RICE PRICE PREDICTION IN EAST SUMBA REGENCY USING THE NEURAL NETWORK ALGORITHM

Renol Bulu Manggal[1*], Arini Aha Pekuwali [2], dan Raynesta Mikaela Indri Malo[3]

[1,2,3]Fakultas Sains dan Teknologi, Universitas Kristen Wira Wacana Sumba, Nusa Tenggara Timur, Indonesia
Email[1*]: renolmanggal@gmail.com
Email[2]: arini.pekuwali@unkriswina.ac.id
Email[3]: raynesta@unkriswina.ac.id

ABSTRACT

Fluctuations in rice prices in East Sumba Regency are an important issue that directly affects farmers, traders, and consumers. Unstable price changes are influenced by weather conditions, supply availability, distribution, and market dynamics. Therefore, a prediction method is needed that can provide accurate estimates of rice prices as a basis for decision making. This study aims to predict the price of medium rice in East Sumba Regency using the Neural Network algorithm, specifically Long Short-Term Memory (LSTM), which is effective in modeling time series data. The data used are monthly rice price data for the period January 2021 to December 2025 obtained from Perum BULOG Waingapu Branch Office, with data processing and analysis carried out after all 2025 data became available. The research stages include data collection, data preprocessing, normalization using Min-Max Scaling, time series dataset formation, division of training and testing data, LSTM model training, and model performance evaluation. The evaluation was carried out using the Root Mean Square Error (RMSE) metric. The results show that the LSTM model is able to predict rice prices with an RMSE value of 360.91 Rp/Kg or around 3.35% of the average rice price. This value indicates that the prediction error of the model is relatively small, so the model can be said to have good prediction performance. Therefore, the developed LSTM model is considered feasible to be used as a tool for predicting rice prices and is expected to help farmers and traders in planning sales and become a consideration for the local government in maintaining rice price stability in East Sumba Regency.
Keywords: rice price prediction, neural network, LSTM, time series, East Sumba.

1. INTRODUCTION

Rice is one of the most important food commodities in the world and is a staple food for nearly half of the population, especially in Asia. In Indonesia, rice has a strategic position because it is the main source of carbohydrates for the community and has an impact on economic and social conditions [1]. The stability of rice prices is often linked to national stability because changes in prices can have a direct impact on people's purchasing power. Therefore, the government has taken various measures, such as maintaining reserve stocks and conducting market operations, to control price fluctuations. However, rice prices in various regions of Indonesia still fluctuate significantly from time to time.

Fluctuations in rice prices also occur in East Sumba Regency, East Nusa Tenggara Province. Geographical conditions and an unpredictable climate cause instability in rice production, which affects the supply of rice in the market. Based on data on medium-grade rice prices for the 2021-2025 period obtained from the Waingapu Branch of Perum BULOG, there has been an upward trend in prices from around IDR 8,600/kg at the beginning of 2021 to around IDR 12,607/kg at the end of 2025, an increase of around 46.6% over five years. This increase indicates that there is pressure on demand and production cost factors that affect rice prices in East Sumba Regency. Data from BULOG was chosen because this institution plays an important role in maintaining the availability and stability of rice prices, so it is considered representative in describing market conditions in the region [2]. This upward trend indicates a pattern of price changes that needs to be analyzed further in order to make more accurate predictions [3].

In facing the dynamics of food commodity prices, the use of artificial intelligence-based prediction technology has become a relevant approach. Data mining techniques are widely used to extract patterns from large data sets so that they can be used in the process of analyzing and predicting economic phenomena and food commodities [4]. Artificial Neural Networks (ANN) are known to be capable of modeling nonlinear relationships in time series data and generating predictions with a high degree of accuracy [5].

***) Corresponding Author**
Submitted : February 2, 2026
Accepted : March 15, 2026
Published : March 31, 2026

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

Compared to conventional methods such as ARIMA or moving average, ANN is superior in capturing complex and non-linear price fluctuation patterns, so that rice price predictions can be closer to actual values, especially when sudden changes occur due to demand or production cost factors. One of the most widely used ANN developments for sequential data is Long Short-Term Memory (LSTM), which has the ability to remember long-term patterns, making it suitable for predicting price data influenced by seasonal factors and trends. Previous studies have shown that the neural network approach provides better results than conventional methods in predicting commodity prices [6]. However, based on a review of previous studies, no research has been found that specifically applies the LSTM method to predict rice prices in East Sumba Regency. Most previous studies have focused on other regions or different types of commodities. Therefore, this study was conducted to fill this gap by utilizing historical rice price data obtained from the Waingapu Branch of Perum BULOG as the basis for developing a rice price prediction model in East Sumba Regency. This study differs from previous studies in terms of its research object, data source, and study area, namely medium-grade rice price data in East Sumba Regency obtained directly from Perum BULOG Waingapu Branch. The application of the LSTM method to regional rice price data is expected to provide a more specific picture of rice price patterns in the region.

Based on these issues, this study aims to develop and analyze a prediction model for medium-grade rice prices in East Sumba Regency using a Neural Network algorithm, specifically Long Short-Term Memory (LSTM), based on historical rice price data from the Waingapu Branch of Perum BULOG for the period January 2021 to December 2025. In addition, this study also aims to measure the accuracy level of the model using the Root Mean Square Error (RMSE) metric as an indicator of the reliability of the prediction results. RMSE is one of the evaluation measures commonly used in predictive modeling because it can show the magnitude of the deviation between the predicted value and the actual value on the same scale as the original data [7]. This research is expected to provide both practical and academic benefits. In practical terms, the prediction results can help farmers and traders in determining planting and harvesting times, as well as sales strategies to minimize the risk of losses due to price fluctuations. For local governments, the results of this study can be used as supporting information in formulating price stabilization policies. Academically, this study contributes to the application of LSTM algorithms for the analysis and prediction of food commodity prices based on historical data at the regional level.

2. MATERIALS AND METHODS

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an extension of Recurrent Neural Network (RNN) designed to overcome the vanishing gradient problem in sequential data. LSTM has a cell state as long-term memory that is controlled by three main gates, namely the input gate, forget gate, and output gate. This mechanism allows the model to store, update, and selectively output information so that it is able to learn complex and non-linear patterns in time series data. The ability to retain long-term information makes LSTM effective in time series forecasting tasks [8]. In this study, LSTM was used to model rice price fluctuation patterns based on historical data. The characteristics of rice prices, which are influenced by trends and seasonal factors, make the LSTM approach relevant for use. Previous studies have shown that the LSTM architecture can improve prediction accuracy in time series data with dynamic variations. Thus, this model is expected to produce more stable and representative price predictions to support analysis and decision-making processes.

Each component of LSTM plays a role in modeling rice prices. The input gate selects new information from historical data, the forget gate removes irrelevant information, and the output gate determines the resulting prediction. The cell state functions as long-term memory, maintaining patterns of fluctuation and seasonal trends in rice prices. With this mechanism, LSTM is able to produce stable and representative price predictions.

The formulas used in the LSTM model are shown in equations 1 to 6, as follows:

$$\text{Forget Gate: } f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right) \tag{1}$$

$$\text{Input Gate: } i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\text{Candidate Cell State: } \widetilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{3}$$

$$\text{Cell State Update: } C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{4}$$

$$\text{Output Gate: } o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$\text{Hidden State (Output LSTM): } h_t = o_t * \tanh(C_t) \tag{6}$$

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

Description:
x_t = input at time t
h_t = hidden state at time t
h_{t-1} = hidden state at time $t-1$
C_t = cell state at time t
C_{t-1} = cell state at time $t-1$
f_t = forget gate at time t
i_t = input gate at time t
o_t = output gate at time t
W_f, W_i, W_c, W_o = weight matrices for forget, input, cell, and output gates
b_f, b_i, b_c, b_o = bias vectors for forget, input, cell, and output gates
σ = sigmoid activation function
$tanh$ = hyperbolic tangent activation function

Method

This research method uses an experimental method with a machine learning approach as the main framework in the process of analyzing and creating rice price prediction models. This method was chosen because it is in line with the research objectives, which focus on the application of the Long Short-Term Memory (LSTM) algorithm to predict time series data. The stages of the research are shown in Figure 1.

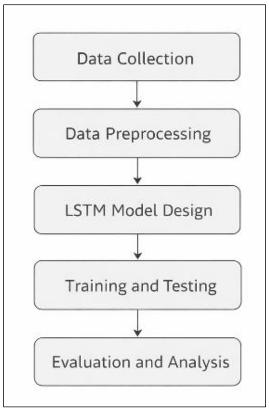

Figure 1. Research Stages

1. **Data Collection**
   At this stage, monthly rice price data for the past five years was collected from the Waingapu Branch of Perum BULOG. The data covers the period from January 2021 to December 2025, with data processing and analysis carried out after all data for 2025 is available on a monthly basis. This data was selected because it reflects rice price fluctuations in the East Sumba region, so it can be used as a basis for building a price prediction model.
2. **Data Preprocessing**
   Data obtained from sources is usually not ready to be used directly for model training, so a data pre-processing stage is necessary. This stage aims to clean, modify, and prepare the data so that it can be used optimally by machine learning algorithms. The pre-processing process includes data cleaning, removal of missing or duplicate values, and data type transformation to suit the needs of the analysis. In addition, the data is also arranged in a time series so that patterns of change can be analyzed more effectively. After this process is complete, the data is then normalized using the Min-Max Scaling technique to convert the data values into a specific range so that the model training process can run more stably and accurately [9].
3. **LSTM Model Design**
   Once the data was ready for use, the LSTM model was designed using the Python programming language, which was run through the Google Colab platform, a cloud-based computing service that allows users to run Python code online without the need to install software on their local computers [10]. At this stage, the model structure is determined, such as the number of layers, the number of neurons, the batch size, and the number of epochs used in the training process so that the model can learn the patterns of rice price changes optimally. Based on experiments, the final configuration of the LSTM model used consists of 2 LSTM layers, each with 50 neurons, followed by 1 Dense layer as the output. The model receives input with 3 times steps and 1 feature (rice price). Training is carried

out for 200 epochs with a batch size of 8, using the Adam optimizer and the Mean Squared Error (MSE) loss function. In addition, early stopping is applied with patience=10 to prevent overfitting and restore the best weights based on val_loss. This configuration allows the model to learn rice price fluctuation patterns optimally and produce stable predictions.

4. **Training and Testing**
The designed model was then trained using training data consisting of 80% of the total data, while the remaining 20% was used as test data to evaluate the model's ability to predict rice prices in the next period. The data was divided in such a way that each input in the test data had a time step of 3, i.e., the previous three periods, so that the model could form a sequential pattern suitable for prediction. After training was complete, the model was tested on the test data to assess the accuracy of the predictions and the ability to capture rice price fluctuations in the next period.

5. **Evaluation and Analysis**
The final stage is to evaluate the model's performance. Error measures such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are used to assess the level of prediction error. The prediction results are then compared with actual data to determine the extent to which the model successfully predicts rice prices. From this analysis, it can be determined whether the LSTM model has been able to adequately capture the patterns of rice price fluctuations in the East Sumba Regency.

3. RESULTS AND DISCUSSION

1. **Research Dataset**
The data used in this study is monthly medium-grade rice price data obtained from the Waingapu Branch Office of Perum BULOG. The data covers the period from January 2021 to December 2025, with a total of 60 observations. The data represents changes in rice prices in East Sumba Regency over the last five years, as shown in Table 1.

Table 1. Rice Price Dataset

No	Date (YYYY-MM)	Price (IDR/Kg)
1	2021-01	8.600
2	2021-02	8.600
..	…	…
12	2021-12	10.895
13	2022-01	10.250
14	2022-02	10.250
…	…	…
24	2022-12	11.498
25	2023-01	10.895
26	2023-02	10.895
..	…	…
36	2023-12	11.500
37	2024-01	10.250
38	2024-02	10.250
…	…	…
48	2024-12	11.300
49	2025-01	11.489
50	2025-02	11.489
..	…	…
60	2025-12	12.607

For the model training process, the data is divided into 80% training data and 20% test data. During training, some of the training data is also used as validation data to monitor model performance, especially when applying early stopping. In this way, the model can automatically stop training if the performance on the validation data does not improve, thereby preventing overfitting and ensuring more stable rice price predictions.

Rice price data is loaded from a Microsoft Excel file using the Python programming language, as shown in Figure 2.

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)


```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Input
from tensorflow.keras.callbacks import EarlyStopping


df = pd.read_excel('/content/Data_Harga_2021_2025.xlsx')

df['Tanggal'] = pd.to_datetime(df['Tanggal (YYYY-MM)'], format='%Y-%m')
df = df.sort_values('Tanggal')
df.set_index('Tanggal', inplace=True)

data = df[['Harga (Rp/Kg)']].values.astype(float)
```

Figure 2. Data Input Script

Next, the date column is converted into a time format and the data is sorted chronologically to ensure that the time series structure is arranged correctly. The date column is used as an index, while the price column is used as the main variable in the modeling process. This stage aims to ensure data consistency and order before entering the pre-processing stage and LSTM model training.

2. Data Preprocessing

Pre-processing is performed so that the data is ready to be used in LSTM model training. This stage includes: data normalization, division of training data and test data, and sequence data formation.

a. Data Normalization

The normalization process is performed so that all rice price values are on a scale of 0–1 using MinMaxScaler from sklearn preprocessing. Normalization is important so that the LSTM model is not biased towards large prices and can learn rice price patterns more stably. The Python script used to normalize the rice price data and create the df_norm dataframe displays a comparison between the original prices and the normalized prices, as well as the first 10 rows to check the transformation results, as shown in Figure 3.

```python
scaler = MinMaxScaler(feature_range=(0, 1))
data_scaled = scaler.fit_transform(data)

df_norm = pd.DataFrame({
    'Original Price (Rp/Kg)': data.flatten(),
    'Normalized Price': data_scaled.flatten()
}, index=df.index)

print("=== First 10 Rows of Normalized Data ===")
print(df_norm.head(10))
```

Figure 3. Data Normalization Script

Table 2 shows the first 10 rows of the normalized rice price data. It can be seen that some initial prices (8600) are normalized to 0 because they are the minimum value, while the price of 10895 is normalized to 0.5727. The vertical axis shows the normalized price value, while the horizontal axis shows the time period (month). With this normalization, the data is ready for the next stage, which is the division of training data and test data.

Table 2. First 10 Rows of Normalized Data

No	Date	Original Price (IDR/Kg)	Normalized Price
1	2021-01-01	8600.0	0.000000
2	2021-02-01	8600.0	0.000000
3	2021-03-01	8600.0	0.000000
4	2021-04-01	8600.0	0.000000
5	2021-05-01	10895.0	0.572748
6	2021-06-01	10895.0	0.572748
7	2021-07-01	10895.0	0.572748
8	2021-08-01	10895.0	0.572748
9	2021-09-01	10895.0	0.572748
10	2021-10-01	10895.0	0.572748

b. Training Data and Test Data Distribution

After the data has been normalized, the next step is to divide the data into 80% training data and 20% test data. The division is done proportionally so that the training data is used to train the LSTM model to recognize price change patterns, while the test data is used to evaluate the model's performance in predicting prices in the next period. Of the total 60 monthly data points, 48 are used as training data and 12 as test data, selected chronologically to maintain temporal patterns.

Figure 4 shows the Python script used to divide the training data and test data, also displays the total amount of data, training data, and test data to ensure that the division is correct.

```
time_step = 3

train_size = int(len(data_scaled) * 0.8)
train_data = data_scaled[:train_size]
test_data = data_scaled[train_size - time_step:]

print("\n=== Training and Testing Data Information ===")
print("Total data        :", len(data_scaled))
print("Training data     :", len(train_data))
print("Testing data      :", len(test_data))
```

Figure 4. Data Distribution Script

Table 3 shows the output information on the total amount of data: 60 training data and 12 test data. From the figure, it can be seen that the data distribution is appropriate, so that the model can learn from most of the data without losing the temporal pattern.

Table 3. Training and Testing Data Distribution

No	Data Type	Amount
1	Total Data	60
2	Training Data	48
3	Testing Data	12

c. Sequence Data Formation (Time Step)

The LSTM model requires input in the form of sequential data in order to learn the temporal relationships between periods. In this study, a time step of 3 was used, meaning that each input uses the prices from the previous three months to predict the price for the following month. After the sequence is formed, the data takes on a three-dimensional form (number of samples, sequence length, number of features) for X, and (number of samples) for y.

Figure 5 shows the Python script used to form the X_train, y_train, X_test, and y_test datasets, ready to be fed into the LSTM model.

```
def create_dataset(dataset, time_step):
    X, y = [], []
    for i in range(len(dataset) - time_step):
        X.append(dataset[i:i+time_step, 0])
        y.append(dataset[i+time_step, 0])
    return np.array(X), np.array(y)

X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

X_train = X_train.reshape(X_train.shape[0], time_step, 1)
X_test = X_test.reshape(X_test.shape[0], time_step, 1)

print("\n=== Shape of Sequence Data (Time Step) ===")
print("X_train :", X_train.shape)
print("y_train :", y_train.shape)
print("X_test  :", X_test.shape)
print("y_test  :", y_test.shape)
```

Figure 5. Sequence Data Formation Script

Figure 6 shows the sequence data formed by X_train (45, 3, 1), y_train (45), X_test (12, 3, 1), and y_test (12). The figure shows that the data is in three dimensions as required by LSTM and is ready for training.

```
=== Shape of Sequence Data (Time Step) ===
X_train : (45, 3, 1)
y_train : (45,)
X_test  : (12, 3, 1)
y_test  : (12,)
```

Figure 6. Sequence Data Formation Results

J-Icon : Jurnal Informatika dan Komputer
Vol. 14 No. 1, March 2026, pp. 9~20
DOI: 10.35508/jicon.v14.i1.27555

3. LSTM Model Design

After the pre-processing stage, the Long Short-Term Memory (LSTM) model was designed with two LSTM layers, each consisting of 50 neurons and one Dense layer as output. This architecture was used to capture temporal patterns in rice price data, which is time series data. The model received input in the form of a sequence of the previous three periods (time_step = 3) to predict prices in the next period. The parameters used in the model are summarized in Table 4.

Table 4. LSTM Model Parameters

No	Parameter	Value	Description
1	Input Shape	(3, 1)	Three time steps and one feature
2	LSTM Layer 1	50 units	With return_sequences=True
3	LSTM Layer 2	50 units	Second LSTM layer
4	Dense Layer	1 unit	Output layer
5	Optimizer	Adam	Optimization algorithm
6	Loss Function	Mean Squared Error	Used to minimize prediction error

Figure 7 shows the script for defining the LSTM model using the Keras Sequential API. The model is compiled using the Adam optimizer and the Mean Squared Error (MSE) loss function to minimize prediction errors during the training process. This configuration was chosen because it is suitable for continuous data prediction problems and is capable of producing stable learning.

```
model = Sequential([
    Input(shape=(time_step, 1)),
    LSTM(50, return_sequences=True),
    LSTM(50),
    Dense(1)
])

model.compile(
    optimizer='adam',
    loss='mean_squared_error'
)
```

Figure 7. LSTM Model Design Script

4. LSTM Model Training and Testing

After the LSTM model was designed, the next step was to train and test the model. The model was trained using training data to learn patterns of rice price changes based on previous time steps. To prevent overfitting and speed up training, an EarlyStopping mechanism was used, which would automatically stop training if the validation loss did not improve for several consecutive epochs. Training is conducted with 200 epochs and a batch size of 8, while validation is performed using test data. During training, the first 5 epochs of training loss and validation loss are displayed as an initial indication of how the model is beginning to learn from the data. A loss graph is also created to visualize the model's learning process in stages.

```
early_stop = EarlyStopping(
    monitor='val_loss',
    patience=10,
    restore_best_weights=True
)

history = model.fit(
    X_train, y_train,
    epochs=200,
    batch_size=8,
    validation_data=(X_test, y_test),
    callbacks=[early_stop],
    verbose=1
)


print("\n=== FIRST 5 EPOCHS OF TRAINING LOSS ===\n")
print("Training Loss   :", history.history['loss'][:5])
print("Validation Loss :", history.history['val_loss'][:5])
print("\n======\n")
```

Figure 8. Model Training Script

Figure 8 shows the LSTM model training script using EarlyStopping to stop training if there is no improvement in validation loss over several epochs. This script uses 200 epochs and a batch size of 8, and validation is performed using test data.

Table 5 shows the output results of the first 5 epochs of the model training process, in the form of training loss and validation loss values. These values provide an initial indication of the model's performance before it is evaluated on the test data.

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

Table 5. Training and Validation Loss for the First 5 Epochs

No	Epoch	Training Loss	Validation Loss
1	1	0.298208	0.577974
2	2	0.195398	0.357684
3	3	0.091213	0.144436
4	4	0.024919	0.016385
5	5	0.028953	0.008203

Figure 9 shows the script for creating a loss graph during training, which compares training loss and validation loss. This graph is used to visualize the model's learning process in stages.

```
plt.figure(figsize=(8,5))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss (MSE)')
plt.title('LSTM Model Training Loss Graph')
plt.legend()
plt.grid(True)
plt.show()
```

Figure 9. Loss Chart Script

Figure 10 shows the loss graph results from the model training process. It can be seen that the training and validation losses decrease as the number of epochs increases, indicating that the model is beginning to learn to recognize rice price patterns from historical data. This graph provides an initial indication of the model's performance before evaluation on the test data.

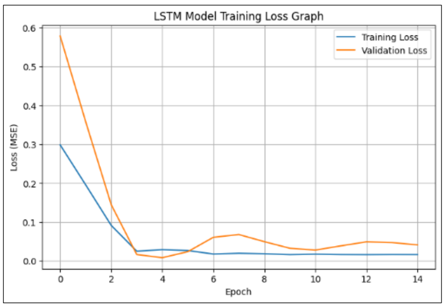

Figure 10. Model Training Loss Graph

5. Evaluation and Analysis of Results

The evaluation stage was conducted to determine how well the LSTM model could predict rice prices. At this stage, the model's predictions were compared with actual data using test data. The prediction error rate was measured using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). In addition, the predictions were visualized so that the differences between the actual data and the predictions could be observed directly.

Figure 11 shows the LSTM model evaluation script on the test data. At this stage, the model is used to make predictions on the test data, then the prediction results are returned to the original scale (inverse transform). Next, the MSE and RMSE values are calculated as measures of the model's prediction error rate.

```
test_predict = model.predict(X_test)
test_predict_inv = scaler.inverse_transform(test_predict)
y_test_inv = scaler.inverse_transform(y_test.reshape(-1, 1))

mse = mean_squared_error(y_test_inv, test_predict_inv)
rmse = math.sqrt(mse)

print("\n==== MSE AND RMSE VALUES ===\n")
print("MSE :", mse)
print("RMSE:", rmse)
print("\n======\n")
```

Figure 11. Model Evaluation Script

Table 6 shows the results of calculating the MSE and RMSE values of the LSTM model predictions on the test data. The RMSE value shows the average prediction error of the model in terms of rice price (Rp/kg). An RMSE value of 360.91 Rp/Kg indicates that the average difference

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

between the actual price and the model prediction is in that range. When compared to the average rice price of around 10,964 Rp/Kg, the model prediction error rate is around 3.35%, so it can be said that the LSTM model has a fairly good level of accuracy.

Table 6. Model Evaluation Results (MSE and RMSE)

No	Metric	Value
1	MSE	131699.730202
2	RMSE	360.904575

Figure 12 shows the Python script used to visualize the LSTM model prediction results on the test data. In this script, the test data prediction results are arranged so that they only appear in the test data period, while the training data period is left empty (NaN). The purpose is to display the model evaluation specifically on the test data.

```
pred_test_plot = np.full((len(data), 1), np.nan)
pred_test_plot[train_size:] = test_predict_inv

plt.figure(figsize=(10,6))
plt.plot(df.index, data, label='Actual Data', linewidth=2)
plt.plot(df.index, pred_test_plot, label='LSTM Prediction Results (Test Data)', linewidth=2)
plt.title('Evaluation of LSTM Prediction Results on Test Data')
plt.xlabel('Year')
plt.ylabel('Rice Price (Rp/Kg)')
plt.legend()
plt.grid(True)
plt.show()
```

Figure 12. Test Data Prediction Visualization Script

The graph in Figure 13 shows a comparison between actual rice price data and LSTM model predictions during the test data period. The blue line represents actual rice price data, while the prediction line only appears in the test data section in accordance with the model evaluation objectives. The prediction line appears shorter because it is only displayed in the test data period (the last 20% of the data). The model does not predict the training data, but only data that has never been seen before to measure the model's generalization ability. Based on the graph, it can be seen that the LSTM model's prediction results are able to follow the pattern of rice price movements in the test data period. Although there are still differences in values at several points, in general, the model successfully captures the trend of rice price changes. This shows that the LSTM model has a fairly good ability to learn the pattern of rice price time series data.

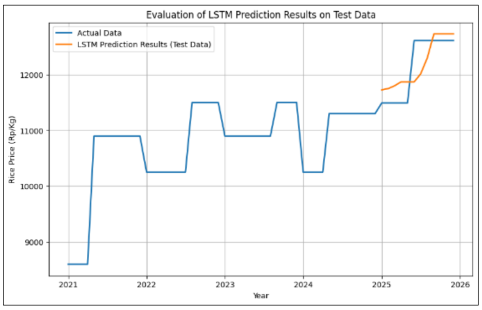

Figure 13. Comparison Chart of Actual Data and Test Data Predictions

Figure 14 shows the Python script used to visualize the results of rice price predictions using the LSTM model on all data, namely a combination of training data and test data. At this stage, the prediction results are displayed comprehensively to see the model's ability to follow rice price patterns from the beginning to the end of the observation period.

```
all_predictions = []

for i in range(len(data_scaled) - time_step):
    x_input = data_scaled[i:i+time_step].reshape(1, time_step, 1)
    pred = model.predict(x_input, verbose=0)
    all_predictions.append(pred[0, 0])

all_predictions = np.array(all_predictions).reshape(-1, 1)
all_predictions_inv = scaler.inverse_transform(all_predictions)

pred_long_plot = np.full((len(data), 1), np.nan)
pred_long_plot[time_step:] = all_predictions_inv

plt.figure(figsize=(10,6))
plt.plot(df.index, data, label='Actual Data', linewidth=2)
plt.plot(df.index, pred_long_plot, label='LSTM Prediction Results', linewidth=2)
plt.title('LSTM Prediction on the Entire Rice Price Dataset')
plt.xlabel('Year')
plt.ylabel('Rice Price (Rp/Kg)')
plt.legend()
plt.grid(True)
plt.show()
```

Figure 14. Overall Data Prediction Visualization Script

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

The graph in Figure 15 shows a comparison between actual rice price data and the LSTM model's predictions for the entire data period. The blue line shows the actual rice price data, while the prediction line shows the LSTM model's estimates of the overall rice price pattern. Based on this graph, it can be seen that the LSTM model's prediction results are generally able to follow the trends and patterns of rice price changes throughout the observation period. The prediction line appears to be close to the actual data, although there are still differences in value at some points. This visualization is used to provide a qualitative overview of the model's performance in studying the overall rice price data pattern. In predicting the overall rice price data, RMSE and MSE values were not calculated because this graph was not used as a quantitative evaluation of the model. The model's accuracy was only evaluated on the test data, while the overall data prediction aimed to show the model's ability to represent rice price patterns visually.

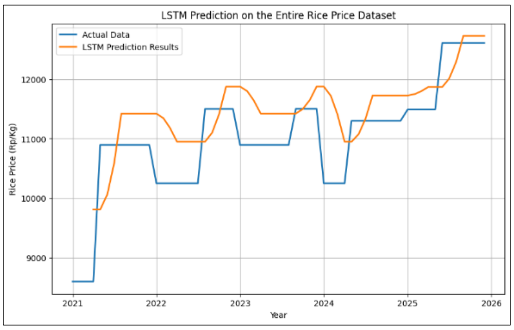

Figure 15. Overall Rice Price Prediction Chart Data

6. Analysis of Rice Price Prediction Results

Based on the prediction results, the LSTM model is able to follow the patterns and trends of rice price changes quite well. In the test data prediction graph, the prediction line appears close to the actual data line, indicating that the model has successfully learned the temporal relationship of rice price data. Although there are differences in certain periods, these differences are still within reasonable limits and do not significantly change the price movement pattern. To display the prediction results in a structured manner, a DataFrame was created containing the date, actual price, and predicted price for the test data period.

Figure 16 shows the process of creating a DataFrame of prediction results. The prediction data, which has been returned to its original scale (inverse transform), is then combined with the date and actual price in the test data period. This step aims to facilitate the analysis and comparison of actual values with predicted values in rupiah per kilogram.

```
df_hasil = pd.DataFrame({
    "Date": df.index[-len(test_predict_inv):],
    "Actual Price": y_test_inv.flatten(),
    "Predicted Price": test_predict_inv.flatten()

})

print(df_hasil)
```

Figure 16. Script for Creating Prediction Results Data

Table 7 shows the results of rice price predictions for the period January 2025 to December 2025, which consists of 12 test data points. It can be seen that the predicted values differ from the actual prices in several months. For example, in January 2025, the actual price was 11,489 rupiah/kg, while the predicted price was 11,676 rupiah/kg. In the middle of the year, when the actual price increased to 12,607 rupiah/kg, the model also showed an increase in the predicted value to approach that figure.

Table 7. Actual and Predicted Rice Prices

No	Date	Actual Price (IDR/Kg)	Predicted Price (IDR/Kg)
1	2025-01-01	11489.0	11676.864258
2	2025-02-01	11489.0	11700.339844
3	2025-03-01	11489.0	11746.462891
4	2025-04-01	11489.0	11813.256836
5	2025-05-01	11489.0	11813.256836
6	2025-06-01	12607.0	11813.256836

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

7	2025-07-01	12607.0	11952.722656
8	2025-08-01	12607.0	12229.168945
9	2025-09-01	12607.0	12634.250000
10	2025-10-01	12607.0	12634.250000
11	2025-11-01	12607.0	12634.250000
12	2025-12-01	12607.0	12634.250000

4. CONCLUSION AND RECOMMENDATIONS

Based on the evaluation results of the test data, the LSTM model produced a Root Mean Squared Error (RMSE) value of 360.91 IDR/kg. When compared to the average rice price of 10,946 IDR/kg, the model's prediction error rate was around 3.35% (Relative RMSE). This indicates that the average prediction error is relatively small compared to the price scale, so that the LSTM model performs well in capturing rice price fluctuations in East Sumba Regency.

In addition, the results of rice price predictions based on the overall data show that the LSTM model is able to follow the patterns and trends of rice price movements over time. Although there are still differences in values in some periods, in general, the model is able to capture the direction of price changes well. Thus, the resulting prediction model can be used as an analytical tool and as a basis for decision-making for farmers and traders in relation to rice price planning and strategy.

Based on the results of the research that has been conducted, further development is recommended using rice price data with a longer period or more detailed frequency, such as weekly or daily data, so that the model is able to capture seasonal patterns and price fluctuation dynamics more optimally. In addition, subsequent research could compare methods with other algorithms, such as Gated Recurrent Unit (GRU) or other machine learning approaches, to obtain models with better prediction performance. The prediction results are also expected to be utilized by farmers, traders, and local governments as consideration in production planning, distribution, and policy-making related to rice price stability in East Sumba Regency.

REFERENCES

[1] L. Habibah, A. Futri, A. P. Khuzaeri, F. Shidqi, W. A. Winata, and D. Desmawan, "Beras Sebagai Makanan Pokok: Faktor Penyebab Ketergantungan Dan Dampaknya Terhadap Perekonomian Indonesia," *Bursa: Jurnal Ekonomi dan Bisnis*, vol. 3, no. 2, pp. 110–114, 2024, doi: 10.24815/jped.v6i2.15000.

[2] A. Amila, Y. Erlina, and M. Masliani, "Pengaruh Pengadaan, Operasi Pasar Dan Jenis Beras Oleh Perum BULOG Terhadap Stabilitas Harga," *J-SEA (Journal Socio Economic Agricultural)*, vol. 20, no. 2, pp. 267–276, 2025, Accessed: Dec. 11, 2025. [Online]. Available: https://doi.org/10.52850/j-sea.v20i2.20498

[3] R. Hidayat and I. Wibisonya, "Rice Price Prediction with Long Short-Term Memory (LSTM) Neural Network," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 5, pp. 658–664, 2024, Accessed: Dec. 11, 2025. [Online]. Available: https://jurnal.iaii.or.id/index.php/RESTI/article/view/6041

[4] Q. Yu, B. A. Tolson, H. Shen, M. Han, J. Mai, and J. Lin, "Enhancing Long Short-Term Memory (LSTM)-Based Streamflow Prediction with A Spatially Distributed Approach," *Hydrol. Earth Syst. Sci.*, vol. 28, no. 9, pp. 2107–2122, May 2024, doi: 10.5194/hess-28-2107-2024.

[5] X. Kong *et al.*, "Deep Learning for Time Series Forecasting: A Survey," *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 7–8, pp. 5079–5112, Aug. 2025, doi: 10.1007/s13042-025-02560-w.

[6] M. A. Setyadji, A. Faqih, and Y. A. Wijaya, "Peramalan Harga Komoditas Beras di Kalimantan Timur Menggunakan Algoritma Neural Network," *Jurnal Mahasiswa Teknik Informatika*, vol. 7, no. 1, 2023, Accessed: Dec. 11, 2025. [Online]. Available: https://www.ejournal.itn.ac.id/index.php/jati/article/view/6327

[7] T. O. Hodson, "Root-Mean-Square Error (RMSE) Or Mean Absolute Error (MAE): When To Use Them Or Not," Jul. 19, 2022, *Copernicus GmbH*. doi: 10.5194/gmd-15-5481-2022.

[8] C. Sun, M. Pei, B. Cao, S. Chang, and H. Si, "A Study on Agricultural Commodity Price Prediction Model Based on Secondary Decomposition And Long Short-Term Memory Network," *Agriculture*, vol. 14, no. 1, p. 60, Jan. 2024, doi: 10.3390/agriculture14010060.

[9] M. A. Rahman, M. S. Hossain, and M. A. Islam, "A Comparative Study of Data Normalization Techniques for Machine Learning Applications," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 733–739, 2021.

ISSN: 2337-7631 (Printed)
ISSN: 2654-4091 (Online)

[10] S. Octarina, F. M. Puspita, E. Yuliza, and I. Indrawati, “Pendampingan Penggunaan Google Colab Pada Pembelajaran Python Dan Machine Learning Bagi Dosen Matematika Di Palembang,” *Jurnal Pepadu*, vol. 6, no. 1, pp. 56–66, Mar. 2025, doi: 10.29303/pepadu.v6i1.6457.