Electronic Journal of Education, Social Economics and Technology
Vol. 6, No. 2, (2025), pp. 1~12, Article ID: 1323
ISSN 2723-6250 (online)
DOI: https://doi.org/10.33122/ejeset.v6i2.1323

Research Article

Comparative Performance Evaluation of ARIMA, SARIMA, and LSTM for Daily Shallot Price Forecasting in Palembang City

Fatreisya Ayu Miranda, Ken Ditha Tania*, Rizka Dhini Kurnia

Department of Information Systems, Universitas Sriwijaya, Sumatera Selatan, Indonesia, 30139
*Corresponding Author: kenya.tania@gmail.com | Phone: +6289677981433

ABSTRACT

Shallots are a food commodity that often experiences price fluctuations and is one of the contributors to inflation in the city of Palembang. This study compares the ARIMA, SARIMA, and LSTM methods in predicting shallot prices using daily data start from January 2020 to October 2025. The Data of shallot price were obtained through the official website of Bank Indonesia. The stages of the study included data collection, pre-processing, visualization and decomposition, split data, modeling, and performance evaluation using the RMSE, MAE, and MAPE metrics. Model performance assessment reveals that ARIMA(1,1,1) method provided the most optimal performance with the lowest error value in comparison with the remaining two other methods, namely SARIMA and LSTM. The SARIMA(1,1,1)(2,1,1)7 model produced a slightly higher error rate, although its performance remains superior than LSTM model. The LSTM method produced the highest error in this study. These findings indicate that the pattern of shallot price data in Palembang tends to follow linear and seasonal trends that are not too complex, so that classical statistical approaches are still superior to deep learning models in capturing these data patterns. This research provides practical contributions as a decision-making support tool for the government and business actors in planning the distribution and stabilization of shallot prices in Palembang City.

Keywords: Shallot Price; ARIMA; SARIMA; LSTM; Time Series Prediction

1. INTRODUCTION

Shallot (*Allium ascalonium* L.) is a horticultural commodity that plays a significant role in national economic activity and in meeting household consumption needs. As an essential culinary ingredient, shallot is widely used and highly consumed by the Indonesian population. National average consumption of shallots remains relatively high, reaching 0,350 kg per capita per week in 2024 (Badan Pusat Statistik Nasional, 2025). Despite strong and relatively stable demand, shallot prices in Indonesia, particularly in Palembang City, tend to fluctuate unpredictably. In 2024, shallot prices in Palembang City experienced a sharp increase during April and May (Bank Indonesia, n.d.). Prices were recorded at Rp38.000/kg in March, before rising to Rp45.000 per kilogram in April. The increase continued in May, with prices reaching a peak of Rp57.000/kg.

Extreme price volatility is driven by several factors, including extreme weather conditions, plant pathogen threats, government regulations, and market transaction dynamics (Triyadi et al., 2024). Furthermore, an imbalance between shallot supply and high market demand can also increase the risk of price fluctuations (Anees et al., 2022). These price changes have significant impacts on farmers, traders, and business actors. For farmers, substantial price declines during the harvest period pose a high risk of financial loss. Conversely, sharp increases in shallot prices lead to a reduction in consumer purchasing power (Al Rosyid et al., 2021). Therefore, the development of an accurate shallot price prediction model is essential to anticipate future price movements and to support informed decision-making.

To dealing with the unpredictable nature of food commodity prices, which commonly exhibit trend and seasonal patterns, an appropriate analytical approach is required. Several forecasting approaches have been applied in time series data, ranging from classical statistical methods to methods with a deep learning approach. The Autoregressive Integrated Moving Average (ARIMA) is known as a statistical approach commonly implemented to predict non-seasonal time series data. Meanwhile, Seasonal Autoregressive Integrated Moving Average (SARIMA) was developed to capture seasonal patterns that often appear in commodity price data (Santoso & Widodo, 2024). As a more recent alternative, the Long Short-Term Memory (LSTM) method employs artificial neural network architectures to capture non-linear patterns and long-term temporal dependencies (Fauzi et al., 2024). These three methods have been implemented in numerous studies to forecast time series data with varying levels of predictive accuracy.

For instance, a research undertaken by Sirisha et al. (2022) compared the ARIMA, SARIMA, and LSTM methods for predicting a company's gross profit. The study employed several evaluation metrics, including RMSE, MAE, and MAPE. The results shows that LSTM model attained the highest level of predictive accuracy at 97.01%, followed by the SARIMA model at 94.38%, and the ARIMA model at 93.84%. Based on these findings, the LSTM model was recommended for profit forecasting. Nevertheless, when computational resources are limited or when the seasonal patterns in the data are relatively simple, the SARIMA model can serve as an effective alternative, as it demonstrates a level of accuracy comparable to the LSTM model. Furthermore, a study by Prasojo & Muludi (2025) compared the performance of the LSTM and SARIMA to forecasting passenger density at airports. The findings reveals that SARIMA model attained superior performance, yielding error values of RMSE = 152.35, MAE = 124.37, and MAPE = 3.81%. In comparison, the LSTM model produced an RMSE = 23.47, an MAE = 228.70, and a MAPE = 6.81%. The results imply that the SARIMA model performs more effectively in forecasting long-term time series data characterized by strong seasonality. Meanwhile, the LSTM method is the preferred choice for responding to sudden changes.

Although numerous previous studies have applied and compared various time series methods to forecast commodity prices, most of these studies are limited to comparisons between two methods and are conducted on different commodities and in different regions. Based on the existing literature, studies that simultaneously compare the ARIMA, SARIMA, and LSTM methods for predicting shallot prices in Palembang City remain limited. Therefore, this study contributes novelty by performing a comparative analysis of three forecasting methods, including ARIMA, SARIMA, and LSTM, to predict shallot prices in Palembang City. The analysis is carried out by evaluating the predictive accuracy of each model using error metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). This study aims to determine the method that delivers the most optimal predictive performance for shallot prices in Palembang City based on objective evaluation results.

2. RESEARCH METHOD

This research process is visualized through a flowchart diagram that shows **Figure 1**.

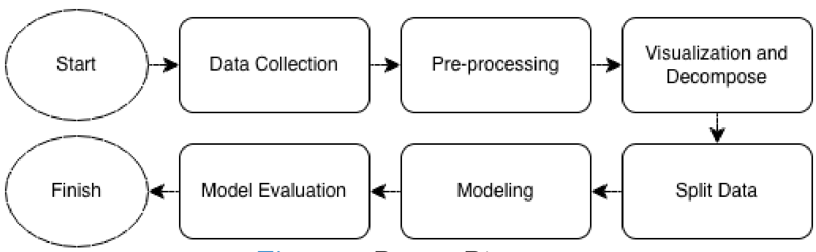

Figure 1. Process Diagram

2.1 Data Collection

The dataset analyzed in this research consist of daily shallot price data in Palembang City covering the period from January 2020 to October 2025. Palembang City was selected as the study area due to its strategic position as capital of South Sumatra, a province with a relatively high average consumption of shallots. In addition, shallots were identified as the largest contributor to inflation in South Sumatra in 2024 (Badan Pusat Statistik Provinsi Sumatera Selatan, 2025). This indicates that fluctuations in shallot prices in Palembang City have a significant role in affecting food price stability and the broader regional economy. The Data of shallot price were obtained through the official website of Bank Indonesia. This data source was selected due to the credibility of Bank Indonesia as the national monetary authority that provides reliable indicators of food commodity prices.

2.2 Data Preprocessing

Before analysis, the data are subjected to a preprocessing stage consisting of data type transformation and data cleaning. The data cleaning process is carried out to address missing values with the aim of enhance the accuracy and validity of the dataset (Koukaras & Tjortjis, 2025). After ensuring that the data are free from missing values, a data transformation process is applied by converting the data format into a datetime data type. This transformation is essential to enable the system to accurately identify the chronological order of the data, thereby facilitating subsequent data visualization and modeling processes.

2.3 Data Visualization and Decompose

After completing the preprocessing stage, the data are visualized in the form of a time series plot. This step aims to identify the initial characteristics and dominant patterns present in the data. Subsequently, data decomposition is conducted to decompose the time series into three components, including trend, seasonality, and residual components.

2.4 Data Split

The next step involves allocating the data into training and testing subsets with ratio of 80:20. Model learning is conducted using the training data, while performance evaluation is carried out using the testing data (MY et al., 2024). This ratio is commonly applied in previous studies. For instance, Putra & Harahap (2025) used an 80% proportion for training data and 20% for testing data when applying the LSTM algorithm to monitor oil palm productivity. The allocation of the dataset with a proportion of 80% for training data and 20% for test data was also applied by Novalia et al. (2024) in their study that examined CNN–LSTM model performance in application review sentiment analysis based on word embedding techniques.

2.5 Modeling

The main stages in this research were the implementation of models using three forecasting methods, namely Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA), and Long Short-Term Memory (LSTM). The application of these three methods enabled researchers to evaluate which model is the most accurate in predicting shallot prices in Palembang City.

2.5.1 ARIMA Modeling

The ARIMA framework comprises three fundamental components, namely the autoregressive (AR) component denoted by p, the differencing (I) component denoted by d, and the moving average (MA) component denoted by q (Kumar Dubey et al., 2021). The autoregressive (AR) component utilizes past values to generate predicted values. The differencing (I) component is needed if the data used is not stationary. Meanwhile, the moving average (MA) component improves the prediction results based on past error values (Ning et al., 2022). ARIMA modeling is conducted through several stages. The first stage is the stationarity test using Augmented Dickey-Fuller (ADF) to determine whether the shallot price data in Palembang City is stationary. The data are considered stationary if the p-value < 0.05 and the test statistic < the critical values at the 1%, 5%, and 10% significance levels (Alawiyah et al., 2024). If the test results indicate that the data are non stationary, then the differencing process needs to be carried out until the data becomes stationary (Alabdulrazzaq et al., 2021). The second stage involves determining the appropriate ARIMA model by analyzing the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots. The determined model then goes through the parameter significance test and Ljung Box test to assess the suitability of the data. The data is considered significant if the value of $P> |z| < 0.05$ (Febiola et al., 2024). Finally, the forecasting process is performed using the selected ARIMA model.

2.5.2 SARIMA Modeling

The SARIMA method, is a conceptual addition to the existing ARIMA model. The SARIMA method complements the ARIMA method by incorporating seasonal components (Ruhiat & Effendi, 2018). These seasonal components consist of P (seasonal autoregressive order), D (seasonal difference order), and Q (seasonal moving average order). SARIMA is commonly formulated as SARIMA(p,d,q)(P,D,Q)s, in which (p,d,q) is the non-seasonal component, (P,D,Q) represent seasonal component, and and s refers to the length of the seasonal cycle (Cintani et al., 2025). The first step of the SARIMA approach involves observing seasonal patterns in the data. Stationarity is then examined through the Augmented Dickey–Fuller (ADF) test. Then, SARIMA model candidates are identified through examination of the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) graphs. These model candidates then undergo significance testing and Ljung Box testing. Finally, predictions are made for shallot prices using the selected SARIMA model.

2.5.3 LSTM Modeling

The Long Short Term Memory (LSTM) model is an extension of the Recurrent Neural Network (RNN) architecture designed to process sequential or ordered data, enabling the model to learn temporal and spatial relationships in the data (Siswanto et al., 2025). The LSTM model comprises three main gating units, namely forget gate, input gate, and output gate (Zahara & Sugianto, 2021). Forget gate plays a role in regulating the information that must be deleted, the input gate plays a role in selecting the information to be stored, and the output gate plays a role in controlling the data to be released from the memory cell (Bahri & Tania, 2025). The first stage of the LSTM model in this study is normalization using MinMaxScaler. Normalization is performed to convert the data scale to between 0 and 1. This stage is important because it can help the model learn objectively without being influenced by unit differences, resulting in more consistent and bias-free predictions (Sofiah et al., 2024). Next, the data is transformed into input–output pairs using the sliding window technique. Then, all data is converted into numpy array format to suit the LSTM model requirements. The next step is to design an LSTM model to make predictions on shallot data. Finally, the prediction results are visualized in graph form.

2.6 Model Evaluation

Model evaluation is performed by comparing several error metrics, namely Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE). These three metrics were chosen as comparative analysis indicators because they can provide a comprehensive assessment of model performance. These metrics primarily serve to assess prediction accuracy by comparing estimated values with observed outcomes (Caesar et al., 2025). Models with the lowest MAPE, MAE, and RMSE values are considered to have better performance (Zhou & Wang, 2021). Root Mean Squared Error (RMSE) is an assessment tool that measures the average error of model predictions. This metric is obtained by taking the square root of the average difference between the prediction results and the actual data (Madhika et al., 2023). The RMSE value is calculated using **Equation 1**. The symbol X_i represents the actual value in observation i, while $\widehat{X}_i$ represents the predicted value in observation i. In addition, the symbol n represents the total data value.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(X_i - \hat{X}_i\right)^2}{n}}$$

Equation 1. RMSE

Mean Absolute Error (MAE) is an evaluation metric to determine the closeness between the predicted number and the actual data (Faisal et al., 2022). The MAE value is calculated using **Equation 2**.

$$\text{MAE} = \frac{\sum_{i=1}^{n}\left|X_i - \hat{X}_i\right|}{n}$$

Equation 2. MAE

Mean Absolute Percentage Error (MAPE) is a commonly used assessment metric to measure the accuracy of a forecasting model. The MAPE value can be calculated by calculating the mean difference between the predicted values and the actual data, then dividing it by the original data (Faisal et al., 2022). The outcome of the calculation is in percentage form. The MAPE value is calculated using **Equation 3**.

$$\text{MAPE} = \frac{\sum_{i=1}^{n}\frac{\left|X_i - \hat{X}_i\right|}{X_i} \times 100}{n}$$

Equation 3. MAPE

3. RESULTS AND DISCUSSION

3.1 Data Collection

The data used in this study consist of daily shallot price observations in Palembang City covering the period from January 2020 to October 2025, with a total of 1505 data points. However, the dataset was not initially suitable for direct analysis, and therefore data preprocessing was required. As shown in **Table 1**, the data format did not conform to the standard data structure necessary for the analysis process. In addition, several missing values were identified in the date and price columns.

Table 1. Daily Red Onion Price Dataset

No	Date	Price
1.	01/01/2020	-
2.	02/01/2020	40,750
3.	03/01/2020	39,500
…	…	…
1503.	03/10/2025	38,500
1504.	06/10/2025	38,500
1505.	07/10/2025	38,500

3.2 Preprocessing

Before analysis, the data first underwent a process of cleaning and transformation. Missing values in the data were removed during the cleaning process. In addition, the data in the date column was transformed from string objects to datetime objects to facilitate the modeling process. The dataset after the cleaning and transformation stages is presented in **Table 2**.

Table 2. Daily Shallot Price Dataset After Preprocessing

No	Date	Price
1.	2020-01-02	40750
2.	2020-01-03	39500
3.	2020-01-06	39250
…	…	…
1503.	2025-10-03	38500
1504.	2025-10-06	38500
1505.	2025-10-07	38500

3.3 Data Visualization and Decompose

Figure 2 shows the visualization and decomposition of the shallot price dataset in Palembang City from 2020 to 2025. Through this graph, irregular fluctuations can be observed, with several sharp spikes occurring during certain periods. This condition indicates a lack of stationarity in the dataset. Trend component in the graph also shows price increases in certain months in 2022 and 2024. In addition, the seasonal component shows a pattern that repeats regularly on a weekly basis. On the other hand, the residual component shows no particular pattern. The information obtained from the results of this visualization and decomposition is important because it can be used to consider whether the data needs to be processed or changed before being applied in modeling (Menteng & Rozi, 2025).

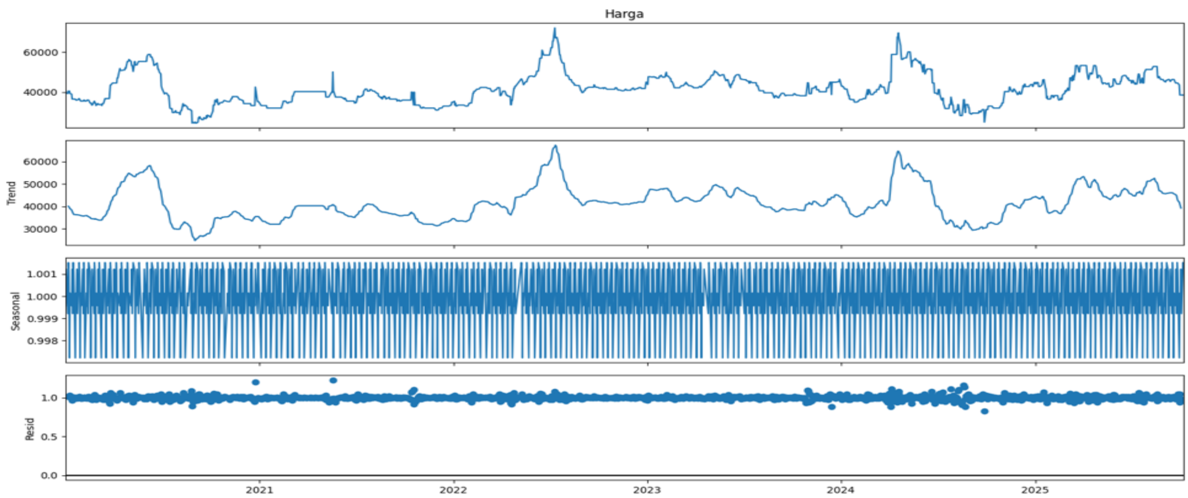

Figure 2. Visualization and Decomposition Results Chart

3.4 Data Split

In this process, the data are allocated into training data and testing data. A total of 1152 data points, or 80% of the total shallot price data, is used as training data. Meanwhile, 288 data points, or 20% of the total shallot data, is used as testing data.

3.5 Modeling

3.5.1 ARIMA Modeling

Stationarity testing was performed using ADF and ACF & PACF plots. The ADF test was used to detect the presence of unit roots, while ACF and PACF graphs were used as diagnostic tools to observe autocorrelation patterns in the data. If one of the instruments showed output indicating that the data was not stationary, differentiation was performed on the dataset. The ADF test in **Table 3** produced a test statistic value of -3.65481. This value is smaller than the critical values at the 1%, 5%, and 10% levels, which are -3.43492, -2.86356, and -2.56785, respectively. Then, the p-value produced is 0,00480. This value is smaller than 0,05, so the data meets the stationarity criteria based on the ADF test.

Table 3. ADF Test Results

Test Component	Value
Test Statistic	-3.65481
P-Value	0.00480
Critical Value (1%)	-3.43492
Critical Value (5%)	-2.86356
Critical Value (10%)	-2.56785

However, to validate the ADF test results, observations were made on the ACF and PACF graphs. **Figure 3** shows the ACF and PACF graphs of the shallot price dataset.The PACF plot exhibits a significant spike at lag 1, followed by a rapid decline at subsequent lags. Meanwhile, the ACF graph shows a slow decay. This is due to the data having a strong relationship with previous values. Although the ADF test results show that the data is stationary, visually, the slowly declining ACF graph indicates that the data is not yet fully stationary. As explained by (Shofwatillah, 2025), the slow decline pattern in the ACF graph is a sign that the data is not yet stationary in terms of average. This pattern reflects the existence of long-term correlations in the dataset. Therefore, first-order differentiation of the dataset is necessary to eliminate the effects of long-term autocorrelation.

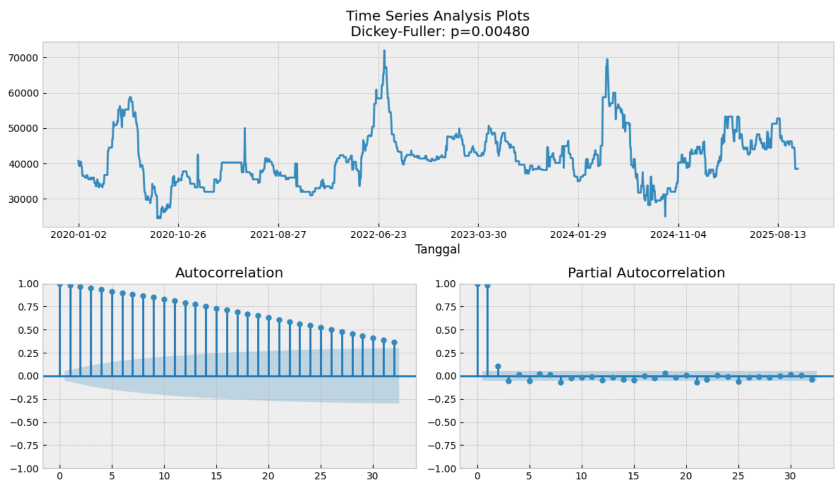

Figure 3. ACF and PACF Graphs

Figure 4 provides an overview of the analysis results after first-order differencing. After differentiation, the p-value obtained was 0.0000. This value is far below 0.05, indicating that the data is stationary in mean. In addition, the ACF graph, which previously experienced slow decay, now has a clear cut-off. This indicates that the long-term autocorrelation effect has disappeared. Therefore, it can be determined that after differentiation, the data meets the stationarity criteria and can be further processed in the next phase.

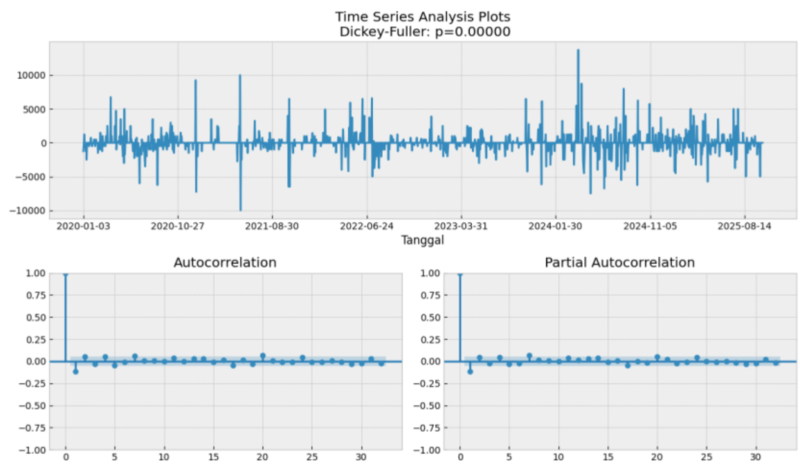

Figure 4. ACF and PACF Graphs After Differentiation

Afterward, to determine the non-seasonal components, observations were made on the ACF and PACF graphs. The moving average (MA) component was identified through analysis of the ACF plot, whereas the autoregressive (AR) component was determined using the PACF plot (M et al., 2024). The analysis of the ACF and PACF plots led to the selection of the ARIMA(1,1,1) model for further analysis. The AR component is obtained from the lag that passes the critical limit on the PACF graph. The d component is derived from a single differentiation. Meanwhile, the MA component is obtained from the lag that passes the critical limit on the ACF graph. Next, a significance test of the parameters in the previously determined ARIMA(1,1,1) model is conducted. The test results are shown in **Table 4**.

Table 4. ARIMA Model Significance Test Results

Variable	$P>\|z\|$	Description	Result
ar.L1	0.002	Significant	Meets the test significance
ma.L1	0.013	Significant	
sigma2	0.000	Significant	

The evaluation of the significance test outcomes shows that ARIMA (1,1,1) model passed the significance test with a value of P> |z| < 0.05. The Ljung Box test results also revealed a p-value > 0.05, specifically 0.51. These findings reveals that the residuals are found to be randomly distributed. and that the model is appropriate for forecasting purposes. After determining the non seasonal components, the forecasting stage was carried out to evaluate the model's predictive performance. The forecasting results obtained using the ARIMA(1,1,1) model are presented in **Figure 5**, where the blue line represents the expected prices and the orange line represents predicted values generated by the ARIMA(1,1,1) model.

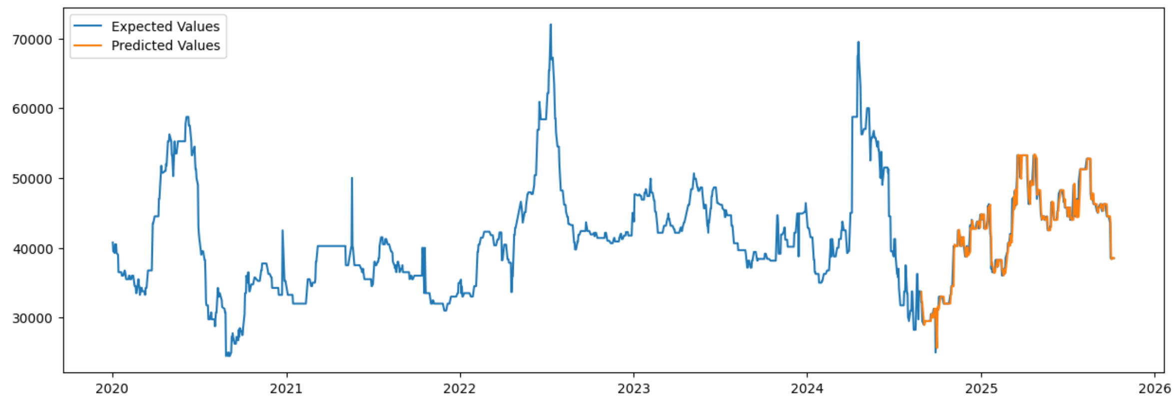

Figure 5. Actual and Predicted Graphs of ARIMA Model

3.5.2 SARIMA Modeling

The ARIMA method is the basis for the development of the SARIMA method. The SARIMA method adds seasonal components to the ARIMA method. The non-seasonal components previously built into the ARIMA method will become the baseline for the SARIMA method. After determining the non-seasonal component, the subsequent step focuses on determing the seasonal component (P, D, Q)s. The initial stage of SARIMA modeling involves examining the decomposition results of shallot prices in Palembang City before differencing, with the aim of identifying trend, seasonal, and residual components. The results of the decomposition of the shallot price data previously shown in **Figure 2** show that the seasonal composition has a clear pattern of fluctuations over a weekly period. In addition, the residual component is still not completely random. Therefore, to handle the seasonal pattern, differentiation is performed at lag 7. The next step is to test stationarity using ADF and ACF & PACF graphs. The ADF test results based on **Table 5** show a p-value < 0.05, specifically 0.00000. Furthermore, the results show that the test statistic value is smaller than the critical value at significance levels of 1%, 5%, and 10%, which are -3.43499, -2.86359, and -2.56786, respectively. Therefore, based on the ADF test, it can be stated that the data meets the stationarity criteria.

Table 5. ADF Test Result

Test Component	Value
Test Statistic	-12.99347
P-Value	0.00000
Critical Value (1%)	-3.43499
Critical Value (5%)	-2.86359
Critical Value (10%)	-2.56786

This is reinforced by the ACF & PACF graphs in **Figure 6**, which have a clear cut-off pattern and no slow decay, indicating stasionarity in terms of the mean level. After the stationarity test, the next step is to determine the SARIMA model with the notation SARIMA(p,d,q) (P,D,Q)s. The selection of an appropriate SARIMA model relies on information obtained from the ACF and PACF plot. The ACF graph shows that lag 7 exceeds the critical limit, so the value of Q = 1.

Meanwhile, the PACF graph shows that lag 7, lag 14, lag 21, and lag 28 exceed the critical limit, so the seasonal AR order candidates are in the range of P = 1 to P = 4. Previously, seasonal differentiation was performed once, so D = 1. The decomposition results also show a seasonal pattern in the weekly period, so s = 7. Next, a parameter significance test is performed on several SARIMA model candidates to determine which model meets the significance test.

Figure 6. ACF and PACF Graphs After Lag 7 Differentiation

According to the significance test results in **Table 6**, it was found that the SARIMA (1,1,1)(2,1,1)_7 and SARIMA (1,1,1)(1,1,1)_7 models passed the significance test with *P*> |z| < 0.05. However, the *Ljung-Box* test on the SARIMA (1,1,1)(1,1,1)_7 model produced a *p-value* < 0.05, specifically 0.00. Meanwhile, the *Ljung Box* test on the SARIMA (1,1,1)(2,1,1)_7 model produced a *p-value* > 0.05, specifically 0.31. This indicates that the residuals are random and the model is suitable for use in prediction. Therefore, the SARIMA (1,1,1)(2,1,1)_7 model was selected as the model to be used in the prediction process.

Table 6. Significance Test Results

Model	Variable	*P*> |*z*|	Description	Results
	ar.L1	0.002	Significant	
(1,1,1)(4,1,1)7	ma.L1	0.013	Significant	Does not meet the significance test
	sigma2	0.000	Significant	
	ar.L1	0.002	Significant	
(1,1,1)(3,1,1)7	ma.L1	0.013	Significant	Does not meet the significance test
	sigma2	0.000	Significant	
	ar.L1	0.002	Significant	
(1,1,1)(2,1,1)7	ma.L1	0.013	Significant	Meets significance test
	sigma2	0.000	Significant	
	ar.L1	0.002	Significant	
(1,1,1)(1,1,1)7	ma.L1	0.013	Significant	Meets significance test
	sigma2	0.000	Significant	

After determining the SARIMA model, forecasting was conducted to evaluate the model's ability to predict shallot price data. The forecasting results obtained using the SARIMA(1,1,1)(2,1,1)_7 model are presented in **Figure 7**. In the figure, the blue line represents the actual prices, while the orange line represents the predicted values generated by the SARIMA(1,1,1)(2,1,1)_7 model.

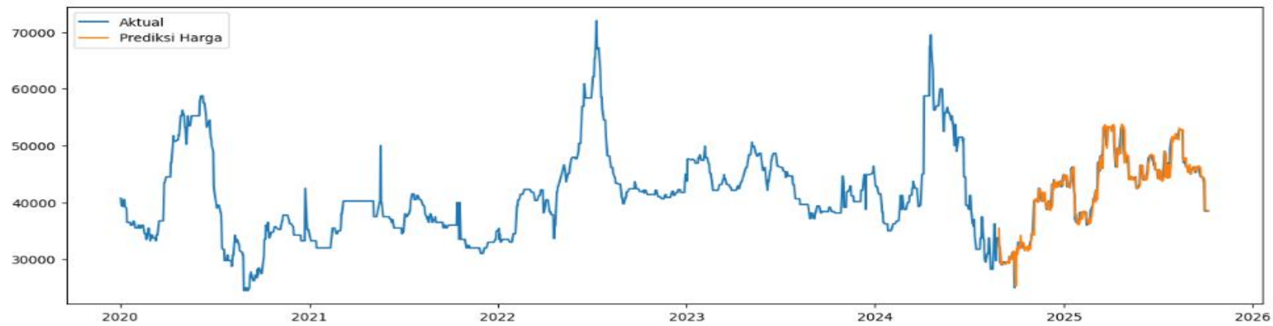

Figure 7. Actual and Predicted Graphs of SARIMA Model

3.5.3 LSTM Modeling

The initial phase in the LSTM modeling process is data normalization. The values in the shallot price column are transformed into a range between 0 and 1, but the proportional ratio of each value is maintained. Since shallot prices are represented on a relatively large numerical scale, normalization helps simplify the computation process and reduces the computational burden during model training. **Figure 8** shows the code used to perform the initial stage of data normalization. The data is normalized using the MinMaxScaler measurement tool, which is a normalization technique based on the minimum and maximum values of the price column. These minimum and maximum values are stored in the scaler object for use in the training and testing processes.

```
scaler = MinMaxScaler()
scaler.fit(df.Harga.values.reshape(-1,1))
```

Figure 8. Data Normalization Code

This study uses a sliding window approach to simplify the prediction process. A sliding window is a technique that converts time series data into sequential data so that the model can capture patterns from previous values to predict subsequent values. This technique works by separating the data into input values, which are the previous values that become the window, and output values, which are the target values. Then, the window is shifted gradually to produce pairs of input and output values. In this research, a window size of 60 was applied, where the prediction at each step is generated using information from the preceding 60 observations. This approach was applied to the training data and test data. The sliding window approach is illustrated in **Figure 9**.

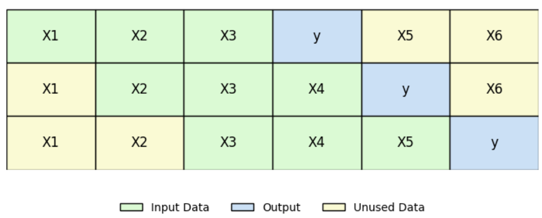

Figure 9. Illustration of the Sliding Window Technique

In the first row, X1, X2, and X3 are the input values used to predict the output value X4. The window then shifts one step in the next row. In the second row, X2, X3, and X4 become the input values used to predict the value X5. And so on, the window shifts one more step in the third row. In the third row, X3, X4, and X5 become the input values utilized for predicting the X6 value. In the next stage, the data is converted into a numpy array with higher dimensions so that it can adjust to the dimension structure required by the LSTM model. The code run on Google Colab to convert the data are presented in **Figure 10**.

```
X_train = np.array(X_train)
X_test  = np.array(X_test)
y_train = np.array(y_train)
y_test  = np.array(y_test)
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
X_test  = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
y_train = np.reshape(y_train, (-1,1))
y_test  = np.reshape(y_test, (-1,1))
```

Figure 10. Data Conversion Code

The next step involves defining the LSTM architecture to be used in the prediction process. **Figure 11** displays the code used to define the LSTM architecture.

```
#model definition
def define_model():
    input1 = Input(shape=(window_size,1))
    x = LSTM(units = 64, return_sequences=True)(input1)
    x = Dropout(0.2)(x)
    x = LSTM(units = 64, return_sequences=True)(x)
    x = Dropout(0.2)(x)
    x = LSTM(units = 64)(x)
    x = Dropout(0.2)(x)
    x = Dense(32, activation='softmax')(x)
    dnn_output = Dense(1)(x)

    model = Model(inputs=input1, outputs=[dnn_output])
    model.compile(loss='mean_squared_error', optimizer='Nadam')
    model.summary()

    return model
```

Figure 11. Code for Defining the LSTM Model

Afterward, model training is conducted to predict shallot prices. The code used to training LSTM model presented in **Figure 12**.

```
#model training
model = define_model()
history = model.fit(X_train, y_train,
                    epochs=150,
                    batch_size=32,
                    validation_split=0.1,
                    verbose=1)
```

Figure 12. Code for Training the LSTM Model

.

Figure 13 is a visualization of the forecasting output using the LSTM method. In the figure, the black line represents the training data, the expected values are illustrated by the blue line, whereas the red line depicts the LSTM predicted results.

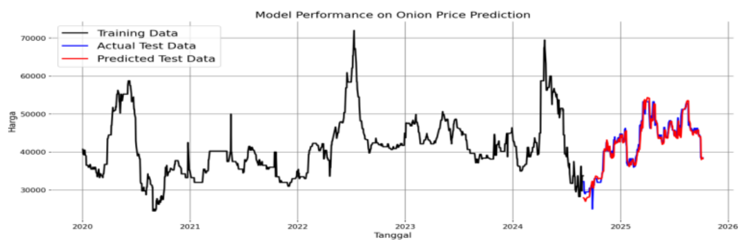

Figure 13. Actual and Predicted Graphs of LSTM Model

3.6 Model Evaluation

Table 7. Evaluation Model Result

Method	RMSE	MAPE	MAE
ARIMA (1,1,1)	1438.47	1.86%	781.52
SARIMA (1,1,1)(2,1,1)7	1444.88	2.01%	836.71
LSTM	1645.66	2.68%	1102.97

Performance of the ARIMA, SARIMA, and LSTM models, as measured by RMSE, MAPE, and MAE, is summarized in **Table 7**. The ARIMA(1,1,1) method produced the smallest combination of errors, yielding error measures of RMSE = 1438.47, MAE = 781.52, and MAPE = 1.86%. These evaluation results indicate that the ARIMA(1,1,1) method is effectively capture linear patterns in shallot price data compared to other methods. The SARIMA(1,1,1)(2,1,1)7 method produced slightly higher errors than the ARIMA(1,1,1) method, with an RMSE value of 1444.88, an MAE of 836.71, and a MAPE of 2.01%. The results show that weekly seasonal component did not improve accuracy. This may be because the seasonal pattern in the shallot price data is not very strong. In contrast, the LSTM method produces higher RMSE, MAPE, and MAE values than other methods, specifically 1860.82, 2.68%, and 1311.87. This may be because the shallot price data does not have a complex pattern, so a simpler model can produce better performance. Overall, the analysis results of this study indicate that conventional statistical methods perform better in handling shallot price data. The results obtained in this study are in agreement with study by Suryawan et al. (2024) which compared the ARIMA, LSTM, and Prophet methods in predicting the sales of a bakery in Bandung. The evaluation results reveal that compared to the other methods, the ARIMA model achieves superior predictive performance because the ARIMA model produced smaller error values.

4. CONCLUSION

Based on the results, it can be inferred that the ARIMA (1,1,1) model has the best performance in predicting shallot prices in Palembang City. This model produces the lowest RMSE, MAE, and MAPE values compared to SARIMA (1,1,1)(2,1,1)7 and LSTM methods. The SARIMA (1,1,1) (2,1,1)7 method also performs quite well, but its performance is still below that of the ARIMA method. Meanwhile, the LSTM method produces the weakest performance compared to other methods. These findings indicate that classical statistical models are still superior to neural network-based models when the data is not highly complex. This study provides a basis for stakeholders to choose the appropriate forecasting method to support decision-making related to stabilizing shallot prices.

REFERENCES

Al Rosyid, A. H., Viana, C. D. N., & Saputro, W. A. (2021). Penerapan Model Box Jenkins (ARIMA) dalam Peramalan Harga Konsumen Bawang Merah di Provinsi Jawa Tengah. Agri Wiralodra, 13(1), 29–37. https://doi.org/https://doi.org/10.31943/agriwiralodra.v13i1.19

Alabdulrazzaq, H., Alenezi, M. N., Rawajfih, Y., Alghannam, B. A., Al-Hassan, A. A., & Al-Anzi, F. S. (2021). On the accuracy of ARIMA based prediction of COVID-19 spread. Results in Physics, 27. https://doi.org/10.1016/j.rinp.2021.104509

Alawiyah, T., Permadi, I., Afuan, L., Maryanto, E., & Rahayu, S. P. (2024). Implementation of Autoregressive Integrated Moving Average (ARIMA) Method for PT XL AXIATA TBK Stock Price Prediction with Website-based Dashboard Visualization. Jurnal Teknik Informatika (Jutif), 5(4), 1205–1211. https://doi.org/10.52436/1.jutif.2024.5.4.2700

Anees, R. A. S., Khoirani, N. M., Silfa, F. P. A., & Hasna, R. A. (2022). Peramalan Harga Bawang Merah di Pasar Tradisional Sulawesi Selatan dengan Metode ARIMA. Jurnal Agribisnis, 24(2), 274–287. https://doi.org/https://doi.org/10.31849/agr.v24i2.8670\

Badan Pusat Statistik Nasional. (2025, January 24). Rata-rata Konsumsi Perkapita Seminggu Menurut Kelompok Sayur-Sayuran Per Kabupaten/kota (Satuan Komoditas). https://www.bps.go.id/id/statistics-table/2/MjEwMCMy/rata-rata-konsumsi-perkapita-seminggu-menurut-kelompok-sayur-sayuran-per-kabupaten-kota--satuan-komoditas-.html

Badan Pusat Statistik Provinsi Sumatera Selatan. (2025). Inflasi Provinsi Sumatera Selatan 2024 (Vol. 11).

Bahri, C. A., & Tania, K. D. (2025). Perbandingan Kinerja LSTM, Random Forest, dan SVR Berbasis Knowledge Discovery untuk Prediksi Harga Beras Sumatera Selatan. JURIKOM (Jurnal Riset Komputer), 12(5), 2407–389. https://doi.org/10.30865/jurikom.v12i5.9140

Bank Indonesia. (n.d.). Tabel Harga Bawang Merah Berdasarkan Daerah. Retrieved November 1, 2025, from https://www.bi.go.id/hargapangan/TabelHarga/PasarTradisionalDaerah

Caesar, B. Y., Hadiono, K., & Ardhianto, E. (2025). Evaluasi empiris model ARIMA dan LSTM dalam konteks peramalan penjualan mobil Toyota. AITI: Jurnal Teknologi Informasi, 22(2), 221–235. https://doi.org/https://doi.org/10.24246/aiti.v22i2.221-235

Cintani, M., Nizar, Y. A., Angraini, Y., Notodiputro, K. A., & Mualifah, L. N. A. (2025). Evaluation of the SARIMA and Prophet Models in Forecasting Ship Passenger Numbers at Balikpapan Port. Jambura Journal of Mathematics, 7(2), 120–128. https://doi.org/https://doi.org/10.37905/jjom.v7i2.31606

Faisal, A. N. M. F., Rahman, A., Habib, M. T. M., Siddique, A. H., Hasan, M., & Khan, M. M. (2022). Neural networks based multivariate time series forecasting of solar radiation using meteorological data of different cities of Bangladesh. Results in Engineering, 13, 100365. https://doi.org/10.1016/j.rineng.2022.100365

Fauzi, F., Aulia, S., Syaifullah, A. R., & Utami, T. W. (2024). Peramalan Harga Emas Menggunakan Pendekatan Long-Short Term Memory (LSTM). JEPIN (Jurnal Edukasi Dan Penelitian Informatika), 10(2), 252–257. https://doi.org/https://doi.org/10.26418/jp.v10i2.78332

Febiola, A., Dewi, A., Fazarin, F. M., Ramadhani, F., Khaffi, M. A., Akbar, R., & Dalimunthe, D. Y. (2024). Perbandingan Metode ARIMA dan SARIMA Dalam Peramalan Jumlah Penumpang Bandara Provinsi Kepulauan Bangka Belitung. Jambura Journal of Mathematics, 6(2). https://doi.org/10.37905/jjom.v6i2.25081

Koukaras, P., & Tjortjis, C. (2025). Data Preprocessing and Feature Engineering for Data Mining: Techniques, Tools, and Best Practices. In AI (Switzerland) (Vol. 6, Issue 10). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/ai6100257

Kumar Dubey, A., Kumar, A., García-Díaz, V., Kumar Sharma, A., & Kanhaiya, K. (2021). Study and analysis of SARIMA and LSTM in forecasting time series data. Sustainable Energy Technologies and Assessments, 47. https://doi.org/10.1016/j.seta.2021.101474

M, H. P., Rehman, M. Z., Dar, A. A., & Wangmo A, T. (2024). Forecasting CO2 Emissions in India: A Time Series Analysis Using ARIMA. Processes, 12(12). https://doi.org/https://doi.org/10.3390/pr12122699

Madhika, Y. R., Kusrini, & Hidayat, T. (2023). Gold Price Prediction Using the ARIMA and LSTM Models. Sinkron : Jurnal Dan Penelitian Teknik Informatika, 7(3), 1255–1264. https://doi.org/10.33395/sinkron.v8i3.12461

Menteng, A. K., & Rozi, A. F. (2025). Prediksi hasil panen padi berdasarkan data curah hujan, suhu, dan kelembapan dengan metode ARIMA. Jurnal Pendidikan Informatika Dan Sains, 14(1), 150–159. https://doi.org/10.31571/saintek.v14i1.9292

MY, H. U., Setyowati, S. L., Notodiputro, K. A., Angraini, Y., & Mualifah, L. N. A. (2024). Comparison of Seasonal ARIMA and Support Vector Machine Forecasting Method for International Arrival in Lombok. Jambura Journal of Mathematics, 6(2), 212–219. https://doi.org/https://doi.org/10.37905/jjom.v6i2.26478

Ning, Y., Kazemi, H., & Tahmasebi, P. (2022). A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet. Computers and Geosciences, 164. https://doi.org/10.1016/j.cageo.2022.105126

Novalia, V., Tania, K. D., Meiriza, A., & Wedhasmara, A. (2024). Knowledge Discovery of Application Review Using Word Embedding's Comparison with CNN-LSTM Model on Sentiment Analysis. ICECOS 2024 - 4th International Conference on Electrical Engineering and Computer Science, Proceeding, 234–238. https://doi.org/10.1109/ICECOS63900.2024.10791113

Prasojo, Y. D., & Muludi, K. (2025). Passenger Density Prediction at the Airport Using LSTM and SARIMA: A Case Study at Radin Inten Airport, Lampung. In Journal of Applied Informatics and Computing (JAIC) (Vol. 9, Issue 4). http://jurnal.polibatam.ac.id/index.php/JAIC

Putra, A. P., & Harahap, A. M. (2025). Implementasi Time Series Forecasting dengan Algoritma LSTM untuk Pemantauan dan Prediksi Produktivitas Kelapa Sawit Berdasarkan Hasil Panen. JURIKOM (Jurnal Riset Komputer), 12(2), 46–56. https://doi.org/10.30865/jurikom.v12i2.8495

Ruhiat, D., & Effendi, A. (2018). Pengaruh Faktor Musiman pada Pemodelan Deret Waktu untuk Peramalan Debit Sungai dengan Metode Sarima. Teorema: Teori Dan Riset Matematika, 2(2), 117–128. https://doi.org/http://dx.doi.org/10.25157/teorema.v2i2.1075

Santoso, A. B., & Widodo, T. (2024). Predicting the Number of Forest and Land Fire Hotspot Occurrences Using the ARIMA and SARIMA Methods. Jurnal Sisfokom (Sistem Informasi Dan Komputer), 13(1), 119–129. https://doi.org/10.32736/sisfokom.v13i1.2018

Shofwatillah, L. (2025). Peramalan Curah Hujan di Provinsi Sumatera Utara Menggunakan Metode Seasonal Autoregressive Integrated Moving Average (SARIMA). Emerging Statistics and Data Science Journal, 3(3), 697–710. https://doi.org/doi.org/10.20885/esds.vol3.iss.3.art22

Sirisha, U. M., Belavagi, M. C., & Attigeri, G. (2022). Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison. IEEE Access, 10, 124715–124727. https://doi.org/10.1109/ACCESS.2022.3224938

Siswanto, J., Hendry, Rahardja, U., Sembiring, I., & Lisangan, E. A. (2025). Predicting Transjakarta Passengers with LSTM-BiLSTM Deep Learning Models for Smart Transportpreneurship. APTISI Transactions on Technopreneurship, 7(1), 84-96-84–96. https://doi.org/10.34306/att.v7i1.440

Sofiah, N. A., Tania, K. D., Meiriza, A., & Wedhasmara, A. (2024). A Comparative Assessment SARIMA and LSTM Models for the Gurugram Air Quality Index's Knowledge Discovery. 2024 International Conference on Electrical Engineering and Computer Science (ICECOS), 26–31. https://doi.org/10.1109/ICECOS63900.2024.10791243

Suryawan, I. G. T., Putra, I. K. N., Meliana, P. M., & Sudipa, I. G. I. (2024). Performance Comparison of ARIMA, LSTM, and Prophet Methods in Sales Forecasting. Sinkron, 8(4), 2410–2421. https://doi.org/10.33395/sinkron.v8i4.14057

Triyadi, A., Suwondo, A., Asmarajati, D., Hasanah, N., & Asnawi, M. F. (2024). Prediksi Harga Bawang Merah Kering di Wonosobo Menggunakan Metode Long Short Term Memory. STORAGE – Jurnal Ilmiah Teknik Dan Ilmu Komputer, 3(2), 133–138. https://doi.org/https://doi.org/10.55123/storage.v3i2.3601

Zahara, S., & Sugianto. (2021). Forecasting Consumer Price Index Data Based on Multivariate Time Series Using Deep Learning. Jurnal RESTI, 5(1), 24–30. https://doi.org/https://doi.org/10.29207/resti.v5i1.2562

Zhou, J., & Wang, S. (2021). A carbon price prediction model based on the secondary decomposition algorithm and influencing factors. Energies, 14(5). https://doi.org/10.3390/en14051328