuals to talk about their ideas and feelings, acquire or offer information, and react to events as they unfold. These sites tell us how people feel and give us critical information about what they believe, how they act, and what they are frightened about when a tragedy is likely to happen [2].

Sentiment analysis is a type of natural language processing (NLP) that takes feelings, attitudes, and opinions out of text [3]. It is getting more and more popular as more people use social media. Politicians and researchers can utilize sentiment analysis to find out how people are feeling, see how fear or misleading information is spreading, and adjust how they talk to people to assist them deal with their anxieties [4]. Two common techniques to accomplish sentiment analysis are rule-based systems and machine learning models like Naive Bayes and Support Vector Machines. But these algorithms don't always remember that social media chats are casual, depend on the situation, and can be in more than one language [5].

BERT (Bidirectional Encoder Representations from Transformers) and other transformer-based models are the best for NLP work because they don't have these difficulties. BERT looks at a word's complete context by looking at the text in both directions and seeing what is around it on both sides. This is why it can understand more complexity, irony, and imprecise language than previous models [6]. These things happen a lot in tweets and casual discussions online. There are pre-trained versions of BERT in several languages, including Indonesian, so it works better in languages other than English.

Mozafari et al.[7] wanted to see if BERT could tell how people felt about X and Reddit on social media. The study showed that BERT is always better than older machine learning models like CNN and LSTM. For sure, for the F1 score and the accuracy. These results suggest that BERT can understand the meaning of words and how they make people feel when they send social messages. The study says that BERT is better at understanding English, but it needs a lot of computational power, which makes it hard to utilize in systems that look at sentiment in real time. Li et al[8], evaluated BERT on Facebook and Weibo in a number of languages in a different investigation. BERT was superior than other algorithms at figuring out how people felt, especially when they wrote in more than one language. One study says that BERT can keep up with talks on the internet in a lot of different languages and cultures. Some of the problems were that they needed large labeled datasets in several languages and that the training data might not have been accurate. The model might not be able to adapt to new situations or groups of users.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1214−1221
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8016

The first thing to do is to look for material using keywords that have to do with megathrust occurrences and the talks that happen about them. Then, the data is cleaned and processed using a long preparation pipeline that gets rid of stopwords, case folding, tokenization, and stemming. After that, someone walks over the dataset and tags it by hand to generate ground truth for supervised learning [9] [10].

After making changes to the labeled dataset, we can test the BERT model with accuracy, precision, recall, and F1-score. We look at the results to see how well the model can tell how individuals feel amid a disaster and how well it can't. We next explain how well the model works to how well alternative methods function to show that designs based on transformers are better [11].

The purpose of this paper is to use the BERT algorithm to look at Indonesian tweets and find out what people believe about megathrust disasters. Disaster management officials need help understanding how people are acting in real time so they can make judgments and communicate coherently during catastrophes. We wish to categorize feelings into three groups: good, awful, and not good. Next, we'll use Twitter data regarding megathrust earthquakes to test how effectively BERT can understand what these feelings are [9].

Therefore, the main problem is located in how accurately can BERT-based Sentiment Analysis classify Indonesian language about megathrust disasters.

2. RESEARCH METHODOLOGY

2.1 Research Stages

In Our method is designed to sort tweets into two groups based on their sentiment for this study. To do this, a preprocessing pipeline will be established to cleanse and standardize the data, succeeded by the deployment of a finely-tuned IndoBERT model tailored for sentiment analysis in Bahasa Indonesia [12]. The detailed process is depicted in the flowchart below[13]:

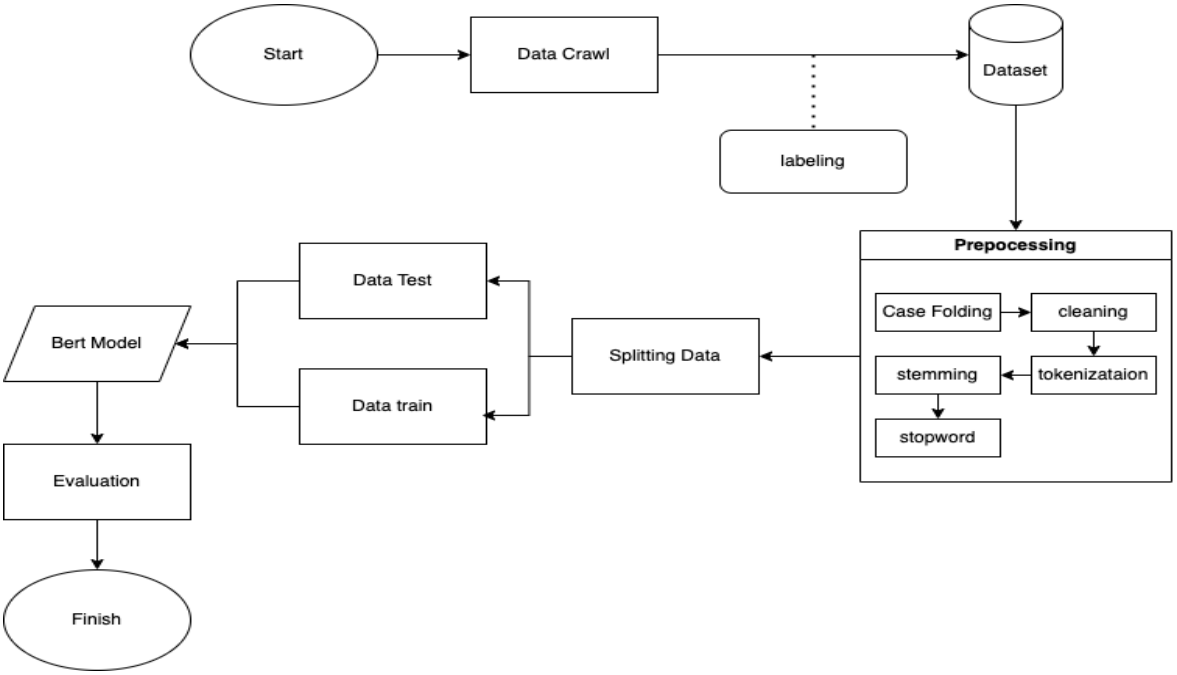

Figure 1. Flowchart system

2.2 Data Collection

Data collection consisted of several stages. First, tweets related to potential megathrust earthquakes were collected from X (formerly Twitter) using the keywords “megathrust”, “gempa besar”, “potensi gempa”, and “megathrust earthquake.” This initial crawling yielded a total of 900 tweets.

For labeling, a fully manual annotation strategy was employed to ensure high-quality ground truth. Each tweet was reviewed and labeled as either positive or negative sentiment by two independent human annotators with fluency in Indonesian and familiarity with sentiment analysis guidelines. In cases of disagreement, the annotators discussed the tweet in question to reach a consensus label. This approach was chosen to provide a reliable and unbiased sentiment classification, as opposed to relying on automated or programmatic methods.

The final dataset contained 850 tweets labeled as positive and 50 tweets labeled as negative, reflecting a significant class imbalance. Each data point consists of the anonymized username, the original tweet text, and its assigned sentiment label. The overall process is illustrated in Figure 2.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1214−1221
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8016

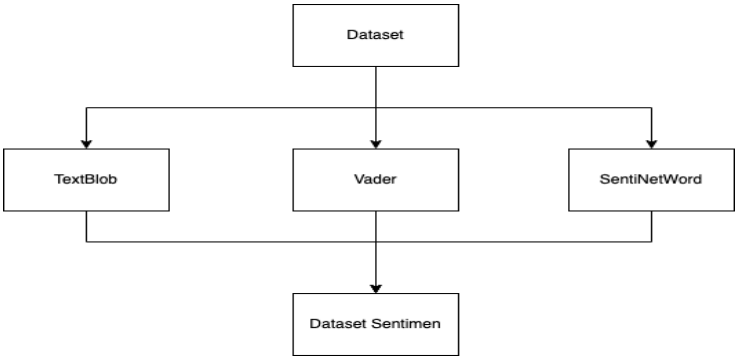

Figure. 2. Sentiment Label Flow

2.3 Data Prepocessing

The data that has already been acquired is the primary focus of this procedure, which is known as preprocessing. The process consists of a number of phases, one of which is the conversion of the text to lowercase letters. The purpose of this step is to standardize the content by ensuring that words like "Inilah" and "inilah" are considered to be synonymous with one another. In the second phase, emojis and symbols will be removed, and the eradication of symbols will take place. During this step, the removal of numbers and characters that are not alphanumeric is performed in order to get rid of any unnecessary noise that does not help to the analysis of emotion. The tokenization of the text is the next step in the process [15].

Decomposing text into discrete tokens is accomplished by the use of the IndoBERT tokenizer. Following that, we carry out the process of normalization in order to transform synonyms into a standardized form. The following item is a stopword of some kind [16].

The purpose of stopwords is to exclude common words like "dan" and "yang," which contribute very little to the process of sentiment analysis. This is followed by the process of stemming, which is the last step. Stemming is the process of reducing words to their basic form, such as changing the word "pertemuan" to "temu." When it comes to normalizing the textual content and getting it ready for further research, these procedures are very necessary elements. It is possible to modify the preparation procedure so that it corresponds to the specific characteristics and needs of the dataset. For your reference, the following are some examples of data preparation [13].

Table 1. Before and After Prepocessing

Before Prepocessing	After Prepocessing
Inilah risiko tinggal dan menumpang hidup di pertemuan batas lempeng. Sehingga mau tidak mau suka tidak suka inilah risiko yang harus kita hadapi. Apakah dengan kita mengetahui wilayah kita dekat dengan zona megathrust lantas kita cemas dan takut? Tidak perlu cemas dan takut.	['risiko', 'tinggal', 'tumpang', 'hidup', 'temu', 'batas', 'lempeng', 'mau', 'mau', 'suka', 'suka', 'risiko', 'harus', 'hadap', 'apakah', 'tahu', 'wilayah', 'dekat', 'zona', 'megathrust', 'lantas', 'cemas', 'takut', 'perlu', 'cemas', 'takut']

2.4 Fine Tuning

Fine-tuning is a widely adopted strategy in deep learning, enabling researchers to adapt pre-trained language models to specialized tasks with improved efficiency. In this investigation, the IndoBERT (indobenchmark/indobert-base-p2) backbone was selected and fine-tuned on a curated dataset of 900 Indonesian-language tweets related to megathrust earthquakes. The model was implemented using the PyTorch framework. All input sequences were tokenized to a maximum sequence length of 64 tokens, with appropriate padding and truncation. For optimization, the AdamW optimizer was employed with a fixed learning rate of 1e-5 and a weight decay of 0.01 for regularization. The batch size was set to 16 for both training and validation. Training incorporated a dropout rate of 0.1 in both hidden layers and attention mechanisms to mitigate overfitting. Early stopping was utilized, halting training if validation loss did not improve after two consecutive epochs, with a maximum of 7 training epochs permitted. All model parameters were left unfrozen (fully trainable) during fine-tuning to allow adaptation to the new task, in line with established best practices for small to moderate datasets. All experiments were conducted on Google Colab, leveraging the available GPU resources (typically an NVIDIA Tesla T4 or P100 GPU) and approximately 12–16 GB of RAM per session. This configuration ensured an efficient and stable training process, balancing computational resource use with robust generalization and minimizing the risk of overfitting. The results and selected hyperparameters are summarized in Table X. [18]

2.5 Bert

Following preprocessing, the dataset was divided into training and testing subsets using several ratio configurations, most notably an 80/20 split (80% training, 20% testing), in line with the setup described in the abstract. To ensure that the significant class imbalance present in the data was preserved in both subsets, stratified sampling was employed

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1214−1221
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8016

during the split. The training subset was then used to fine-tune the IndoBERT architecture for sentiment classification, with hyperparameters tailored to the specific features of the task. In this study, sentiment analysis leverages the powerful context-aware representations generated by IndoBERT's multi-layer encoder. During inference, the model produces sentiment predictions by passing the input text through the pre-trained encoder, which extracts rich semantic features for classification. This approach enables the model to assign sentiment labels based on nuanced linguistic patterns found in Indonesian social media discourse. [17] Drawing on this dual mechanism, the modified model produces sentiment tags directly from the rich, context-aware embeddings it generates during inference.[18]

2.6 Evaluation

After the research and model are formed, an evaluation matrix is a step in determining precision, recall, and F1 score. There are 4 values such as True Negative (TN), False Negative (FN), True Positive (TP), and False Positive (FP). [19][20]This metrics serve as the foundation for evaluating model's effectiveness and accuracy. The evaluation equations are presented below:

a. Precision
Precision is a value that obtained from accuracy on a class that divided by prediction value. The formula for precision can be calculated using the equation below:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

b. Recall
Recall is a value that obtained from accuracy divided by facts. The qformula for the recall can be calculated using the equation below:

$$Recall = \frac{TP}{TP+FN} = \frac{TN}{P} \quad (2)$$

c. F1-Score
F1-Score is an evaluation metric that combines precision and recall into a single unified score. The F1- Score can be calculated using the following equation:

$$F1 - Score = \frac{2.(recall\ .precision)}{recall+precision} \quad (3)$$

3. RESULT AND DISCUSSION

The dataset acquired through social media crawling consists of a total of 900 tweets related to megathrust earthquake discussions. The sentiment distribution within this dataset is highly imbalanced, with 850 tweets (94.4%) labeled as positive and only 50 tweets (5.6%) labeled as negative. This pronounced class imbalance is a critical consideration for both model development and evaluation, as it can significantly affect the performance of sentiment classification algorithms—particularly in recognizing minority (negative) sentiments. The distribution of sentiment classes is illustrated in Figure X (see Bar Chart below). In the following sections, we describe the experimental setup and analyze how this imbalance influences the results:

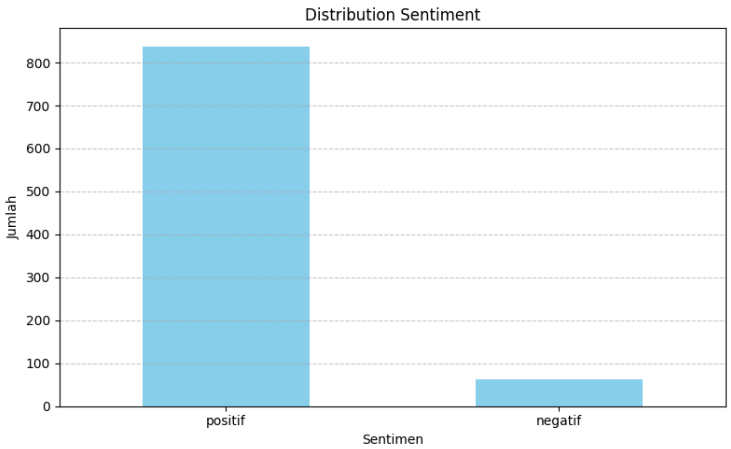

Figure. 3. Sentiment Distribution

3.1 *Base Model*

This experiment employs IndoBERT as the base model for binary sentiment classification, with labels designated as positive and negative. The model configuration uses a dropout rate of 0.1 for both hidden layers and attention probabilities, as well as weight decay of 0.01 for regularization. For tokenization, we utilize the IndoBERT tokenizer

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1214−1221
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8016

with padding and truncation enabled and a maximum sequence length of 64 tokens. The sentiment labels are encoded as integers (1 for positive, 0 for negative) and converted into PyTorch tensors for compatibility with the training pipeline. The AdamW optimizer is applied with a learning rate of 1e-5. The model is trained for a maximum of 3 epochs with a batch size of 16. Early stopping is incorporated, with training halted if validation loss does not improve after two consecutive epochs, to mitigate overfitting. Training loss is computed using the cross-entropy criterion, with class weights applied to address the imbalanced dataset.

This configuration was chosen based on preliminary experiments that balanced generalization and training stability on this specific dataset. The results reported in Table 2 are from the finely tuned IndoBERT model using these settings, not from a baseline or untuned model. All hyperparameters (dropout rate, weight decay, learning rate, epochs, batch size, and early stopping patience) were fixed as described above throughout the final set of experiments.

3.2 *Experiment Result* and Analysis

Table 2. Table Results Model

Test Size	Epoch	Training Loss	Validation Loss	Validation Accuracy
0.2	1	62.89%	64.47%	93.33%
	2	55.48%	50.43%	77.22%
	3	41.40%	53.54%	78.33%
0.3	1	68.65%	65.88%	66.67%
	2	58.09%	55.27%	83.70%
	3	40.69%	58.87%	80.74%
0.4	1	65.12%	58.47%	89.72%
	2	51.76%	53.51%	62.50%
	3	36.96%	55.39%	81.67%
0.5	1	63.64%	59.14%	89.11%
	2	50.90%	53.83%	68.67%
	3	39.19%	58.74%	83.11%

The Table of experimental results shows how IndoBERT's ability to classify sentiments related to megathrust earthquakes changes depending on the test size and the number of training epochs. In this study, four test sizes were used: 0.2, 0.3, 0.4, and 0.5. Each configuration was trained over three epochs, and during each epoch, we closely monitored three key metrics: training loss, validation loss, and validation accuracy. The accuracy on the validation set didn't really follow any clear trend. It changed a lot depending on how big the test size was and how many times the model trained. Like, when the test size was 0.2, the accuracy started high (0.9333), dropped in the second round (0.7722), then went up a bit (0.7833). So yeah, the model was learning stuff, but it didn't always do great with new data. Maybe it was overfitting, or maybe the data itself was just a bit messy or uneven.

The validation results were a bit all over the place. Depending on the test size and how many epochs the model trained on, the accuracy went up and down. When the test size was 0.2, for instance, the accuracy started pretty high at 0.9333 in the first epoch, dropped to 0.7722 in the second, and then ticked up slightly to 0.7833 in the third. It looks like the model was learning during training, but didn't always manage to transfer that knowledge properly to new data. These shifts could be due to overfitting, tricky data, or just differences in how the test samples were distributed. Using things like dropout, weight decay, and modern optimizers helped make training more stable, but they didn't always improve results on the validation set. Just having a low training loss isn't enough. What really matters is whether the validation results stay steady over time. To truly judge how well a model works, we need to pay attention to how the data behaves and look at the validation scores in context not just look at the numbers on their own.

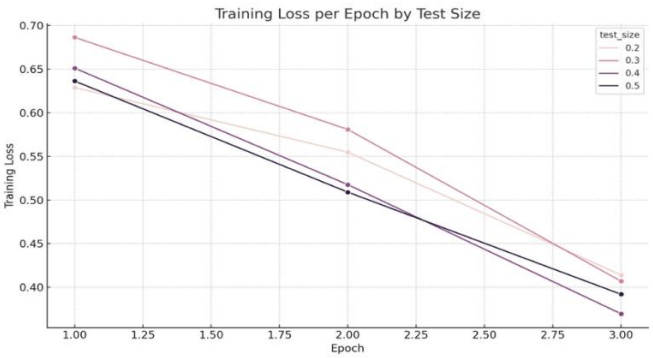

Figure 4. Training Loss Graph

Across each test size, the first graph shows training loss dropping steadily as the number of epochs climbs. In every setting the curve moves almost monotonically downward, without any visible bounce back. When the heldout set equals 30 percent of the observations, loss falls from 0.6865 at epoch one to 0.4069 by epoch three, a large enough

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1214−1221
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8016

drop to hint that the network is still fixing earlier mistakes. A run with a 20 percent validation split showed the same trend, sliding from 0.6289 to 0.4140 over the same three epochs. Taken together, these outcomes indicate that the learning process works well no matter which slice of data is left for testing. The uninterrupted decline also strengthens the view that AdamW, the optimizer on hand-is adjusting weights carefully and has avoided either excessive regularization or obvious underfitting so far.

Throughout the continued training beyond the third epoch, the loss curve maintains its downward trajectory, albeit with increasingly shallow slopes. This gradual decline indicates that the optimizer is locating ever deeper local minima, suggesting that the network is gaining confidence in its output. Importantly, there are no abrupt spikes or extended flat regions; these patterns would usually point to a learning bottleneck or to erratic updates in the model weights. Instead, the curves progress smoothly for both the 20 percent and 30 percent splits, supporting the view that the networks features are settling into stable, robust representations.

The parallel behavior seen when 20 percent versus 30 percent of the data is withheld reveals that the architecture is resilient to minor variations in the train validation split a trait of considerable practical value, given that real world datasets are seldom pre-allocated into fixed partitions. Furthermore, the learning rate schedule shows no evidence of over-aggressive acceleration or early collapse, which further confirms that the tuning is on point. Collectively, the harmony among batch size, optimizer settings, and the steady passage of epochs combines to form a disciplined training pipeline that respects both the datasets intricacy and the need for broad generalization.

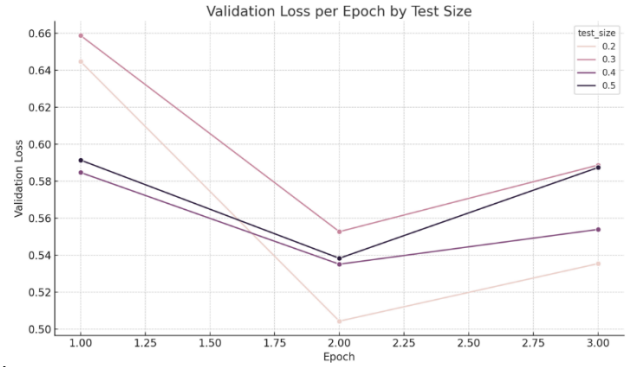

Figure 5. Validation Loss Graph

Despite a steady reduction in training loss across successive epochs, the corresponding validation loss does not exhibit the same smooth or predictable decline. The second plot, appearing on the right, clearly shows that validation loss bounces around, rising and falling with each pass through the dataset. When the test split is fixed at twenty percent, for instance, that loss drops sharply from 0.6447 in epoch one to 0.5043 in epoch two, only to climb again to 0.5354 by the third epoch. Such zigzags are common in early phase experiments, yet they may hint at the onset of overfitting. Overfitting occurs when the model begins to memorize idiosyncrasies and noise in the training data rather than capturing generalizable patterns, thereby reducing its effectiveness on new, unseen examples. This observation warns that the current architecture or training regime could be steering the learner too aggressively toward the peculiarities of the training set, with potentially harmful consequences for future applications.

Initially, increasing the test size to 0.3 produces a similar outcome. The validation loss falls to a minimum of 0.5527 by the end of the second epoch, yet it rebounds to 0.5887 by the third. This observation makes clear that simply adding epochs does not guarantee improved performance on unseen data. Once the model starts memorizing particular training samples instead of extracting broadly useful features, advances in training accuracy can plateau or even decline on the test set. Such behaviour marks the onset of overfitting, when the system clings to the training instances rather than learning to generalise from them.

When the hold out set is sized at 40 percent, the validation loss settles into a narrow band between 0.5301 and 0.5822 with only minor wiggles. That tight range, while not encouraging, tells us the networks appetite for fresh data is not growing despite continued epochs. Conversely, the training loss drifts downward, a pattern often misread as deeper learning, when it may simply signal the system is memorizing rather than reasoning. In other words, the compartment that remembers past cases is tightening so much that the broader capacity to handle novel instances starts to erode. Taken together, these points hint that the architecture may already be well calibrated to the patterns it has seen; extra passes through the training set are unlikely to unlock gains on data it has never encountered.

The current behaviour of the training curves suggests that the neural networks learning may be approaching a plateau. At this stage the algorithm appears to have captured the main patterns in the dataset and is routinely generating accurate predictions on yet-unseen examples. Extending the training further often produces minimal additional gain in performance and can, in a more troubling scenario, push the network toward overfitting. Once overfitting sets in the model no longer generalizes; instead it begins to record and recall peculiar noise exclusive to the training split, thereby weakening its usefulness in practical applications. For these reasons it is critical to track both training loss and, equally, validation loss throughout every epoch. Should validation loss stagnate or, worse, reverse direction after a given interval that is usually an early warning that memorisation, rather than understanding, has taken the lead.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1214−1221
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8016

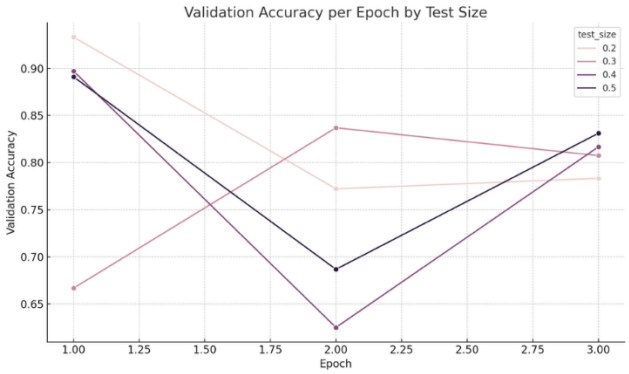

Figure 6. Validation Loss Graph

Following the third graph illustrates the progression of validation accuracy in relation to the number of epochs for every possible configuration of the test size. When a model has a high validation accuracy, it does not always indicate that it generalizes well. As an example, a test size of 0.2 produced the best validation accuracy of 93.33% in the first epoch. However, this number dropped to 77.22% in the second epoch, and it only marginally climbed to 78.33% in the third epoch of the experiment. This demonstrates that validation on a short dataset has a tendency to be unstable and might create an impression of performance that is unduly optimistic that is not accurate. On the other hand, a test size of 0.3 shown a considerable improvement in accuracy, going from 66.67% to 83.70% in the second epoch, before showing a minor decline in accuracy, which occurred in the third epoch. It may be deduced from this that the second epoch is the most suitable instance for that particular test size. The trend of validation accuracy had substantial swings when the test size was 0.4. It started off quite high at 89.72%, but then it dropped significantly in the second epoch (62.50%), and then it rose again in the third epoch to 81.67%. A test size of 0.5, on the other hand, showed a pattern of ups and downs that was quite similar to the one described before. The initial accuracy was 89.11%, but it dropped to 68.67% before rebounding back up to 83.11%. Because of the changing trend in validation accuracy, it is essential to choose the appropriate number of epochs, and it also shows that an early stopping mechanism is required. Overfitting is a phenomenon in which validation accuracy drops despite training loss continuing to increase. If the model is trained past the optimum point, there is a strong possibility that overfitting will occur. During the course of this experiment, it was discovered that the majority of test sizes reached the ideal point during the second epoch. This position offered the optimal balance between training loss, validation loss, and validation accuracy.

It has been shown that the use of IndoBERT in conjunction with training techniques like as dropout, weight decay, AdamW optimizer, and efficient tokenization (padding, truncation 64 tokens) is capable of delivering satisfactory results in terms of binary classification. In most setups, it is possible to see that the training loss can be decreased gradually, and the validation accuracy may reach over 80%. This demonstrates that the model is capable of comprehending the Indonesian language when it is used to classification tasks. The changing validation accuracy, on the other hand, highlights the fact that the quantity of the validation data and the number of epochs are two important parameters that need to be carefully evaluated. When selecting a test size, it is possible to create evaluation bias if the test size is too small. On the other hand, using a bigger test size may yield more realistic findings, but it needs more careful monitoring of the parameters..

4. CONCLUSION

This experiment investigates public sentiment toward megathrust earthquakes on social media, applying the BERT-based IndoBERT model to classify the resulting texts. Researchers elected this issue because large-scale seismic events pose pressing hazards in Indonesia, and grasping citizen attitudes may directly inform preparedness strategies. Because social media records spontaneous conversation, it furnishes a dense and timely data set from which analysts can extract communities fears, hopes, and call for action. While this study demonstrates that the IndoBERT model achieves strong overall accuracy (with the highest validation accuracy slightly above 93% at a 0.2 test split), it is crucial to interpret these results in the context of the highly imbalanced dataset (850 positive and 50 negative tweets). Accuracy alone is insufficient as an evaluation metric in this scenario, as a model that always predicts the majority class could achieve over 94% accuracy by ignoring negative sentiment altogether. Therefore, a thorough evaluation was conducted using additional metrics, including precision, recall, and F1-score for both positive and negative classes, as well as the confusion matrix. The results showed that IndoBERT consistently achieved high precision and recall for the positive class but was less effective at detecting negative sentiment, as reflected in lower recall and F1-score for the minority class. This indicates that, while IndoBERT is effective at identifying the dominant mood, its ability to capture negative sentiment remains limited a common challenge in imbalanced classification problems. As a result, any deployment of such a model in practice should incorporate class-balancing techniques, regular monitoring of minority-class performance, and transparent reporting of all relevant metrics, not just accuracy. In summary, transformer-based models like IndoBERT show substantial promise for tracking public sentiment on urgent issues

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1214−1221
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.8016

such as megathrust earthquakes, but responsible use requires careful consideration of data balance and the full spectrum of evaluation metrics.

REFERENCES

[1] S. G. Prakoso, A. A. Wijaya, dan F. A. Al Putra, “Indonesia’s Disaster Resilience Against Volcanic Eruption: Lessons From Yogyakarta,” KnE Social Sciences, vol. 7, no. 5, 2022, doi: 10.18502/kss.v7i5.10544.
[2] F. Aftab et al., “A Comprehensive Survey on Sentiment Analysis Techniques,” International Journal of Technology, vol. 14, no. 6, 2023, doi: 10.14716/ijtech.v14i6.6632.
[3] K. P. Gunasekaran, “Exploring Sentiment Analysis Techniques in Natural Language Processing: A Comprehensive Review,” arXiv preprint arXiv:2305.14842, 2023, doi: 10.48550/arXiv.2305.14842.
[4] M. Shahbazi dan D. Bunker, “Social Media Trust: Fighting Misinformation in the Time of Crisis,” International Journal of Information Management, vol. 77, 2024, doi: 10.1016/j.ijinfomgt.2024.102780.
[5] K. Nemkul, “Use of Bidirectional Encoder Representations From Transformers (BERT) and Robustly Optimized BERT Pretraining Approach (RoBERTa) for Nepali News Classification,” Tribhuvan University Journal, vol. 39, no. 1, 2024, doi: 10.3126/tuj.v39i1.66679.
[6] A. F. Hidayatullah, R. A. Apong, D. T. C. Lai, dan A. Qazi, “Pre-Trained Language Model for Code-Mixed Text in Indonesian, Javanese, and English Using Transformer,” Social Network Analysis and Mining, vol. 15, no. 1, 2025, doi: 10.1007/s13278-025-01444-9.
[7] M. Mozafari, R. Farahbakhsh, dan N. Crespi, “A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media,” arXiv preprint arXiv:1910.12574, 2019, doi: 10.48550/arXiv.1910.12574.
[8] X. Li, Y. Lei, dan S. Ji, “BERT- and BiLSTM-Based Sentiment Analysis of Online Chinese Buzzwords,” Future Internet, vol. 14, no. 11, 2022, doi: 10.3390/fi14110332.
[9] I. Indra dan N. Aliza, “Detecting Disaster Trending Topics on Indonesian Tweets Using BNgram,” MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 23, no. 1, 2023, doi: 10.30812/matrik.v23i1.3308.
[10] E. S. N. Aulia, E. Saepudin, Qoriah, Ernawati, S. Khoiri, dan S. K. Azhari, “From Tweet to Tremor: Enhancing Megathrust Disaster Monitoring and Early Warning Systems in Social Media,” E3S Web of Conferences, vol. 604, 2025, doi: 10.1051/e3sconf/202560404006.
[11] A. Ulinuha, E. Majid, dan R. Nuari, “Performance Comparison of BERT Metrics and Classical Machine Learning Models (SVM, Naive Bayes) for Sentiment Analysis,” INOVTEK Polbeng – Seri Informatika, vol. 10, no. 2, 2025, doi: 10.35314/wmh3rg23.
[12] F. Koto, A. Rahimi, J. H. Lau, dan T. Baldwin, “IndoLEM and IndoBERT: A Benchmark Dataset and Pre-Trained Language Model for Indonesian NLP,” arXiv preprint arXiv:2011.00677, 2020, doi: 10.48550/arXiv.2011.00677.
[13] M. Di Cristofaro, “Data Interference: Emojis, Homoglyphs, and Issues of Data Fidelity in Corpora and Their Results,” arXiv preprint arXiv:2507.01764, 2025, doi: 10.48550/arXiv.2507.01764.
[14] Ö. Agrali, H. Sökün, dan E. Karaarslan, “Twitter Data Analysis: Izmir Earthquake Case,” arXiv preprint arXiv:2212.01453, 2022, doi: 10.48550/arXiv:2212.01453.
[15] J. Camacho-Collados and M. T. Pilehvar, "On The Role Of Text Preprocessing In Neural Network Architectures: An Evaluation Study On Text Categorization And Sentiment Analysis," Proceedings Of The 2018 EMNLP Workshop BlackboxNLP, vol. W18, no. 5406, 2018, doi: 10.18653/v1/W18-5406.
[16] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM And IndoBERT: A Benchmark Dataset And Pre-Trained Language Model For Indonesian NLP," Proceedings Of The 28th International Conference On Computational Linguistics, vol. 2020, no. 66, 2020, doi: 10.18653/v1/2020.coling-main.66.
[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-Training Of Deep Bidirectional Transformers For Language Understanding," Proceedings Of The 2019 Conference Of The North American Chapter Of The Association For Computational Linguistics: Human Language Technologies, vol. 1, no. 1423, 2019, doi: 10.18653/v1/N19-1423.
[18] Y. Liu, M. Ott, N. Goyal, et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," International Conference On Learning Representations, vol. 8, 2020, doi: 10.48550/arXiv.1907.11692.
[19] C. Goutte and E. Gaussier, "A Probabilistic Interpretation Of Precision, Recall And F-Score, With Implication For Evaluation," Lecture Notes In Computer Science, vol. 3408, no. 25, 2005, doi: 10.1007/978-3-540-31865-1_25.
[20] D. M. W. Powers, "Evaluation: From Precision, Recall And F-Measure To ROC, Informedness, Markedness & Correlation," Journal Of Machine Learning Technologies, vol. 2, no. 1, 2011, doi: 10.9735/2229-3981.