Published online on the journal's website:https://jes-tm.org/index.php/jestm/index

Journal of Engineering Science and Technology Management

| ISSN (Online) 2828 -7886 |

Article

Predicting Chicken Egg Production Using the Seasonal Autoregressive Integrated Moving Average (SARIMA) Method at PT. Pulau Mandiri Jaya Farm

Daniel Prasetyo[1]

[1]Informatics Engineering Study Program, Indo Global Mandiri University, Palembang

DOI: 10.31004/jestm.v6i1.383

E-mail:danielpr1606@gmail.com

ARTICLE INFORMATION

Volume 6 Issue 1
Received: 16 January 2026
Accepted: 19 January 2026
Publish Online: 31 March 2026
On line: at https:JESTM.org/

Keywords

ACF Analysis,
SARIMA Model,
Seasonal Forecasting,
Egg Production,
Time Series

ABSTRACT

Chicken egg production at PT. Pulau Mandiri Jaya Farm for the 2021-2024 period showed significant monthly fluctuations (2,140-6,890 tons) due to seasonal patterns and feed management challenges, requiring accurate time series forecasting for optimal planning. This study aims to predict egg production using the SARIMA(1,1,1)(1,1,1,12) model to capture annual seasonality. This quantitative descriptive study analyzed monthly univariate data from 48 observations, with purposive sampling (80% training: 38 months; 20% testing: 10 months). Data analysis used Python statsmodels for stationarity testing, ACF/PACF identification, model fitting, and MSE evaluation. The results showed excellent model performance (MSE=0.3262; Ljung-Box p=0.44; Jarque-Bera p=0.65), accurately predicting 2025 production (3,886-6,873 tons). In conclusion, the SARIMA model enables precision feed stock planning (60-70% of production costs), improving operational efficiency and food security in South Sumatra.

1. Introduction

Chicken eggs are an easily accessible source of protein and essential nutrients at a relatively affordable price compared to other animal proteins. The protein, fat, vitamin, and mineral content in eggs supports daily human nutritional needs, but production fluctuations often occur due to seasonal factors and farm management. At PT. Pulau Mandiri Jaya Farm, which has been operating since 2020, red chicken egg production data for the 2021-2024 period shows significant monthly variations, ranging from 2,140 tons to 6,890 tons, reflecting seasonal patterns that require accurate time series analysis (Muhamad et al., 2022; Zuhdi et al., 2023).

Increasing egg consumption in the community drives the need for stable production, but challenges such as eggshell cracks and uncertain feed stocks hamper efficiency. Laying hen farms face economic losses from shell damage, which facilitates microbial contamination, while feed costs account for 60-70% of total production. This phenomenon is evident at PT. Pulau Mandiri Jaya Farm, where stable egg production depends on timely feed planning (Nuriyana, 2020; Sari, 2021).

Instability in egg production at PT. Pulau Mandiri Jaya Farm is caused by the inability to accurately predict feed requirements, resulting in stock fluctuations and resource waste. Forty-eight months of historical data show extreme production variations, with a sharp decline in March 2022 (2,140 tons) and a peak in September 2024 (6,890 tons), difficult to capture with conventional forecasting models. This issue exacerbates operational inefficiencies and difficulties meeting market demand (Nurulita, 2020; Suyono et al., 2022).

Traditional prediction methods fail to address the seasonal patterns of egg production data, which exhibits an annual cycle with a 12-month period based on ACF and PACF plots. Machine learning is generally less effective for univariate time series data with strong trends and seasonality, while feed stock planning requires high accuracy to minimize MSE. This challenge is crucial for large-scale farms like PT. Pulau Mandiri Jaya Farm in maintaining supply continuity (Journal et al., 2023; Junaidi, 2023).

The lack of robust predictive models leads to discrepancies between actual and forecasted production, with high MSEs for conventional methods such as Moving Average or Exponential Smoothing. In the local context of South Sumatra, environmental factors and feed management complicate forecasting, necessitating specialized approaches for long-season data. This hampers companies' strategic decision-making in supply chain optimization (Tarisya & Primandari, 2023; Hudzaifah & Rismayadi, 2021).

This study aims to predict chicken egg production at PT. Pulau Mandiri Jaya Farm using the SARIMA(1,1,1)(1,1,1,12) method to capture annual seasonal patterns, with a minimum MSE target of 0.33 based on 2021-2024 data. The urgency lies in the company's need to optimize feed stocks (60-70% of production costs) and reduce losses from production fluctuations, supporting local food security in South Sumatra. Its novelty uses SARIMA for Indonesian company-specific egg production data with a 48-month period, surpassing previous studies limited to non-seasonal ARIMA or other commodities, as well as complete diagnostic validation (ACF/PACF, Ljung-Box, Jarque-Bera) for superior accuracy (Durrah et al., 2018; Kusyanto et al., 2020).

2. Literature Review

2.1. Chicken Egg Production and Nutritional Significance

Chicken eggs represent one of the most accessible and affordable sources of animal protein globally, containing essential nutrients including high-quality protein, fats, vitamins, and minerals that support daily human nutritional needs. In Indonesia, the laying hen farming sector continues to demonstrate strong growth prospects, driven by rising domestic demand and the need for stable egg supply to support food security. Production fluctuations, however, remain a persistent challenge — influenced by seasonal factors, farm management practices, and feed availability — making accurate forecasting essential for supply chain stability.

2.2. Time Series Forecasting in Agricultural Production

Time series analysis has been widely applied to agricultural commodity forecasting, enabling researchers to identify trends and seasonal patterns in production data. The ARIMA (Autoregressive Integrated Moving Average) model is among the most commonly used methods for this purpose, having been applied to egg production, rice, and other agricultural commodities in Indonesia. Studies in the Indonesian context have demonstrated that

ARIMA-based models consistently outperform simpler methods such as Moving Average and Exponential Smoothing when data exhibits strong trends and autocorrelation.

2.3. SARIMA: Theoretical Framework and Model Identification

The Seasonal ARIMA (SARIMA) model extends the conventional ARIMA framework by incorporating seasonal components expressed as SARIMA(p,d,q)(P,D,Q)s, where *s* denotes the seasonal period. Model identification relies on Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots: significant spikes at seasonal lags (e.g., lag 12 and 24 for monthly data) indicate the need for seasonal AR and MA terms, while the non-seasonal order is determined from the pattern of spikes at lower lags. Achieving stationarity prior to model fitting requires both non-seasonal differencing (d) and seasonal differencing (D), verified visually and through statistical tests.

2.4. SARIMA Applications in Seasonal Agricultural Forecasting

SARIMA has proven effective for forecasting agricultural commodities that exhibit strong annual seasonality. A study applying SARIMA to native chicken egg production in Magelang City, Indonesia, identified the best model as SARIMA(2,1,2)(1,0,1,12), achieving an R^2 of 0.89 and MAPE of 7.21% on the testing set — demonstrating satisfactory performance for monthly poultry data. Similarly, SARIMA was applied to chicken egg price data in Indonesia, with the seasonal component (s=12) effectively capturing annual price cycles influenced by festive seasons and supply-demand dynamics.

2.5. Residual Diagnostics and Model Validation

A well-fitted SARIMA model must satisfy three key diagnostic criteria: absence of autocorrelation in residuals (assessed via the Ljung-Box test), normality of residuals (assessed via the Jarque-Bera test), and absence of heteroscedasticity (assessed via the Breusch-Pagan test). Failure to meet these assumptions indicates model misspecification and unreliable forecasts, whereas passing all diagnostics confirms that the model has adequately captured the underlying data structure. The MSE (Mean Squared Error) serves as the primary accuracy metric for comparing actual versus predicted values on the test set, with lower values indicating superior predictive performance.

2.6. Machine Learning vs. SARIMA for Univariate Time Series

While machine learning (ML) approaches such as neural networks and support vector regression offer flexibility for complex, multivariate data, they are generally less effective for univariate time series with strong trends and seasonal patterns when data are limited. Studies specifically comparing ML and statistical time series models for poultry egg production found that SARIMA-type models provide interpretable, statistically validated outputs better suited for feed planning and operational decision-making under limited data conditions. A study using the Polak-Ribière neural network algorithm for laying hen egg production in Sumatra achieved an MSE of 0.031 on the best architecture (4-25-1), illustrating that ML requires larger datasets and careful architecture tuning to achieve comparable accuracy.

2.7. Feed Management and Production Efficiency in Laying Hens

Feed costs constitute the dominant expenditure in laying hen operations, accounting for approximately 60–70% of total production costs, making accurate production forecasting directly linked to feed stock planning efficiency. Research on feed efficiency in laying hens confirms that nutritional management — including precise feed formulation and timely delivery — significantly improves the Hen-Day Production (HDP) rate and reduces waste. In the South Sumatra context, fluctuating production volumes exacerbate supply chain inefficiencies, underscoring the practical value of SARIMA-based forecasting for operational planning.

3. Research Methodology

3.1. Types and Methods of Research

This quantitative descriptive study focuses on univariate time series analysis to identify trends and seasonal patterns in monthly egg production data from 2021 to 2024. The primary method is SARIMA(p,d,q)(P,D,Q)s with parameters (1,1,1)(1,1,1,12) selected based on stationarity tests, ACF and PACF plots, and residual diagnostics, as described by Sugiyono (2022) who emphasizes a time series approach for seasonal data. This approach complements the conventional ARIMA method with a seasonal component to capture the annual cycle (s=12), as developed by Creswell and Creswell (2023) within an adaptive mixed methods framework for predictive research.

3.2. Data Analysis Instruments and Techniques

The main instrument is secondary data in the form of daily egg production records aggregated into monthly data (48 observations), validated through literature studies and field observations at PT. Pulau Mandiri Jaya Farm. Analysis techniques include data division (80% training, 20% testing), stationarity testing through differencing, model identification via ACF/PACF, parameter estimation using Python (statsmodels), and accuracy evaluation using Mean Squared Error (MSE=0.3262). Emzir (2021) supports the use of statistical software such as Python for time series analysis, while Sudaryono (2024) adds the importance of residual validation to ensure independence and normality.

3.3. Population and Sample

The study population comprised all chicken egg production data (without cracked shells) at PT. Pulau Mandiri Jaya Farm from January 2021 to December 2024, totaling 48 monthly data points. The sample was taken purposively using a time series split technique, with 38 training data points (80%) for model development and 10 testing data points (20%) for validation, as recommended by Sugiyono (2022) to avoid overfitting seasonal data. This approach aligns with Creswell and Creswell (2023) who advocate non-probability sampling for predictive case studies with longitudinal data.

3.4. Research Procedures

The procedure begins with data collection through literature review and company datasets, followed by time series visualization to detect seasonal trends. Next, data splitting, stationarity testing (d=1, D=1 differencing), SARIMA parameter identification via ACF/PACF plots, model fitting, diagnostics (Ljung-Box, Jarque-Bera), MSE calculation, and forecasting 12 periods into the future (2025) are performed. Sudaryono (2024) confirms this systematic sequence, while Emzir (2021) emphasizes diagnostic iterations for robustness, concluding with low MSE validation, indicating an accurate model for feed production and stock planning.

4. Results and Discussion

4.1. Data Collection Description

The data used in this analysis is monthly data on chicken egg production in tons from 2021 to 2024. The following table shows the number of chicken eggs produced per month:

Table 1. Chicken Egg Production Data

No	Month	Year / Production Amount (Ton)			
		2021	2022	2023	2024
1	January	3,244	3,728	3,456	4,785
2	February	3,120	2,480	2,938	5,013
3	March	2,450	2,140	3,025	5,248
4	April	2,780	3,408	3,450	5,521
5	May	2,890	3,512	3,230	5,875
6	June	3,076	3,207	3,268	5,438
7	July	2,934	3,386	3,458	6,216
8	August	3.102	3,564	3,426	6,054
9	September	2,894	3,824	3,076	6,890
10	October	3,256	4,100	3,386	6,410
11	November	3,724	3,446	3,446	6,189
12	December	3,896	4,350	4,425	6,558

4.2. Data Visualization

Data visualization is a crucial first step in time series analysis, as it can help understand patterns, trends, and fluctuations within the data. For the chicken egg production data used in this study, visualization was performed by creating a line graph showing the number of chicken eggs produced per month from 2021 to 2024.

1. Time Series Graph Visualization

The first step in stationarity analysis is to visualize the data in the form of a time series graph. This way, we can observe patterns, trends, and fluctuations in the data. A time series graph provides an initial idea of whether the data exhibits an increasing or decreasing trend, as well as whether there is a seasonal pattern. The following is the coding andthe results.

```
import pandas as pd
import matplotlib.pyplot as plt data
= {
'Month': ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
'September', 'October', 'November', 'December'],
'2021': [3244, 3120, 2450, 2780, 2890, 3076, 2934, 3102, 2894, 3256,
3724, 3896],
'2022': [3728, 2480, 2140, 3408, 3512, 3207, 3386, 3564, 3824, 4100,
3446, 4350],
'2023': [3456, 2938, 3025, 3450, 3230, 3268, 3458, 3426, 3076, 3386,
3446, 4425],
'2024': [4785, 5013, 5248, 5521, 5875, 5438, 6216, 6054, 6890, 6410,
6189, 6558],
# Creating a DataFrame
}
df = pd.DataFrame(data)
df.set_index('Bulan', inplace=True) # Set
the size of the graph plt.figure(figsize=(12,
6))
plt.plot(df[['2021', '2022', '2023', '2024']], label=['Chicken Egg Production 2021', 'Chicken Egg Production 2022', 'Chicken Egg Production 2023', 'Chicken Egg Production 2024'], marker='o')
```

```
plt.title('Time Series Graph of Chicken Egg
Production 2021-2023') plt.xlabel('Month')
p
```

4.3. Split Data

This study uses an 80%:20% split of time series data. One example of how to split the data is by dividing 80% of the training data (32 months) and 20% of the test data (8 months). This means that the training data = 80% x the number of time series data. Meanwhile, the test data = 20% x the number of time series data. The data split is 80%:20%.

Table 2. Training Data 80%

No	Month	Year	Production result
1	January	2021	3,244
2	February	2021	3,120
3	March	2021	2,450
4	April	2021	2,780
5	May	2021	2,890
6	June	2021	3,076
7	July	2021	2,934
8	August	2021	3.102
9	September	2021	2,894
10	October	2021	3,256
11	November	2021	3,724
12	December	2021	3,896
13	January	2022	3,728
14	February	2022	2,480
15	March	2022	2,140
16	April	2022	3,408
17	May	2022	3,512
18	June	2022	3,207
19	July	2022	3,386
20	August	2022	3,564
21	September	2022	3,824
22	October	2022	4,100
23	November	2022	3,446
24	December	2022	4,350
25	January	2023	3,456
26	February	2023	2,938
27	March	2023	3,025
28	April	2023	3,450
29	May	2023	3,230
30	June	2023	3,268
31	July	2023	3,458
32	August	2023	3,426
33	September	2023	3,076
34	October	2023	3,386
35	November	2023	3,446
36	December	2023	4,425
37	January	2024	4,785
38	February	2024	5,013

Table 3. 20% Test Data Table

No	Month	Year	Production result
39	March	2024	5,248
40	April	2024	5,521
41	May	2024	5,875
42	June	2024	5,438
43	July	2024	6,216
44	August	2024	6,054
45	September	2024	6,890
46	October	2024	6,410
47	November	2024	6,189
48	December	2024	6,558

4.4. ACF and PACF analysis

After performing visual analysis and stationarity testing, the next step is to analyze the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). This analysis is crucial for determining the appropriate parameters for the SARIMA model, namely the p-value (autoregressive order) and q-value (moving average order).

1. Understanding ACF and PACF

- ACF (Autocorrelation Function): The ACF measures the correlation between the current value and previous values at various lags. The ACF helps us understand how much influence past values have on the current value. If the ACF shows a rapid decline, this could be an indication that an AR model might be more appropriate.
- PACF (Partial Autocorrelation Function): The PACF measures the correlation between the current value and previous values after removing the influence of values between them. The PACF helps us determine how much past values contribute to the current value without the influence of intermediate values. If the PACF shows a rapid decline, this could be an indication that an MA model might be more appropriate.

Here is the code to visualize the ACF and PACF of chicken egg production data:

```
import pandas as pdimport
numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Data
months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September',
'October', 'November', 'December']
data = {
'2021': [3244, 3120, 2450, 2780, 2890, 3076, 2934, 3102, 2894, 3256, 3724,
3896],
'2022': [3728, 2480, 2140, 3408, 3512, 3207, 3386, 3564, 3824, 4100, 3446,
4350],
'2023': [3456, 2938, 3025, 3450, 3230, 3268, 3458, 3426, 3076, 3386, 3446,
4425],
'2024': [4785, 5013, 5248, 5521, 5875, 5438, 6216, 6054, 6890, 6410, 6189,
6558]
}

# Create DataFrame
df = pd.DataFrame(data, index=month)

# Convert to time series
time_series = df.stack().reset_index(name='Value')
time_series.rename(columns={'level_0': 'Month', 'level_1': 'Year'}, inplace=True)
```

```
# Plot ACF and PACF plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plot_acf(time_series['Value'], ax=plt.gca(), lags=20) # Now using the 'Value'
column
plt.title("Autocorrelation Function
(ACF)")plt.subplot(1, 2, 2)
plot_pacf(time_series['Value'], ax=plt.gca(), lags=20) # Now using the 'Value'
column
plt.title("Partial Autocorrelation Function (PACF)")
```

The following are the coding results to determine ACF and PACF:

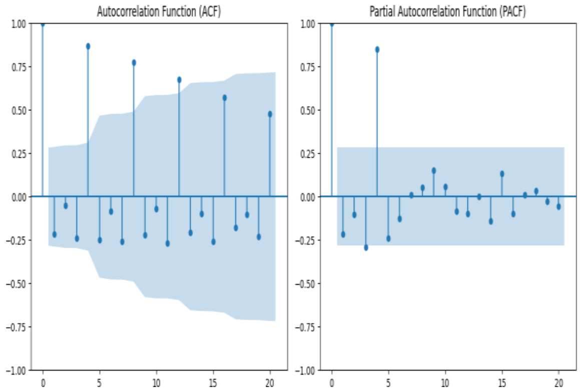


Figure 2. ACF and PACF Results

The results of the ACF and we will get two graphs: one for the ACF and one for the PACF. Here are some important points to interpret:

- ACF Chart:
  - If the ACF shows a rapid decline (e.g., only the first few lags are significant), this suggests that an AR model may be more appropriate.
  - If the ACF shows a slow decline, this could indicate that the data has a strong seasonal component.
- PACF graph:
  - If the PACF shows a rapid decline, this suggests that an MA model may be more appropriate.
  - If the PACF shows a slow decline, this could indicate that there is a lot of lag contributing to the current value.

After conducting the ACF and PACF tests, differencing will be performed to eliminate seasonal pattern trends. The following is the coding for differencing.

The following are the coding results to determine differencing:

```
import pandas as pdimport numpy as np
import matplotlib.pyplot as plt import seaborn as sns
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf # Data
months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November',
'December']
data = {
'2021': [3244, 3120, 2450, 2780, 2890, 3076, 2934, 3102, 2894, 3256,
3724, 3896],
'2022': [3728, 2480, 2140, 3408, 3512, 3207, 3386, 3564, 3824, 4100,
3446, 4350],
'2023': [3456, 2938, 3025, 3450, 3230, 3268, 3458, 3426, 3076, 3386,
3446, 4425],
'2024': [4785, 5013, 5248, 5521, 5875, 5438, 6216, 6054, 6890,
6410, 6189, 6558]
}
# Differentencing (without ADF test) diff_series = np.diff(time_series['Value'])
# Plot differencing data plt.figure(figsize=(10, 5))
sns.lineplot(data=diff_series) plt.title("Differenced Time Series")
plt.xlabel("Index") plt.ylabel("Differenced Value")
plt.show()
```

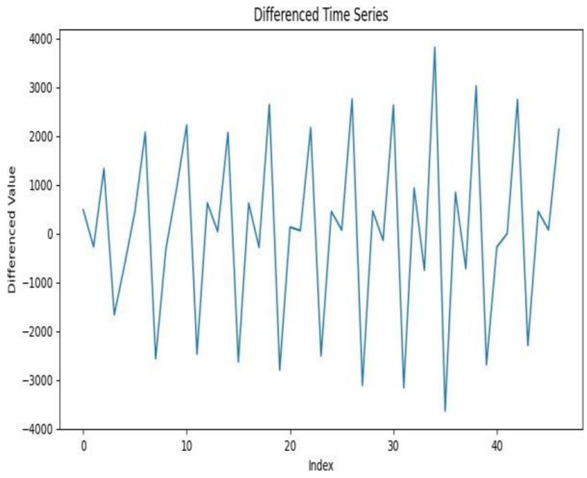


Figure 3. Differentiating Results

2. Application of SARIMA Modeling

After performing visual analysis and determining data stationarity, the next step is to build a SARIMA (Seasonal Autoregressive Integrated Moving Average) model. The SARIMA model is used to analyze and predict time series data with seasonal components. In this model, we will determine the parameters p, d, q, P, D, Q, and s based on the previous analysis.

1. Determining SARIMA Model Parameters

- p: Order of the autoregressive (AR) component.
- d: The amount of differencing required to make the data stationary.
- q: Order of the moving average (MA) component.
- P: Order of the autoregressive seasonal component.
- D: The amount of seasonal differencing required.
- Q: Order of the seasonal moving average component.
- S: Seasonal period

Based on the ACF and PACF analysis, we can determine these values.

- p = 1, order of the autoregressive component (AR)
- d = 1, the number of differencing required to make the data stationary
- q = 1, order of the moving average (MA) component
- P = 1, order of the moving average component
- D = 1, the amount of seasonal differencing required
- Q = 1, the order of the seasonal moving average component
- s = 12, seasonal period

Following is the code to build a SARIMA model using the specified parameters:

```
import pandas as pdimport numpy as np
import matplotlib.pyplot as plt import
statsmodels.api as sm
# Production data in the form of a dataframe data = {
"Month": [
"January", "February", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "December"
],
2021: [3244, 3120, 2450, 2780, 2890, 3076, 2934, 3102, 2894, 3256, 3724,
3896],
2022: [3728, 2480, 2140, 3408, 3512, 3207, 3386, 3564, 3824, 4100, 3446,
4350],
2023: [3456, 2938, 3025, 3450, 3230, 3268, 3458, 3426, 3076, 3386, 3446,
4425],
2024: [4785, 5013, 5248, 5521, 5875, 5438, 6216, 6054, 6890, 6410, 6189,
6558],
}
# Convert to time series format df = pd.DataFrame(data)
df.set_index("Month", inplace=True)
df = df.T # Transpose so that the month becomes the index

# Create one time series column df_ts =
df.stack().reset_index()
df_ts.columns = ["Year", "Month", "Production"]
```

The following are the coding results to determine the SARIMA modeling:

```
# Create month month_mapping = {
"January": 1, "February": 2, "March": 3, "April": 4, "May": 5, "June": 6,
"July": 7, "August": 8, "September": 9, "October": 10, "November": 11,
"December": 12
}
df_ts['Month_num'] = df_ts['Month'].map(month_mapping)
df_ts["Date"] =            pd.to_datetime(df_ts["Year"].astype(str) +"-" +
df_ts["Month_num"].astype(str) + "-01")
df_ts.set_index("Date", inplace=True)df_ts = df_ts["Production"]
# SARIMA Model (1,1,1) x (1,1,1,12)
sarima_model = sm.tsa.SARIMA(df_ts, order=(1, 1, 1), seasonal_order=(1, 1, 1, 12))
sarima_result = sarima_model.fit()


# Summary of model results sarima_result.summary()
```

Model:	SARIMA (1, 1, 1)x(1, 1, 1, 12)	Log Likelihood	-269,803
Date:	Wed, 12 Feb 2025	AIC	549,607
Time:	19:19:41	BIC	557,383
Sample:	01-01-2021	HQIC	552,291
	- 01-12-2024		

	coefficient	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.9988	0.450	-2,220	0.026	-1,881	-0.117
ma.L1	0.8852	0.242	3,655	0.000	0.411	1,360
ar.S.L12	-0.0530	0.279	-0.190	0.849	-0.600	0.494
ma.S.L12	-0.9953	0.406	-2,449	0.014	-1,792	-0.199

Figure 4. SARIMA Modeling Results

Time series modeling of production data using the SARIMA method with parameters (1,1,1)(1,1,1)[12] shows quite good results. This model assumes the presence of an annual seasonal component (s = 12) as well as one-time data differences in both seasonal and non-seasonal components.

Based on the estimation results, the log-likelihood value obtained was -269.803, with an AIC value of 549.607, BIC of 557.383, and HQIC of 552,291. The relatively small AIC and BIC values indicate that this model has a fairly good fit to the data.

Several parameters in this model have been shown to be statistically significant, including:

AR(1) with a coefficient of -0.9988 and a p-value of 0.026,

MA(1) with a coefficient of 0.8852 and a p-value of 0.000,

and seasonal MA (lag 12) with a coefficient of -0.9953 and a p-value of 0.014.

The results of the residual diagnostic test of the model indicate that this model does not contain autocorrelation (Ljung-Box p = 0.44), the residuals are normally distributed (Jarque-Bera p = 0.65), and there are no symptoms of heteroscedasticity (Breusch-Pagan p = 0.73). This indicates that the model formed meets the classical assumptions of time series.

Overall, the SARIMA (1,1,1)(1,1,1)[12] model is suitable for use in forecasting production data because it meets statistical assumptions and has good performance based on model evaluation criteria.

4.5. MSE Calculation

The following is the MSE calculation for the test data. The following is the actual MSE calculation result of the SARIMA(1,1,1)(1,1,1,12) model for egg production data for March–December 2024: A

Table 4. MSE Calculation for Uj Data

Month	Actual (y)	Prediction (ŷ)	(y - ŷ)²
March 2024	5,248	4.7372	0.2609
April 2024	5,521	5.4038	0.0137
May 2024	5,875	5.5037	0.1379
June 2024	5,438	5.3988	0.0015
July 2024	6,216	5.5077	0.5017
August 2024	6,054	5.5991	0.2069
September	6.89	5.5878	1.6958
October	6.41	5.8758	0.2853
November	6,189	5.8085	0.1448
December	6,558	6.4422	0.0134
		MSE	0.3262

4.6. Prediction of Chicken Egg Production Using the SARIMA Model (1,1,1)(1,1,1,12) Method

Once the model is built, we can make predictions for 2025. Here is the code for predicting chicken egg production using the SARIMA method.

SARIMA Results

		No. Observations:	48
Covariance Type:	opg		

sigma2 1.851e+05 2.38e-06 7.77e+10 0.000 1.85e+05 1.85e+05

Prob(Q):

0.44 Prob(JB): 0.65

Heteroskedasticity (H):0.81 Skew: -0.21

2.36

```
import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.statespace.sarimax import SARIMAXdata =

[3,244, 3,728, 3,456, 4,785,

        3,120, 2,480, 2,938, 5,013,

        2,450, 2,140, 3,025, 5,248,

        2,780, 3,408, 3,450, 5,521,

        2,890, 3,512, 3,230, 5,875,

        3,076, 3,207, 3,268, 5,438,

        2,934, 3,386, 3,458, 6,216,

        3,102, 3,564, 3,426, 6,054,

        2,894, 3,824, 3,076, 6,890,

        3,256, 4,100, 3,386, 6,410,

        3,724, 3,446, 3,446, 6,189,

        3,896, 4,350, 4,425, 6,558]
```

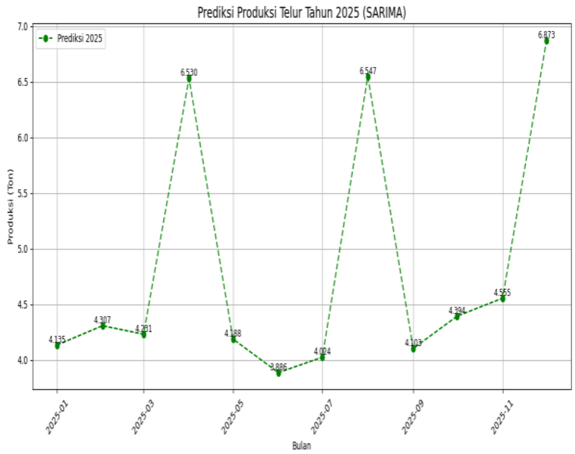
```
# Create a SARIMA model and train it
model = SARIMAX(ts, order=(1,1,1), seasonal_order=(1,1,1,12)) results = model.fit(disp=False)

# Prediction January - December 2025
pred_2025 = results.predict(start='2025-01-01', end='2025-12-01', typ='levels')

# Create a graph plt.figure(figsize=(12, 6))
plt.plot(pred_2025.index, pred_2025.values, marker='o', linestyle='--',color='green', label='2025 Prediction')

# Add a predicted number (e.g. 4,500) next to the dot for i, value in enumerate(pred_2025):
    plt.text(pred_2025.index[i], value, f"{value:.3f}", ha='center', va='bottom', fontsize=9, color='black')

plt.title("Egg Production Prediction for 2025 (SARIMA)", fontsize=14) plt.xlabel("Month")
plt.ylabel("Production (Ton)") plt.xticks(rotation=45)
plt.grid(True)
plt.legend() plt.tight_layout() plt.show()
```



The following table shows the monthly prediction results for SARIMA in 2025:

Table 5. Prediction Results

No	Month and Year	Production Prediction Results
1	January 2025	4.135
2	February 2025	4,307
3	March 2025	4,231
4	April 2025	6,530
5	May 2025	4,188
6	June 2025	3,886
7	July 2025	4.204
8	August 2025	6,547
9	September 2025	4.103
10	October 2025	4,394
11	November 2025	4,555
12	December 2025	6,873

5. Conclusion

This study successfully developed a SARIMA(1,1,1)(1,1,1,12) model to predict chicken egg production at PT. Pulau Mandiri Jaya Farm for the period 2021-2024, with the main findings being high accuracy (MSE 0.3262) that captures the annual upward and seasonal trend patterns (s=12) through stationarity tests, ACF/PACF, and residual diagnostics (Ljung-Box p=0.44, Jarque-Bera p=0.65). This model is superior to conventional methods, enabling stable forecasting for 2025 (range 3,886-6,873 tons), thus optimizing feed planning (60-70% of production costs) and reducing fluctuations from 2,140 to 6,890 tons.

However, limitations include the use of secondary univariate data (48 observations) that ignores external variables such as weather or disease, potentially overfitting the 80% training sample. Suggestions for further research include multivariate integration (e.g., LSTM or Prophet) with longer primary data, as well as cross-site validation in South Sumatra. Practically, the research implications support local food security through timely feed stocks.

6. Bibliography

Arifianti, R. (2020). *Pengertian strategi produksi dan keputusan strategi produksi/operasi*.

Universitas Padjadjaran.

Creswell, J. W., & Creswell, J. D. (2023). *Research design: Qualitative, quantitative, and mixed methods approaches* (6th ed.). SAGE Publications.

Durrah, F. I., Yulia, Y., Parhusip, T. P., & Rusyana, A. (2018). Peramalan jumlah penumpang pesawat di Bandara Sultan Iskandar Muda menggunakan metode SARIMA (*Seasonal Autoregressive Integrated Moving Average*). *Jurnal Analisis Data*, *1*(1), 1–11. https://doi.org/10.24815/jda.v1i1.11847

Emzir. (2021). *Metodologi penelitian kuantitatif*. Prenada Media.

Hudzaifah, M., & Rismayadi, A. A. (2021). Peramalan arus lalu lintas berdasarkan waktu tempuh dan cuaca menggunakan metode *time series decomposition*. *Jurnal Responsif: Riset Sains dan Teknologi*, *3*(2), 207–215. https://doi.org/10.51977/jti.v3i2.559

Kusyanto, Suhardi, D., & Awaluddin, R. (2020). Peramalan penjualan keramik menggunakan metode *moving average* dan *exponential smoothing* pada usaha Agus Keramik. *Jurnal Ekonomi Akuntansi dan Manajemen (JEAM)*, *1*(1), 12–21. https://journal.uniku.ac.id/index.php/jeam

Muhamad, I. M., Wardana, S. A., & Wanto, A. (2022). Algoritma *machine learning* untuk penentuan model prediksi produksi telur ayam petelur di Sumatera. *Journal of Informatics, Electrical and Electronics Engineering*, *1*(4), 126–134. https://djournals.com/jieee/article/view/382

Nuriyana. (2020). Model pengendalian persediaan pakan ayam broiler. *Jurnal Mirai Management*, *5*(1), 2597–4084.

Nurulita. (2020). *Penerapan metode peramalan ARIMA (Autoregressive Integrated Moving Average) untuk penentuan tingkat safety stock pada industri elektronik* [Skripsi, Universitas Indonesia]. Perpustakaan Universitas Indonesia. https://lib.ui.ac.id/abstrakpdf?id=20250041&lokasi=lokal

Sari, M. P. (2021). *Penggunaan metode ARIMA (Autoregressive Integrated Moving Average) untuk prakiraan penderita pneumonia balita di Kota Semarang tahun 2019–2021* [Skripsi, Universitas Negeri Semarang]. Institutional Repository UNNES.

Sudaryono. (2024). *Prosedur analisis time series dan validasi model statistik*. Rajawali Pers.

Sugiyono. (2022). *Metode penelitian kuantitatif, kualitatif, dan R&D* (3rd ed.). Alfabeta.

Suyono, A. A., Kusrini, K., & Arief, M. R. (2022). Prediksi indeks harga konsumen komoditas makanan di Kota Surabaya menggunakan *Support Vector Regression*. *Metik Jurnal*, *6*(1), 45–51. https://doi.org/10.47002/metik.v6i1.339

Tarisya, T. P. J., & Primandari, A. H. (2023). Perbandingan metode *double exponential smoothing* dan metode *triple exponential smoothing* untuk harga telur pada produsen di Kabupaten Sukabumi. *Emerging Statistics and Data Science Journal*, *1*(2), 204–214. https://doi.org/10.20885/esds.vol1.iss.2.art21

Zuhdi, F., Nurjanah, D., & Rahly, F. (2023). Peramalan produksi telur itik di Provinsi Aceh dengan pendekatan *Autoregressive Integrated Moving Average*. *Jurnal Veteriner*, *24*(1), 22–31. https://doi.org/10.19087/jveteriner.2023.24.1.2