Implementation of ARIMA with Min-Max Normalization for predicting the Price and Production Quantity of Red Chili Peppers in North Sumatra Province considering Rainfall and Sunlight Duration Factors

Ifan Prihandi
Faculty of Informatics Technology, Satya Wacana Christian University Salatiga, Indonesia
982022023@student.uksw.edu (corresponding author)

Sutarto Wijono
Faculty of Informatics Technology, Satya Wacana Christian University Salatiga, Indonesia
sutarto.wijono@uksw.edu

Irwan Sembiring
Faculty of Informatics Technology, Satya Wacana Christian University Salatiga, Indonesia
irwan@uksw.edu

Evi Maria
Faculty of Informatics Technology, Satya Wacana Christian University Salatiga, Indonesia
evi.maria@uksw.edu

Received: 27 December 2024 | Revised: 25 January 2025 | Accepted: 11 February 2025

 DOI: https://doi.org/10.48084/etasr.9875

ABSTRACT

Red chili peppers are a vital agricultural commodity in the North Sumatra province, playing a significant role in Indonesia's economy. Fluctuations in chili prices affect farmers, consumers, and overall economic stability. This study leverages time series forecasting using the ARIMA model to predict red chili pepper prices and production, incorporating weather factors such as rainfall and sunlight duration. The dataset spans March 2021 to December 2023 and includes historical records of chili prices, production levels, and weather conditions. The analysis reveals a strong correlation between price fluctuations and production trends: Prices tend to rise when production declines and fall when yields increase. Additionally, production is influenced by weather conditions, where excessive rainfall damages crops and reduces yields, while balanced rainfall and sunlight duration support optimal growth. The ARIMA model demonstrates its effectiveness in capturing these patterns, providing actionable insights for farmers and policymakers to predict price changes and optimize production strategies. By integrating data-driven forecasting with weather analysis, this research contributes to more adaptive and informed decision-making in the agricultural sector.

Keywords-time series prediction; machine learning; ARIMA; agricultural analytics; climate impact; data-driven decision-making

I. INTRODUCTION

Agricultural commodities play an essential role in our daily lives and significantly influence economic stability [1]. Among these, chili peppers, particularly red chili peppers, are vital commodities in the Indonesian economy [2]. The monthly consumption per capita of red chili reaches 0.15 kg. In 2021, Indonesia's red chili production reached 1.36 million tons, indicating a 7.62% increase compared to the previous year. The West Java province emerged as the largest producer, with a production volume of 343,070 tons, followed by North Sumatra with 210,220 tons and Central Java with 169,280 tons [3]. Fluctuations in chili prices can directly affect the livelihoods of farmers, the cost to consumers, and overall economic stability [4]. Consequently, forecasting agricultural commodity prices has become a critical focus in both economic and agricultural research [5]. A thorough understanding of the factors that affect chili prices and the ability to predict them is crucial to mitigate their impact and support economic resilience.

Chili prices are affected by various factors, including production levels, weather conditions, and natural phenomena [6]. Weather, especially rainfall and sunlight duration, plays a critical role in chili production. Adequate rainfall is essential for the growth of chili plants. However, excessive rainfall can damage crops, increase disease risks, and reduce yields. Variations in the number of rainy days and daily rainfall intensity can affect economic production, including the agricultural sector [7]. The optimal rainfall for chili production ranges from 600 to 800 mm per year. The duration of sunlight affects photosynthesis and plant development. Chili plants require sufficient sunlight to produce optimal yields. The ideal sunlight duration for chili plants is around 6-8 hours per day. Several studies have shown a positive relationship between weather factors and agricultural production. For instance, research by the Namibia Economist outlined that high rainfall intensity could damage crops and hinder pollination processes. A well-distributed rainfall throughout the growing season is crucial to ensure adequate water availability for plants [8]. Further research has indicated that changes in rainfall patterns due to climate change could have significant economic consequences, especially in agriculture-dependent sectors [8]. For example, a 10% increase in sunlight exposure can increase chili production by 5%. In North Sumatra, the price of red chili peppers experienced dramatic changes in 2024, with a notable spike on February 9, when prices soared to Rp59,150 per kg, an increase of 863% in just one week [9]. This volatility reflects the broader challenges of rising and fluctuating chili prices that occur every year. Between 1983 and 2018, red chili pepper prices at both the producer and consumer levels have shown a consistent upward trend, with annual growth rates of 12.68% and 13.73%, respectively, over the past five years [10].

North Sumatra is a key chili production region in Indonesia, producing approximately 210,220 tons of red chili peppers in 2021, making it the second largest producer after West Java [11]. The importance of the province in meeting national chili demand plays an important role in shaping market prices. Its diverse geography and climate make North Sumatra an ideal region for studying how changes in rainfall patterns and sunlight duration significantly affect chili production and pricing. For example, the province experiences weather variations, such as increased rainfall due to disruptions in wind circulation and reduced rainfall in some areas caused by wind shifts earlier in 2024 [12, 13]. These climatic fluctuations highlight the challenges faced by chili farmers and markets in maintaining stable production and prices. By focusing on North Sumatra, researchers can gain valuable insights into how the weather influences chili production and price dynamics. This research is expected to provide valuable insights for other regions with similar characteristics [10, 11].

Machine learning, a subset of artificial intelligence, provides robust algorithms capable of learning and performing specific tasks with high precision. These algorithms have demonstrated significant potential to enhance the accuracy of forecasting commodity prices [1]. This study employs the ARIMA model to predict the price and production quantity of red chili peppers in North Sumatra province, taking into account rainfall and sunlight duration. ARIMA was chosen for its superior performance compared to machine learning algorithms such as the Support Vector Machine (SVM) and Wavelet Neural Network (WNN) in studies involving weather-related data, along with its simplicity, interpretability, and reliability in capturing patterns in time series data [14]. The ARIMA model addresses serial dependency in time series data, where the Autoregressive (AR) component models, the interdependence within the dependent variable, and the Moving Average (MA) component captures the dependency of the dependent variable on previous errors. This model is generally applied to stationary time series data, where the mean, variance, and autocorrelation function remain constant over time. This model can also be applied to non-stationary time series, particularly after making the data stationary through transformations such as differencing and logging. This flexibility underscores ARIMA's adaptability as a robust model to effectively analyze and forecast both stationary and non-stationary time series data. The ARIMA model is represented by the AR, difference (I), and MA components denoted by p, d, and q, respectively. Historical data are decomposed into an AR process that retains past occurrences, an Integrated (I) process that makes the data stationary to reduce forecasting complexity, and an MA process that handles forecasting errors. The p component in AR represents the linear relationship between the dependent variable and its lagged values [15]. The standard form of the ARIMA model is as follows:

$$Y_t = C + \sum_{i=1}^{p} \varphi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j e_{t-j} \quad (1)$$

where t denotes time, p is the number of lag values of the AR, component, q is the number of lag errors of the MA component, Y_t is the value of the series at time t, C is the intercept coefficient, Y_{t-i} denotes the past values of the series at time $t-i$, e_{t-j} denotes the past residual values of the series at time $t-j$, φ_i denotes the AR model coefficients, and θ_j denotes the MA model coefficients. The ARIMA model has demonstrated significant accuracy in agricultural forecasting studies, such as predicting wheat and maize production in China and India [13, 15]. Despite its strengths, ARIMA is inherently univariate and cannot account for interactions between multiple variables.

In contrast, the Vector Autoregressive (VAR) model is specifically designed for multivariate time series analysis to study the dynamic relationships between multiple variables without imposing strict assumptions about causality. Each variable in a VAR model is treated symmetrically and expressed as a function of its own lags and those of other variables in the system. This flexibility makes VAR models particularly useful in macroeconomic and financial studies to analyze and forecast the interconnected behavior of indicators such as GDP, inflation, and interest rates. Additionally, VAR models facilitate impulse-response analysis to evaluate the impact of shocks to one variable on others and forecast error variance decomposition to understand the contribution of each variable to forecast errors. This study uses VAR to compare ARIMA, allowing a rigorous evaluation of their respective strengths in time series forecasting. This comparison aims to highlight ARIMA's superior performance in handling time-series data effectively. VAR models treat all the variables involved symmetrically, with each model comprising two or more variables, where the right-hand side includes the lag vector of the dependent variable [16]. The following equation is used to mathematically represent the structure of the VAR model:

$$Y_t = C + \left(\sum_{i=1}^{p} \varphi_i \, Y_{t-i}\right) + e_t \qquad (2)$$

where t denotes time, p is the number of lag lengths, Y_t is the vector representing the value of the series at time t, C is the vector of intercept coefficients, e_t is the vector representing the error term at time t, φ_i is the matrix of model coefficients, and Y_{t-i} is the past vector representing the value of the series at time $t-i$. VAR has been applied in various macroeconomic and agricultural studies to analyze interdependencies between factors such as rainfall, sunlight duration, and production levels. For example, in [17], VAR was used to analyze the prices of food ingredients such as rice, cooking oil, chicken eggs, chicken meat, and cayenne pepper. This study showed that VAR could capture the influence of one staple food's price on another's within a multivariate framework. The results highlighted that although some prices were stable, others, such as cayenne pepper, were more volatile. This showcases the ability of the VAR model to handle complex interdependencies in multivariate time series data.

Although existing research often focuses on applying ARIMA or VAR for forecasting, direct comparisons between the two models are scarce, particularly in the context of agricultural commodities influenced by weather variability. This gap is critical since understanding the relative strengths and limitations of these models can inform more accurate forecasting strategies. By comparing ARIMA and VAR in predicting chili prices and production in North Sumatra, this study aims to address this gap and provide insights into which model performs better under varying conditions.

Data on chili prices, production, rainfall, and sunlight duration were collected from March 2021 to December 2023 and analyzed to identify patterns and trends. Before analysis, the data were cleaned and normalized using the min-max method to ensure comparable scales between the variables. The min-max method is a widely used standardization technique that scales data values to fall within the range of 0 to 1. This approach ensures that differences in index dimensions do not interfere with the accuracy of model training or predictions [18]. By constraining data values within a specific range, the min-max normalization technique enhances the precision of time series model predictions by limiting the prediction outputs to a narrower interval, thereby providing a more accurate depiction of prediction uncertainty [19]. The normalization formula for the min-max method is:

$$x_{norm} = \frac{x' - \min(x)}{\max(x) - \min(x)} \qquad (3)$$

where x_{norm} is the normalized value, $\max(x)$ is the maximum value in the selected column dataset, $\min(x)$ is the minimum value in the selected column dataset, and x' is the old value in the selected column dataset [20].

The ARIMA model was constructed through stationarity testing and parameter determination using the Augmented Dickey-Fuller (ADF) test and Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots. The resulting ARIMA model was then evaluated using Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) to measure prediction accuracy. MAPE and MAE are effective metrics to evaluate prediction models, especially in the context of time series data [21]. MAPE is a relative error metric that calculates absolute values to prevent positive and negative errors from offsetting each other. Additionally, it employs relative errors to facilitate the comparison of forecast accuracy across different time-series models [22]. MAPE can be calculated using:

$$MAPE(\%) = \frac{\sum_{t=1}^{n} \left|\frac{A_t - F_t}{A_t}\right|}{n} \times 100 \qquad (4)$$

where A_t denotes the actual data at time t, F_t denotes the forecasted data at time t, and N is the number of forecasted data points. MAE is a statistical metric used to calculate the average of the absolute differences between the predicted values generated by a model and the actual target values [23], calculated using:

$$MAE = \frac{\sum_{t=1}^{n} |p_t - q_t|}{N} \qquad (5)$$

where p_t denotes the predicted data at time t, q_t denotes the actual data at time t, and N is the number of data pairs.

This study investigates the effectiveness of ARIMA and VAR models in predicting red chili prices and production in North Sumatra by integrating weather factors such as rainfall and sunlight duration. The processed data are then analyzed to identify patterns and relationships among variables, utilizing visualizations and statistical techniques. Following this, the ARIMA and VAR models are implemented. The results section provides a comparative analysis of these models, highlighting their strengths and weaknesses in forecasting chili prices and production. These findings have significant implications for chili production and distribution, as well as for policy-making related to the stability of agricultural commodity prices in Indonesia. It is anticipated that this research will offer substantial benefits to Indonesia in designing strategies to manage chili price volatility that affects economic stability, and providing in-depth insights into how weather factors affect chili production and prices, allowing the National Food Agency (BPN) to formulate more effective policies in managing chili supply and prices.

II. METHODOLOGY

This study employs a quantitative research approach using time series analysis. This method was chosen to analyze numerical data over time to understand the patterns, trends, and relationships between variables. According to [24], quantitative research refers to the systematic collection and analysis of numerical data using mathematical or statistical methods to explain real-world phenomena and obtain conclusive evidence concerning hypotheses. Time series analysis is particularly suitable for this research, as it allows analysis of historical data to predict future trends.

The datasets used included retail prices and production data of red chili peppers in North Sumatra province from March 2021 to December 2023. Chili price data were sourced from the openly accessible BPN public price monitoring platform (panelharga.badanpangan.go.id). Production data were obtained from the BPN Data and Information Center, which requires permission to access the dataset. Weather datasets, including rainfall and sunlight duration, were sourced from the Meteorology, Climatology, and Geophysics Agency (BMKG) through its data portal (data.bmkg.go.id). Although publicly accessible, creating an account is necessary to retrieve this data. The data were analyzed using the ARIMA model. Figure 1 illustrates the research stages.

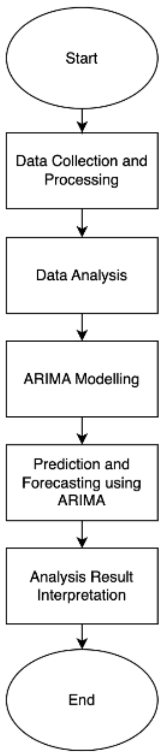

Fig. 1. Research stages.

A. *Data Collection and Processing*

Data on retail prices and red chili pepper production in North Sumatra were collected for the period from March 2021 to December 2023. Daily retail price data, which vary by city, were averaged across the province. Monthly production data were provided by BPN. Additionally, daily data on rainfall and sunlight duration were sourced from BMKG, specifically from the Aek Godang and Silangit meteorological stations. The data were cleaned, with missing values imputed using monthly averages, normalized using min-max, and transformed as needed for analysis.

B. *Data Analysis*

Processed data were analyzed to identify patterns, trends, and relationships between variables. Descriptive analysis was performed to understand data characteristics and explore correlations between retail prices, red chili pepper production, rainfall, and sunshine duration. The data analysis steps included visualizing data patterns using matplotlib, computing descriptive statistics with pandas, and performing correlation analysis.

C. *ARIMA Modeling*

ARIMA is used to capture temporal patterns in time series data and build predictive models based on historical data. ARIMA modeling steps included testing data stationarity and determining d using the ADF test [25], determining p and q using ACF and PACF plots, and building the ARIMA model using statmodels [26]. However, a significant challenge in building predictive models lies in balancing model complexity and data fit to avoid over- or under-fitting. An overly complex model might perfectly fit the training data but fail to generalize to new data, resulting in overfitting. In contrast, a simple model might generalize well but fail to accurately capture patterns in both training and future data, leading to underfitting and poor predictive performance. Effective modeling requires finding the right balance to ensure robust predictions without compromising generalization [27].

D. *Prediction and Forecasting Using ARIMA*

The constructed ARIMA model is then used to forecast future retail prices and production of red chili peppers. These predictions are crucial to provide insights into future trends in prices and production. The prediction steps include forecasting using the ARIMA model and evaluating model performance using MAE and MAPE [28].

E. *Analysis of Results and Interpretation*

Results show an improvement in accuracy for the ARIMA model when using min-max normalization. The results of the forecasting are further analyzed to provide interpretation. These findings can offer insights into the factors that influence the price and production of red chili peppers, as well as how weather conditions, such as rainfall and sunlight exposure, play a role. Interpretation steps include analyzing the main trends in the prediction results and evaluating the impact of weather variables on the price and production of red chili peppers in North Sumatra.

III. RESULTS AND DISCUSSION

A. *Data Collection and Processing*

The datasets include production data in quintals, rainfall data in mm, and sunshine duration data in hours. All data were provided in table files in .xlsx format. Price data consisted of daily retail prices for commodities in North Sumatra from March 9, 2021, to December 31, 2023. These data were processed to retain only two columns: date and price in Rp per kg, focusing specifically on red chili peppers. The average values were used to fill missing data on certain days. The resulting dataset contained 1,036 rows. Figure 2 shows an example of the first five rows of the resulting price dataset.

	date	price_idr
0	2021-03-01	48214.13549
1	2021-03-02	48214.13549
2	2021-03-03	48214.13549
3	2021-03-04	48214.13549
4	2021-03-05	48214.13549

Fig. 2. Price data.

The production data consisted of monthly production volumes for red chili peppers in North Sumatra from January 2021 to December 2023. These data were clean at the time of acquisition, so the only processing step was to remove the first two rows. This adjustment was made to align its structure with the price data beginning from March 2021. The dataset contained 35 rows and 2 columns, representing the monthly date and production quantity in quintals. Figure 3 shows an example of the first five rows of the resulting production dataset.

	date	production_quintals
0	2021-03	71534.70
1	2021-04	52169.50
2	2021-05	59343.46
3	2021-06	65186.19
4	2021-07	66465.75

Fig. 3. Production data.

Rainfall data consisted of daily rainfall intensity per month in north Sumatra, with two columns, date and RR, representing the rainfall intensity in mm. The data covered the period from March 2021 to December 2023. RR indicates the height of rainfall that would accumulate over an area of one square meter if none of the water was absorbed, flowed away, or evaporated. Any missing values or incorrectly formatted entries were replaced with a value of 0. The resulting dataset included 1,036 rows and 2 columns. Figure 4 shows an example of the first five rows of the resulting rainfall intensity dataset.

	date	rain_mm
0	2021-03-01	0.0
1	2021-03-02	8.5
2	2021-03-03	0.0
3	2021-03-04	12.3
4	2021-03-05	0.0

Fig. 4. Rainfall intensity data.

Sunlight duration data consisted of daily sunlight intensity per month in North Sumatra, with two columns, date and SS, representing sunlight intensity in hours. The data processing procedure was the same to the rainfall data. The resulting dataset included 1,036 rows and 2 columns. Figure 5 shows an example of the first five rows of the resulting dataset.

	date	sun_hours
0	2021-03-01	7.0
1	2021-03-02	8.0
2	2021-03-03	8.8
3	2021-03-04	9.1
4	2021-03-05	8.5

Fig. 5. Sunshine duration data.

B. Data Analysis

Data analysis is crucial to identify the relationships between variables. Converting data to diagrams or graphs facilitates analysis compared to tables. In line graphs, identifying the highest and lowest values is easier. For instance based on Figure 6, in November 2023, the retail price of red chili peppers peaked at Rp79,028.67 per kg, while the lowest price was recorded in May 2023 at Rp28,323.55. For red chili pepper production data, the highest production occurred in November 2022 with 97,568.8 quintals, while the lowest production occurred in April 2021 with 52,169.5 quintals. Rainfall data showed the highest intensity in August 2023 at 13.5806 mm and the lowest in July 2022 at 0.7322 mm. In contrast, sunshine duration data showed the longest duration in April 2021 at 7.7067 hours and the shortest in May 2023 at 3.9607 hours.

Price and production data are interconnected through the law of supply. This law states that ceteris paribus, if the price of a good increases, the quantity supplied by producers will increase, and vice versa, if the price decreases, the quantity supplied will decrease [29]. Therefore, the price and production data patterns will inversely correlate as depicted in the line graph. However, an anomaly was observed from November 2022 to the end of 2023, where the data trends were relatively directly correlated. However, at several points, the patterns still align with the law of supply.

Rainfall and sunshine duration can influence production amounts and indirectly affect prices. The line graphs of rainfall and sunshine duration show inversely correlated patterns and trends. This is because an increase in rainfall intensity reduces the duration of sunlight exposure. These climatic effects relate to red chili pepper production, where rainfall inversely correlates with production amounts, while sunshine duration directly correlates with production.

C. ARIMA Modeling

Before implementing the ARIMA model, all cleaned data were converted using min-max normalization. This step aimed to expedite the computation process in model creation and minimize prediction errors. Figure 7 shows the normalization process using min-max scaling by preparing a MinMaxScaler object from the sklearn preprocessing library for each dataset. These scaler objects are then used to normalize each dataset to a value between 0 and 1. Normalization was carried out using the fit_transform method, which applies the predetermined scale to the original data. The normalized data are then stored in the variables scaled_price, scaled_production, scaled_rainfall, and scaled_sunshine.

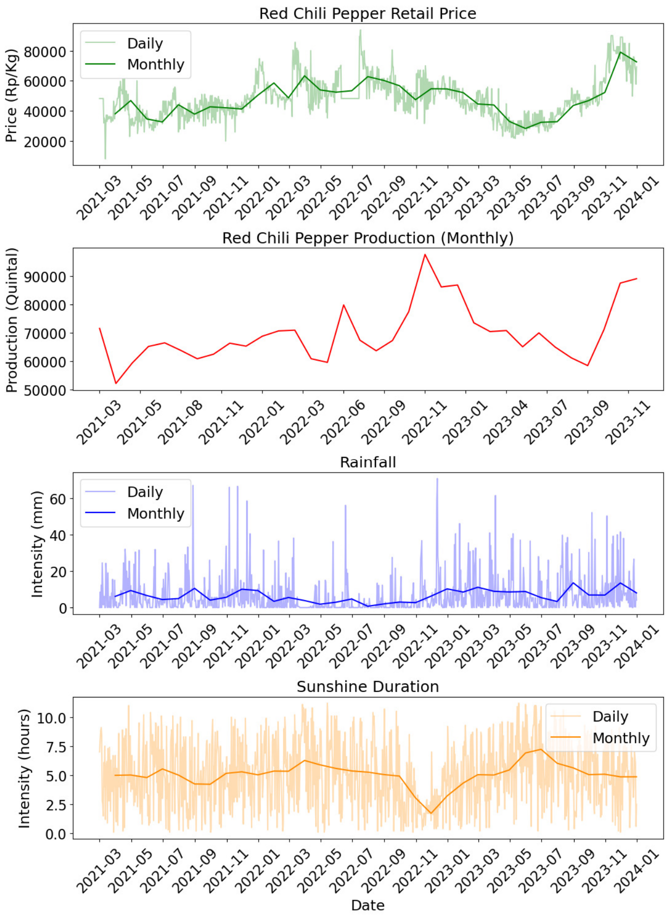

Fig. 6. Graphs of red chili pepper price and production, rainfall, and sunshine duration.

```
from sklearn import preprocessing

# Price Data
price = df_price['price_idr']
# Production Data
production = df_production['production_quintals']
# Rainfall Data
rainfall = df_rain['rain_mm']
# Sunshine Duration Data
sunshine = df_sunshine['sun_hours']

# Prepare MinMaxScaler objects for each data
scaler_price = preprocessing.MinMaxScaler()
scaler_production = preprocessing.MinMaxScaler()
scaler_rainfall = preprocessing.MinMaxScaler()
scaler_sunshine = preprocessing.MinMaxScaler()

# Apply MinMaxScaler to the data
scaled_price = scaler_price.fit_transform(
                                price.values.reshape(-1, 1))
scaled_production = scaler_production.fit_transform(
                                production.values.reshape(-1, 1))
scaled_rainfall = scaler_rainfall.fit_transform(
                                rainfall.values.reshape(-1, 1))
scaled_sunshine = scaler_sunshine.fit_transform(
                                sunshine.values.reshape(-1, 1))
```

Fig. 7. Min-max normalization process.

Several sequential steps must be followed to construct the ARIMA model, including the identification of data stationarity, the determination of the differentiation value or d, and the determination of p and q.

1) Identifying Data Stationarity

The ADF test was used to test data stationarity. The test criterion states that if the *p*-value is less than or equal to the significance level ($\alpha = 5\%$), the data can be considered stationary. Conversely, if the *p*-value is greater than the significance level ($\alpha = 5\%$), the data are considered non-stationary, requiring transformation or differencing.

TABLE I. ADF TEST RESULTS

Data	P-Value
Price	0.38927
Production	0.14011
Rainfall	3.24e-17
Sunshine duration	0.0006

Table I shows the results of the ADF test using the adfuller() function from statsmodels library. The price and production data were non-stationary and therefore, require further differencing. In contrast, the rainfall and sunlight duration data were already stationary, so differencing was not necessary.

2) Determining Differentiation Value (Parameter d)

Differencing can be applied to non-stationary data to achieve stationarity. This process can be repeated until the data become stationary, with stationarity verified using the ADF test after each differencing step.

```
from statsmodels.tsa.stattools import adfuller

def diff_list(df_col):
    list = {}
    data = df_col
    for i in range (1, 11):
        data = data.diff().dropna()
        adf = adfuller(data)
        p_value = adf[1]
        list[i] = p_value

    return list
```

Fig. 8. Differencing level function.

The code in Figure 8 defines a function named diff_list to determine the differencing level needed for data to become stationary. This function takes a DataFrame column (df_col) as argument. It initializes an empty dictionary to store *p*-values from the ADF test at each differencing level. The data from the column are copied into the variable data. In each iteration of a loop running from 1 to 10, the function differences the data using the diff() method, which calculates the changes between consecutive values and removes resulting NaN values with dropna(). After differencing, the ADF test is performed on the data using the adfuller() function to extract the *p*-value from the. This *p*-value is then stored in the list dictionary with the current differencing level as the key. The function finally returns the list dictionary containing *p*-values for each differencing level from 1 to 10.

```
# P-Value List for Price Data
list_diff_price = diff_list(pd.Series(scaled_price.flatten()))
list_diff_price
```

```
{1: 7.979772447136233e-27,
 2: 2.08813374720155e-23,
 3: 8.402750927863155e-30,
 4: 2.446281719001665e-30,
 5: 0.0,
 6: 0.0,
 7: 0.0,
 8: 0.0,
 9: 0.0,
 10: 0.0}
```

Fig. 9. Application of the diff_list function.

Figure 9 displays the results of applying the diff_list() function to the price data. These data became stationary after a single differencing, with a *p*-value of 7.97977e-27. The same function was also applied to other non-stationary data. Table II shows the results of the analysis to determine the degree of differencing required. The price and production data achieve stationarity after a single differencing. In contrast, the rainfall and sunlight duration data do not require differencing, with their differencing value set to zero.

TABLE II. OPTIMAL DIFFERENTIATION VALUES

Data	Number of differentiations (d)	p-value
Price	1	7.97977e-27
Production	1	3.67078e-08
Rainfall	0	3.24e-17
Sunshine duration	0	0.0006

3) *Determining p and q*

The parameters p and q in the ARIMA model can be determined through the identification of PACF and ACF plots on the stationary data. Historical data PACF and ACF plots are used to depict the correlation between observed data and one or more previous periods (lags). In this process, the parameter p is determined based on significant lag values in the PACF plot, while the parameter q is determined based on significant lag values in the ACF plot.

Figure 10 shows the PACF and ACF plots for the stationary data on prices, production, rainfall, and sunlight duration. The parameter values for p and q are as follows: 1 for price data, 6 and 12 for production data, 6 for rainfall data, and 1 for sunlight duration data. Table III shows the results of the analysis for determining the parameters p, d, and q. The optimal ARIMA models for each dataset are as follows: ARIMA(1, 1, 1) for price data, ARIMA(6, 1, 12) for production data, ARIMA(6, 0, 6) for rainfall data, and ARIMA(1, 0, 1) for sunlight duration data.

TABLE III. OPTIMAL ARIMA PARAMETERS

No.	Data	Model
1	Price	ARIMA (1, 1, 1)
2	Production	ARIMA (6, 1, 12)
3	Rainfall	ARIMA (6, 0, 6)
4	Sunshine duration	ARIMA (1, 0, 1)

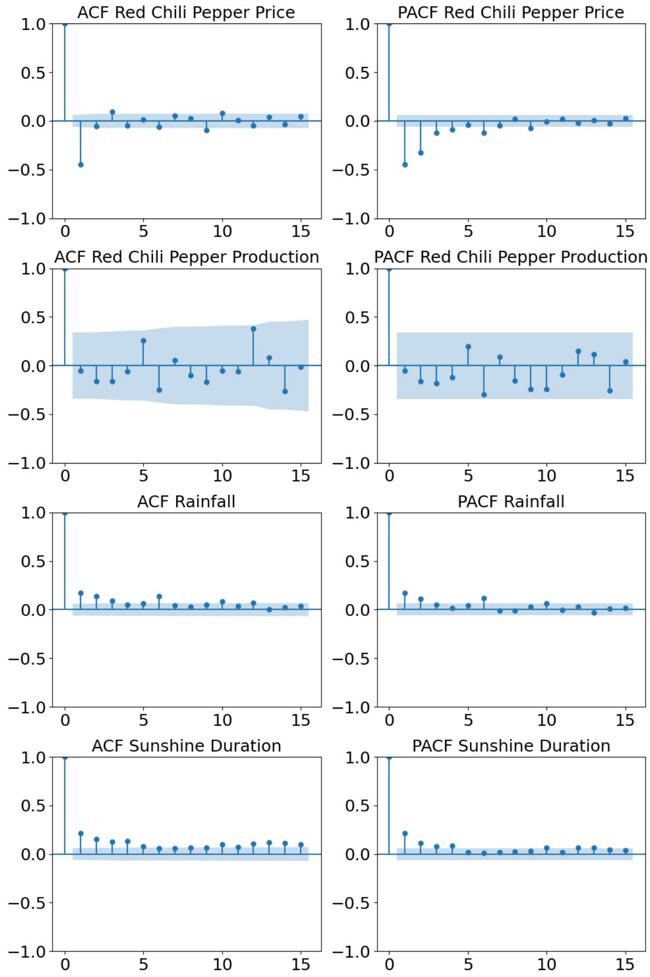

Fig. 10. PACF and ACF plots.

4) *Prediction and Forecasting using ARIMA*

This study treats prediction and forecasting as distinct concepts. Prediction refers to the values generated by the model at each corresponding point in time within the examined data, while forecasting pertains to the values generated by the model for multiple future time periods that are not present within the studied data. All historical data under analysis are in a daily time range from March 1, 2021, to December 31, 2023 (1036 rows), except for production data, which are structured monthly (35 rows). Each training and forecasting process uses the respective time units of each dataset. Subsequently, the final results of the daily data are averaged by month, and error values are calculated by comparing them with the actual data, also averaged on a monthly basis. For this study, the forecast extended 3 months into the future, covering January 2024 to March 2024.

As shown in Figure 11, the calculate_forecast_steps() function was designed to calculate the number of days required for the forecast period based on a specified number of months. This function uses the pandas.tseries.offsets.MonthEnd module to identify the end of each month and calculate the total days in each successive month. The calculation begins from a given start date (start_date) and iterates according to the number of months defined by the forecast_months parameter, summing the days in each month encountered. The output of this function is the total number of days corresponding to the desired forecast period in months, making it applicable for daily data forecasting.

```python
from pandas.tseries.offsets import MonthEnd

# Calculate the number of days in the upcoming months
def calculate_forecast_steps(start_date, forecast_months):
    current_date = pd.Timestamp(start_date) + MonthEnd(0)
    total_days = 0

    # Loop through each month to forecast
    for _ in range(forecast_months):
        next_month_end = current_date + MonthEnd(1)
        days_in_month = (next_month_end - current_date).days
        total_days += days_in_month
        current_date = next_month_end

    return total_days
```

Fig. 11. Monthly-to-daily time conversion function.

```python
from statsmodels.tsa.arima.model import ARIMA

# Forecasting for up to 3 months ahead for daily data
forecast_months = 3

# ARIMA Model for Red Chili Price
last_date_price = df_price['date'].max()
forecast_steps_price = calculate_forecast_steps(
    last_date_price, forecast_months) # Convert forecast_months to days
model_price = ARIMA(scaled_price, order=param_price)
model_fit_price = model_price.fit()
predicted_price = scaler_price.inverse_transform(
    model_fit_price.fittedvalues.reshape(-1, 1))
forecasted_price = scaler_price.inverse_transform(
    model_fit_price.forecast(steps=forecast_steps_price).reshape(-1, 1))

# ARIMA Model for Red Chili Production (Monthly Data)
forecast_steps_production = forecast_months
model_production = ARIMA(scaled_production, order=param_production)
model_fit_production = model_production.fit()
predicted_production = scaler_production.inverse_transform(
    model_fit_production.fittedvalues.reshape(-1, 1))
forecasted_production = scaler_production.inverse_transform(
    model_fit_production.forecast(steps=forecast_steps_production).reshape(-1, 1))

# ARIMA Model for Rainfall
last_date_rain = df_rain['date'].max()
forecast_steps_rain = calculate_forecast_steps(
    last_date_rain, forecast_months) # Convert forecast_months to days
model_rain = ARIMA(scaled_rainfall, order=param_rain)
model_fit_rain = model_rain.fit()
predicted_rain = scaler_rainfall.inverse_transform(
    model_fit_rain.fittedvalues.reshape(-1, 1))
forecasted_rain = scaler_rainfall.inverse_transform(
    model_fit_rain.forecast(steps=forecast_steps_rain).reshape(-1, 1))

# ARIMA Model for Sunshine Duration
last_date_sun = df_sunshine['date'].max()
forecast_steps_sun = calculate_forecast_steps(
    last_date_sun, forecast_months) # Convert forecast_months to days
model_sun = ARIMA(scaled_sunshine, order=param_sun)
model_fit_sun = model_sun.fit()
predicted_sun = scaler_sunshine.inverse_transform(
    model_fit_sun.fittedvalues.reshape(-1, 1))
forecasted_sun = scaler_sunshine.inverse_transform(
    model_fit_sun.forecast(steps=forecast_steps_sun).reshape(-1, 1))
```

Fig. 12. Application of the ARIMA model.

In Figure 12, the ARIMA model is applied to forecast all datasets, including red chili pepper prices, red chili pepper production, rainfall, and sunlight duration. For daily data (such as chili prices, rainfall, and sunlight duration), the code uses the latest date in the dataset as the starting point and converts the forecast period from months into days using the calculate_forecast_steps function. The ARIMA model is then built with the normalized data and predefined model parameters, fitted to the data, and the results are transformed back to their original scale using the inverse_transform method from the scaler used during data normalization. For the monthly data (red chili pepper production), the forecast period remains in months without conversion to daily units. The ARIMA model is built with normalized data, fitted to it, and then the predictions and forecasts are reverted to the original scale for ease of interpretation. This approach ensures that the ARIMA model can provide accurate forecasts aligned with the daily and monthly data requirements for the designated 3-month forecast period.

The forecast results are visualized using line charts alongside the actual data to observe how close they are. In Figure 13, the actual solid line represents the historical data, while the dashed line represents the historical prediction and forecast data from the ARIMA model. The forecasting data lines for price, production, and rainfall do not start concurrently with the actual data due to the elimination of rows with missing values (NaN) corresponding to the degree of differencing applied. Visually, the ARIMA model produces results that are highly aligned with the actual data, especially for red chili pepper prices, production, rainfall, and sunshine duration. Although minor discrepancies are observed during sudden fluctuations, ARIMA consistently captures the overall trends.

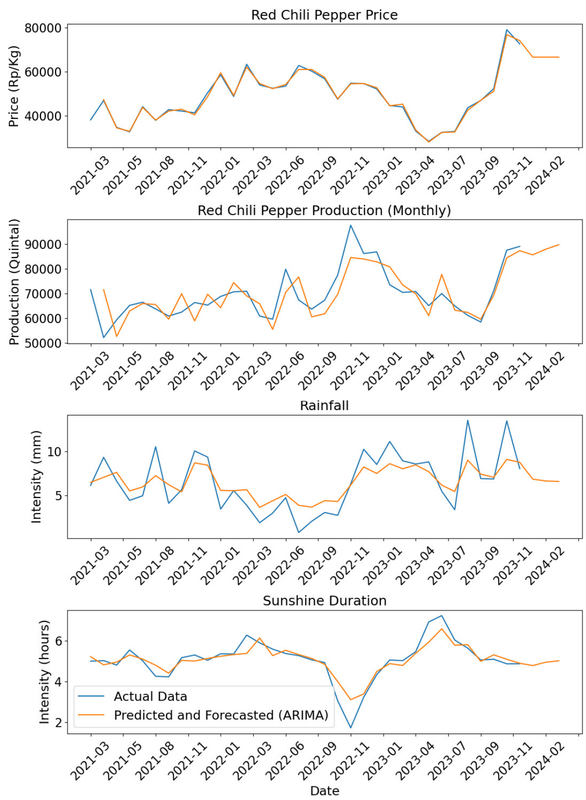

Fig. 13. Actual data and ARIMA prediction graphs.

On the other hand, the VAR model, shown in Figure 14, while effective in identifying general patterns, demonstrates larger deviations, particularly in production and rainfall data. Thus, the ARIMA model outperforms the VAR model, offering more precise predictions and better alignment with actual data, making it a more reliable choice for forecasting complex variables such as agricultural prices and production influenced by weather conditions.

Fig. 14. Actual data and VAR prediction graphs.

In addition to visual methods, MAE and MAPE were used to test the model's performance. The code in Figure 15 describes the calculation of MAE and MAPE to assess the performance of the ARIMA model in predicting data. For each dataset, MAE and MAPE are calculated using the mean_absolute_error and mean_absolute_percentage_error functions from the sklearn.metrics module. The index slices param_price[1]:, param_production[1]:, param_rain[1]:, and param_sun [1]: are used to ensure comparisons are made on data corresponding to the previously applied differencing to avoid NaN values from skewing the calculations. This evaluation provides an overview of the model's accuracy in predicting the variables based on the resulting error values.

Based on Tables IV and V, which show MAE and MAPE for each dataset, the price data for the ARIMA model has scores of 687.61 and 1.34% with normalization and 691.11 and 1.34% without normalization. On the other hand, the VAR model produces scores of 14,516.09 and 32.23% with normalization, which increase significantly to 19,561.34 and 41.92% without normalization. For production data, the ARIMA model achieves 4,796.57 and 7.07% with normalization, and 5,501.05 and 7.89% without normalization. The VAR model shows identical scores with ARIMA for both normalized and non-normalized production data. For rainfall data, the ARIMA model achieves 1.47 and 38.59% with normalization and 1.44 and 38.54% without normalization. The VAR model produces identical scores of 1.65 and 45.18% for both normalized and non-normalized data. Meanwhile, the sunlight duration data for the ARIMA model records identical scores of 0.29 and 7.33% regardless of normalization. Similarly, the VAR model achieves consistent scores of 0.39 and 10.34% with and without normalization. Therefore, normalization proves to be moderately effective for the price and production data, with the ARIMA model showing the best performance when paired with min-max normalization. This combination consistently outperforms the VAR model in terms of accuracy and reliability across all datasets, establishing it as the superior approach in this study.

TABLE IV. MAE AND MAPE SCORES FROM ARIMA

Data	MAE ARIMA		MAPE ARIMA	
	min-max	Without min-max	min-max	Without min-max
Price	687.61	691.11	1.34%	1.34%
Production	4796.57	5501.05	7.07%	7.89%
Rainfall	1.47	1.44	38.59%	38.54%
Sunshine duration	0.29	0.29	7.33%	7.33%

TABLE V. MAE AND MAPE SCORES FROM VAR

Data	MAE VAR		MAPE VAR	
	min-max	Without min-max	min-max	Without min-max
Price	14516.09	19561.3444	32.23%	41.92%
Production	4796.57	5501.05	7.07%	7.89%
Rainfall	1.65	1.6546	45.18%	45.18%
Sunshine duration	0.39	0.39	10.34%	10.34%

To further validate the effectiveness of min-max normalization compared to other normalization methods, a comparative analysis between min-max and Z-score normalization was performed. This evaluation aimed to assess whether the performance of the ARIMA model could be enhanced by using an alternative normalization method. By examining MAE and MAPE values, this analysis provides a comprehensive overview of the advantages and limitations of each normalization technique. The aim is to determine the most suitable normalization method to achieve consistent and reliable predictions across various datasets.

TABLE VI. COMPARISON OF MAE AND MAPE SCORES FROM ARIMA WITH MIN-MAX AND Z-SCORE

Data	MAE ARIMA		MAPE ARIMA	
	min-max	Z-score	min-max	Z-score
Price	687.61	687.61	1.34%	1.34%
Production	4796.57	4771.61	7.07%	7.02%
Rainfall	1.47	1.49	38.59%	39.28%
Sunshine duration	0.29	0.29	7.33%	7.33%

Based on the results presented in Table VI, the price and sunshine duration data exhibit identical performance scores for both min-max and Z-score normalization methods. However, notable differences arise when examining the production and rainfall data. For the production data, Z-score normalization slightly outperforms min-max, as evidenced by a 0.52% lower MAE score. Conversely, for the rainfall data, min-max normalization demonstrates superior performance, with a 1.36% lower MAE compared to Z-score normalization. Although both methods achieve similar overall results, the larger margin of improvement observed in rainfall data with min-max normalization indicates its relative advantage. As a result, the combination of the ARIMA model with min-max normalization proves to be the optimal approach to achieve superior predictive performance in this study.

```python
import numpy as np
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_absolute_percentage_error

# Model performance in monthly data
# MAE, NMAE, and MAPE for Price
mae_price_monthly = mean_absolute_error(
    df_actual_monthly['price_idr'][param_price[1]:],
    df_prediction_forecast_monthly['prediction_forecast_price']
        [param_price[1]:-forecast_months])
nmae_price_monthly = mae_price_monthly / np.mean(
        df_actual_monthly['price_idr'][param_price[1]:])
mape_price_monthly = mean_absolute_percentage_error(
    df_actual_monthly['price_idr'][param_price[1]:],
    df_prediction_forecast_monthly['prediction_forecast_price']
        [param_price[1]:-forecast_months])

# MAE, NMAE, and MAPE for Production
mae_production_monthly = mean_absolute_error(
    df_actual_monthly['production_quintals'][param_production[1]:],
    df_prediction_forecast_monthly['prediction_forecast_production']
        [param_production[1]:-forecast_months])
nmae_production_monthly = mae_production_monthly / np.mean(
        df_actual_monthly['production_quintals'][param_production[1]:])
mape_production_monthly = mean_absolute_percentage_error(
    df_actual_monthly['production_quintals'][param_production[1]:],
    df_prediction_forecast_monthly['prediction_forecast_production']
        [param_production[1]:-forecast_months])

# MAE, NMAE, and MAPE for Rainfall
mae_rain_monthly = mean_absolute_error(
    df_actual_monthly['rain_mm'][param_rain[1]:],
    df_prediction_forecast_monthly['prediction_forecast_rain']
        [param_rain[1]:-forecast_months])
nmae_rain_monthly = mae_rain_monthly / np.mean(
        df_actual_monthly['rain_mm'][param_rain[1]:])
mape_rain_monthly = mean_absolute_percentage_error(
    df_actual_monthly['rain_mm'][param_rain[1]:],
    df_prediction_forecast_monthly['prediction_forecast_rain']
        [param_rain[1]:-forecast_months])

# MAE, NMAE, and MAPE for Sunshine Duration
mae_sunshine_monthly = mean_absolute_error(
    df_actual_monthly['sun_hours'][param_sun[1]:],
    df_prediction_forecast_monthly['prediction_forecast_sun']
        [param_sun[1]:-forecast_months])
nmae_sunshine_monthly = mae_sunshine_monthly / np.mean(
        df_actual_monthly['sun_hours'][param_sun[1]:])
mape_sunshine_monthly = mean_absolute_percentage_error(
    df_actual_monthly['sun_hours'][param_sun[1]:],
    df_prediction_forecast_monthly['prediction_forecast_sun']
        [param_sun[1]:-forecast_months])
```

Fig. 15. Model testing with MAE and MAPE.

D. Interpretation of Results

Based on the results and analysis, it can be concluded that applying ARIMA to the price, production, rainfall, and sunshine duration data results in models with different parameters, leading to varying predictions and forecasts tailored to each dataset. The model parameters are determined through two stages: identifying the degree of differencing to achieve data stationarity and determining significant lag values using PACF and ACF plots. As shown in Figure 16, an important consideration during the differencing stage is that data with a differencing degree of one or more will contain NaN values in the top rows corresponding to the degree of differencing. These NaN rows should be removed to prevent them from affecting model performance and data visualization.

```
Differenced price data
without dropping NaN values
0          NaN
1     0.000000
2     0.000000
3     0.000000
4     0.000000
5     0.000000
6     0.000000
7     0.000000
8    -0.189241
9     0.038866
```

Fig. 16. Example of data corrupted by differencing.

Figure 13 shows a combination of ARIMA and min-max normalization forecasts for prices, production, rainfall, and sunshine duration. The prediction lines closely follow the actual data patterns, although there are some differences and smoother fluctuations compared to the actual data, which shows sharp spikes. The price of red chili peppers tends to increase when production decreases due to reduced market supply and vice versa. However, predictions for weather variables such as rainfall show greater deviations, indicating that weather factors may be more challenging to predict with this model.

According to the MAPE classification criteria in [30], shown in Table VII, the MAPE scores produced by each ARIMA model indicate that the ARIMA (1, 1, 1) model for price data is highly accurate in forecasting, the ARIMA (6, 1, 12) model for production data is also highly accurate, the ARIMA (6, 0, 6) model for rainfall data is reasonably accurate, and the ARIMA (1, 0, 1) model for sunlight duration data is highly accurate.

TABLE VII. MAPE CRITERIA ACCORDING TO [30]

No.	MAPE	Interpretation
1.	< 10%	Very accurate in prediction
2.	10% - 20%	Accurate in prediction
3.	20% - 50%	Fairly accurate in prediction
4.	> 50%	Not accurate in prediction

IV. CONCLUSION

A. Conclusion

The novelty of this research lies in the implementation of ARIMA with min-max normalization on price, production, rainfall, and sunlight duration data. Based on the analysis conducted, the forecasting results closely follow the actual data patterns, despite minor differences. The model evaluation shows:

- Price data has an MAE of Rp687.6107 and MAPE of 1.34%, indicating that the ARIMA (1, 1, 1) model is very accurate in predicting prices.
- Production data has an MAE of 4796.5704 quintals and MAPE of 7.07%, indicating that the ARIMA (6, 1, 12) model is very accurate in predicting production.
- Rainfall data has an MAE of 1.4676 mm and MAPE of 38.59%, indicating that the ARIMA (6, 0, 6) model is fairly accurate in predicting rainfall.
- Sunshine duration data has an MAE of 0.2894 hours and MAPE of 7.33%, indicating that the ARIMA (1, 0, 1) model is very accurate in predicting sunshine duration.

These results demonstrate that min-max normalization as part of data preprocessing plays a significant role in enhancing the accuracy of the ARIMA model. By applying min-max normalization, data variability can be minimized, making the model more sensitive to actual pattern changes. This approach allows the model to capture data fluctuations more accurately and improve its estimates. Therefore, the ARIMA model, using min-max normalization, is effective in forecasting the prices and production quantities of red chili peppers in the North Sumatra province. These predictions provide valuable insights for farmers and policymakers in anticipating price and production changes, allowing the development of more effective strategies in the face of weather variability. For farmers, the results of this study can support production planning, weather risk mitigation, and strengthening the bargaining position through optimal harvest and sales timing. Meanwhile, for policymakers, these predictions provide an empirical basis for developing price stabilization policies, improving agricultural infrastructure, and managing resources that are more adaptive to climate change. Furthermore, this method can be extended to other agricultural commodities, such as rice or corn, and applied in different regions by adjusting local parameters, thus supporting a specific and data-driven approach. This research also opens up opportunities for integration with advanced technologies such as machine learning or IoT sensors to improve prediction accuracy and support the sustainability of the agricultural sector.

B. Suggestions

There are several suggestions for future research. First, the grid search method can be applied to automatically determine the optimal ARIMA parameters. Next, the Normalized Mean Absolute Error (NMAE) metric can be used as a model performance evaluation metric instead of MAPE, as it can address MAPE's limitation of occasionally providing excessively high or low values for data with small values.

Furthermore, the limited dataset period of two years in this study is due to data availability constraints. Chili pepper price data from the BPN price monitoring platform are only accessible starting from March 2021, while the production data provided by the BPN Data and Information Center are available only up to December 2023. Expanding data availability in the future will allow researchers to conduct analyses over longer periods, potentially improving the reliability of predictions.

Additionally, it is recommended that the BMKG Database Center complete the availability of rainfall and sunshine duration data at each station to obtain valid research results. Following these suggestions, future research is expected to provide more accurate and beneficial results.

REFERENCES

[1] H. Rana, M. U. Farooq, A. K. Kazi, M. A. Baig, and M. A. Akhtar, "Prediction of Agricultural Commodity Prices using Big Data Framework," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12652–12658, Feb. 2024, https://doi.org/10.48084/etasr.6468.

[2] C. S. e Souza, S. A. Duah, A. Neményi, Z. Pék, and L. Helyes, "The impact of cultivar and irrigation on yield, leaf surface temperature and SPAD readings of chili pepper," *Acta Agraria Debreceniensis*, no. 2, pp. 103–108, Dec. 2020, https://doi.org/10.34101/ACTAAGRAR/2/4286.

[3] A. Lukas *et al.*, "Fresh Chili Agribusiness: Opportunities and Problems in Indonesia," in *Agricultural Economics and Agri-Food Business*, IntechOpen, 2023.

[4] R. Rahma, R. F. Sari, and S. Dur, "Multivariate Singular Spectrum Analysis Model in Forecasting Red Chili and Cayenne Pepper Prices," *Al Ulum: Jurnal Sains Dan Teknologi*, vol. 10, no. 1, pp. 14–22, Apr. 2024, https://doi.org/10.31602/jst.v10i1.14281.

[5] E. Mpaata, N. Koskei, and E. Saina, "Financial Literacy and Saving Behavior Among Micro and Small Enterprise Owners in Kampala, Uganda: the Moderating Role of Social Influence," *Journal of Economics, Finance and Accounting Studies*, vol. 2, no. 1, pp. 22–34, Jun. 2020.

[6] D. Devianto, E. Wahyuni, M. Maiyastri, and M. Yollanda, "The seasonal model of chili price movement with the effect of long memory and exogenous variables for improving time series model accuracy," *Frontiers in Applied Mathematics and Statistics*, vol. 10, Jul. 2024, https://doi.org/10.3389/fams.2024.1408381.

[7] H. Sasai, "The Effects of Rainfall Distribution and Intensity on Crop Production," *Namimbia Economist*. https://economist.com.na/57635/columns/the-effects-of-rainfall-distribution-and-intensity-on-crop-production/.

[8] M. Kotz, A. Levermann, and L. Wenz, "The effect of rainfall changes on economic production," *Nature*, vol. 601, no. 7892, pp. 223–227, Jan. 2022, https://doi.org/10.1038/s41586-021-04283-8.

[9] S. M. Khasanah, M. Maksum, and E. Suwondo, "Trend Analysis of Red Chili Price-Formation Models," *agriTECH*, vol. 40, no. 1, pp. 57–63, Mar. 2020, https://doi.org/10.22146/agritech.45946.

[10] D. Novita, T. Supriana, Sirozujilam, and S. N. Lubis, "Strategy of Development of Sustainable Red Chili Agribusiness Areas in North Sumatra Province," *Journal of Ecohumanism*, vol. 3, no. 7, pp. 4983–4997, Nov. 2024, https://doi.org/10.62754/joe.v3i7.4607.

[11] S. Mardiyati and M. Natsir, "Fluctuations and trends in the prices of red chilies and cayenne peppers in the traditional markets of Makassar City," *IOP Conference Series: Earth and Environmental Science*, vol. 1302, no. 1, Oct. 2024, Art. no. 012124, https://doi.org/10.1088/1755-1315/1302/1/012124.

[12] Y. D. S. Saragih, R. Pambudy, and T. G. Dewi, "Dampak Luas Lahan terhadap Kinerja Usahatani Cabai Merah (Kasus Provinsi Sumatera Utara)," *JIA (Jurnal Ilmiah Agribisnis) : Jurnal Agribisnis dan Ilmu Sosial Ekonomi Pertanian*, vol. 8, no. 6, pp. 486–496, Dec. 2023, https://doi.org/10.37149/jia.v8i6.914.

[13] T. Surbakti, T. Supriana, and I. Iskandarini, "Asymmetric Price Transmission of Red Chili Market in North Sumatra Province, Indonesia," *Agro Bali : Agricultural Journal*, vol. 5, no. 1, pp. 156–165, Feb. 2022, https://doi.org/10.37637/ab.v5i1.896.

[14] Y. Zhang, H. Yang, H. Cui, and Q. Chen, "Comparison of the Ability of ARIMA, WNN and SVM Models for Drought Forecasting in the Sanjiang Plain, China," *Natural Resources Research*, vol. 29, no. 2, pp. 1447–1464, Apr. 2020, https://doi.org/10.1007/s11053-019-09512-6.

[15] "Comparison of ARIMA, ANN and Hybrid ARIMA-ANN Models for Time Series Forecasting," *Information Sciences Letters*, vol. 12, no. 2, pp. 1003–1016, Feb. 2023, https://doi.org/10.18576/isl/120238.

[16] L. M. Hamzah, S. U. Nabilah, E. Russel, M. Usman, E. Virginia, and Wamiliana, "Dynamic Modelling and Forecasting of Data Export of Agricultural Commodity by Vector Autoregressive Model," *Journal of Southwest Jiaotong University*, vol. 55, no. 3, 2020, Art. no. 41, https://doi.org/10.35741/issn.0258-2724.55.3.41.

[17] N. Nainggolan, H. A. H. Komalig, and T. Manurung, "Vector autoregressive time series model in predicting food prices in Manado city," *AIP Conference Proceedings*, vol. 2694, no. 1, Apr. 2023, Art. no. 050005, https://doi.org/10.1063/5.0119696.

[18] Y. Peng, "Construction and Evaluation of Credit Risk Early Warning Indicator System of Internet Financial Enterprises Based On AI and Knowledge Graph Theory," *Procedia Computer Science*, vol. 243, pp. 918–927, 2024, https://doi.org/10.1016/j.procs.2024.09.110.

[19] Y. S. Kim, M. K. Kim, N. Fu, J. Liu, J. Wang, and J. Srebric, "Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models," *Sustainable Cities and Society*, vol. 118, Jan. 2025, Art. no. 105570, https://doi.org/10.1016/j.scs.2024.105570.

[20] H. Henderi, T. Wahyuningsih, and E. Rahwanto, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *International Journal of Informatics and Information Systems*, vol. 4, no. 1, pp. 13–20, Mar. 2021, https://doi.org/10.47738/ijiis.v4i1.73.

[21] I. D. Aulia and I. Pratama, "Analysis of Forecasting Methods on Rice Price Data at Milling Level According to Quality," *Edu Komputika Journal*, vol. 11, no. 1, pp. 1–10, Aug. 2024, https://doi.org/10.15294/edukom.v11i1.4763.

[22] R. Konda, V. S. S. Kumar, R. C. Bagadi, and N. S. Kumar, "A Novel Ensemble Linear Regression Scheme Based on Incorporation of the Linear Model Strength Contributed by Each Data Point Co-ordinate Using a Special Weighted Error Metrics Ensembling Scheme Spanning Error Metric Types and Considered Linear Regression Models – An Example of Construction Accident (Fatal Falls) Prediction," *Asian Research Journal of Current Science*, pp. 84–106, Apr. 2023.

[23] Y. Ledmaoui, A. El Maghraoui, M. El Aroussi, R. Saadane, A. Chebak, and A. Chehri, "Forecasting solar energy production: A comparative study of machine learning algorithms," *Energy Reports*, vol. 10, pp. 1004–1012, Nov. 2023, https://doi.org/10.1016/j.egyr.2023.07.042.

[24] D. F. McCaffrey, "Volume 14: Quantitative Research and Educational Measurement," in *International Encyclopedia of Education*, 4th ed., Elsevier, 2023, pp. xix–xxiv.

[25] S. K. Filipova-Petrakieva and V. Dochev, "Short-Term Forecasting of Hourly Electricity Power Demand: Reggresion and Cluster Methods for Short-Term Prognosis," *Engineering, Technology & Applied Science Research*, vol. 12, no. 2, pp. 8374–8381, Apr. 2022, https://doi.org/10.48084/etasr.4787.

[26] R. Ospina, J. A. M. Gondim, V. Leiva, and C. Castro, "An Overview of Forecast Analysis with ARIMA Models during the COVID-19 Pandemic: Methodology and Case Study in Brazil," *Mathematics*, vol. 11, no. 14, Jul. 2023, Art. no. 3069, https://doi.org/10.3390/math11143069.

[27] C. Aliferis and G. Simon, "Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI," in *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences: Best Practices and Pitfalls*, G. J. Simon and C. Aliferis, Eds. Springer International Publishing, 2024, pp. 477–524.

[28] F. Alshammari, N. Aljojo, A. Tashkandi, A. Alghoson, A. Banjar, and N. K. E. Abbadi, "A Hybrid Time-Series Prediction of the Greater Riyadh's Metropolitan Area Expansion," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11890–11897, Oct. 2023, https://doi.org/10.48084/etasr.6350.

[29] N. E. Samosir, N. Siagian, R. R. Nst, S. Frisnoiry, and K. Kunci, "The Effect of Demand And Supply On The Economy," *Jurnal Ilmiah Multi Disiplin Indonesia*, vol. 2, no. 8, pp. 1799–1805, Jun. 2023.

[30] C. D. Lewis, Industrial and Business Forecasting Methods: A Practical Guide to Exponential Smoothing and Curve Fitting. Butterworth Scientific, 1982.

AUTHORS PROFILE

Ifan Prihandi is a PhD student in the Faculty of Information Technology in Universitas Kristen Satya Wacana, Indonesia. He earned his B.S. and M.Sc. degrees from Universitas Budi Luhur in 2012 and 2014. Currently, he serves as an Assistant Expert at the Information Systems Study Program, Faculty of Computer Science, Universitas Mercu Buana. His research interests include computational techniques and applied informatics.

Sutarto Wijono is a Professor at the Faculty of Psychology, Universitas Kristen Satya Wacana, Indonesia, known for his contributions to the field of Industrial and Organizational Psychology. He earned his B.S. degree in Education from Universitas Kristen Satya Wacana in 1987, followed by a M.Sc. in Industrial Psychology from Universiti Kebangsaan Malaysia in 1997. He later completed his Ph.D. studies in Industrial Psychology at Universitas Indonesia in 2005. Prof. Sutarto was appointed as the first Professor of Psychology at Universitas Kristen Satya Wacana on 2011, and currently serves as the Head of the Master's Program in Psychological Sciences at Universitas Kristen Satya Wacana. His academic contributions include several books and research articles.

Irwan Sembiring was born in 1976. He earned his B.S. degree in Engineering (S.T.) from Universitas Pembangunan Nasional Veteran Yogyakarta in 2001, followed by a M.Sc. degree in Computer Science (M.Kom.) from Universitas Gadjah Mada in 2004. He later completed his Ph.D. studies at Universitas Gadjah Mada, obtaining in 2015. Dr. Irwan serves as the Head of the Doctoral Program in Computer Science at the Faculty of Information Technology, Universitas Kristen Satya Wacana, Indonesia. His research focuses on network security, information systems, and blockchain technology.

Evi Maria is an Associate Professor of accounting at the Information Systems Department, Universitas Kristen Satya Wacana, Indonesia. She earned her B.S. degree from Universitas Kristen Satya Wacana in 2002. She obtained her M.Sc. in Accounting (2009) and Ph.D. in Accounting Science (2018) from Universitas Gadjah Mada, where she also received her professional accountant certification in 2008. Her research focuses on public accounting and accounting information systems.