Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Twitter Sentiment Analysis to Assess Public Opinion on Jokowi's Performance Over Two Periods using the Recurrent Neural Network (RNN) Method

Sagah Ageng Penggalih[1]*, Siti Mujilahwati[2], Azza Abidatin Bettaliyah[3]

[1,2,3] Informatics Engineering Study Program, Faculty of Science and Technology, Islamic University of Lamongan, East Java, Indonesia

galihbockrad@gmail.com[1], moedjee@unisla.ac.id[2], azzabettaliyah@unisla.ac.id[3]*

Abstract

This study aims to analyze public sentiment toward President Joko Widodo's performance over his two terms in office by utilizing data from the social media platform Twitter. Twitter was chosen due to its real-time nature and popularity among users for expressing opinions openly. The method employed is a Recurrent Neural Network (RNN) with a Long Short-Term Memory (LSTM) architecture, which is well-suited for processing sequential text data. The research process involves several preprocessing stages, including text cleaning, normalization, tokenization, stopword removal, and stemming, to prepare the textual data for model input. The dataset consists of 1,006 Indonesian-language tweets categorized into three sentiment classes: positive, negative, and neutral. After simplifying the classification into two classes, the model achieved an Accuracy of 63.33%. Evaluation using a Confusion matrix and classification report indicates that the model performs better in identifying positive sentiment, though it still struggles with misclassifying negative sentiment. The results demonstrate that the RNN method is fairly effective for social media-based sentiment analysis. It is expected that this study can serve as a reference for government institutions or policy researchers to understand public perceptions and support more data-driven and responsive policymaking processes.

***Keywords**: Sentiment Analysis, Recurrent Neural Network, Jokowi*

1. Introduction

The development of information technology and social media has changed the way people express their opinions, particularly regarding political and governmental issues. One of the most popular social media platforms in Indonesia is Twitter. Twitter was chosen for this study because of its unique characteristics, such as character limits, the use of hashtags, and the rapid and widespread dissemination of information.

The focus on President Jokowi's performance was chosen because Jokowi is a prominent figure in the Indonesian government, whose policies and actions are widely discussed in the public sphere, particularly on social media. Twitter is a dynamic discussion space, where people freely voice their opinions, both in support and criticism, resulting in a vast and complex data set. However, the sheer volume of data on Twitter makes manual analysis inefficient and inaccurate. Therefore, an automated system capable of processing and analyzing public sentiment quickly and accurately is needed.

This sentiment analysis is important for providing an overview of public perception of government performance and can inform the formulation of more responsive public policies. To address the complexity of Twitter data, the Recurrent Neural Network (RNN) method was used in this study. RNNs are capable of understanding the context and relationships between words in sentences, including informal language structures such as abbreviations, mixed language, or slang frequently used on social media. This capability makes RNNs superior in processing sequential and contextual text data, such as tweets. Using this approach, this study aims to assess public opinion on President Jokowi's performance during his two terms based on Twitter data and to provide a useful overview of public perception for evaluation and future government policy formulation.

Several previous studies are relevant to this issue. Research The results showed a peak accuracy of 73.3% (Precision 75.8%, Recall 73.3%, F-measure 73.4%), with an average accuracy of 62.7% over five trials and an average accuracy increase of 2.7% per fold. The researchers recommend using more and more diverse training data to improve the analysis results. Research [1]"Twitter Sentiment Analysis of Public Figures Using the Naive Bayes Algorithm and Support Vector Machine," analyzed public opinion on Twitter toward Nadiem Makarim. The results showed that Naive Bayes achieved 91.48% accuracy, higher than SVM at 85.47%, with netizen sentiment consisting of 69.72% positive and 30.28% negative. The limitations of the study lie in the size and variety of the dataset and the focus on only one public figure, so the results are less generalizable. Research [2] "Sentiment Analysis on the Influence of Shopping Interest Based on Comments in the Marketplace Using the Recurrent Neural Network (RNN) Method" uses the RNN method modified into LSTM and GRU to analyze the

sentiment of purchase reviews. The results of the study showed an accuracy of 88% with an F-Measure of 58.20% using the stemming procedure, as well as an accuracy of 94.8% in short-term PV power forecasts. The weakness lies in the limitations of traditional RNNs in storing memory and handling unstructured data. The purpose of this study is to classify positive and negative sentiments to improve products, services, and user shopping interest.

Current research updates the sentiment analysis method from Naive Bayes to Recurrent Neural Networks (RNN) to analyze public opinion on President Jokowi's performance. RNN is more effective in capturing context and word order, making it suitable for analyzing social media texts that often use informal and mixed language. The focus of this research shifted from the New Normal policy during the pandemic to a discussion of President Jokowi's performance and opinions. Furthermore, RNN automatically selects relevant features, increasing the accuracy of the analysis results. By using a larger training dataset, this research is expected to achieve higher accuracy and provide deeper insights into public sentiment analysis. Through this research, it is hoped that a bridge will be created between public aspirations on social media and the government's policy-making process.

This research was conducted through systematic stages, starting with collecting tweet data related to President Jokowi's performance using the Twitter API. The data was then processed through text cleaning, normalization, and tokenization, followed by sentiment labeling (positive, negative, neutral). A Recurrent Neural Network (RNN) model was trained for sentiment classification and evaluated using accuracy, precision, recall, and F1-Score. The analysis results were used to understand public opinion on President Jokowi's performance, and conclusions and recommendations were then drawn as a basis for developing future sentiment analysis methods.

2. Methodology

2.1. Analysis Method

2.1.1. Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is an artificial neural network algorithm designed to process sequential data, such as text or time series, with the ability to remember context through internal memory (hidden layers). The mechanism involves playing back the output as input for the next step, thus understanding temporal relationships. RNNs are trained using Backpropagation Through Time (BPTT) to minimize errors. However, RNNs have weaknesses in long sequences, such as vanishing and exploding gradients, which were later addressed with the development of LSTM and GRU.

RNN has several basic architectural types that are tailored to data processing needs, including :

1. One to One:
   The most basic architecture where one input produces one output.
2. One to Many:
   One input produces multiple outputs, as in an image captioning application.
3. One to Many:
   One input produces multiple outputs, as in an image captioning application.
4. Many to One:
   Accepting multiple inputs to produce a single output, it is relevant for sentiment classification tasks, where a set of words in a sentence is analyzed to produce a single sentiment label (e.g., positive, negative, or neutral).
5. Many to Many:
   Capable of processing multiple inputs into multiple outputs, for example in language translation machine applications.

In this study, RNNs were used because of their ability to understand sequential text data. By remembering the context of previous words in a tweet, RNNs can recognize language patterns that reflect sentiment toward Jokowi's performance. This allows the model to generate more accurate sentiment predictions based on the historical information of each word or phrase.

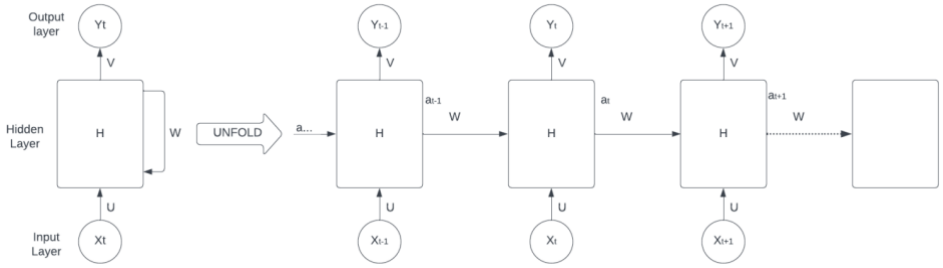

Fig. 1: RNN architecture

Figure 1 shows the architecture of an RNN sourced from GeeksforGeeks, a website that provides tutorials on various emerging technologies. An RNN takes input X and produces output Y by sequentially scanning the data from left to right, with each step updating the hidden state and producing an output

2.1.2. Model Performance Evaluation

In developing classification systems such as text-based sentiment analysis, model performance evaluation is crucial for determining how well the model performs in predicting previously unseen data (test data). This evaluation aims not only to measure accuracy but also to assess the balance of the model's ability to recognize each class, including positive, negative, and neutral sentiments. By considering multiple evaluation metrics such as precision, recall, and F1-score, researchers can gain a more comprehensive understanding of the model's strengths and weaknesses, ensuring that the system does not favor one class over the others and can provide reliable results in real-world applications.Some classification evaluation metrics commonly used in testing sentiment classification models, especially those based on Recurrent Neural Network (RNN), are explained as follows:

1. Accuracy
   Accuracy is the most basic metric, measuring the ratio of the number of correct predictions to the total number of predictions made by the model. Accuracy is appropriate when the data is evenly distributed, but can be misleading when the data is dominated by a single class. In Data Mining[3]: Concepts and Techniques, accuracy is often the initial metric in assessing model performance, but it must be combined with other metrics in the context of imbalanced data[4].
2. Precision
   Precision measures the proportion of predictions classified as positive that are actually positive. This metric is important when the cost of false positives is higher. Presisi tinggi menunjukkan bahwa model jarang mengklasifikasikan data negatif sebagai positif.
3. Recall (Sensitivity or True Positive Rate)
   Recall measures how many positive data points the model successfully recognizes out of all the data points that should be classified as positive. This is crucial when false negatives need to be minimized. In the context of public opinion analysis, Recall is important so that true positive/negative opinions are not missed by the model.
4. F1-Score
   F1-Score is a harmonization of Precision and Recall, used when the data is imbalanced and we want to consider the trade-off between the two metrics. The F1 score is between 0 and 1, where a value close to 1 indicates a good balance between precision and recall. According to [5] in Introduction to Information Retrieval, the F1-Score is often used in NLP tasks because real-world data is often imbalanced.
5. Confusion matrix
   A confusion matrix is a table that displays the model's classification results for each class. Each row in the matrix represents the actual class, while each column represents the model's prediction. This matrix provides comprehensive information on the number of correct predictions (True Positives, True Negatives) and incorrect predictions (False Positives, False Negatives) for each sentiment class[6].

2.2. Research Stages

2.2.1. Functional Requirements

Functional requirements refer to the software components that are essential in the system design phase, as they define what the system should be able to do. Python is a popular programming language widely used for machine learning because of its simplicity, flexibility, and strong community support. One key reason is the large number of Python-based libraries[7], which provide ready-to-use tools for data processing, model building, and evaluation.

Table 1: Training Data

No.	name	Function
1.	Python	Used as a foundation for running all operations and analysis.
2.	Jupyter Notebook	Provides a platform for interactive development.
3.	Numpy	Used for numerical data calculations.
4.	Pandas	Used for data manipulation and analysis in tabular form.
5.	Matplotlib	Used for data visualization.
6.	Sastrawi	Used for stemming in Indonesian text.
7.	Natural Language Toolkit	Includes a variety of tools for tokenization, stopword removal, stemming, and other text analysis.
8.	Stopword	Used to remove common words that do not provide significant meaning in text analysis.

2.2.2. Data Collection

During the data collection phase, information was gathered from Twitter, a social media platform that provides a wide variety of relevant content for research and data analysis, particularly in the field of sentiment analysis. The dataset used in this study contained tweet content related to public satisfaction with Jokowi's leadership and performance during his presidency. The following is an example of the collected tweet data. To ensure the availability of structured and accessible data, the dataset was taken from Kaggle, a well-known platform that provides ready-to-use datasets for various research purposes. The review data obtained came from Twitter, covering a total of 1,006 tweets related to Jokowi from 2023 to 2024, which served as the basis for building and evaluating the sentiment analysis model.

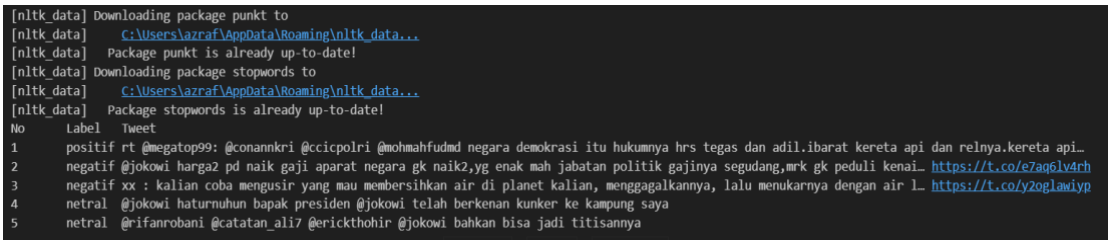

Fig. 2: Tweet Data

In the image, the main focus is on the content variable, which contains user review text and tweet labels. Some of these processes will be carried out in the data preprocessing stage.

2.2.3. Training Data

Table 2: Training Data

No.	Text	Label
1	rt @megatop99: @conannkri @ccicpolri @mohmahfudmd negara demokrasi itu hukumnya hrs tegas dan adil.ibarat kereta api dan relnya.kereta api…	Positif
2	@jokowi harga2 pd naik gaji aparat negara gk naik2,yg enak mah jabatan politik gajinya segudang,mrk gk peduli kenai… https://t.co/e7aq6lv4rh	Negatif
3	xx : kalian coba mengusir yang mau membersihkan air di planet kalian, menggagalkannya, lalu menukarnya dengan air l… https://t.co/y2oglawiyp	Negatif
4	@jokowi haturnuhun bapak presiden @jokowi telah berkenan kunker ke kampung saya	Netral
…	….	…
1005.	@jokowi @budimandjatmiko @Kemenperin_RI,dengan memanfaat teknologi jangan heran kalau rrt mampu antisipasi krisis pangan mulai dari persiapan media tanam pemilihan bibit persemaian penanaman perawatan lahan pencegahan hama hingga pemanenan memanfaatkan teknologi,	Positif

The table above represents the training data for the sentiment analysis that will be tested in this study. There are 1,006 training data sets available.

2.2.4. Pre-Processing

Preprocessing is the initial stage in text mining preparation, which aims to transform raw text into data ready for further processing. This process involves several steps, such as normalization, tokenization, stopword removal, and stemming. This stage serves to generate a collection of relevant words and support sentiment analysis. The steps involved in the preprocessing process are shown in Figure 2

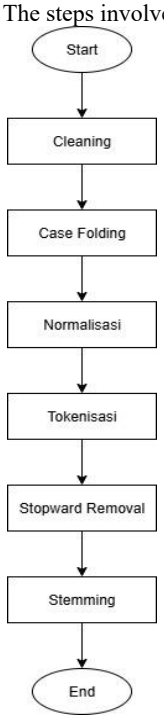

Fig. 2: Preprocessing Stages

1. Cleaning
   Cleaning removes elements such as URLs, mentions, hashtags, punctuation, numbers, and Indonesian stopwords from tweet text. The cleaning results are stored in a new column, "Cleaned_Tweet," and a comparison between the original tweet and the cleaned tweet is displayed for further analysis. The following is a cleaning segment and an image of the resulting process.

Table 3: Cleaning Results

No.	Before	After
1	rt @megatop99: @conannkri @ccicpolri @mohmahfudmd negara demokrasi itu hukumnya hrs tegas dan adil.ibarat kereta api dan relnya.kereta api…	rt negara demokrasi hukumnya hrs adil kereta api relnya kereta api
2	@jokowi harga2 pd naik gaji aparat negara gk naik2,yg enak mah jabatan politik gajinya segudang,mrk gk peduli kenai…https://t.co/e7aq6lv4rh	harga pd gaji aparat negara gk yg enak mah jabatan politik gajinya segudang mrk gk peduli kenai
3	xx : kalian coba mengusir yang mau membersihkan air di planet kalian, menggagalkannya, lalu menukarnya dengan air … https://t.co/y2oglawiyp	xx coba mengusir membersihkan air planet menggagalkannya menukarnya air

The table above is the result of text cleaning or cleaning text which removes several unimportant parts in a sentence to avoid punctuation and special characters.

2. Case Folding

Case folding is the process of converting all letters in a text to lowercase. The goal is to standardize text formatting so that words with both uppercase and lowercase letters are treated the same, thus avoiding duplication of meaning during analysis such as word searches, text classification, or modeling. For example, the words "Jokowi," "jokowi," and "JOKOWI" will be treated as the same word after case folding.

Table 4: Case Folding

No.	Before	After
1	rt negara demokrasi hukumnya hrs adil kereta api relnya kereta api	rt negara demokrasi hukumnya hrs adil kereta api relnya kereta api
2	harga pd gaji aparat negara gk yg enak mah jabatan politik gajinya segudang mrk gk peduli kenai	harga pd gaji aparat negara gk yg enak mah jabatan politik gajinya segudang mrk gk peduli kenai
3	xx coba mengusir membersihkan air planet menggagalkannya menukarnya air l	xx coba mengusir membersihkan air planet menggagalkannya menukarnya air l

3. Normalization

The function of normalization in text processing is the process of standardizing the forms of words that have the same meaning but different spellings, so that they form a consistent format. For example, non-standard words, abbreviations, or spelling variations such as "gk," "gak," and "tidak" are all normalized to "tidak." The purpose of normalization is to reduce unnecessary word variations and improve accuracy in text analysis, such as classification or sentiment analysis. Normalization is usually performed after case folding and before tokenization or feature extraction.

Table 5: Normalization Results

No.	Before	After
1	rt negara demokrasi hukumnya hrs adil kereta api relnya kereta api	negara demokrasi hukum harus adil kereta api relnya kereta api
2	harga pd gaji aparat negara gk yg enak mah jabatan politik gajinya segudang mrk gk peduli kenai	harga pada gaji aparat negara tidak yang enak sih jabatan politik gajinya segudang mereka tidak peduli kena
3	xx coba mengusir membersihkan air planet menggagalkannya menukarnya air l	coba usir bersih air planet gagal tukar air

The table above represents the data before normalization. The text still contains many non-standard words, abbreviations, or informal forms commonly used in casual conversations, such as "hrs" (harus), "gk" (tidak), "pd" (pada), "yg" (yang), and "mrk" (mereka). These words can confuse text processing models because they differ in form from their standard equivalents, even though they carry the same meaning. After normalization, these words are converted into their proper standardized forms according to the official Indonesian spelling system, for example, "hrs" becomes "harus," "gk" becomes "tidak," and "mrk" becomes "mereka." As a result, the text becomes cleaner, more consistent, and aligned with correct linguistic rules, making it more suitable for linguistic analysis or use in machine learning model training.

4. Tokenization

Tokenization is a process in Natural Language Processing (NLP) that aims to break down text into smaller units called tokens. These tokens are usually words, phrases, or even characters depending on the method used. For example, the sentence "Presiden Jokowi berpidato di istana" will be tokenized into ["Presiden", "Jokowi", "berpidato", "di", "istana"]. The purpose of tokenization is to simplify the analysis of the structure and meaning of the text, since NLP models or algorithms work more effectively when the input consists of structured tokens. Tokenization is also an important initial step before other stages such as stemming, sentiment analysis, or text classification.

Table 6: Tokenization Results

No.	Before	After
1	negara demokrasi 1864okow harus adil kereta api rel kereta api	[negara, demokrasi, 1864okow, harus, adil, kereta, api, rel, kereta, api]
2	harga pd gaji 1864okowi1864 negara gk yang enak mah jabatan politik gajinya segudang mrk gk peduli kenai	[harga, pd, gaji, 1864okowi1864, negara, gk, yang, enak, mah, jabatan, politik, gajinya, segudang, mrk, gk, peduli, kenai]
3	xx coba mengusir membersihkan air planet menggagalkannya menukarnya air l	[coba, mengusir, membersihkan, air, planet, menggagalkannya, menukarnya, air, l]

Before tokenization, the data in the "Normalized" column still consists of complete sentences in string format, so each word is merged into a single block of text. This makes analysis difficult because natural language processing models or algorithms cannot identify words individually. After tokenization is applied, each sentence is split into a list of individual words called tokens. For example, the text "presiden jokowi berpidato di istana negara" will be transformed into ['presiden', 'jokowi', 'berpidato', 'di', 'istana', 'negara']. The results of this tokenization are stored in a new column called "Tokenized," which allows each word to be analyzed in a more structured way. This process is crucial as an initial step before further text analysis such as stemming, classification, or feature extraction.

5. Stopword Removal

Stopwords are common words that frequently appear in text, such as “dan” (and), “yang” (which/that), “di” (in/at), “ke” (to), and so on. The main function of stopword removal is to eliminate these words, as they usually do not carry significant meaning in text analysis.

Table 7: Stopword Removal Results

No.	Before	After
1	negara demokrasi hukum harus adil kereta api rel kereta api	negara demokrasi hukum adil kereta api rel kereta api
2	harga pd gaji aparat negara gk yang enak mah jabatan politik gajinya segudang mrk gk peduli kenai	harga gaji aparat negara enak jabatan politik gajinya segudang peduli kenai
3	xx coba mengusir membersihkan air planet menggagalkannya menukarnya air l	coba mengusir membersihkan air planet menggagalkannya menukarnya air

The table above shows the stopword removal process, which is the elimination of non-essential words from a sentence. The "Before" column contains the original text, while the "After" column contains the text after stopwords have been removed using both the basic stopword list and additional stopwords (such as: "gk", "yang", "mah", "mrk", "pd", etc.).

6. Stemming
Stemming is the process of converting derived words into their root form. In Indonesian, stemming is used to remove affixes (prefixes, suffixes, and infixes) so that words such as "berlari," "pelari," and "berlarian" are reduced to a single root form, namely "lari." The following is a segment of the stemming process

Table 8: Stemming Results

No.	Before	After
1	negara demokrasi hukum adil kereta api rel kereta api	negara demokrasi hukum adil kereta api rel kereta api
2	harga gaji aparat negara enak jabatan politik gajinya segudang peduli kenai	harga gaji aparat negara enak jabat politik gaji gudang peduli kena
3	xx coba mengusir membersihkan air planet menggagalkannya menukarnya air l	coba usir bersih air planet gagal tukar air l

The table above shows the results of the stemming process, which converts affixed words into their root forms. For example, "jabatan" becomes "jabat," "gajinya" becomes "gaji," and "mengusir" becomes "usir." This process aims to simplify the text, making it more consistent and easier to analyze, especially in tasks such as classification or information retrieval.

7. Tokenization & Padding
Tokenization converts text into numbers or tokens so they can be processed by the model. Padding equalizes the length of the input so the neural network model can process data with consistent dimensions.

Table 9: Tokenization and Padding Results

No.	Tokenization	Padding
1	[20, 409, 23, 125, 307, 410, 608, 307, 410],	[20 409 23 125 307 410 608 307 410 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
2	[5, 308, 309, 20, 609, 31, 183, 308, 411, 234, 104],	[5 308 309 20 609 31 183 308 411 234 104 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
3	[1052, 126, 1053, 610, 105, 1054, 1055, 611, 105, 235].	[1052 126 1053 610 105 1054 1055 611 105 235 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

The table above shows the results of the tokenization and padding process on text data in natural language processing (NLP) using deep learning. In the Tokenization column, each sentence has been converted into a sequence of numbers using a tokenizer. Meanwhile, in the Padding column, each sequence of tokens of varying lengths is standardized to a fixed length, namely 30 digits (in this case). If the number of tokens in a sentence is less than the maximum length, the system adds a zero at the end (post-padding).

8. Neutral Data Deletion
In this segment, a data cleaning process is carried out by removing all entries with the label 'netral'. This step aims to prepare the dataset so that it only consists of two main sentiment classes, namely 'positif' and 'negatif', which are commonly used in binary sentiment classification. The process begins by displaying the initial label distribution using value_counts(). Then, data with the label 'netral' is removed using case-insensitive string filtering (str.lower()) to ensure that all variations of the word (e.g., 'Netral', 'NETRAL', etc.) are also removed. After that, the label distribution is displayed again to confirm that the 'netral' class has indeed been completely removed from the dataset.

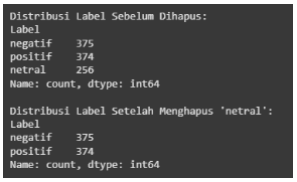

Fig. 3: Neutral Data Deletion Process Results

9. Embedding
The goal of embedding in text processing is to transform words that have been converted to numbers (through tokenization) into a more mathematically meaningful vector representation. With embedding, words with similar meanings will have adjacent vector representations in a vector space. This allows neural network models, such as LSTMs, to more effectively understand patterns and meaning in text data.

Table 8. Embedding Results

Vektor Word Embedding
[0.00513763 -0.03108134 0.00268158 -0.02178457 -0.00404441 -0.03872202
-0.00593043 -0.02505587 0.03890372 0.00064378 -0.01872915 -0.01709757
0.04747904 0.03449509 0.00136055 0.02986464 -0.04825434 -0.00519288
0.03091046 0.04539538 0.04012704 -0.03263779 0.02291216 -0.04553105
0.04221678 -0.03230178 0.04660193 -0.0330909 0.00044904 -0.00518575
……
-0.04344139 0.04939736 -0.03599025 -0.01684254 -0.01337906 0.01992743
-0.01594273 0.01014487 -0.03775709 0.0251721 -0.04059501 -0.03522197
0.04298567 -0.03910162 -0.00122388 -0.0452718 -0.02872851 0.04592545
-0.00569765 -0.0137877 -0.00685445 -0.00188285 -0.0265271 -0.04123045
-0.00775756 -0.04894583 -0.00771647 -0.027215 -0.04797603 -0.00444683
0.03225097 0.04141685 -0.03299888 -0.02651365 -0.016075 0.01384679
0.02715314 0.04364024 -0.03085764 0.0048156 -0.04777241 0.0278829
0.01547048 0.0462052 -0.04811347 0.04393904 0.03028536 -0.0064792
-0.04942241 -0.01766938 0.0140059 -0.03682026]

2.2.5. LSTM Implementation Architecture

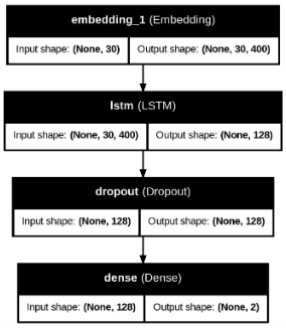

Fig. 3: LSTM Implementation Architecture

The image above shows an LSTM (Long Short-Term Memory) model built using a sequential approach with four main layers. First, the Embedding layer is used to convert word tokens into 400-dimensional vectors. Next, the data is processed by an LSTM layer with 128 units to capture the relationships between words in the sequence. After that, a Dropout layer with a ratio of 0.5 is used as regularization to prevent overfitting. Finally, the results are forwarded to the Dense layer with softmax activation to classify the data into three sentiment classes. This model is designed to process text input with a fixed length (30 tokens).

3. System Implementation

3.1. Training Model

3.1.1. Epouch

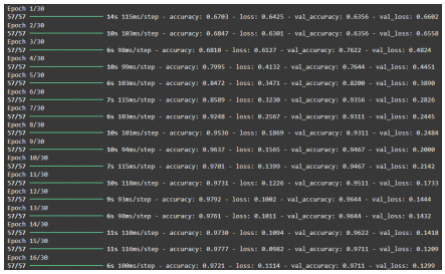

Fig. 4: Epouch LSTM

The image above shows the model training process using 30 epochs. Each epoch represents one run of the model learning the entire training data, then updating the network weights through forward propagation and backpropagation. The training results show that the accuracy value continues to increase from the first epoch to the last, while the loss value decreases, both on the training and validation data. This indicates that the model is able to learn to recognize data patterns well and generalize to the test data.

3.1.2. Evaluation of LSTM Model

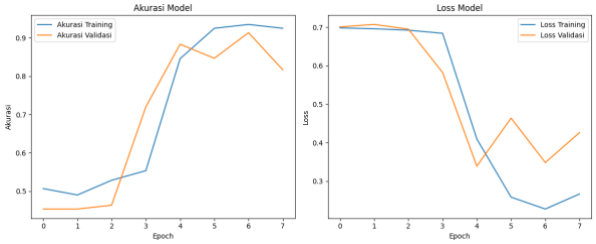

Fig. 5: LSTM Model Evaluation

The figure above shows a graph of the LSTM model's evaluation of accuracy and loss during the training process for 30 epochs. The graph above compares model performance on training and validation data. In the accuracy graph, the blue line indicates training accuracy, while the orange line indicates validation accuracy. It can be seen that model accuracy tends to increase from the 3rd to the 5th epoch, with a sharp increase in the final epoch. This indicates that the model begins to better understand patterns in the data as the number of epochs increases. Meanwhile, the graph on the right shows the loss, or model prediction error. Both the loss on the training and validation data show a downward trend, especially significant after the 3rd epoch.

3.1.3. Confusion Matriks

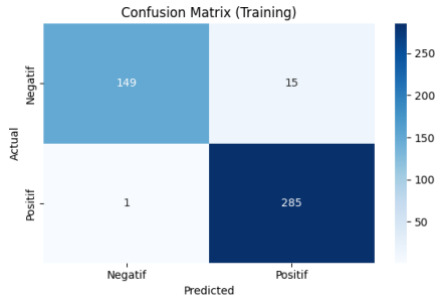

Fig. 6: Confusion Matriks

The image above shows a confusion matrix of the LSTM model's prediction results for two classes of negative and positive sentiment. Rows represent the actual labels, while columns represent the predicted labels.

3.1.4. Accuracy Results

Classification Report:	precision	recall	f1-score	support
Negatif	0.99	0.91	0.95	164
Positif	0.95	1.00	0.97	286
accuracy			0.96	450
macro avg	0.97	0.95	0.96	450
weighted avg	0.97	0.96	0.96	450

Fig. 7: Accuracy Results

The image above presents the evaluation results of the classification model, which achieved an accuracy of 96%. The model demonstrated stronger performance in identifying the Positive class with a Recall of 1.00, indicating that all Positive instances were successfully detected. For the Negative class, Precision was high (0.99), although Recall was slightly lower (0.91), suggesting that some Negative instances were not correctly identified. Meanwhile, the F1-Score values were also high, at 0.95 for Negative and 0.97 for Positive, reflecting that the model's overall performance was both balanced and reliable.

3.2. Testing Model

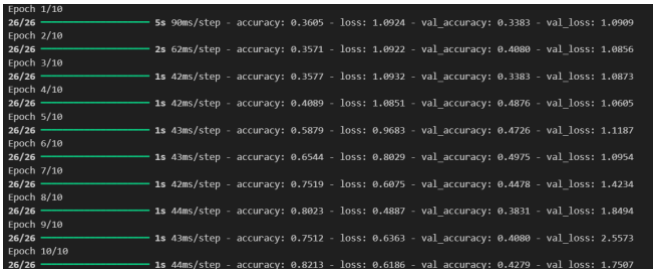

Fig. 8: 10 Epoch Testing Data Results

This is the result of model testing for 10 epochs, which clearly shows how the model accuracy and loss gradually develop on the testing data (Accuracy, Loss) as well as on the validation data (Val_Accuracy, Val_Loss) during each epoch.

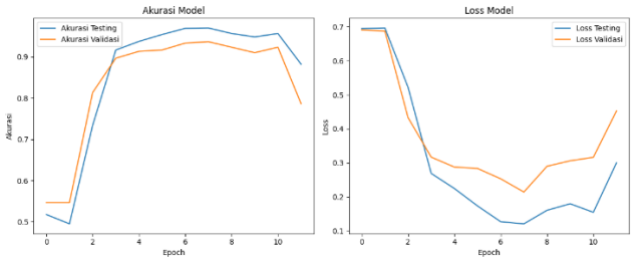

Fig. 9: Model Accuracy and Los Model Accuracy

The image above shows the model's performance during the 10-epoch training process, consisting of accuracy (left) and loss (right) graphs on the training and validation data. The accuracy graph shows the model's prediction accuracy for the training and validation data at each epoch. Meanwhile, the loss graph is used to assess the extent of the model's prediction errors. Comparing these two graphs reveals whether the model is learning well, underfitting, or overfitting.

3.3. Model Evaluation

3.3.1. Evaluasi Akurasi

The first stage of evaluation uses model.evaluate() to calculate loss and accuracy. Loss measures prediction error, while accuracy indicates the percentage of correct predictions on the test data.

```
=== Evaluasi Model (Data Test) ===
Loss     : 0.0748
Akurasi  : 97.62%
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
```

Fig. 10: Model Evaluation Results With Neutrality

The image above shows the results of model evaluation on test data using the model.evaluate() method. A loss value of 0.0748 indicates that the model's prediction error rate is relatively low. Meanwhile, an accuracy of 97.62% indicates that the model successfully classified the test data with a very high degree of accuracy. These results indicate that the model performs well in recognizing patterns in the test data.

3.3.2. Confusion Matrix

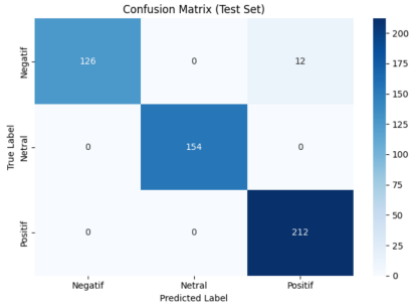

Fig. 11: Confusion Matrix

The image above is a confusion matrix showing the model's prediction results for three sentiment classes: Negative, Neutral, and Positive. The model successfully classified most of the data correctly: 126 Negative data, 154 Neutral data, and 212 Positive data. However, there were 12 misclassifications of Negative data predicted as Positive. There were no misclassifications for the Neutral and Positive classes. Overall, this indicates excellent model performance, especially in recognizing Neutral and Positive sentiments.

3.3.3. Classification Report

```
=== Classification Report ===
              precision    recall  f1-score   support

     Negatif       1.00      0.91      0.95       138
      Netral       1.00      1.00      1.00       154
     Positif       0.95      1.00      0.97       212

    accuracy                           0.98       504
   macro avg       0.98      0.97      0.98       504
weighted avg       0.98      0.98      0.98       504
```

Fig. 12: Classification Report

The image above shows a classification report displaying model evaluation metrics for each sentiment class: Negative, Neutral, and Positive. The model achieved a total accuracy of 98%, with very high Precision, recall, and f1-score values across all three classes. Notably, the Neutral class was perfectly classified (1.00 across all metrics), while the Negative class had a slightly lower recall (0.91), indicating some Negative data was misclassified. Overall, the model demonstrated excellent and balanced performance.

4. Conclusion

Based on the research results, it can be concluded that sentiment analysis of public opinion regarding President Jokowi's performance can be performed automatically using the Recurrent Neural Network (RNN) method. The RNN model has been proven capable of recognizing patterns and context in sequential texts such as tweets, and classifying sentiment into three categories: negative, neutral, and positive. With systematic preprocessing stages and a customized RNN architecture, the system is able to produce sentiment classification from unstructured data efficiently. The effectiveness of the RNN model in sentiment classification is obtained through evaluation using metrics such as accuracy, precision, recall, and F1-score. The test results show that the model has the best performance in the negative sentiment class with a high recall value, but is still relatively low in the neutral and positive classes. Therefore, improvements are needed through dataset balancing and the development of contextual features to optimize classification results. Overall, the main objective of this research has been achieved, namely developing an RNN-based sentiment analysis system and evaluating its performance in the context of classifying public opinion regarding the President's performance. The resulting system can serve as an initial reference in monitoring public opinion and be used as a consideration in making more responsive policy decisions.

Acknowledgement

With gratitude, the author expresses his gratitude to Allah SWT for His grace and blessings, enabling the successful completion of this thesis research, entitled "Twitter Sentiment Analysis to Assess Public Opinion on Jokowi's Performance During Two Terms Using the Recurrent Neural Network (RNN) Method." The author also expresses his deepest gratitude to his parents and his supervisor for their guidance, support, and direction throughout the process of preparing this thesis. Furthermore, the author expresses his gratitude to all parties who have assisted, directly or indirectly, by providing data, input, and opportunities to support the successful completion of this research.

References

[1] T. T. Widowati and M. Sadikin, "ANALISIS SENTIMEN TWITTER TERHADAP TOKOH PUBLIK DENGAN ALGORITMA NAIVE BAYES DAN SUPPORT VECTOR MACHINE," *J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 11, no. 2, pp. 626–636, 2021.

[2] G. S. Lasatiraa, K. D. Hartomo, and I. Sembiring, "Analisis Sentimen Terhadap Pengaruh Minat Belanja Berdasarkan Komentar di Marketplace Menggunakan Metode Recurrent Neural Network (RNN," *J. Sist. Inf. Bisnis*, vol. 13, no. 2, pp. 112–119, 2023.

[3] S. Mujilahwati, M. Sholihin, R. Wardhani, and M. R. Zamroni, "Python Based Machine Learning Text Classification," *J. Phys. Conf. Ser.*, vol. 2394, no. 1, 2022, doi: 10.1088/1742-6596/2394/1/012015.

[4] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2012.

[5] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[6] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Pearson, 2021.

[7] S. Mujilahwati, M. Sholihin, and R. Wardhani, "Optimasi Hyperparameter TensorFlow dengan Menggunakan Optuna di Python: Study Kasus Klasifikasi Dokumen Abstrak Skripsi," *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 1084, 2021, doi: 10.30865/mib.v5i3.3090.