JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 1, JANUARY, 2025, Pp. 33 – 47
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Implementation of Long Short-Term Memory for Chili Price Prediction in East Java Province

Fata Nabil Fikri [(1)*], Nurochman [(2)]
Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga, Yogyakarta, Indonesia
e-mail : fatanabil88@gmail.com, nurochman@uin-suka.ac.id.
* Corresponding author.
This article was submitted on 23 February 2024, revised on 8 April 2024, accepted on 15 April 2024, and published on 31 January 2025.

Abstract

Groceries prices often experience fluctuations in several regions in Indonesia, such as East Java Province and one of the commodities is chilies, both red chilies and rawit chilies. Predictive steps that utilize machine learning such as Long-Short Term Memory (LSTM) can be taken to estimate the next price of chili with expectations that the appropriate strategy can be taken by the authorities. LSTM is a network that developed from RNN networks in previous times by offering a longer cell memory so that more information can be stored. This research focuses on finding out whether the LSTM network can be applied to the case of chili price prediction and what architecture and hyperparameter configuration is appropriate for this case. For this reason, the experimental method is used by testing several predetermined variables to obtain the right architecture and hyperparameter configuration. The results of this research show that the LSTM network can be applied in this case and the architecture and best hyperparameter configuration obtained are the same for both types of chilies, namely red chilies and rawit chilies. For red chili, the best RMSE value that can be produced is 1751.890 and 1888.741 for rawit chili.

Keywords: LSTM, Prediction, RMSE, Chili Prices, Groceries

Abstrak

Harga bahan pangan sering terjadi fluktuasi di beberapa daerah di Indonesia seperti di Provinsi Jawa Timur dan salah satu komoditasnya yaitu cabai, baik cabai merah maupun cabai rawit. Langkah prediksi yang memanfaatkan pembelajaran mesin seperti Long-short Term Memory (LSTM) dapat ditempuh untuk memperkirakan harga cabai selanjutnya dengan harapan strategi yang tepat dapat diambil oleh pihak yang berwenang. LSTM merupakan bentuk jaringan hasil pengembangan dari jaringan RNN pada masa-masa sebelumnya dengan menawarkan memori sel yang lebih panjang sehingga lebih banyak infromasi yang dapat disimpan. Penelitian ini berfokus untuk mengetahui apakah jaringan LSTM dapat diterapkan pada kasus prediksi harga cabai serta arsitektur dan konfigurasi hyperparameter apa yang tepat untuk kasus ini. Untuk itu, metode eksperimen ditempuh dengan menguji beberapa variabel yang telah ditentukan untuk mendapatkan arsitektur serta konfigurasi hyperparameter yang tepat. Hasil dari penelitian ini menunjukkan bahwa jaringan LSTM dapat diterapkan pada kasus ini dan arsitektur serta konfigurasi hyperparameter terbaik yang didapat itu sama untuk kedua jenis cabai yaitu cabai merah dan cabai rawit. Pada data cabai merah nilai RMSE terbaik yang dapat dihasilkan yaitu 1751,890 dan 1888,741 pada data cabai rawit.

Kata Kunci: LSTM, Prediksi, RMSE, Harga Cabai, Bahan Pangan

1. INTRODUCTION

The demand for basic commodities in Indonesia is highly significant for its population. Particularly, the prices of these basic commodities frequently experience instability or fluctuation, as is evident with chili, a key commodity. Yanwardhana, reporting in CNBC Indonesia, noted that the average price of chili in Indonesia can reach Rp 106,764 per kilogram, with prices even exceeding Rp 150,000 in some regions (Yanwardhana, 2022).

In East Java Province, the price of red chili soared by 241.48%, increasing from Rp 24,840 per kilogram to Rp 84,823 per kilogram as of June 7, 2022. A similar increase was observed in red chili prices, which rose approximately 78.58%.

Such instability can be attributed to several factors, including weather conditions, fuel prices, and significant festive seasons. As quoted from CNN Indonesia, the price of chili rose ahead of the Ramadan period from Rp 55,000 per kilogram to Rp 60,000 per kilogram (CNN Indonesia, 2022).

With advancements in computational technology, predictive models leveraging machine learning can be developed. One widely used model for prediction and forecasting is the Long Short-Term Memory (LSTM). LSTM is a type of artificial neural network categorized under Recurrent Neural Networks (RNNs). RNNs are capable of addressing time-series data problems due to their inherent “memory” within their cells. However, RNNs face limitations such as the vanishing gradient problem, which slows training progress. LSTM, introduced in the 1990s, was developed to address this challenge by providing longer memory capabilities (Yadav et al., 2020).

Hochreiter and Schmidhuber in their study highlighted that LSTM resolves issues arising from Back-Propagation Through Time (BPTT) and Real-Time Recurrent Learning (RTRL) algorithms, where error signals in these algorithms either blow up or vanish as they propagate backward in time. The former causes weight oscillation, while the latter extends model training duration significantly. LSTM is specifically designed to handle such errors, enabling the model to learn data spanning over 1,000 steps (Hochreiter & Schmidhuber, 1997).

Several past studies have employed machine learning for prediction purposes. For instance, Chairurrachman I conducted research on PT Indofood CBP Sukses Makmur Tbk stock prices using the LSTM network. The study revealed that CNN-LSTM architecture achieved the best MAE value of 74.1365 (Chairurrachman, 2022).

Additionally, research by Arfan and Lussiana compared LSTM and Support Vector Regression (SVR). Using stock prices from various companies as the dataset, their findings demonstrated that LSTM produced better accuracy than SVR, particularly for longer time-series data (Arfan & ETP, 2020).

Similarly, Riyantoko et al. (2020) analyzed predictions for banking sector stock prices using the LSTM algorithm. This study examined the impact of varying epochs and optimization techniques on computation time, RMSE, and loss levels. It compared the optimizations Adam, RMSprop, and SGD, finding that Adam and RMSprop achieved similar accuracy levels ranging from 89% to 95%, while SGD lagged behind at 49% to 61%. Additionally, changing the number of epochs affected computation time but had little impact on the resulting RMSE.

Another relevant study by Syaidah et al. (2020) focused on predicting staple food prices in Jakarta using Artificial Neural Networks (ANN). Their results indicated that the chosen alpha and threshold values influenced accuracy, with lower values enhancing accuracy.

Further, a study conducted by Suradiradja (2022), titled “Machine Learning Algorithms: Multi-Layer Perceptron and Recurrent Neural Network for Predicting Large Red Chili Prices in Tangerang City,” showed notable results. The study achieved a low MAPE value of 3.79%, indicating significant accuracy, using the Multi-Layer Perceptron algorithm. However, it compared only two algorithms: Recurrent Neural Networks and Multi-Layer Perceptron.

Currently, LSTM algorithms are frequently used in research for time-series data predictions, owing to their superior accuracy compared to earlier algorithms. The primary focus of this study is to implement LSTM, determine its architecture, and optimize several hyperparameters for effective data preparation and training for predicting chili prices in East Java Province.

2. METHODS

2.1 Long Short-Term Memory (LSTM)

LSTM, or Long Short-Term Memory, is a type of artificial neural network architecture commonly applied in cases involving time-series data, text, video, or audio. LSTM represents a significant advancement over Recurrent Neural Networks (RNNs), which are ineffective at handling long-term dependencies due to their lack of persistent "memory" cells. As a result, LSTM outperforms RNNs in such scenarios. Figure 1 illustrates the structure of an LSTM cell.

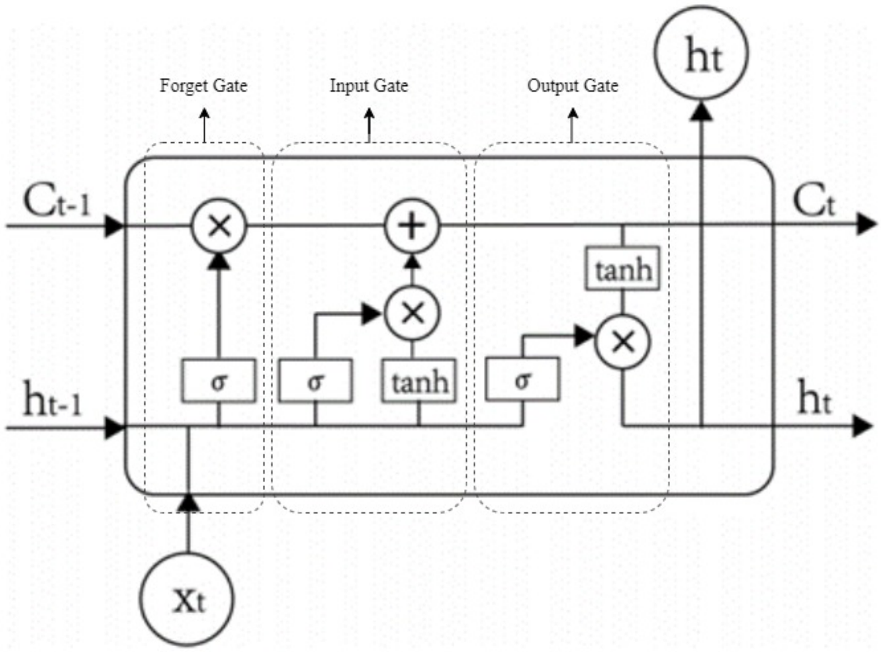


Figure 1 LSTM Structure (Qiu et al., 2020a)

Based on Figure 1, there are several components, often referred to as "gates," within an LSTM cell (Saxena, 2024):

2.1.1 Forget Gate

This gate, represented by a dashed box labeled "Forget Gate," determines whether the information from the previous state should be retained or discarded. The forget gate computes the hidden state from the previous step and the current input state using a Sigmoid activation function.

2.1.2 Input Gate

This gate, also represented by a dashed box labeled "Input Gate," evaluates whether newly incoming information is important. If deemed important, the information is added to the current state; otherwise, it is discarded. The input gate involves two computations: the input gate value using the Sigmoid activation function and the memory cell candidate value using the Tanh activation function.

2.1.3 Output Gate

This gate, shown as a dashed box labeled "Output Gate," determines the output value based on the processed information from the forget and input gates. The value of the output gate is computed, and the result becomes the value for the next hidden state.

2.2 Tools

The tools utilized in this study include both software and hardware, described as follows:

2.2.1 Python Programming Language

Python is frequently used in machine learning, data analysis, and various aspects of computer science. It can execute without the need for compilation, unlike many other programming languages (Sanner, 1999). Additionally, Python offers a package manager known as PIP, which provides access to numerous libraries related to computer science.

2.2.2 TensorFlow and Keras

TensorFlow is one of the most renowned libraries in the machine learning community. It is used for building computational models applicable in pattern recognition, image recognition, prediction, and more. TensorFlow is popular due to its multi-level abstraction, flexible coding, and comprehensive ecosystem (Gifari & Widya, 2020).

Keras API is one limitation of TensorFlow is its steep learning curve for beginners. To address this, the Keras API was developed to simplify building, training, and evaluating computational models. Keras is built on top of TensorFlow and focuses on user-friendly machine learning implementations (Dhadse, 2021).

2.2.3 Hardware

The hardware used includes a notebook equipped with a GPU (Graphics Processing Unit). Machine learning benefits from the GPU's Compute Unified Device Architecture (CUDA) to enhance computational performance. CUDA is a parallel computing platform developed by NVIDIA (Oh, 2012).

2.3 Experimental Methodology

The experimental method involves investigating relationships between variables. Variables in the experimental methodology are categorized as follows:

a) **Independent variables:** These are manipulated to observe the response of dependent variables under specific scenarios.
b) **Control variables:** These remain constant throughout the experiment to neutralize their impact on dependent variables.
c) **Dependent variables:** These are observed to understand the effect of changes in independent variables.

The flow diagram illustrating the experimental stages in this research is presented in Figure 2. The steps are as follows (Campbell & Stanley, 1963):

2.3.1 Data Collection and Preparation

In this phase, the dataset for training is collected and prepared as input for the LSTM model. This process involves several stages such as cleaning, filling in missing data, normalizing using MinMaxScaling, and splitting the dataset into training and testing data.

2.3.2 Experiment Design (Defining Variables)

This stage ensures that the experiment is conducted consistently and with focus, determining the variables relevant to the execution of the experiment.

2.3.3 Execution and Data Logging

The experiment is carried out according to the scenarios planned in the previous stage. The resulting experimental data is recorded for later analysis.

2.3.4 Analysis and Discussion of Experimental Results

The recorded data is analyzed to identify which scenario produces the best results based on evaluation metrics. Additionally, this step explains the findings of the conducted experiments.

2.3.5 Conclusion Drawing

Conclusions are drawn to succinctly summarize the experimental results, making them easier to comprehend and addressing the problem statements identified.

Figure 2 Experimental Workflow Diagram

2.4 Data collection and preparation

The dataset used in this study consists of chili price data from East Java Province over approximately three years (January 1, 2020 – June 1, 2023). It includes two types of commodities: red chili and bird's eye chili. This price data was obtained from the PIHPS Nasional website (National Strategic Food Price Information Center), managed by Bank Indonesia (Rahmadini et al., 2023). The dataset is presented in a .xlsx file format containing daily chili prices in East Java. Figure 3 shows the price trends of red chili and bird's eye chili in East Java from January 1, 2020, to June 1, 2023.

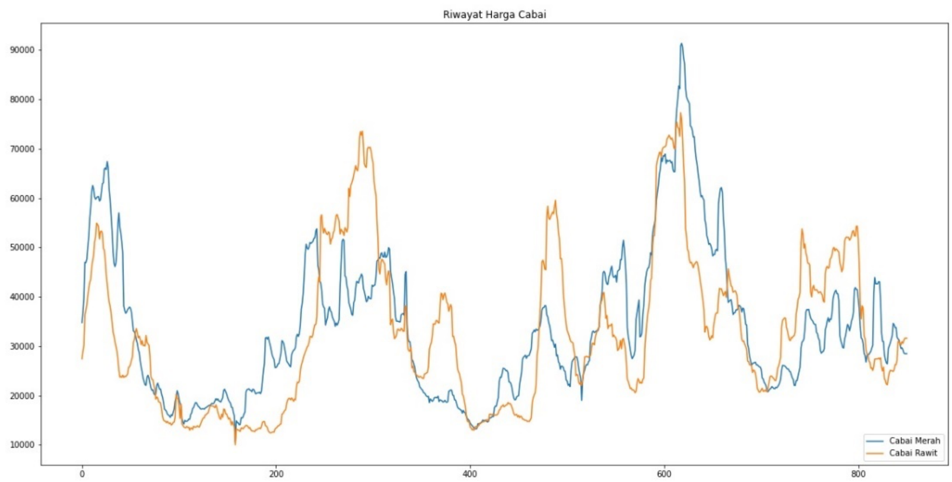


Figure 3 Red Chili and Bird's Eye Chili Price Trends

The blue line in the graph represents red chili prices, while the orange line represents bird's eye chili prices. The x-axis indicates the sequence of dates from the start to the end of data collection, and the y-axis represents the actual prices of red and bird's eye chili on their respective dates.

The total chili price data collected for the specified date range amounted to 894 records. However, this figure includes invalid data, such as missing or non-numeric values (e.g., a "-" character). Therefore, data cleaning was conducted to eliminate such anomalies, resulting in 851 valid data entries.

The next preparation step involved filling or replacing missing or undesirable data points. This step is crucial as predictive processes require uniform data intervals. For example, if the interval between data points is one day, all intervals should consistently adhere to this pattern. In this dataset, missing entries caused inconsistencies in intervals, either due to unwanted data values or entirely missing values on specific dates. For instance, no reports were available on Saturdays and Sundays because the PIHPS Nasional website does not update data on these days.

Several methods can be applied to address missing data, including using the value from the previous date, calculating the average of the values before and after the missing data, or using the value from the next date, among others. In this study, missing data was filled using the value from the preceding day. This method was chosen because the website updates data daily. Using the average of preceding and succeeding values would be impractical as succeeding values were not always available at the time of calculation. This data-filling process resulted in 1,250 values, which were then further processed.

After data cleaning and filling missing values, the next preparation step was data normalization. Normalization aims to optimize the model's training process. A commonly used method for time-series dataset normalization is Min-Max Scaling. This method adjusts the data range to a specific interval, typically between 0 and 1, though other ranges can also be used if necessary (Maity, 2021). The Min-Max Scaling equation is shown in Equation 1.

$$y = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

In this equation, $\boldsymbol{y}$ represents the normalized value within the desired range, typically from 0 to 1. The variable $\boldsymbol{x}$ is the original value from the dataset, while $\boldsymbol{x_{min}}$ and $\boldsymbol{x_{max}}$ are the minimum and maximum values in the dataset, serving as the lower and upper bounds, respectively. By transforming all data points using this formula, the values are rescaled proportionally within the specified range, optimizing the dataset for machine learning algorithms by reducing the risk of bias due to varying magnitudes in data points. This process ensures more uniform data distribution, simplifying the model's task of processing data during training.

The next step involved splitting the dataset into training and testing data. The split is typically based on a specific ratio between training and testing datasets. A common ratio used is 80:20, where 80% of the data is used to train the model, while the remaining 20% is reserved for testing its performance. With a total of 1,250 data points, this ratio yielded 1,000 entries for training and 250 entries for testing. However, in this study, the training-to-testing ratio was treated as an independent variable to be evaluated and discussed further in subsequent sections.

After splitting the data into training and testing sets, the training data was further processed into input-label pairs. Each input consists of a sequence of values of a certain length, referred to as the sequence length. The label generally consists of a single value.

For example, if the sequence length is 7, the values from Day 1 to Day 7 form one sequence or input, while the label is the value on Day 8. This sequence formation process shifts sequentially, with each new sequence beginning at the next data point in the dataset. For instance, the second sequence consists of values from Day 2 to Day 8, with the label being the value on Day 9, and so on until all training data has been processed. However, in this study, the sequence length was not fixed as it was considered an independent variable to be further examined. An example of the final format of the sequence and label formation for training data is shown in Figure 4.

```
# Split train data and validation data
train, validation = train_val_split(scaled_data, sequence_length, train_data_ratio)
# train, validation = train_val_split(dataset, sequence_length, train_data_ratio)

X_train, Y_train = x_y_split_sequence(train, sequence_length)
X_validation, Y_validation = x_y_split_sequence(validation, sequence_length)

print(X_train[0], Y_train[0])
print(X_train[1], Y_train[1])
```

```
[32] ✓ 0.0s
[0.29993816 0.29993816 0.32529375 0.36239951 0.36239951 0.36239951
 0.45145331] 0.4502164502164502
[0.29993816 0.32529375 0.36239951 0.36239951 0.36239951 0.45145331
 0.45021645] 0.45949288806431665
```

Figure 4 Example of Sequence Formation and Labeled Data from Training Data

2.5 Experimental Scenario Design

The design of experimental scenarios aims to achieve valid, relevant, and accountable results. Additionally, this design simplifies the experimental process and ensures that it remains controlled. In this study, seven scenarios were designed, applied to both types of chili data, namely red chili and bird's eye chili. The first three scenarios focus on LSTM architecture, as detailed in Tables 1, 2, and 3, while the remaining four scenarios relate to data, training, and model optimization, as explained in Tables 4, 5, 6, and 7.

Table 1 Scenario 1 (Number of Units)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	10	0	Linear	80:20	30	30	Adam
2	20	0	Linear	80:20	30	30	Adam
3	30	0	Linear	80:20	30	30	Adam
4	40	0	Linear	80:20	30	30	Adam
5	50	0	Linear	80:20	30	30	Adam
6	60	0	Linear	80:20	30	30	Adam
7	70	0	Linear	80:20	30	30	Adam
8	80	0	Linear	80:20	30	30	Adam
9	90	0	Linear	80:20	30	30	Adam
10	100	0	Linear	80:20	30	30	Adam

Table 2 Scenario 2 (Hidden Layers)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	30	30	Adam
2	50	1	Linear	80:20	30	30	Adam
3	50	2	Linear	80:20	30	30	Adam
4	50	3	Linear	80:20	30	30	Adam

Table 3 Scenario 3 (Activation Function)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	30	30	Adam
2	50	0	Relu	80:20	30	30	Adam
3	50	0	Leaky Relu	80:20	30	30	Adam
4	50	0	SELU	80:20	30	30	Adam

Table 4 Scenario 4 (Training Data Ratio)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	70:30	30	30	Adam
2	50	0	Linear	80:20	30	30	Adam
3	50	0	Linear	90:10	30	30	Adam

Table 5 Scenario 5 (Sequence Length)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	7	30	Adam
2	50	0	Linear	80:20	14	30	Adam
3	50	0	Linear	80:20	21	30	Adam
4	50	0	Linear	80:20	30	30	Adam

Table 6 Scenario 6 (Epoch)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	30	10	Adam
2	50	0	Linear	80:20	30	20	Adam
3	50	0	Linear	80:20	30	30	Adam
4	50	0	Linear	80:20	30	40	Adam
5	50	0	Linear	80:20	30	50	Adam

Table 7 Scenario 7 (Model Optimization)

No.	Number of Units	Hidden Layers	Activation Function	Training Data Ratio	Sequence Length	Epochs	Model Optimization
1	50	0	Linear	80:20	30	30	Adam
2	50	0	Linear	80:20	30	30	RMSprop
3	50	0	Linear	80:20	30	30	SGD

Scenario 1 evaluates the number of LSTM units (independent variable) with varying values, while other variables are treated as control variables, kept constant across all trials in this scenario. **Scenario 2** examines the number of hidden layers between the input and output layers, testing variations of 1, 2, and 3 layers, as well as a condition without hidden layers. Other variables remain as controls. **Scenario 3** focuses on testing different activation functions applied to the output layer, including Linear, Relu, Leaky Relu, and SELU.

Scenario 4 investigates aspects of data, training, and model optimization by treating the training-to-testing ratio as an independent variable with several values tested. **Scenario 5** evaluates the sequence length with various values such as 7, 14, 21, and 30, representing time periods of 1 week, 2 weeks, 3 weeks, and 1 month, respectively. **Scenario 6** tests the number of epochs run during training to observe its effect on model performance. The final scenario, **Scenario 7**, tests various model optimization methods, including Adam, RMSprop, and SGD, which are commonly used in machine learning to enhance model performance.

2.6 Evaluation metrics

Evaluation metrics are measures used to assess the outcomes of an experiment. In this study, a key metric is the accuracy level of a model. Additionally, the training duration of a model is considered to evaluate performance. However, accuracy has greater significance compared to training duration, which is only considered further when models yield similar accuracy levels. These evaluation metrics also serve as dependent variables in this experiment. Below are the metrics used in this study:

2.6.1 Root Mean Squared Error (RMSE)

The Root Mean Squared Error (RMSE) equation measures the discrepancy between the predicted values and actual values in a dataset. It is calculated by first squaring the differences between each actual value (y_i) and its corresponding predicted value ($\hat{y_i}$), then summing these squared differences across all data points. This total is divided by the number of data points (n) to calculate the mean squared error, and finally, the square root is taken to obtain the RMSE. A smaller RMSE value indicates better model performance, as it reflects a lower prediction error.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{n}} \quad (2)$$

2.6.2 Mean Absolute Error Percentage (MAPE)

The Mean Absolute Error Percentage (MAPE) equation measures prediction error as a percentage, making it easier to interpret across datasets of different scales. For each data point, the absolute difference between the actual value (y_i) and the predicted value ($\hat{y_i}$) is calculated, divided by the actual value (y_i), and multiplied by 100 to convert it into a percentage. These percentages are then averaged across all data points (n) to yield the MAPE. Lower MAPE values indicate higher model accuracy and better prediction performance.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y_i}}{y_i}\right| \cdot 100 \quad (3)$$

2.6.3 Training Duration

The final evaluation metric used in this experiment is training duration. It refers to the time taken for the LSTM model to complete training, from the first epoch to the last. Training duration is measured in seconds, starting right before the training process begins and ending upon its completion.

2.7 Experiment Execution and Results Recording

The experiment was conducted by running all scenarios sequentially, from the initial to the final scenario, for both red chili and bird's eye chili data. Each independent variable value was tested three times, and the results of each test were recorded. The average RMSE value from the three tests was then calculated, and the lowest RMSE value was selected as the basis for comparing scenarios.

The experiment results were automatically recorded each time a scenario was executed, saved in a .txt file format with columns separated by commas (","). This results recording process is illustrated in Figure 5.

To facilitate the readability of the results, the data was transformed into a tabular format using spreadsheet software (Microsoft Excel) and appropriate column headers were added. Figure 6 shows the experimental results in tabular format.

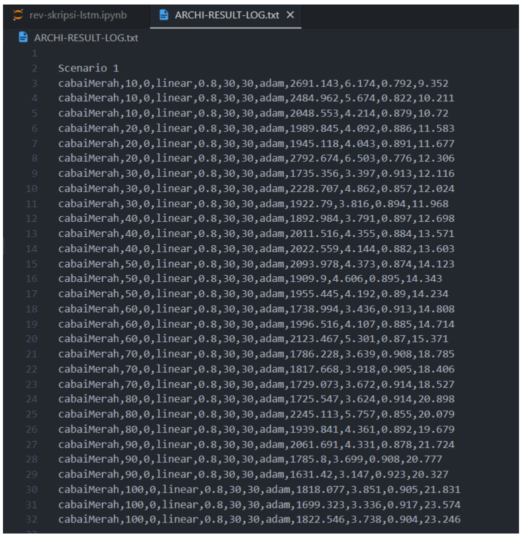
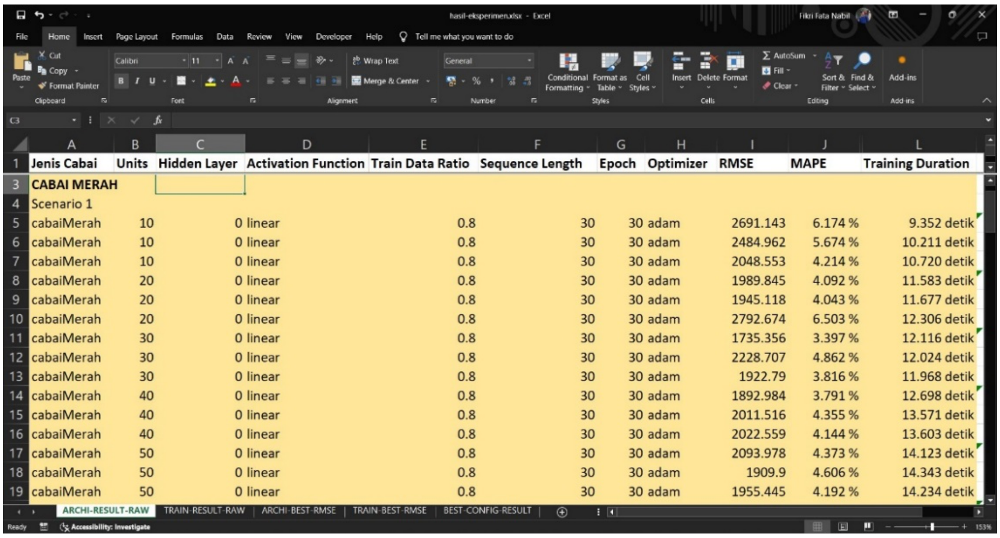


Figure 5 Experimental Results Recording



Jenis Cabai	Units	Hidden Layer	Activation Function	Train Data Ratio	Sequence Length	Epoch	Optimizer	RMSE	MAPE	Training Duration
CABAI MERAH										
Scenario 1										
cabaiMerah	10	0	linear	0.8	30	30	adam	2691.143	6.174 %	9.352 detik
cabaiMerah	10	0	linear	0.8	30	30	adam	2484.962	5.674 %	10.211 detik
cabaiMerah	10	0	linear	0.8	30	30	adam	2048.553	4.214 %	10.720 detik
cabaiMerah	20	0	linear	0.8	30	30	adam	1989.845	4.092 %	11.583 detik
cabaiMerah	20	0	linear	0.8	30	30	adam	1945.118	4.043 %	11.677 detik
cabaiMerah	20	0	linear	0.8	30	30	adam	2792.674	6.503 %	12.306 detik
cabaiMerah	30	0	linear	0.8	30	30	adam	1735.356	3.397 %	12.116 detik
cabaiMerah	30	0	linear	0.8	30	30	adam	2228.707	4.862 %	12.024 detik
cabaiMerah	30	0	linear	0.8	30	30	adam	1922.79	3.816 %	11.968 detik
cabaiMerah	40	0	linear	0.8	30	30	adam	1892.984	3.791 %	12.698 detik
cabaiMerah	40	0	linear	0.8	30	30	adam	2011.516	4.355 %	13.571 detik
cabaiMerah	40	0	linear	0.8	30	30	adam	2022.559	4.144 %	13.603 detik
cabaiMerah	50	0	linear	0.8	30	30	adam	2093.978	4.373 %	14.123 detik
cabaiMerah	50	0	linear	0.8	30	30	adam	1909.9	4.606 %	14.343 detik
cabaiMerah	50	0	linear	0.8	30	30	adam	1955.445	4.192 %	14.234 detik

Figure 6 Experimental Results in Tabular Format

3. RESULTS AND DISCUSSION

After running all scenarios, the best results (lowest averages) for each scenario, for both red chili and bird's eye chili, are presented. The best results for red chili are shown in Table 8. The RMSE values across all scenarios range from 1600 to 1900, with the lowest RMSE observed in Scenario 7 during RMSprop model optimization, resulting in an RMSE of 1635.065. Regarding MAPE, all scenarios achieved values below 4%, with the smallest MAPE also found in Scenario 7 during RMSprop optimization.

The longest training duration occurred in Scenario 6 when testing the number of epochs, as it used the highest value of 50 epochs. However, the smallest average RMSE was observed in Scenario 6, at 1754.700. Conversely, Scenario 7, which had the lowest RMSE value, produced a higher average RMSE compared to Scenario 6. Table 9 compares the average RMSE values in Scenarios 6 and 7. In Scenario 7, the results of RMSprop optimization tests were inconsistent, as not all tests achieved low RMSE values (close to the minimum RMSE). As a result, the average RMSE in Scenario 7 was relatively high. On the other hand, the RMSE values in Scenario 6 were more consistent, staying within the 1700 range, and thus yielded a lower average RMSE than Scenario 7.

Table 8 Best Results for Red Chili Price Prediction

No.	Scenario	Independent Variable	Variable Value	RMSE	MAPE (%)	Training Duration (s)	Average RMSE
1	Scenario 1	Number of LSTM Units	70	1729.073	3.67%	18.527	1777.656
2	Scenario 2	Hidden Layers	0	1663.543	3.21%	13.744	1781.956
3	Scenario 3	Activation Function	Linear	1861.336	3.90%	14.337	1953.674
4	Scenario 4	Training Data Ratio	80:20	1710.004	3.52%	10.886	1975.861
5	Scenario 5	Sequence Length	7	1932.669	3.93%	5.431	1923.455
6	Scenario 6	Epochs	50	1739.348	3.52%	19.127	1754.700
7	Scenario 7	Optimization Method	RMSprop	1635.065	3.16%	14.777	1931.978

Table 9 Comparison of Average RMSE Between Scenario 6 and Scenario 7 for Red Chili Price Prediction

No.	Scenario	Independent Variable	RMSE 1	RMSE 2	RMSE 3	Average RMSE
1	Scenario 6	50 Epochs	1778.592	1746.161	1739.348	1754.700
2	Scenario 7	RMSprop Optimization	1987.332	1635.065	2173.538	1931.978

Table 10 Best Results for Bird's Eye Chili

No.	Scenario	Independent Variable	Independent Variable Value	RMSE	MAPE	Training Duration	Average RMSE
1	Scenario 1	Number of Units	80	1913.818	3.59 %	19.936 seconds	1969.866
2	Scenario 2	Hidden Layers	0	1929.001	3.55 %	11.222 seconds	2099.155
3	Scenario 3	Activation Function	Linear	2056.248	4.01 %	14.694 seconds	2104.328
4	Scenario 4	Training Data Ratio	90:10	1941.933	3.384 %	15.183 seconds	2067.900
5	Scenario 5	Sequence Length	7	1936.939	3.485 %	6.531 seconds	2061.045
6	Scenario 6	Epochs	50	1837.712	3.419 %	22.351 seconds	1878.329
7	Scenario 7	Model Optimization	RMSprop	1834.365	3.423 %	15.001 seconds	1983.437

Table 10 presents the best results for bird's eye chili in each scenario. Across all scenarios, RMSE values for bird's eye chili ranged higher than those for red chili, with values between 1800 and 2000. The lowest RMSE occurred in Scenario 7 during RMSprop optimization testing. The smallest MAPE, however, was achieved in Scenario 4, during the 90:10 training-to-test data ratio test, with a MAPE of 3.384. This indicates that MAPE does not always correlate with RMSE values.

The training durations required for bird's eye chili were similar to those for red chili. The longest duration for bird's eye chili occurred in the 50-epoch test in Scenario 6, lasting 22.351 seconds. The training duration difference between the two datasets was only around 3 seconds, suggesting similar training time requirements for both data types.

As with red chili, scenarios with the smallest RMSE values did not always result in the smallest average RMSE for bird's eye chili. In Scenario 7, RMSprop optimization testing produced the lowest RMSE (1834.365), but the smallest average RMSE was observed in Scenario 6, during the 50-epoch test. The inconsistency of RMSE values in Scenario 7 contributed to a higher average RMSE.

Table 11 Comparison of Average RMSE Between Scenario 6 and Scenario 7 for Bird's Eye Chili Price Prediction

No.	Scenario	Independent Variable	RMSE 1	RMSE 2	RMSE 3	Average RMSE
1	Scenario 1	Number of Units	80	1913.818	3.59 %	19.936 seconds
2	Scenario 2	Hidden Layers	0	1929.001	3.55 %	11.222 seconds

Table 11 shows that the RMSE range from RMSprop optimization testing is quite broad, with a minimum value of 1834.365 and a maximum of 2174.004. In contrast, the RMSE range during the 50-epoch test was narrower, with a minimum of 1837.712 and a maximum of 1974.733. Based on this discussion, the scenario with the smallest average RMSE is Scenario 6 during the 50-epoch test, for both red chili and bird's eye chili. This variable configuration is shown in Table 6. Scenario 6 with 50 epochs was retested five times to confirm it as the best scenario. The retesting was performed for both chili types, and the results are shown in Table 12.

Table 12 Re-Test Results for Scenario 6 Epoch 50

No.	Type of Chili	RMSE	MAPE	Training Duration	Average RMSE
1	Red Chili (Cabai Merah)	1540.898	2.923 %	22.270 seconds	1751.690
2	Bird's Eye Chili (Cabai Rawit)	1816.318	3.503 %	28.906 seconds	1888.741

Table 12 indicates that the retest results for Scenario 6 with 50 epochs produced average RMSE values similar to those observed in the initial test, as seen in Tables 9 and 11. However, Table 12 provides enough evidence that Scenario 6 with 50 epochs yields more consistent RMSE values than the other scenarios.

Figures 7 and 8 illustrate the retest results for Scenario 6 with 50 epochs for red chili and bird's eye chili. These show good accuracy levels, with a narrow gap between the prediction curve (orange) and the actual curve (blue). Although there are some wider gaps visible, as in Figure 6, the RMSE results in Table 12 show that red chili has a lower RMSE than bird's eye chili.

For further testing, the trained model configured with Scenario 6 was tested on data outside the training dataset (price data for red chili and bird's eye chili from June 1, 2023, to October 18, 2023). After applying the same data preparation process used for the training dataset, 140 data points were obtained, and the model produced RMSE and MAPE values, as shown in Table 13.

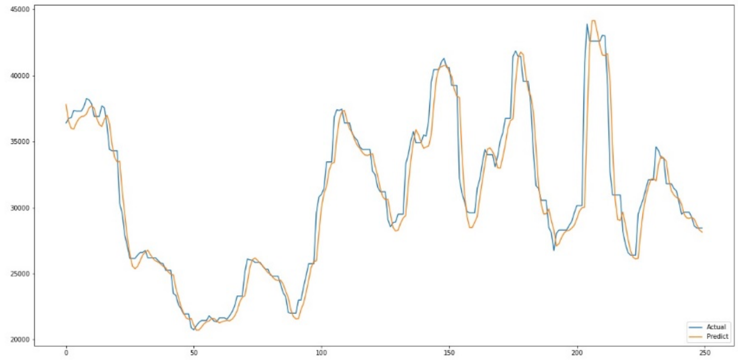


Figure 7 Graph of Re-Test Results for Scenario 6 Epoch 50 on Red Chili

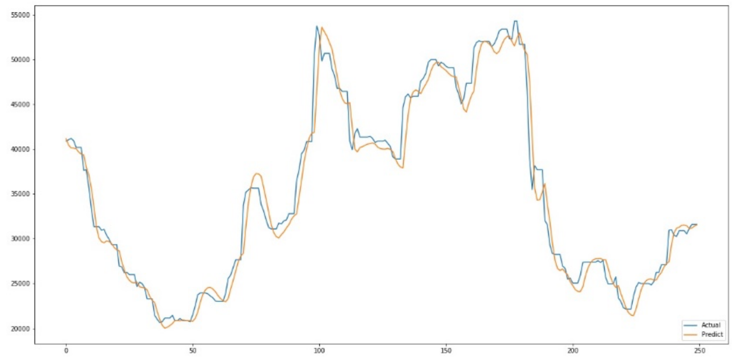


Figure 8 Graph of Re-Test Results for Scenario 6 Epoch 50 on Bird's Eye Chili

Table 13 Results of the Trained Model Testing

No.	Type of Chili	RMSE	MAPE
1	Red Chili (Cabai Merah)	1160.695	2.280 %
2	Bird's Eye Chili (Cabai Rawit)	816.052	2.256 %

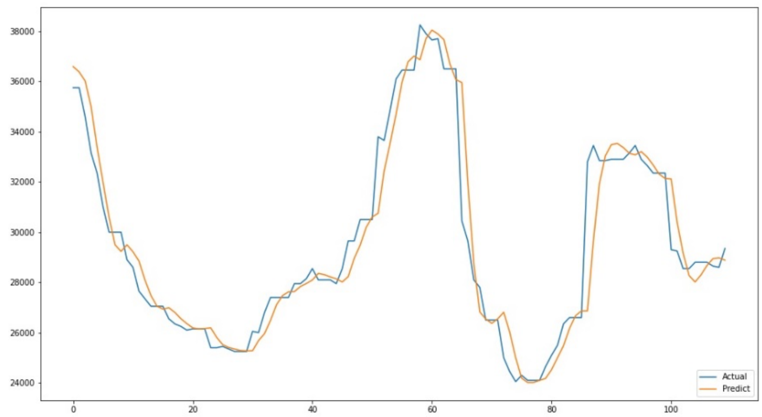


Figure 9 Grafik Hasil Pengujian Model yang Sudah Dilatih pada Data Cabai Merah

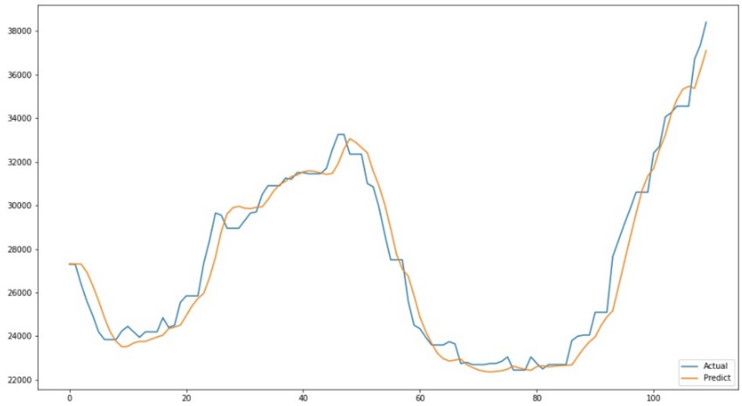


Figure 10 Grafik Hasil Pengujian Model yang Sudah Dilatih pada Data Cabai Rawit

Both types of chili, red chili and bird's eye chili, produced very low RMSE values compared to those from the previous scenarios, as well as low MAPE values below 2.5%. This indicates that the trained model does not suffer from overfitting and generalizes the price patterns of both chili types well. Overfitting occurs when a model performs well on the training dataset but poorly on the test dataset because it focuses too much on the training details without recognizing general patterns in the test data (Ying, 2019). In this case, the trained model did not overfit, as the RMSE values produced on the test data were not significantly different from those on the training data, as evidenced in Tables 12 and 13, which show test results on data beyond the training dataset range.

Graphs depicting the test results on the external dataset for both chili types can be seen in Figures 9 and 10. The prediction curve (orange) and the actual curve (blue) are almost overlapping, with only slight gaps between the two. This demonstrates that the trained model produces highly accurate predictions when tested on a different dataset, indicating its strong ability to generalize patterns from unseen data.

4. CONCLUSIONS

In this study, the researchers attempted to predict chili prices in East Java Province using the Long Short-Term Memory (LSTM) network, a method in machine learning. To achieve this goal, the researchers designed and conducted experiments to determine the LSTM architecture and effective variable configurations for this case. The results show that the LSTM network is well-suited for chili price data, producing satisfactory results in predicting chili prices in East Java. Among all scenarios tested, Scenario 6, with 50 epochs, achieved the best results with the smallest RMSE and MAPE. The average RMSE values from the retest of Scenario 6 were 1751.690 for red chili and 1888.741 for bird's eye chili. These results suggest that the developed model has strong predictive capabilities and generalizes well for both chili types.

For future research, it is recommended to test the LSTM architecture on other types of time-series data, such as stock prices or weather data, to expand its application. Furthermore, it is advised to develop an information system that implements this chili price prediction model, enabling stakeholders to plan chili pricing and distribution more efficiently. With these measures, the results of this study could provide broader and more impactful contributions to the field of agricultural commodity price prediction.

REFERENCES

Arfan, A., & ETP, L. (2020). Perbandingan Algoritma Long Short-Term Memory dengan SVR Pada Prediksi Harga Saham di Indonesia. *PETIR*, *13*(1), 33–43. https://doi.org/10.33322/petir.v13i1.858

Campbell, D. T., & Stanley, J. C. (1963). *Experimental And Quasi-Experiment Al Designs For Research*. Houghton Mifflin Company.

CNN Indonesia. (2022, March 8). *Harga Cabai Rawit di Pasar Wilayah Jatim Naik Jelang Ramadan*. CNN Indonesia. https://www.cnnindonesia.com/ekonomi/20220307190530-92-767987/harga-cabai-rawit-di-pasar-wilayah-jatim-naik-jelang-ramadan

Dhadse, A. (2021, May 23). *Introduction to TensorFlow with Keras API*. Analytics Vidhya. https://medium.com/analytics-vidhya/introduction-to-tensorflow-with-keras-api-36cbeeb562d5

Gifari, J., & Widya, A. (2020, November 17). *Belajar Data Science : Apa yang dimaksud dengan Tensorflow dan Bagaimana Penggunaannya?* DQLab. https://dqlab.id/belajar-data-science-pahami-tensflow

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Chairurrachman, I. (2022). *Penerapan Long Short-Term Memory Pada Data Time Series Untuk Prediksi Harga Saham Pt. Indofood Cbp Sukses Makmur Tbk (ICBP)* [UIN SUNAN KALIJAGA YOGYAKARTA]. https://digilib.uin-suka.ac.id/id/eprint/53738/

Maity, R. (2021, June 24). *Min Max Scaler*. Medium. https://medium.com/@ranjitmaity95/min-max-scaler-b2411ab3136d

Oh, F. (2012, September 10). *What Is CUDA?* NVIDIA. https://blogs.nvidia.com/blog/what-is-cuda-2/

Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLOS ONE*, *15*(1), e0227222. https://doi.org/10.1371/journal.pone.0227222

Rahmadini, R., LorencisLubis, E. E., Priansyah, A., N, Y. R. W., & Meutia, T. (2023). Penerapan Data Mining Untuk Memprediksi Harga Bahan Pangan Di Indonesia Menggunakan Algoritma K-Nearest Neighbor. *Jurnal Mahasiswa Akuntansi Samudra*, *4*(4), 223–235. https://doi.org/10.33059/jmas.v4i4.7074

Riyantoko, P. A., Fahruddin, T. M., Hindrayani, K. M., & Safitri, E. M. (2020). Analisis Prediksi Harga Saham Sektor Perbankan Menggunakan Algoritma Long-Short Terms Memory (LSTM). *Seminar Nasional Informatika (SEMNASIF)*, *1*(1), 427–435. http://www.jurnal.upnyk.ac.id/index.php/semnasif/article/view/4135

Sanner, M. F. (1999). Python: a programming language for software integration and development. *Journal of Molecular Graphics & Modelling*, *17*(1), 57–61. http://www.ncbi.nlm.nih.gov/pubmed/10660911

Saxena, S. (2024, December 30). *What is LSTM? Introduction to Long Short-Term Memory*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

Suradiradja, K. H. (2022). Algoritme Machine Learning Multi-Layer Perceptron dan Recurrent Neural Network untuk Prediksi Harga Cabai Merah Besar di Kota Tangerang. *Faktor Exacta*, *14*(4), 194. https://doi.org/10.30998/faktorexacta.v14i4.10376

Syaidah, K., Chrisnanto, Y. H., & Abdillah, G. (2020). Prediksi Harga Sembako di DKI Jakarta Menggunakan Artificial Neural Network. *JUMANJI (Jurnal Masyarakat Informatika Unjani)*, *3*(02), 136. https://doi.org/10.26874/jumanji.v3i02.63

Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science*, *167*, 2091–2100. https://doi.org/10.1016/j.procs.2020.03.257

Yanwardhana, E. (2022, June 15). *Ternyata Gegara Ini Harga Cabai Terbang Sampai Ratusan Ribu*. CNBC Indonesia. https://www.cnbcindonesia.com/news/20220615103728-4-347195/ternyata-gegara-ini-harga-cabai-terbang-sampai-ratusan-ribu

Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, *1168*(2), 022022. https://doi.org/10.1088/1742-6596/1168/2/022022