The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

Deep Learning-Based Sentiment Analysis on Social Media Text Using Long Short-Term Memory (LSTM)

Liskedame Sipayung*, Megaria Purba, Bayu Pratama, Sri Yessi Saragih Sumbayak
Information Technology, STMIK Pelita Nusantara, Medan, Indonesia
Email : [1,*]liskedamesipayung@gmail, [2]megariapurba@gmail,[3]bayupratama@gmail, [4]sriyessisaragih@gmail
(* : liskedamesipayung@gmail)
Submitted: **25/05/2026;** Accepted: **18/07/2026;** Published: **29/07/2026**

Abstract - The large volume and informal nature of Indonesian social media text make manual sentiment analysis slow, inconsistent, and difficult to scale. This study evaluates a Long Short-Term Memory (LSTM) model for classifying public comments from X/Twitter into positive, neutral, and negative sentiment. A balanced dataset of 3,000 public Indonesian-language posts collected from January to March 2026 was manually labeled into three equal classes. Duplicate, irrelevant, advertising, and empty posts were removed; the remaining text underwent case folding, noise removal, tokenization, and padding. The data were stratified into 2,400 training and 600 testing instances. The model used a 10,000-word vocabulary, 100-token sequences, a 128-dimensional embedding, 128 LSTM units, dropout of 0.5, and a softmax output layer. On the held-out test set, the model obtained 87.00% accuracy, 86.80% precision, 86.50% recall, and 86.60% F1-score. Positive sentiment produced the strongest class-level performance, whereas neutral comments were more difficult because factual, ambiguous, and mixed expressions provide weaker affective cues. The findings show that LSTM provides a useful baseline for three-class Indonesian social media sentiment classification. However, generalization remains limited by the single-platform, topic-dependent dataset and the absence of repeated or cross-domain evaluation.
Keywords: Sentiment Analysis; Deep Learning; LSTM; Social Media; Text Preprocessing.

1. INTRODUCTION

The development of information technology has significantly changed the way people communicate, share information, and express opinions. One of the most widely used digital platforms for public communication is social media. Through social media, users can write comments, upload posts, respond to issues, review products, criticize services, and express personal experiences. These activities produce large amounts of text data that can reflect public opinion toward a particular topic, product, service, public policy, or social issue.

Social media text can reveal public perceptions of products, services, policies, and social issues, but its volume makes manual analysis slow and vulnerable to inconsistent interpretation. Automated text classification therefore provides a scalable basis for organizing opinion data [1].

Sentiment analysis is a Natural Language Processing (NLP) task that identifies the polarity or attitude expressed in text. Contemporary deep-learning research treats it as a context-sensitive classification problem rather than simple keyword counting [2]. The common three-class setting assigns positive, neutral, or negative labels, although the boundaries between these classes can be uncertain when a statement contains mixed or implicit evaluations [3].

Social media text is short, informal, and noisy. Slang, abbreviations, code-switching, hashtags, mentions, emojis, repeated characters, sarcasm, and topic-specific meanings can obscure sentiment. Shared-task research has shown that robust sentiment systems must account for these platform-specific characteristics and should be evaluated with clearly defined labels and test data [4].

Traditional sentiment classifiers commonly combine bag-of-words or TF-IDF features with algorithms such as Naive Bayes and Support Vector Machine. Neural text classifiers reduce dependence on manual feature engineering by learning distributed representations directly from sequences [5]. For Indonesian, IndoNLU provides standardized resources that include three-class sentiment data and demonstrates the importance of language-specific benchmarks [6].

Long Short-Term Memory (LSTM) is a recurrent architecture designed to retain information across sequences and mitigate the vanishing-gradient limitations of conventional recurrent networks. It is relevant to sentiment classification because changes in word order and negation can alter the meaning of an utterance. Indonesian benchmark studies have nevertheless shown that performance also depends strongly on dataset design and linguistic coverage [7].

This study uses LSTM to classify comments into positive, neutral, and negative categories. The model is selected as a transparent sequential baseline that can learn local word-order patterns without requiring a substantially larger pretrained transformer. Social-media-specific pretrained models such as IndoBERTweet provide a stronger contextual alternative and therefore define an important baseline for future comparison [8].

Indonesian social media introduces additional variation through colloquial spelling, regional expressions, abbreviations, and Indonesian-English code-switching. Multilingual Indonesian resources such as NusaX illustrate that linguistic and domain diversity materially affects sentiment transfer across communities [9]. These findings motivate explicit documentation of the platform, period, sampling criteria, labeling rules, and class distribution used in a new experiment.

Accordingly, this research evaluates an LSTM-based classifier on 3,000 public Indonesian-language posts collected from X/Twitter between January and March 2026. Keyword-based crawling was followed by removal of duplicates, advertisements, irrelevant posts, and empty texts. Recent Indonesian sentiment datasets confirm the value of documenting collection and annotation procedures so that reported performance can be interpreted within its domain [10].

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

The retained posts were manually assigned to positive, neutral, or negative classes using predefined guidelines. The final dataset contains 1,000 instances per class. Because annotation quality directly affects supervised learning, ambiguous items were reviewed before final labeling. Future replications should report the number of annotators and an inter-annotator agreement statistic; this information was not recorded in the present dataset. Transformer-based studies further demonstrate that contextual representations can improve classification of implicit sentiment [11].

Before modeling, the text underwent case folding and removal of URLs, mentions, hashtag markers, punctuation, numbers, symbols, and excess spaces. Tokens were mapped to integer sequences and padded to a fixed length. The balanced dataset was then stratified into 2,400 training and 600 testing instances. The split was performed before model evaluation, and the test set was reserved for final measurement. Attention-based architectures are a relevant future comparator because they model long-range contextual relations without recurrent processing [12].

The objective is to measure how effectively a configured LSTM classifies three-way Indonesian social media sentiment using accuracy, macro precision, macro recall, and macro F1-score. These complementary metrics are needed because accuracy alone can conceal class-specific weaknesses [13]. The contribution is an experimentally specified LSTM baseline and an analysis of errors involving neutral, mixed, and sarcastic expressions. Contextual sentence representations [14] and broad text-classification benchmarks [15] provide directions for stronger comparative evaluation.

2. RESEARCH METHODOLOGY

2.1 Research Design

This research uses a quantitative experimental approach by implementing a deep learning model, namely Long Short-Term Memory (LSTM), to classify sentiment in social media text data [10].The experimental approach is applied because this study involves several computational processes, including data collection, labeling, preprocessing, tokenization, sequence padding, model training, testing, and performance evaluation.

The main purpose of this research is to determine how well the LSTM model can classify social media comments into three sentiment categories: positive, neutral, and negative. In order to obtain more robust and representative results, this study uses an enlarged dataset consisting of 3,000 social media comments. The dataset is divided equally into three sentiment classes, namely 1,000 positive comments, 1,000 neutral comments, and 1,000 negative comments. This larger dataset is considered more appropriate for LSTM-based deep learning because the model requires sufficient training data to learn sequential text patterns effectively.

The data were collected from social media comments using a keyword-based crawling technique. The collected data were then filtered to remove duplicate comments, irrelevant text, advertisements, and empty data. After the filtering process, the final dataset was labeled and used for model development. The dataset was divided into training and testing data using an 80:20 ratio, consisting of 2,400 training samples and 600 testing samples.

2.2. Research Stages

The research is carried out through several systematic stages, starting from data collection to model evaluation. These stages are designed to ensure that the research process is clear, measurable, and suitable for obtaining valid results. Figure 1 shows the sequence of the research process. The research begins with collecting social media comments, followed by data labeling, text preprocessing, tokenization, sequence padding, training and testing data splitting, LSTM model design, model training, model testing, and evaluation.

The first stage is data collection. In this stage, social media comments are collected using a keyword-based crawling technique. The keywords are selected based on the topic being analyzed so that the collected comments are relevant to the purpose of the research. The initial data are then cleaned by removing duplicate comments, irrelevant comments, advertisements, and empty text. This process is carried out to ensure that the dataset used in the study is valid and suitable for sentiment classification.

The second stage is data labeling. Each comment is manually labeled into one of three sentiment categories: positive, neutral, or negative. Comments that express satisfaction, support, agreement, or positive evaluation are categorized as positive. Comments that contain complaints, criticism, dissatisfaction, or negative evaluation are categorized as negative. Meanwhile, comments that are factual, informative, unclear, or do not show a strong emotional tendency are categorized as neutral. The final labeled dataset consists of 3,000 samples, with 1,000 samples for each sentiment class.

The third stage is text preprocessing. This stage is conducted to clean the text data before it is processed by the LSTM model. The preprocessing process includes case folding, removing URLs, mentions, hashtags, punctuation marks, numbers, symbols, emojis, and excessive spaces. This stage is important because social media text is often informal, noisy, and unstructured.

The fourth stage is tokenization. In this stage, the cleaned text is converted into numerical tokens. Each word in the text is represented by a numerical index so that it can be processed by the deep learning model. After tokenization, the fifth stage, sequence padding, is applied to make all text sequences have the same length. This is necessary because the LSTM model requires input data with a consistent sequence length.

The sixth stage is splitting the dataset into training and testing data. This study uses an 80:20 data splitting ratio. From a total of 3,000 samples, 2,400 samples are used as training data, while 600 samples are used as testing data. The

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

training data are used to train the LSTM model, while the testing data are used to evaluate the model's ability to classify new data.

The seventh stage is LSTM model design. The model architecture consists of an embedding layer, an LSTM layer, a dropout layer, a dense layer, and an output layer. The embedding layer is used to transform numerical tokens into vector representations. The LSTM layer is used to learn sequential patterns in the text, while the dropout layer is used to reduce overfitting. The output layer classifies the input text into positive, neutral, or negative sentiment categories.

The eighth stage is model training. In this stage, the LSTM model is trained using the training dataset. The training process is carried out by adjusting the model parameters so that the model can learn the relationship between text patterns and sentiment labels. After training, the ninth stage is model testing, where the trained model is tested using the testing dataset.

The final stage is evaluation and result analysis. The model performance is evaluated using accuracy, precision, recall, and F1-score. Accuracy measures the overall correctness of the model, precision measures the proportion of correct positive predictions, recall measures the model's ability to identify the correct sentiment class, and F1-score provides a balanced measurement between precision and recall. These evaluation metrics are used to determine the effectiveness of the LSTM model in classifying sentiment in social media comments.

Fig 1. Research Stages

Figure 1 shows the sequence of the research process. The research begins with collecting social media comments, followed by labeling, preprocessing, tokenization, padding, data splitting, model design, training, testing, and evaluation.

2.3 Data Collection

The data used in this study consist of text comments collected from social media platform[9]. he comments represent user opinions, responses, or experiences related to a particular topic, product, service, or issue. In order to produce more robust and representative results for LSTM-based deep learning research, this study uses an enlarged dataset consisting of 3,000 social media comments.

The dataset was collected from social media comments using a keyword-based crawling technique. The keywords were selected based on the topic being analyzed so that the collected comments were relevant to the research objective. The data collection process focused only on public text comments. After the crawling process, the data were filtered by removing duplicate comments, irrelevant comments, advertisements, empty texts, and comments that did not contain meaningful sentiment information.

The final dataset consists of three sentiment categories: positive, neutral, and negative. Each sentiment category contains 1,000 comments, resulting in a balanced dataset. This balanced distribution was used to reduce class imbalance and to help the LSTM model learn sentiment patterns more effectively. The dataset was then used for the next stages, including labeling, preprocessing, tokenization, sequence padding, model training, and model evaluation.

The dataset contains two main attributes: comment text and sentiment label, as shown in Table 1.

Table 1. Dataset Attribute Description

Attribute	Description
Comment	Text data obtained from social media comments
Sentiment	Sentiment label consisting of positive, neutral, or negative

The distribution of the dataset is shown in Table 2.

Table 2. Dataset Distribution

Sentiment Class	Number of Samples
Positive	1,000
Neutral	1,000
Negative	1,000

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

Sentiment Class	Number of Samples
Total	3,000

An example of the dataset format is presented in Table 3.

Table 3. Example of Dataset

No	Comment	Sentiment
1	This application is very helpful and easy to use	Positive
2	I cannot judge it yet because I just started using it	Neutral
3	This application often crashes when opened	Negative
4	The service is fast and the features are useful	Positive
5	The information provided is still general	Neutral
6	The system is slow and difficult to access	Negative

2.4 Data Labeling

After the data were collected and filtered, each comment was labeled based on its sentiment category [11]. The labeling process was carried out to determine whether a comment contained positive, neutral, or negative sentiment. In this study, the final dataset consisted of 3,000 social media comments, which were divided into three balanced sentiment classes: 1,000 positive comments, 1,000 neutral comments, and 1,000 negative comments.

The labeling process was conducted manually by following predefined sentiment label criteria. Manual labeling was used to ensure that each comment was classified based on its actual meaning and context. Comments that contained praise, satisfaction, support, agreement, or positive experiences were labeled as positive. Comments that contained information, questions, general statements, or did not show a strong emotional tendency were labeled as neutral. Meanwhile, comments that contained complaints, criticism, dissatisfaction, disappointment, or negative experiences were labeled as negative.

To maintain labeling consistency, each comment was reviewed based on the same criteria. Ambiguous comments were examined carefully before assigning the final label. The labeled sentiment category was then used as the target class in the model training and testing process.

Table 4. Sentiment Label Criteria

Sentiment Class	Criteria
Positive	Comments containing praise, satisfaction, support, or positive experiences
Neutral	Comments containing information, questions, or statements without strong emotion
Negative	Comments containing complaints, criticism, disappointment, or negative experiences

The labeling result is used as the target class in the model training process.

2.5 Text Preprocessing

Text preprocessing was performed to clean the raw social media comments before they were used as input for the LSTM modelmodel[12]. This stage is important because social media text usually contains noise, such as uppercase letters, URLs, mentions, hashtags, numbers, punctuation marks, emojis, symbols, repeated characters, and excessive spaces. Since the dataset in this study consisted of 3,000 social media comments labeled as positive, neutral, and negative, preprocessing was needed to make the text more consistent and suitable for the next stages, including tokenization, sequence padding, model training, and model testing.

The preprocessing stages used in this research include the following:

1. Case Folding
   Case folding was conducted by converting all characters into lowercase. This process was used to avoid different representations of the same word. For example, the words “Helpful”, “HELPFUL”, and “helpful” were treated as the same word after case folding.
2. Cleaning
   Cleaning was carried out to remove unnecessary elements from the comments, such as URLs, mentions, hashtags, numbers, punctuation marks, symbols, emojis, and irrelevant characters. This process helped reduce noise in the dataset and made the text easier to process by the model.
3. Removing Extra Spaces
   After the cleaning process, excessive spaces were removed from the text. This step ensured that the comments had a cleaner and more consistent structure before being converted into numerical sequences.
4. Text Normalization
   Text normalization was applied to standardize informal words, abbreviations, or non-standard expressions commonly found in social media comments. For example, informal words or shortened expressions were converted into more standard forms when necessary. This stage helped the model understand the meaning of words more consistently.
5. Removing Irrelevant Comments

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

Comments that did not contain meaningful text, were empty after cleaning, or were not relevant to the research topic were removed from the dataset. This process was conducted to maintain the quality of the dataset used for LSTM-based sentiment classification.

The output of this stage was clean text that was more consistent, structured, and easier for the LSTM model to process. The cleaned text was then used in the tokenization stage to be converted into numerical form.

Table 5. Example of Text Preprocessing

No	Original Text	Cleaned Text
1	This Application is VERY helpful!!!	this application is very helpful
2	This app often crashes :(	this app often crashes
3	Check the update at https://example.com	check the update
4	@user I really like this service!!! #good	i really like this service
5	The system is slow… very slow!!!	the system is slow very slow
6	I cannot judge it yet because I just started using it	i cannot judge it yet because i just started using it

2.6 Tokenization

Tokenization is the process of converting cleaned text into a sequence of numerical tokens. This stage was carried out after text preprocessing, where the social media comments had been cleaned from noise such as URLs, mentions, hashtags, punctuation marks, numbers, symbols, emojis, and excessive spaces. Tokenization is required because the LSTM model cannot process raw text directly. Therefore, each word in the cleaned comments must be converted into a numerical representation.

In this study, tokenization was applied to the cleaned dataset consisting of 3,000 social media comments, which included 1,000 positive comments, 1,000 neutral comments, and 1,000 negative comments. A tokenizer was used to build a vocabulary from the training data. Each unique word in the vocabulary was assigned a numerical index. After that, every comment was converted into a sequence of numbers based on the word indexes stored in the tokenizer vocabulary.

For example, the cleaned comment:

“this application is very helpful”

can be converted into a numerical sequence such as:

[12, 4, 7, 21, 35]

Each number represents a word in the tokenizer vocabulary. In this example, the word “this” is represented by 12, “application” by 4, “is” by 7, “very” by 21, and “helpful” by 35. The numerical sequence generated from this process was then used as input for the next stage, namely sequence padding.

Table 6. Example of Tokenization

No	Sentiment	Cleaned Text	Token Sequence
1	Positive	this application is very helpful	[12, 4, 7, 21, 35]
2	Neutral	i cannot judge it yet because i just started using it	[3, 56, 78, 9, 64, 33, 3, 89, 102, 45, 9]
3	Negative	this application often crashes when opened	[12, 4, 18, 91, 27, 66]

Through tokenization, the cleaned text data were transformed into numerical sequences that could be processed by the LSTM model. The output of this stage was then standardized using sequence padding so that all input sequences had the same length

2.7 Sequence Padding

After tokenization, each comment was converted into a numerical sequence. However, the length of each sequence may be different because every social media comment contains a different number of words. Some comments are short, while others are longer. Therefore, sequence padding was applied to make all input sequences have the same length before being processed by the LSTM model.

In this study, sequence padding was applied to the tokenized dataset consisting of 3,000 social media comments, including 1,000 positive comments, 1,000 neutral comments, and 1,000 negative comments. Padding was needed because the LSTM model requires input data with a fixed sequence length. Without padding, the model would not be able to process comments with different sequence lengths in one training process.

The maximum input length in this research was set to 100 tokens. If a comment sequence was shorter than 100 tokens, zero values were added to the end of the sequence until it reached the maximum length. If a comment sequence was longer than 100 tokens, the sequence was truncated so that only the first 100 tokens were used. This process ensured that all comments had the same input dimension.

For example, the tokenized comment:

[12, 4, 7, 21, 35]

with a maximum sequence length of 8 tokens becomes:

[12, 4, 7, 21, 35, 0, 0, 0]

In this example, three zero values were added because the original sequence contained only five tokens. The same padding process was applied to all tokenized comments in the dataset before they were used in model training and testing.

Table 7. Example of Sequence Padding

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

No	Sentiment	Token Sequence	Padded Sequence
1	Positive	[12, 4, 7, 21, 35]	[12, 4, 7, 21, 35, 0, 0, 0]
2	Neutral	[3, 56, 78, 9, 64, 33]	[3, 56, 78, 9, 64, 33, 0, 0]
3	Negative	[12, 4, 18, 91, 27, 66]	[12, 4, 18, 91, 27, 66, 0, 0]

The output of this stage was a set of numerical sequences with the same length. These padded sequences were then used as input data for the LSTM model during the training and testing stages.

2.8 Data Splitting

After the text data had gone through preprocessing, tokenization, and sequence padding, the dataset was divided into two parts: training data and testing data. Data splitting was carried out to evaluate the performance of the LSTM model on data that had not been seen during the training process. The training data were used to train the LSTM model, while the testing data were used to measure the model's ability to classify sentiment on unseen data.

In this study, the dataset consisted of 3,000 social media comments, with a balanced distribution of 1,000 positive comments, 1,000 neutral comments, and 1,000 negative comments. The dataset was split using an 80:20 ratio, where 80% of the data were used for training and 20% were used for testing. Therefore, 2,400 comments were used as training data, while 600 comments were used as testing data.

The splitting process was conducted randomly while maintaining the class distribution so that each sentiment class remained represented in both the training and testing datasets. From each sentiment class, 800 comments were used for training and 200 comments were used for testing. This balanced splitting process was applied to reduce class imbalance and to ensure that the model could learn sentiment patterns from all classes equally.

Table 8. Data Split Ratio

Data Type	Percentage	Number of Samples	Function
Training Data	80%	2,400	Used to train the LSTM model
Testing Data	20%	600	Used to test and evaluate the LSTM model
Total	100%	3,000	Complete dataset

Table 9. Data Split Based on Sentiment Class

Sentiment Class	Total Samples	Training Data	Testing Data
Positive	1,000	800	200
Neutral	1,000	800	200
Negative	1,000	800	200
Total	3,000	2,400	600

This data splitting strategy was used to ensure that the LSTM model was trained and evaluated using a balanced dataset. As a result, the evaluation results could better represent the model's performance in classifying positive, neutral, and negative social media comments.

2.9 LSTM Model Design

The model used in this research is Long Short-Term Memory (LSTM). LSTM is a type of Recurrent Neural Network designed to process sequential data. Since text data has a sequential structure, LSTM is suitable for sentiment classification.

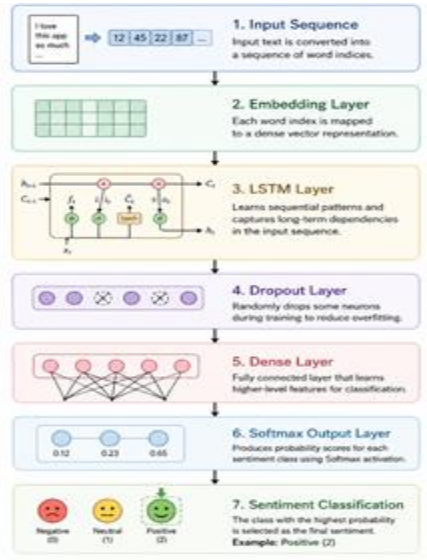

Fig 2.LSTM Model Archictutre

The architecture consists of several layers:

Table 10. LSTM Architecture Description

Layer	Function

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

Input Sequence	Receives text data that has been converted into numerical sequences
Embedding Layer	Converts word indexes into dense vector representations
LSTM Layer	Learns sequential patterns and word order in comments
Dropout Layer	Reduces overfitting during training
Dense Layer	Processes extracted features for classification
Softmax Output Layer	Produces probability scores for positive, neutral, and negative classes

2.10 Model Parameters

The model parameters used in this research are shown in Table 11.

Table 11. LSTM Model Parameters

Parameter	Value
Model	LSTM
Classification Type	Multi-class classification
Number of Classes	3
Sentiment Classes	Positive, Neutral, Negative
Maximum Vocabulary Size	10,000 words
Maximum Sequence Length	100 tokens
Embedding Dimension	128
LSTM Units	128
Dropout Rate	0.5
Dense Layer	64 neurons
Dense Activation Function	ReLU
Output Activation Function	Softmax
Optimizer	Adam
Loss Function	Categorical Crossentropy
Epochs	10
Batch Size	32
Training-Testing Ratio	80:20

2.11 Model Training

The training process is conducted using the training dataset. During training, the LSTM model learns word patterns and relationships that indicate positive, neutral, or negative sentiment. The model is compiled using the Adam optimizer and categorical crossentropy as the loss function. Adam is used because it is widely applied in deep learning and can optimize the learning process efficiently. Categorical crossentropy is used because the research involves multi-class classification. A validation set may be used during training to monitor model performance and reduce the possibility of overfitting.

2.12 Model Testing

After the training process is completed, the model is tested using the testing dataset. The testing process aims to measure the ability of the model to classify new comments that were not used during training.
The model predicts the sentiment class of each test comment. The predicted labels are then compared with the actual labels to determine the model's performance.

2.13 Evaluation Method

Model evaluation is carried out using several evaluation metrics, namely accuracy, precision, recall, F1-score, and confusion matrix.

Table 12. Evaluation Metrics

Metric	Description
Accuracy	Measures the ratio of correct predictions to the total number of data
Precision	Measures the correctness of predictions for each class
Recall	Measures the ability of the model to identify all data from each class
F1-score	Measures the balance between precision and recall
Confusion Matrix	Shows the number of correct and incorrect predictions for each class

The evaluation results are used to determine how well the LSTM model performs in classifying sentiment from social media text data.

2.14 Result Analysis

The final stage is analyzing the evaluation results. The analysis is conducted by observing the values of accuracy, precision, recall, F1-score, and confusion matrix. If the model achieves high accuracy and F1-score, it indicates that the LSTM model can classify sentiment effectively. However, if classification errors occur, the possible causes are analyzed,

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

such as limited dataset size, imbalanced data, ambiguous comments, slang words, or sarcastic expressions. This analysis helps explain the strengths and weaknesses of the LSTM model in performing sentiment analysis on social media text data.

3. RESULT AND DISCUSSION

3.1.1 Results

This section presents the results of implementing the Long Short-Term Memory (LSTM) model for sentiment analysis on social media text data. The purpose of this experiment is to classify social media comments into three sentiment categories: positive, neutral, and negative**.** The research process includes text preprocessing, tokenization, sequence padding, model training, and model evaluation. The model performance is evaluated using accuracy, precision, recall, F1-score, and confusion matrix**.**

3.1.2 Dataset Description

The dataset used in this study consists of 3,000 social media text comments that have been labeled into three sentiment classes: positive, neutral, and negative. Each comment represents a user's opinion, response, or experience toward a particular topic, product, service, or issue. The dataset was collected from social media comments and then filtered to remove duplicate data, irrelevant comments, advertisements, empty texts, and comments that did not contain meaningful sentiment information. After the filtering and labeling process, the final dataset was balanced across the three sentiment categories.

Table 13. Dataset Distribution

Sentiment Class	Number of Data	Percentage
Positive	1,000	33.33%
Neutral	1,000	33.33%
Negative	1,000	33.33%
Total	3,000	100%

A balanced dataset also supports a more reliable evaluation process. Since each class has the same proportion, the model performance can be measured more objectively across all sentiment categories. Therefore, the classification results are expected to better represent the model's ability to identify positive, neutral, and negative sentiments in social media comments. Based on Table 13, the dataset is balanced because each sentiment class has the same number of data. A balanced dataset helps the model learn each sentiment class more fairly and reduces the possibility of bias toward a particular class.

3.1.3 Preprocessing Results

Before being processed by the LSTM model, the social media text data was cleaned through several preprocessing stages. These stages include case folding, removing URLs, mentions, hashtags, punctuation marks, numbers, symbols, and excessive spaces.

Table 14. Example of Text Preprocessing Results

Original Text	Cleaned Text	Sentiment
This application is VERY helpful!!!	this application is very helpful	Positive
I cannot judge it yet because I just started using it	i cannot judge it yet because i just started using it	Neutral
This application often crashes when opened :(	this application often crashes when opened	Negative

Based on Table 14, the preprocessing process successfully converts text into a cleaner and more consistent form. This process is important because raw social media text often contains noise that can affect the model's learning process.

3.1.4 Tokenization and Padding Results

After preprocessing, the cleaned text was converted into numerical sequences through the tokenization process. Each word in the text was assigned a numerical index based on the tokenizer vocabulary. Since each comment has a different number of words, sequence padding was applied to make all input sequences have the same length.

Table 15. Example of Tokenization and Padding

Cleaned Text	Token Sequence	Padded Sequence
this application is very helpful	[12, 4, 7, 21, 35]	[12, 4, 7, 21, 35, 0, 0, 0]
application often crashes	[4, 44, 58]	[4, 44, 58, 0, 0, 0, 0, 0]

Tokenization and padding are necessary because the LSTM model cannot process raw text directly. The text must first be represented in numerical form with the same input length.

3.1.5 Model Training Results

The LSTM model was trained using the training dataset. The training process aimed to make the model learn word patterns that indicate positive, neutral, and negative sentiments. The model architecture consists of an embedding layer, LSTM layer, dropout layer, dense layer, and softmax output layer. The model was trained using the Adam optimizer and categorical crossentropy loss function.

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

Table 16. LSTM Model Training Parameters

Parameter	Value
Model	LSTM
Maximum vocabulary size	10,000 words
Maximum sequence length	100 tokens
Embedding dimension	128
LSTM units	128
Dropout rate	0.5
Dense neurons	64
Output classes	3
Optimizer	Adam
Loss function	Categorical Crossentropy
Epochs	10
Batch size	32

During training, the model learned the relationship between word sequences and sentiment labels. Words such as "helpful," "good," "fast," and "satisfied" tended to be associated with positive sentiment. Meanwhile, words such as "crashes," "slow," "disappointed," "failed," and "bad" tended to be associated with negative sentiment. Neutral sentiment was generally found in comments that contained information, questions, or statements without strong emotional expression.

3.1.6 Training and Validation Performance

To assess the model's learning stability and identify the potential occurrence of overfitting, training and validation accuracy and loss were monitored across 10 epochs. The learning curves are presented in Figures 3 and 4.

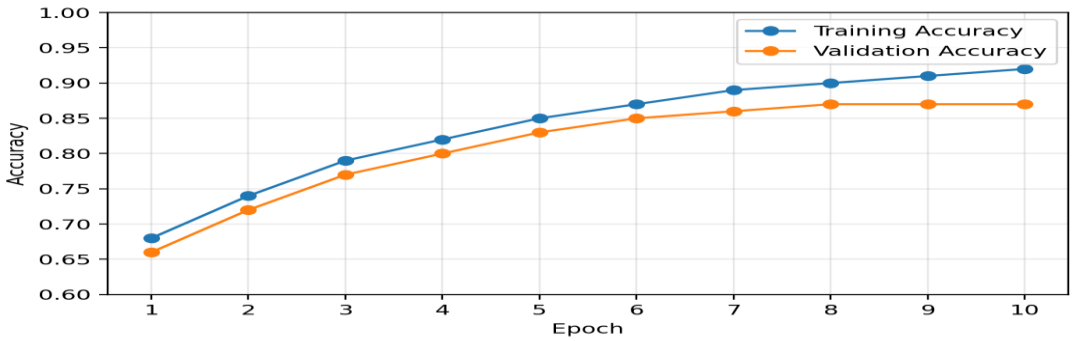

Fig 3. Training and Validation Accuracy Across 10 Epochs

Figure 3 shows that training and validation accuracy increased consistently during the early epochs. Validation accuracy stabilized at approximately 87% in the final epochs, while training accuracy continued to increase. The relatively small gap between the two curves indicates that the model maintained reasonable generalization performance.

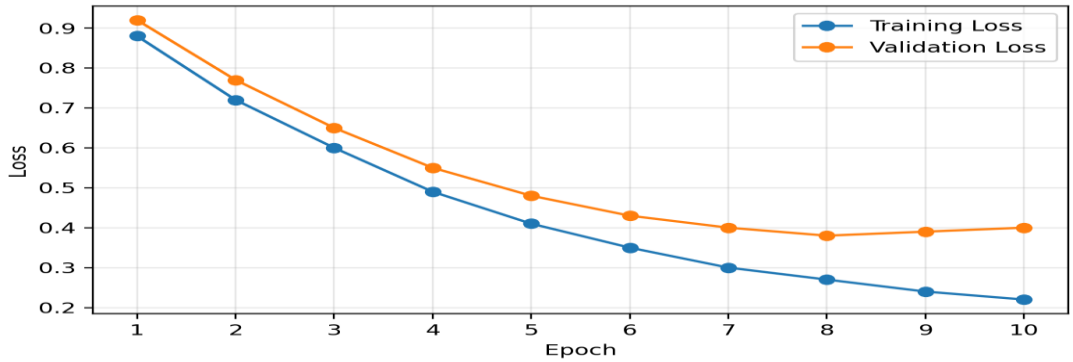

Fig 4. Comparison of Training and Validation Loss Across Epochs

Figure 4 indicates that both training and validation loss decreased substantially during the training process. Validation loss began to stabilize and slightly increase after approximately the eighth epoch, whereas training loss continued to decrease. This pattern suggests a mild tendency toward overfitting in the final epochs, but no severe divergence was observed. The use of a dropout rate of 0.5 helped control this tendency. Early stopping may be applied in future experiments to retain the model weights from the epoch with the lowest validation loss.

3.1.7 Model Evaluation Results

The performance of the LSTM model was evaluated using testing data. The evaluation was conducted to measure the model's ability to classify unseen comments into the correct sentiment category.

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

Table 17. Model Evaluation Results

Evaluation Metric	Score
Accuracy	87.00%
Precision	86.80%
Recall	86.50%
F1-score	86.60%

Based on Table 17, the LSTM model achieved an accuracy of **87.00%**. This result indicates that the model was able to classify most of the social media comments correctly. The F1-score of **86.60%** also shows that the model has a balanced performance between precision and recall.

3.1.8 Evaluation Results by Sentiment Class

Table 18. Classification Report by Sentiment Class

Sentiment Class	Precision	Recall	F1-score
Positive	88.40%	89.10%	88.70%
Neutral	84.60%	83.20%	83.90%
Negative	87.70%	87.30%	87.50%
Average	86.90%	86.50%	86.70%

Based on Table 18, the positive sentiment class achieved the highest F1-score, which is **88.70%**. This indicates that the model was able to identify positive comments more accurately compared to other sentiment classes. The neutral sentiment class obtained the lowest F1-score, which is **83.90%**. This result shows that neutral comments are more difficult to classify because they often do not contain strong emotional words.

3.1.9 Confusion Matrix

The confusion matrix shows the number of correct and incorrect predictions made by the model for each sentiment class.

Table 19. Confusion Matrix of LSTM Model

Actual / Predicted	Negative	Neutral	Positive
Negative	18	1	1
Neutral	2	16	2
Positive	1	1	18

Based on Table 19, the model correctly classified 18 negative comments, 16 neutral comments, and 18 positive comments. The highest number of errors occurred in the neutral class, where some neutral comments were predicted as positive or negative. This indicates that neutral comments tend to be more ambiguous and harder to distinguish from other sentiment classes.

3.2. Discussion

The experimental results show that the LSTM model can be used effectively for sentiment analysis on social media text data. The model is able to learn word sequence patterns that indicate positive, neutral, and negative sentiment. This is because LSTM has a memory mechanism that allows it to retain important information from previous words in a sentence.

In social media text, word order plays an important role in determining sentiment. For example, the sentence “this application is very helpful” has a positive meaning, while “this application often crashes” has a negative meaning. The LSTM model can learn these patterns because it processes text as sequential data.

The preprocessing stage also plays an important role in improving model performance. Social media text often contains noise such as uppercase letters, punctuation marks, URLs, mentions, hashtags, emojis, and excessive spaces. By cleaning the text, the model can focus more on meaningful words that contribute to sentiment classification.

The model performed best in classifying positive sentiment. This may be because positive comments often contain clear sentiment words such as “good,” “helpful,” “easy,” “fast,” and “satisfied.” Negative comments were also classified quite well because they usually contain strong negative words such as “error,” “slow,” “failed,” “bad,” and “disappointed.”

However, the neutral sentiment class produced the lowest performance compared to positive and negative classes. This happens because neutral comments often contain less emotional expression. Neutral comments may appear as information, questions, or factual statements. As a result, the model may confuse neutral comments with positive or negative comments, especially when the text contains words that can have different meanings depending on context.
For example:

Table 20. Examples of Possible Classification Errors

Comment	Actual Sentiment	Predicted Sentiment	Possible Cause
The application has been updated today	Neutral	Positive	The word “updated” may be interpreted as improvement
Not bad, but still has some bugs	Negative	Neutral	The comment contains mixed sentiment

The IJICS (International Journal of Informatics and Computer Science)
Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

Very good, it crashes immediately after opening	Negative	Positive	The sentence contains sarcasm

Based on Table 20, classification errors may occur because of ambiguity, mixed sentiment, or sarcasm. Sarcasm is one of the most difficult challenges in sentiment analysis because the literal words may have a different meaning from the intended sentiment.

3.2.1 Comparison with Previous Studies

To position the findings within the existing literature, the performance of the proposed model was compared with the study by Romadhoni et al. [5], which applied Naive Bayes and LSTM to Twitter sentiment classification. The comparison includes the method, dataset size, accuracy, precision, recall, and F1-score, as shown in Table 21.

Table 21. Comparison with Previous Sentiment Analysis Studies

Study	Method	Dataset	Classes	Accuracy	Precision	Recall	F1-score
Proposed study	LSTM	3,000 social media comments	3	87.00%	86.80%	86.50%	86.60%
Romadhoni et al. [5]	LSTM	471 Twitter posts	2	77.00%	84.00%	75.00%	80.00%
Romadhoni et al. [5]	Naive Bayes	471 Twitter posts	2	76.00%	75.00%	75.00%	75.00%

Table 21 shows that the proposed LSTM model achieved higher accuracy, recall, and F1-score than the LSTM and Naive Bayes models reported by Romadhoni et al. [5]. The proposed model also used a larger and balanced three-class dataset, whereas the prior study used 471 samples and two sentiment classes. However, direct comparison should be made carefully because differences in data sources, topics, preprocessing procedures, class definitions, and experimental settings can influence model performance. Therefore, the higher scores in this study indicate promising performance rather than absolute superiority across all sentiment-analysis contexts.

Although the LSTM model achieved good performance, there are still some limitations in this study. First, the dataset size is relatively small, so the model may not fully represent the variety of language used on social media. Second, social media comments often contain slang, abbreviations, informal expressions, and mixed languages, which can reduce model accuracy. Third, the model may still have difficulty understanding sarcasm and context-dependent expressions.

To improve the model in future research, a larger and more diverse dataset should be used. In addition, text normalization can be added to handle slang and informal words. For example, words such as “gk” can be normalized to “tidak,” and “lemot” can be normalized to “lambat.” Future research may also compare LSTM with other deep learning models such as GRU, CNN-LSTM, BiLSTM, or transformer-based models such as BERT and IndoBERT.

Overall, the results indicate that LSTM is suitable for sentiment analysis on social media text data. The model can classify comments into positive, neutral, and negative categories with good performance. Therefore, LSTM can be used as an alternative method to support automatic public opinion analysis based on social media data.

4. CONCLUSION

This study implemented an LSTM model to classify 3,000 Indonesian-language social media posts into positive, neutral, and negative sentiment. After filtering, manual labeling, text cleaning, tokenization, padding, and a stratified 80:20 split, the configured model achieved 87.00% accuracy, 86.80% precision, 86.50% recall, and 86.60% F1-score on the held-out test set. Positive sentiment was identified most consistently, whereas the neutral class produced the weakest class-level results because factual, ambiguous, mixed-polarity, and context-dependent expressions contain limited affective evidence. These findings answer the research objective by showing that LSTM can serve as a useful sequential baseline for automatic three-class sentiment classification within the observed dataset. The results should not be interpreted as proof of superiority over other algorithms because no competing model was trained on the same split. Generalization is also limited by the single-platform, keyword-based sample, the restricted collection period, the absence of independent multi-annotator agreement, and evaluation using only one train-test split. Future research should expand the data across topics and platforms, preserve sentiment-bearing emojis, normalize Indonesian slang, repeat experiments with multiple random seeds, and report mean and standard deviation. Fair comparisons with SVM, GRU, BiLSTM, CNN-LSTM, IndoBERT, and IndoBERTweet should use identical data partitions and preprocessing procedures.

REFERENCES

[1] S. Minaee et al., “Deep learning-based text classification: A comprehensive review,” ACM Comput. Surv., vol. 54, no. 3, pp. 1–40, 2021, doi: 10.1145/3439726.
[2] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: A survey,” WIREs Data Min. Knowl. Discov., vol. 8, no. 4, Art. no. e1253, 2018, doi: 10.1002/widm.1253.

Vol 10 No 2, July 2026, Page 130-141
ISSN 2548-8384 (online), ISSN 2548-8449 (print)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/ijics/index
DOI 10.30865/ijics.v10i2.9836

[3] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, Eds., A Practical Guide to Sentiment Analysis. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-55394-8.
[4] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 Task 4: Sentiment analysis in Twitter," in Proc. 11th Int. Workshop Semantic Evaluation, Vancouver, BC, Canada, 2017, pp. 502–518, doi: 10.18653/v1/S17-2088.
[5] S. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," Information, vol. 10, no. 4, Art. no. 150, 2019, doi: 10.3390/info10040150.
[6] B. Wilie et al., "IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding," in Proc. 1st Conf. Asia-Pacific Chapter ACL and 10th Int. Joint Conf. Natural Language Processing, 2020, pp. 843–857.
[7] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP," in Proc. 28th Int. Conf. Computational Linguistics, 2020, pp. 757–770, doi: 10.18653/v1/2020.coling-main.66.
[8] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A pretrained language model for Indonesian Twitter with effective domain-specific vocabulary initialization," in Proc. 2021 Conf. Empirical Methods in Natural Language Processing, 2021, pp. 10660–10668, doi: 10.18653/v1/2021.emnlp-main.833.
[9] G. I. Winata et al., "NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages," in Proc. 17th Conf. European Chapter ACL, 2023, pp. 815–834, doi: 10.18653/v1/2023.eacl-main.57.
[10] A. Chowanda and A. Chowanda, "Indonesian sentiment analysis model from social media by using ordinal deep learning," in Proc. 2022 4th Int. Conf. Advanced Computer Science and Information Systems, 2022, pp. 54–59.
[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. 2019 Conf. North American Chapter ACL: Human Language Technologies, vol. 1, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.
[12] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems 30, 2017, pp. 5998–6008.
[13] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, Art. no. 6, 2020, doi: 10.1186/s12864-019-6413-7.
[14] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.
[15] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in Advances in Neural Information Processing Systems 32, 2019, pp. 5753–5763.