OPEN ACCESS 13

ISSN 2356-5462
http://journals.telkomuniversity.ac.id/ijoict/

Intl. Journal on ICT
Vol. 11, No. 1, Jun 2025. pp. 13-25
doi:10.21108/ijoict.v11i1.9176

Analysis of Public Sentiment Regarding the 2024 Jakarta Election on Platform X Using Deep Learning

Moch. Rizki Khaerul Muhaemin [1*], Fitriyani [2], Lazuardy Syahrul Darfiansa [3]

[1,2,3]*School of Computing, Telkom University*
Jl. Telekomunikasi No. 1 Terusan Buah Batu, Bandung, Jawa Barat, Indonesia, 40257

[*] rkkhaerul@student.telkomuniversity.ac.id

Abstract

The 2024 Jakarta Regional Head Election (Pilkada Jakarta) is a critical issue that requires an in-depth understanding of public sentiment. This platform generates complex, unstructured text with informal language and ambiguity, posing challenges alongside the lack of local context-specific datasets and inaccuracies in traditional sentiment analysis models. Analyzing sentiment for the Pilkada is crucial for evaluating public response to policies, aiding political strategy, and improving governance. Current systems struggle with complex data and class imbalance (dominant neutral sentiment), leading to underrepresented information. This study addresses these issues by constructing a sentiment analysis system using four deep learning models: IndoBERT, LSTM, CNN, and GRU. The procedure encompassed data acquisition from X, preprocessing, model training, and assessment based on accuracy, precision, recall, and F1-score. The CNN model achieved the highest accuracy of 83.37%, followed by LSTM at 82.61%, GRU at 82.30%, and IndoBERT at 80.77%. All models achieved the accuracy benchmark of a minimum of 80%, however the neutral class continues to pose a challenge. Research contributions include a deep learning-based sentiment classification system that can be implemented in local political opinion analysis, as well as recommendations for using hybrid models like IndoBERT + CNN for further research.

Keywords: Sentiment Analysis, Deep Learning, Jakarta Election

I. INTRODUCTION

Regional Head Elections (Pilkada) are an important moment in Indonesia's democratic system, especially in Jakarta. Elections not only serve to elect leaders, but also reflect the social, political and economic aspects of society. Understanding people's attitudes towards the 2024 elections is crucial, especially when social media platforms such as X are becoming the main channel for individuals to express their opinions and perspectives [1]. With the increasing use of social media, X offers extensive data on users' unfiltered thoughts on several topics, including the 2024 Jakarta elections, making it an optimal platform for sentiment analysis [2].

This research is important because of the need to understand the process of public opinion in the context of local democracy, particularly in Jakarta. The 2024 Jakarta election is a political event that significantly influences government policies, the growth of the city and the well-being of the people. Analyzing the mood of the public enables an understanding of individual reactions to politicians, programs and policies presented

Received on 02 Jun 2025. Revised on 19 Jun 2025. Accepted and Published on 30 Jun 2025

during election campaigns. This is important because public sentiment can influence voters' choices, both during elections and in assessing the performance of elected officials. In addition, sentiment research can uncover significant community concerns, which stakeholders can evaluate to improve the quality of public policies and services in the future.

Numerous previous studies have attempted to assess sentiment on social media, mainly through the application of deep learning techniques. Mollah developed an LSTM model for sentiment analysis on Twitter, which is adept at understanding emotional subtleties in text [1]. Murthy et al. used LSTM to assess sentiment in text, achieving good accuracy in distinguishing between positive and negative attitudes [4]. In another study, Gandhi et al. used CNN and LSTM to assess sentiment on Twitter, demonstrating the significant potential of these methodologies to understand individual sentiment [5]. Research conducted by Andriawan et al. [6] used the Gated Recurrent Unit (GRU) algorithm along with Natural Language Processing to classify sentiment about political parties on Twitter, illustrating the effectiveness of GRU in understanding political issues. At the same time, Geni et al. conducted sentiment analysis on tweets before the 2024 General Election in Indonesia using the IndoBERT language model, yielding substantial insights into popular perceptions and assisting the government in election preparation [7].

This study seeks to evaluate the mood of the Jakarta populace concerning the 2024 Election through the application of four Deep Learning models: IndoBERT, Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Gated Recurrent Unit (GRU). The four models were selected for their capacity to identify long-term correlations in textual data, essential for sentiment analysis. This research will enhance data preprocessing and choose pertinent elements to augment the model's accuracy.

II. LITERATURE REVIEW

Sentiment analysis has emerged as a prolific domain of inquiry within Natural Language Processing (NLP), particularly due to the vast array of user opinions accessible online. Research on sentiment analysis has spanned a variety of fields, including economics, politics, and medicine. IndoBERT is a pre-trained transformer-based language model designed to understand Indonesian. It can generate vector representations that comprehensively capture the semantics of tokens, phrases, sentences, or text. IndoBERT has demonstrated its efficacy in sentiment analysis on certain topics, such as electric vehicles. Research by Merdiansah et al [8] utilized IndoBERT to analyze the sentiment of users of platform X (formerly Twitter) in Indonesia regarding electric vehicles. The research assessed the efficacy of the IndoBERT model by contrasting models trained with and without IndoNLU data, employing conventional assessment measures like accuracy, precision, recall, and F1-score. The results demonstrate IndoBERT's capability in identifying and understanding user sentiment, providing important insights into the public's views on green technology.

Long Short-Term Memory (LSTM) is an enhancement of Recurrent Neural Networks (RNN) that aims to solve the shortcomings of conventional RNNs, such as the problem of vanishing and exploding gradients. LSTM excels at managing long-term dependencies in sequential data, including text. LSTM has been widely used to analyze sentiment on social media, especially on platforms such as Twitter. Research conducted by Nistor et al [9] demonstrated that LSTM may attain an accuracy rate of up to 80.74% in binary sentiment categorization. This study employs an attention mechanism to enhance model performance by directing the network's focus toward the most pertinent segments of the text. The dataset used was also very large, consisting of 1.5 million labeled tweets, which helped optimize the model results. This research highlights the importance of integrating attention mechanisms to capture sentiment information more effectively [9].

Convolutional Neural Networks (CNN) are widely used in various applications, such as sentiment analysis. CNN were initially used primarily for visual data processing; however, lately they have demonstrated their efficacy with text data as well. Diwan and Tembhurne [10] stated that the capacity of CNN to extract spatial and contextual information from visual and textual inputs makes it an indispensable instrument in sentiment analysis. In sentiment analysis, CNN can extract significant characteristics from text used for sentiment categorization [11]. A primary advantage of CNN is its capacity to autonomously learn feature hierarchies from

data. On text data, CNN can identify patterns of words or phrases relevant for sentiment, without requiring extensive manual feature engineering [10].

The Gated Recurrent Unit (GRU) is an algorithm utilized in Recurrent Neural Networks (RNN) that enables each recurrent unit to adaptively capture information at varying temporal scales. An artificial neural network (RNN) is a type of neural network that uses internal memory to analyze inputs. Therefore, GRU overcomes the RNN's constraint on long-term memory retention. GRU consists of two gates: the update gate and the reset gate. The update gate sets the memory retention rate, while the reset gate determines the memory erasure rate. GRU has been used extensively in sentiment analysis applications. Aakash et al [19] used GRU for sentiment analysis of product reviews sourced from URLs, showing that the GRU model consistently achieved high scores across various performance metrics. In the realm of tweet sentiment analysis.

This comparative research is essential to understand the advantages and disadvantages of various models in the unique context of data and sentiment analysis tasks, especially when dealing with problems such as class imbalance (multi-class) classification. Overall, the purpose of this research is to build a public opinion sentiment classification system related to the 2024 Jakarta Election on Platform X using four Deep Learning models. This study aims to evaluate and compare the efficacy of four models in classification and performance analysis based on accuracy, precision, recall, as well as F1-score.

III. RESEARCH METHOD

The design of the system to be built is represented in the form of a flowchart which can be seen in Fig. 1.

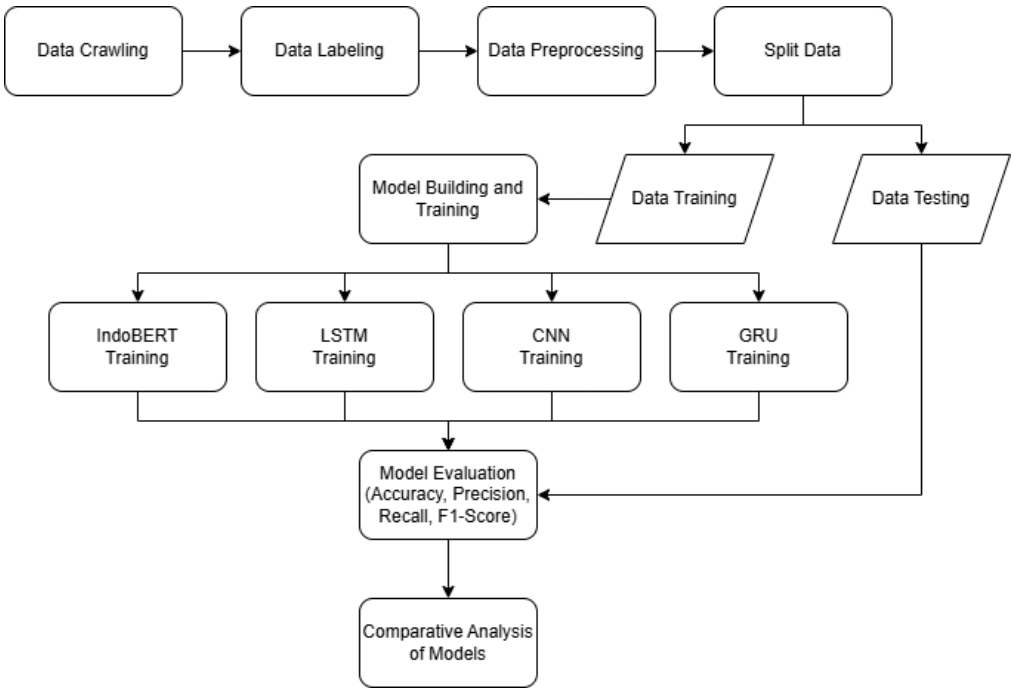

Fig. 1. Sentiment Analysis Flowchart Design

A. Data Crawling and Labeling

The dataset for this research was acquired via a crawling procedure from Platform X via the platform's Twitter authentication token, concentrating on tweets pertinent to the 2024 Jakarta elections. Data collection was conducted over the period July 1, 2024 to November 30, 2024, resulting in a dataset of information including username, date, location, full text of tweets, as well as various other metadata relevant for sentiment analysis. The data collection process was designed to ensure a balanced representation of Jakartans' various perspectives

on the 2024 elections, taking into account criteria such as content relevance, use of election-related hashtags, and significant user interaction. This dataset contains full_text columns with a total of 22,462 tweets.

The data labeling process in this research is a crucial stage to prepare the tweet dataset. Each tweet related to the 2024 Jakarta election was categorized into three sentiments: positive, negative, and neutral as shown in Table I. This labeling was done automatically using IndoBERT [17] to minimize errors, by scoring each tweet based on the emotional charge, intensity of expression, and context of user statements from platform X (formerly Twitter). After automatic labeling, a validation stage was conducted to ensure consistency and accuracy, where adjustments were made if inconsistencies were found as shown in Table II. This research does not involve manual labeling by multiple people, but rather relies on an automated approach with validation. This validation was performed by the researchers to ensure the quality of the dataset for subsequent model training [17].

TABLE I
RESULT OF DATA CRAWLING AND LABELING

Label	Amount	Ratio %
Positive	1,680	7.48
Negative	4,992	22.22
Neutral	15,785	70.30
Total	22,462	100

TABLE II
LABELING EXAMPLE

Sentiment	Label	Explanation
Jakarta butuh perubahan nyata dan mas pram amp bang doel adalah jawabannya pilkada satu putaran fix ini mah satusatunya jagoan gua yang siap membawa banyak kemajuan di jakarta gaspol	Positive	Sentences contain positive phrases such as “perubahan nyata”, “siap membawa banyak kemajuan”, and “jagoan gua” that show support and good wishes for the candidate.
Kubu jokowi kenapa dah ribet banget sama pilkada Jakarta katanya mau pindah ikn gaje	Negative	Sentences contain negative words such as “ribet banget” and “gaje” (unclear) that indicate frustration or disagreement.
Jika nanti pilkada Jakarta berlangsung putaran pasti ada kubu yang tidak terima dengan hasil itu	Neutral	This sentence is more of a prediction or factual statement about possible reactions to the election results (“tidak terima dengan hasil itu”) without showing any positive or negative emotions from the writer. This indicates objectivity or impartiality.

B. Preprocessing Data

The data preprocessing step is an important stage in preparing the dataset for sentiment analysis using Deep Learning models. This stage begins with case folding to clean the text from irrelevant elements, such as URLs, mentions (@username), hashtags (#), special characters, emoticons and unnecessary punctuation. After the cleaning stage, text normalization is performed by converting all letters to lowercase and handling informal words or abbreviations that are often used in social media. For this process, a special normalization dictionary designed for informal Indonesian is used. This step ensures that variations in word writing can be uniformed. The final stage of pre-processing includes tokenization, the removal of stop words. Tokenization seeks to decompose text into smaller pieces (tokens), whereas stop words removal eliminates common terms that lack substantial relevance for sentiment analysis as shown in Table III.

1) *Data Cleansing:* Data cleansing is a crucial step in text pre-processing. Garg and Sharma [15] state that data cleaning involves the removal of noise such as special characters, numbers, punctuation marks, excess spaces, removal of duplicate data, and irrelevant single characters. This procedure seeks to enhance data quality

by eliminating components that may disrupt sentiment analysis, allowing the model to concentrate on pertinent information.

2) *Case Folding:* Letter folding denotes the transformation of all characters in the text into lowercase letters. According to Alzami et al. [13], case folding helps homogenize text by converting all words to lowercase, which reduces unnecessary word variations and improves data consistency. It is important that sentiment analysis algorithms treat the same words in the same way, regardless of capitalization in the original text.

3) *Tokenizing:* Tokenization is the process of breaking down text into smaller units, referred to as tokens, which can be words, phrases, or symbols. Duong and Nguyen-Thi [14] explain that tokenization is an important step in text pre-processing as it allows analysis to be performed on individually meaningful language units.

4) *Word Normalization:* Word normalization refers to the procedure of converting words into their standardized form. Garg and Sharma [15] explained that this process includes the correction of nonstandard words, abbreviations, and spelling variations. The purpose of normalization is to reduce the variation of words that are different but mean the same thing, thus improving data consistency and analysis effectiveness.

5) *Stopword Removal:* Stopword elimination is the procedure of eliminating ubiquitous words that regularly occur yet lack substantial meaning in the context of sentiment analysis. According to Rosid et al. [16], the purpose of stopword removal is to reduce data dimensionality and improve processing efficiency. In this study, the removal is performed using the Sastrawi library for Bahasa Indonesia, which initializes and applies the stopword remover to the normalized text.

6) *Stemming:* Stemming is the process of reducing words to their base form. Rosid et al. [16] explained that stemming unifies words that have the same root but different forms, such as “running”, “running-running”, and ‘runner’, into “running”. In this research, stemming is performed using the algorithm from Sastrawi after the stopword removal process to simplify the morphological variation of words and maintain the core meaning of the text.

TABLE III
RESULT OF DATASET AFTER PREPROCESSING

Label	Data before preprocessing	Data after preprocessing
Positive	Jakarta butuh perubahan nyata dan mas pram amp bang doel adalah jawabannya pilkada satu putaran fix ini mah satusatunya jagoan gua yang siap membawa banyak kemajuan di jakarta gaspol	jakarta butuh ubah nyata mas pram amp abang doel jawab pilkada satu putar fix mah satusatunya jago gua siap bawa banyak maju jakarta gaspol
Negative	Kubu jokowi kenapa dah ribet banget sama pilkada Jakarta katanya mau pindah ikn gaje	kubu jokowi sudah ribet banget sama pilkada jakarta kata mau pindah ikn tidak jelas
Neutral	Jika nanti pilkada Jakarta berlangsung putaran pasti ada kubu yang tidak terima dengan hasil itu	nanti pilkada jakarta langsung putar ada kubu tidak terima hasil

C. Split Data

The dataset is divided into two portions for optimal model training and assessment. 80% of the dataset, comprising 17,969 tweets, is utilized to train the model to identify linguistic patterns and political contexts in tweets related to the Jakarta election. Twenty percent of the data, comprising 4,493 tweets, was designated for testing to assess the model's capacity to predict sentiment on novel data. The 80:20 division was established to

avert overfitting and offer an accurate representation of the model's efficacy in real-world scenarios, while guaranteeing a sufficient quantity of samples in both segments.

D. Model for Sentiment Analysis

This research employed many model architectures that were meticulously built to attain best outcomes. The parameters utilized for each model, specifically IndoBERT, LSTM, CNN, and GRU, are thoroughly delineated below. The consistency in setting certain parameters across these models demonstrates a standardized methodology to enable fair and valid performance comparisons.

1) IndoBERT Architecture: The IndoBERT model employed in this study is based on the BertForSequenceClassification framework, a pre-trained transformer model fine-tuned for Indonesian language processing. The foundational BertModel comprises three main components: an embeddings layer, an encoder, and a pooler. The embeddings layer is responsible for converting input tokens into dense vector representations, consisting of word_embeddings (vocabulary size 50,000, dimension 768), position_embeddings (maximum sequence length 512, dimension 768), and token_type_embeddings (2 types, dimension 768). This is followed by a LayerNorm for stabilization and a Dropout layer (p=0.2). The encoder is a stacked architecture of 12 BertLayer modules. Each BertLayer includes a BertAttention mechanism with BertSdpaSelfAttention for computing self-attention weights (query, key, and value linear layers each with 768 output features, and a dropout of 0.2). The attention output is then processed by a BertSelfOutput (dense layer with 768 features, LayerNorm, and dropout of 0.2). An intermediate BertIntermediate layer with a dense output of 3072 features and GELUActivation provides non-linearity, which is then fed into a BertOutput (dense layer with 768 features, LayerNorm, and dropout of 0.2). The pooler applies a Linear layer (input 768, output 768) followed by a Tanh activation to derive a fixed-size representation of the input sequence. Finally, for sentiment classification, a dropout layer (p=0.2) is applied before a classifier Linear layer, which transforms the 768 features into 3 output classes, corresponding to positive, negative, and neutral sentiments. The total number of parameters for this IndoBERT architecture is 124,443,651.
2) LSTM (Long Short-Term Memory) Architecture: The LSTM model is designed to process sequential data and effectively capture long-term dependencies, overcoming challenges faced by traditional Recurrent Neural Networks. The architecture begins with an input_layer configured to accept sequences of up to 40 tokens. An embedding layer then converts these discrete tokens into continuous vector representations, with an output dimension of 128, contributing 2,560,000 parameters. To enhance training stability and performance, a batch_normalization layer with 512 parameters is applied to the embedded input. Following this, a dropout layer with a rate of 0.2 is included to mitigate overfitting during training. The core of the model is an LSTM layer, which processes the sequential embedded data and outputs a vector of 256 dimensions, encompassing 394,240 parameters. A subsequent dropout_1 layer (p=0.2) is applied to the output of the LSTM layer. The final stage involves a dense layer with 3 output units and 771 parameters, responsible for producing the classification probabilities for positive, negative, and neutral sentiments. The total number of parameters for the LSTM model is 2,955,523, with 2,955,267 being trainable.
3) CNN (Convolutional Neural Network) Architecture: While commonly used in computer vision, CNNs are highly effective in natural language processing tasks like sentiment analysis by identifying local patterns within text. The CNN model architecture in this research starts with an input_layer configured for sequences of 40 tokens. An embedding layer maps these tokens to 128-dimensional dense vectors, accounting for 2,560,000 parameters. This is followed by a batch_normalization layer (512 parameters) and a dropout layer (p=0.2). A distinguishing feature of this CNN architecture is the use of three parallel Conv1D layers, enabling the capture of various n-gram features:
   - The first Conv1D layer generates an output shape of (None, 38, 128) with 49,280 parameters.
   - The second Conv1D layer produces an output shape of (None, 37, 128) with 65,664 parameters.

- The third Conv1D layer results in an output shape of (None, 36, 128) with 82,048 parameters. Each Conv1D layer's output is then fed into a GlobalMaxPooling1D layer, which extracts the most significant feature from each filter, yielding a (None, 128) output shape. The outputs from these three global max-pooling layers are then concatenated into a single feature vector of 384 dimensions. A final dropout_1 layer (p=0.2) is applied before a dense layer, which takes the 384 concatenated features and outputs 3 units with 1,155 parameters for the sentiment classification. The CNN model has a total of 2,758,659 parameters, with 2,758,403 being trainable.

4) GRU (Gated Recurrent Unit) Architecture: The GRU model, a lighter yet effective variant of Recurrent Neural Networks, is employed to handle sequential text data with improved memory retention compared to basic RNNs. The model's input begins with an input_layer accepting sequences of 40 tokens. These tokens are transformed into 128-dimensional dense vectors by an embedding layer, contributing 2,560,000 parameters. A batch_normalization layer (512 parameters) and a dropout layer (p=0.2) are then applied to the embedded sequence. The core recurrent component is a GRU layer, which processes the sequence and produces an output vector of 256 dimensions, involving 296,448 parameters. The GRU layer utilizes its update and reset gates to efficiently manage information flow through time. Following the GRU layer, another dropout_1 layer (p=0.2) is included. The final dense layer, with 3 output units and 771 parameters, is responsible for predicting the sentiment class (positive, negative, or neutral). This GRU architecture comprises a total of 2,857,731 parameters, with 2,857,475 parameters being trainable.

All models used dropout (0.2) and L2 regularization (0.001) to prevent overfitting, and were trained with 10 epochs, as well as an Adam optimizer with a low learning rate (0.0001) for training stability. Other parameters that were also uniformed for experimental consistency include data split (80:20 data split for training and testing data) and batch size (Batch Size 32).

E. Evaluation

Model evaluation involves juxtaposing the model's predictions with the actual sentiment labels in the test dataset. The confusion matrix is a widely utilized evaluation technique that offers a detailed assessment of a classification model's performance by juxtaposing expected and actual labels. Confusion matrix is very useful to understand how the model classifies each sentiment class (positive, negative, and neutral) and identify possible misclassifications [18].

The confusion matrix is a tabular representation that specifies the number of correct and incorrect predictions for each category. For a three-class classification (positive, negative, neutral), the confusion matrix will be a 3×3 matrix with rows representing actual labels and columns representing model predictions. The main diagonal elements indicate the number of correct predictions, while the off-diagonal elements indicate misclassification [18].

TABLE IV
CONFUSION MATRIX

Confusion Matrix	Positive Prediction	Negative Prediction	Neutral Prediction
Actual Positive	TP	FN	FN
Actual Negative	FP	TN	FN
Actual Neutral	FP	FN	TN

As shown in Table IV, the system performance table includes four key terms: True Positive (TP), which indicates a correct positive prediction; False Positive (FP), which indicates an incorrect positive prediction; False Negative (FN), which indicates an incorrect negative prediction for the positive class; and True Negative (TN), which indicates a correct negative prediction. These four terms are used to calculate performance evaluation metrics.

1) Accuracy: Accuracy measures the percentage of correct predictions overall.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

2) Precision: Precision evaluates the accuracy of a model's positive predictions by comparing real positive predictions to the overall number of positive predictions.

$$precision = \frac{TP}{TP + FP} \tag{2}$$

3) Recall: Recall evaluates the model's ability to recognize all positive instances in the dataset.

$$recall = \frac{TP}{TP + FN} \tag{3}$$

4) F1-Score: The F1-score represents the harmonic mean of precision and recall, beneficial in situations with imbalanced classes.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

IV. Results And Discussion

This research evaluated four constructed models using the dataset. According to the test findings of the four models: IndoBert, LSTM, CNN, and GRU. This is a comprehensive evaluation of the efficacy of four deep learning models (IndoBERT, LSTM, CNN, and GRU) in sentiment classification of the 2024 Jakarta Pilkada on Platform X. The analysis includes a comparison of architectural parameters, evaluation metrics, and interpretation of test results to understand the advantages and disadvantages of each model [12].

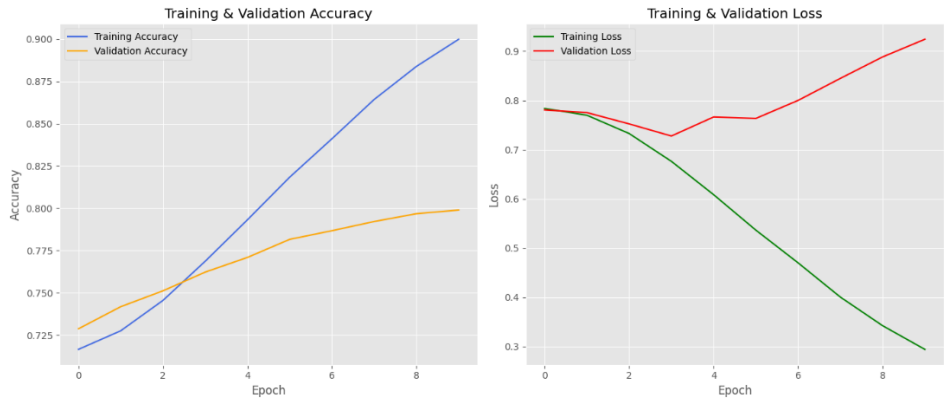

Fig. 2. Visualization of Training & Validation of IndoBERT model

The initial scenario involves evaluating the Indobert Model, with the dataset partitioned into 80% for training and 20% for testing. As shown in Fig. 2, the IndoBert Model assessed in this study demonstrates commendable efficacy in comprehending the sentiment dynamics of Jakarta's populace concerning the 2024 Pilkada. The visualization of training and model validation indicates that the model effectively assimilates data patterns, despite minor changes during the initial epochs. The accuracy graph shows a consistent increase until it reaches around 80% at the 10th epoch, while the loss function drops significantly, indicating that the model has succeeded in optimizing parameters to reduce classification errors. The stable accuracy at the final epoch also shows that the model does not experience excessive overfitting, despite the high complexity of social media text data.

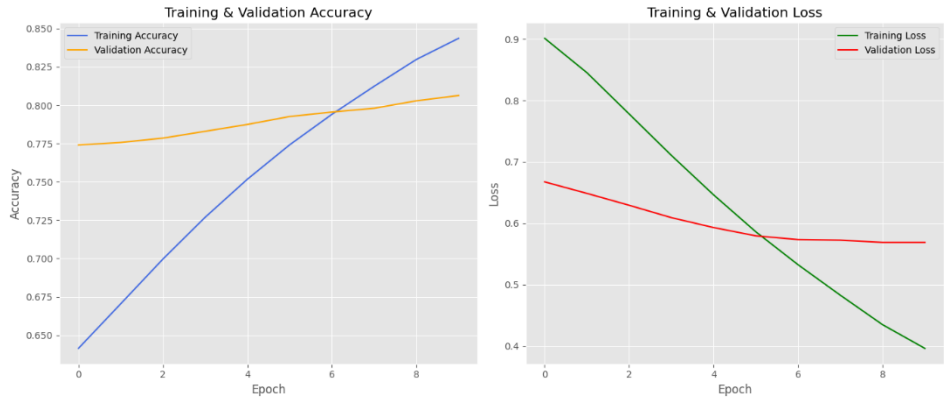

Fig. 3. Visualization of Training & Validation of LSTM model

The second scenario entails evaluating the LSTM model with same or analogous parameters, but the dataset is divided into two segments: 80% for training and 20% for testing. Upon developing the LSTM model, the concluding process entails testing and assessing the model. During the LSTM model's testing phase, the training and validation visualizations demonstrate that the model proficiently discerns sentiment patterns in textual input. As shown in Fig. 3, graphs illustrating the progression of accuracy and loss throughout the training process indicate that the model can learn proficiently from the available data. The training accuracy increases consistently as the epochs increase, reaching a high value at the 10th epoch. Similarly, the validation accuracy shows a positive trend and stabilizes at a value that indicates good generalization of the model. The loss function for both training and validation markedly diminished as the epochs advanced, signifying that the model successfully optimized the parameters to reduce sentiment misclassification.

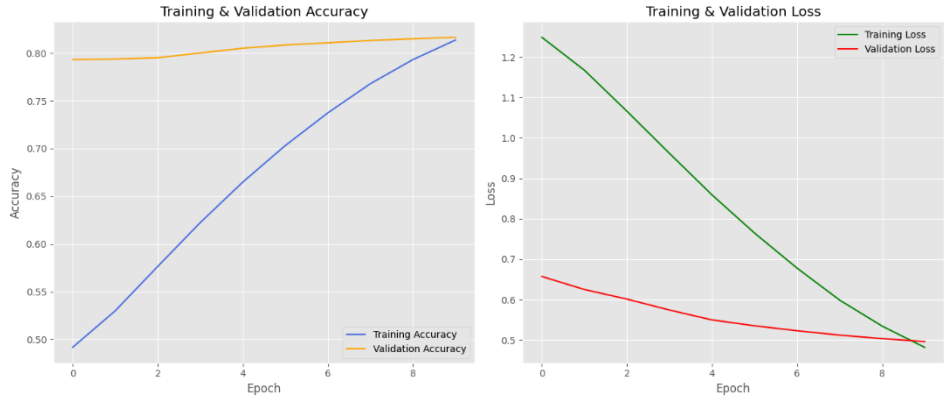

Fig. 4. Visualization of Training & Validation of CNN model

The third case entails evaluating the CNN model with identical parameters. The dataset is divided into two segments: 80% designated for training and 20% for testing. Upon completion of the CNN model construction, the final phase involves testing and evaluating the model. The implemented CNN model demonstrates commendable performance; nonetheless, certain features require enhancement to augment accuracy and consistency in sentiment classification as shown in Fig. 4. The training and validation visualization of the CNN model's accuracy and loss indicates that the model learns effectively from the data. The training accuracy steadily rises with the progression of epochs, attaining a peak at the 10th epoch. The validation accuracy exhibits an upward trend and stabilizes at a level indicative of the model's effective generalization. The loss function for both training and validation markedly declined as the epochs advanced, signifying that the model effectively optimized the parameters to reduce sentiment misclassification.

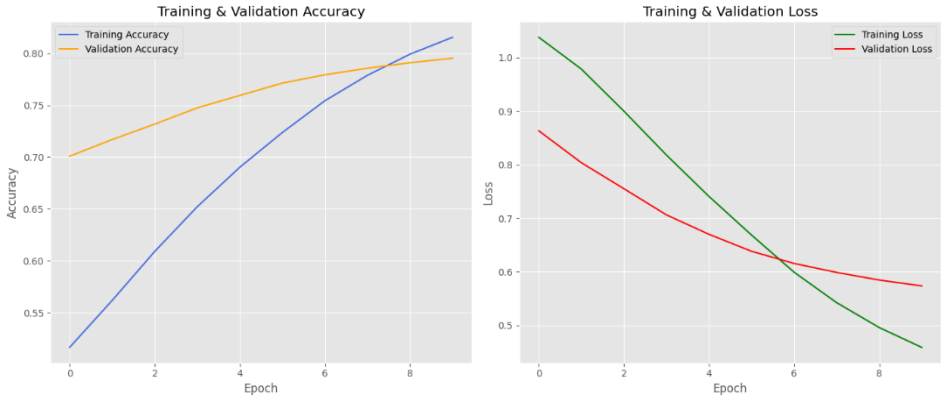

Fig. 5. Visualization of Training & Validation of GRU model

The last scenario entails evaluating and testing the GRU model. The dataset is divided into two segments: 80% designated for training and 20% for testing. Following the training and validation phase, the GRU model exhibits notable effectiveness in categorizing sentiment related to the 2024 Jakarta Pilkada. The training and validation visualizations for accuracy and loss indicate that the GRU model effectively learns from the employed dataset. Fig. 5 demonstrates that the GRU model proficiently acquires knowledge from the training data during both the training and validation phases. The accuracy graph demonstrates steady enhancement until the 10th epoch, with training accuracy achieving almost 90% and validation accuracy stabilizing between 82% and 83%. The reduction in loss for both training and validation datasets suggests that the model does not exhibit substantial overfitting, despite a minor fluctuation in the last epoch. This confirms that the GRU architecture with dropout and normalization layers successfully stabilizes the learning process, despite the high complexity of social media text data.

TABLE V
RESULT OF ACCURACY MODELS

Model	Accuracy
IndoBERT	80.77%
LSTM	82.61%
CNN	**83.37%**
GRU	82.30%

TABLE VI
RESULT OF MODEL PERFORMANCE FOR NEGATIVE CLASSES

Model	Precision	Recall	F1-Score
IndoBERT	93.12%	85.24%	89.01%
LSTM	95.69%	84.91%	89.97%
CNN	94.29%	**87.01%**	**90.51%**
GRU	**95.94%**	84.47%	89.78%

TABLE VII
RESULT OF MODEL PERFORMANCE FOR POSITIVE CLASSES

Model	Precision	Recall	F1-Score
IndoBERT	65.25%	75.45%	69.98%
LSTM	71.05%	81.98%	75.73%
CNN	70.69%	79.90%	75.01%
GRU	**71.90%**	**82.45%**	**76.35%**

TABLE VIII
RESULT OF MODEL PERFORMANCE FOR NEUTRAL CLASSES

Model	Precision	Recall	F1-Score
IndoBERT	40.94%	54.46%	46.74%
LSTM	40.77%	**65.98%**	50,40%
CNN	**45.41%**	60.06%	**51.72%**
GRU	38.34%	62.72%	47.59%

As shown in Table 5, metrics analysis showed that CNN achieved the highest accuracy of 83.37%, slightly ahead of LSTM (82.61%) and GRU (82.30%) thanks to its ability to extract local features with different kernels that improve classification accuracy. On the negative precision aspect as shown in Table VI, GRU recorded the highest value of 95.94%, indicating minimal errors in classifying negative sentiments as other classes, while on the positive recall as shown in Table VII, GRU also excelled with a value of 82.45% as it was able to capture more true positive samples. However, major weaknesses were seen in all models in classifying neutral sentiments as shown in Table VIII, where IndoBERT had the lowest precision of 40.94%, which is thought to be due to data imbalance (with the dominant neutral class reaching 15,785 samples) as well as the presence of ambiguous neutral expressions.

The analysis of the results, grounded in the model architecture, indicates that IndoBERT excelled in the negative class with a precision of 93.12%, attributable to its pretrained capacity to comprehend the context of pivotal terms like “korupsi” and “gagal.” However, its performance in the neutral class was subpar, owing to insufficient training data pertinent to Jakarta politics. LSTM achieved a high recall in the positive class (81.98%) due to its long-term memory capability in recognizing recurring positive patterns like “semangat” or “dukung”; CNN attained the highest accuracy of 83.37% by effectively identifying critical phrases such as “pilkada adil” or “janji palsu”; whereas GRU recorded the highest positive F1-score of 76.35% as its reset gate efficiently organized pertinent information from lengthy texts, although it was less effective with unbalanced data.

V. CONCLUSION

This research effectively achieves the goal of creating a dependable sentiment categorization system, contributing academically to the use of deep learning technology for public opinion analysis within the realm of local politics. The CNN model had the highest performance with an accuracy of 83.37%, succeeded by LSTM at 82.61%, GRU at 82.30%, and IndoBERT at 80.77%. Each model has specific advantages: GRU excels in negative class precision (95.94%), LSTM has the highest recall for positive class (81.98%). However, all models face challenges in distinguishing neutral sentiment, which is due to data imbalance (neutral class dominance of 15,785 samples) and context ambiguity of neutral expressions. Nevertheless, the limitations in handling imbalanced data and the complexity of political texts suggest the need for further research, such as a combination of ensemble learning (e.g. IndoBERT + CNN) or Jakarta issue-specific data augmentation. These results also provide practical recommendations: CNN is suitable for real-time systems with high accuracy, GRU is ideal for negative sentiment detection, while LSTM is relevant for long context analysis.

To provide a broader context to these findings, a comparison is made with previous relevant research. Research by Nistor et al. [9] showed that their LSTM model can achieve an accuracy rate of up to 80.74% in binary sentiment categorization. The study used a very large dataset of 1.5 million labeled tweets, and applied an attention mechanism to improve the model's performance. In the current study, the developed LSTM model

achieved an accuracy of 82.61%. This direct comparison shows that the LSTM model in this study shows higher performance compared to Nistor et al.'s LSTM model, although the sentiment classification task in this study is more complex. The Nistor et al. study focused on binary classification (two classes), while this study performed multi-class classification (positive, negative, neutral). Multi-class classification is inherently more challenging as it increases the number of possible misclassifications and requires finer distinctions between categories.

DATA AND COMPUTER PROGRAM AVAILABILITY

Data and program used in this paper can be accessed on the Telkom University Dataverse, available at the following site: https://doi.org/10.34820/FK2/FEIT6A

ACKNOWLEDGMENT

The author would like to thank the lecturers who have guided the author.

REFERENCES

[1] R. Vindua and A. U. Zailani, "Analisis Sentimen Pemilu Indonesia Tahun 2024 Dari Media Sosial Twitter Menggunakan Python", *JURIKOM (Jurnal Riset Komputer)*, 10(2), 479-487. https://doi.org/10.30865/jurikom.v10i2.5945, 2023.

[2] D. A. Firdlous and R. Andrian, "Analisis Sentimen Publik Twitter terhadap Pemilu 2024 menggunakan Model Long Short Term Memory", *SISTEMASI: Jurnal Sistem Informasi*, 12(1), 52-60, https://doi.org/10.32520/stmsi.v12i1.2145, 2023.

[3] M. P. Mollah, "An LSTM model for Twitter Sentiment Analysis", https://doi.org/10.48550/arXiv.2212.01791, 2022.

[4] G. S. N. Murthy, S. R. Allu, B. Andhavarapu, M. Bagadi, and M. Belusonti, "Text based Sentiment Analysis using LSTM", *International Journal of Engineering Research & Technology (IJERT)*, 9(05), 299-303, https://doi.org/10.17577/IJERTV9IS050290, 2020.

[5] U. D. Gandhi, P. M. Kumar, G. C. Babu, and G. Karthick, "Sentiment Analysis on Twitter Data by Using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM)", *Wireless Personal Communications*, https://doi.org/10.1007/s11277-021-08580-3, 2021.

[6] A. R. Andriawan, Mustakim and R. Novita, "Sentiment Analysis Classification Of Political Parties On Twitter Using Gated Recurrent Unit Algorithm And Natural Language Processing", *JITE (Journal of Informatics and Telecommunication Engineering)*, 7(2), 514-522, https://doi.org/10.31289/jite.v7i2.10709, 2024.

[7] L. Geni, E. Yulianti, and D.I. Sensuse, "Sentiment Analysis of Tweets Before the 2024 Elections in Indonesia Using IndoBERT Language Models", *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, 9(3), 746-757. https://doi.org/10.26555/jiteki.v9i3.26490, 2023.

[8] R. Merdiansah, Siska, and A.A. Ridha, "Analisis Sentimen Pengguna X Indonesia Terkait Kendaraan Listrik Menggunakan IndoBERT", *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, 7(1), 221-228, https://doi.org/10.55338/jikomsi.v7i1.2895, 2024.

[9] S.C. Nistor, M. Moca, D. Moldovan, D. B. Oprean, and R.L. Nistor, "Building a Twitter Sentiment Analysis System with Recurrent Neural Networks", *Sensors*, 21(7), 2266, https://doi.org/10.3390/s21072266, 2021.

[10] T. Diwan, and J. V. Tembhurne, "Sentiment analysis: a convolutional neural networks perspective", *Multimedia Tools and Applications*, 81, 44405-44429, https://doi.org/10.1007/s11042-021-11759-2, 2022.

[11] A.W. Subagio, A. P. Sari, and A. N. Sihananto, "Klasifikasi Lexicon-Based Sentiment Analysis Tragedi Kanjuruhan pada Twitter Menggunakan Algoritma Convolutional Neural Network", *Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer (JUISIK)*, 4(1), 166-177, https://doi.org/10.55606/juisik.v4i1.759, 2024.

[12] A. Agarwal, P. Dey, and S. Kumar, "Sentiment Analysis using Modified GRU", In *2022 Fourteenth International Conference on Contemporary Computing IC3-2022* (pp. 356-361), ACM, https://doi.org/10.1145/3549206.3549270, 2022.

[13] F. Alzami, E.D. Udayanti, D. P. Prabowo, and R. A. Megantara, "Document preprocessing with TF-IDF to improve the polarity classification performance of unstructured sentiment analysis", *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 5(3), 235-242, https://doi.org/10.22219/kinetik.v5i3.1066, 2020.

[14] H. T. Duong, and T. A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis", *Computational Social Networks*, 8(1), Article 1, https://doi.org/10.1186/s40649-020-00080-x, 2021.

[15] N. Garg, and K. Sharma, "Text pre-processing of multilingual for sentiment analysis based on social network data", *International Journal of Electrical and Computer Engineering (IJECE)*, 12(1), 776-784, https://doi.org/10.11591/ijece.v12i1.pp776-784, 2022.

[16] M. A. Rosid, A. S. Fitrani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, "Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi", In *IOP Conference Series: Materials Science and Engineering* (Vol. 874, p. 012017), IOP Publishing, https://doi.org/10.1088/1757-899X/874/1/012017, 2020.

[17] Tarwoto, R. Nugroho, N. Azka, and W. S. R. Graha, "Analisis Sentimen Ulasan Aplikasi Mobile JKN di Google PlayStore Menggunakan IndoBERT", *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 9(2), 495-505, https://doi.org/10.35870/jtik.v9i2.3340, 2025.

[18] Yulia Ery Kurniawati, "What is Confusion Matrix?", *School of Information Systems*, https://sis.binus.ac.id/2024/10/31/what-is-confusion-matrix/ (accessed May 10, 2025).

[19] S. Aakash, S. Gupta, and A. Noliya, "URL-Based Sentiment Analysis of Product Reviews Using LSTM and GRU", In *International Conference on Machine Learning and Data Engineering (ICMLDE 2023)*, Procedia Computer Science, 235, 1814–1823, https://doi.org/10.1016/j.procs.2024.04.172, 2024.