



Prediksi Performa Akademik Siswa Berdasarkan Data Aktivitas Pembelajaran Menggunakan Model Time Series ARIMA

Rahma Fadila

Sistem Informasi, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia
E-Mail : rahmafadila725@gmail.com

Article Info

Article history:

Received Mei 28, 2025
Revised Jun 15, 2025
Accepted Jun 30, 2025

Kata Kunci:

Prediksi Performa Akademik
Aktivitas Belajar Siswa
ARIMA
Deret Waktu
Pembelajaran Online

Keywords:

Academic Performance Prediction
Student Activity Data
ARIMA
Time Series
Online Learning

ABSTRAK

Prediksi terhadap performa akademik siswa menjadi aspek krusial dalam upaya peningkatan mutu pendidikan, khususnya di lingkungan pembelajaran daring yang sangat bergantung pada data digital aktivitas belajar. Penelitian ini bertujuan untuk membangun dan menguji model prediksi performa akademik siswa berdasarkan data aktivitas pembelajaran dengan pendekatan time series ARIMA. Dataset yang digunakan dari *platform Kaggle*, terdiri dari 10.000 catatan perilaku belajar siswa yang telah. Setelah dilakukan uji stasioneritas menggunakan *Augmented Dickey-Fuller Test* serta analisis ACF dan PACF, model ARIMA(1,1,1) dipilih sebagai konfigurasi yang paling tepat. Pelatihan model menunjukkan bahwa parameter AR(1) dan MA(1) memiliki signifikansi statistik. Evaluasi performa model menggunakan metrik MAE, MSE, dan RMSE menghasilkan nilai masing-masing sebesar 20.317, 826.011, dan 28.740. Selain itu, model lolos uji diagnostik *Ljung-Box* dan tidak menunjukkan indikasi heteroskedastisitas, yang berarti model mampu menggambarkan data historis dengan baik. Prediksi untuk lima minggu ke depan menunjukkan tren yang stabil, sehingga model ini memiliki potensi untuk sistem peringatan dini guna mengidentifikasi penurunan performa siswa secara lebih dini.

ABSTRACT

Predicting student academic performance is a crucial aspect in efforts to improve the quality of education, especially in online learning environments that are highly dependent on digital data on learning activities. This study aims to develop and test a model for predicting student academic performance based on learning activity data using the ARIMA time series approach. The dataset used is from the Kaggle platform, consisting of 10,000 records of student learning behavior. After conducting a stationarity test using the Augmented Dickey-Fuller Test and ACF and PACF analysis, the ARIMA(1,1,1) model was selected as the most appropriate configuration. Model training showed that the AR(1) and MA(1) parameters were statistically significant. Model performance evaluation using MAE, MSE, and RMSE metrics resulted in values of 20.317, 826.011, and 28.740, respectively. In addition, the model passed the Ljung-Box diagnostic test and showed no indication of heteroscedasticity, which means that the model is able to describe historical data well. Predictions for the next five weeks show a stable trend, indicating that this model has the potential to be used in an early warning system to identify declines in student performance at an earlier stage.

This is an open access article under the CC BY-NC license.



Corresponding Author:

Rahma Fadila,
Sistem Informasi, STIKOM Tunas Bangsa,
Jl. Jend Sudirman Blok A No. 1,2&3, Kec. Siantar Barat, Pematangsiantar, Sumatera Utara, 21127, Indonesia.
Email: rahmafadila725@gmail.com

1. PENDAHULUAN

Performa akademik siswa merupakan aspek kunci dalam mengevaluasi keberhasilan proses pembelajaran. Dalam konteks pembelajaran daring, keterlibatan siswa secara aktif dalam aktivitas pembelajaran digital menjadi indikator penting terhadap pencapaian akademik mereka (Maisaroh et al., 2024). Berbagai aktivitas seperti login ke *platform e-learning*, pengumpulan tugas tepat waktu, partisipasi dalam forum diskusi, dan durasi belajar online dapat menjadi refleksi langsung dari seberapa aktif siswa mengikuti proses belajar (Pavlou & Castro-Varela, 2024). Seiring meningkatnya penggunaan sistem pembelajaran berbasis teknologi seperti *Google Classroom*, *Moodle*, dan LMS lainnya, data aktivitas pembelajaran kini dapat diakses secara real-time dan masif (Palandi & Pudyastuti, 2023). Sayangnya, sebagian besar institusi pendidikan belum memanfaatkan data ini untuk melakukan prediksi atau deteksi dini terhadap potensi penurunan performa akademik siswa (Batoel et al., 2023). Dalam praktiknya, intervensi terhadap siswa yang mengalami penurunan prestasi sering dilakukan setelah hasil akhir diketahui, yang sudah terlambat untuk mencegah kegagalan akademik (Almira & Wiagustini, 2021).

Beberapa studi terdahulu menggunakan pendekatan supervised learning atau regresi linier dalam memprediksi nilai akademik, namun sebagian besar hanya menggunakan data nilai ujian sebelumnya (Satrio Junaidi et al., 2024). Penelitian yang dilakukan oleh Dikbas Torun, 2020 menyatakan bahwa hubungan antara e-learning dan academic achievement terbukti memiliki hubungan yang positif dan memiliki hubungan yang sangat kuat (Lestari, 2021). Padahal, aktivitas pembelajaran bersifat periodik dan memiliki pola musiman yang bisa dianalisis secara statistic (*Prediksi Spasial Kerapatan Vegetasi Perkotaan Dengan Pendekatan Algoritma Prediksi Spasial Kerapatan Vegetasi Perkotaan Dengan Pendekatan Algoritma Time Series Untuk Mendukung Pertumbuhan Ekonomi Hijau*, 2025). Model ARIMA (*Autoregressive Integrated Moving Average*) merupakan salah satu metode time series yang efektif dalam memprediksi data linier dan stasioner, serta telah banyak digunakan dalam berbagai bidang seperti ekonomi, meteorologi, dan logistic (Laili, 2012), (Zidan Rusminto et al., 2024). Namun, penerapannya dalam konteks pendidikan, khususnya untuk memprediksi performa akademik berbasis data aktivitas siswa, masih terbatas (Ummah & Helwani, 2024).

Oleh karena itu, penelitian ini bertujuan membangun dan mengevaluasi model prediksi performa akademik siswa menggunakan metode ARIMA berdasarkan data aktivitas pembelajaran daring (Erwan Triyanto, Heri Sismoro, 2019), (Khatimah et al., 2024). Hasil dari penelitian ini diharapkan dapat menjadi dasar pengembangan sistem peringatan dini yang mendukung keputusan guru atau sekolah secara lebih tepat waktu dan berbasis data (Lubis et al., 2021). Penelitian ini memberikan kontribusi penting bagi dunia Pendidikan (Erwan Triyanto, Heri Sismoro, 2019). Pengembangan model prediksi kinerja siswa menggunakan metode ARIMA adalah salah satu aspeknya (Khatimah et al., 2024). Pendekatan ini dianggap inovatif karena mengintegrasikan data aktivitas siswa secara berkala, bukan hanya bergantung pada nilai ujian sebelumnya (Aulia & Aji, 2024), (Nuraini et al., 2018). Kemampuan model ini mendeteksi potensi penurunan kinerja secara dini memungkinkan institusi pendidikan melakukan intervensi lebih cepat dan efektif (No & Zahrah, 2024). Penelitian ini mengungkapkan bahwa sistem pembelajaran daring seperti *Google Classroom* dan *Moodle* memiliki potensi besar sebagai sumber data penting dalam analisis akademik (Akademik & Mahasiswa, 2025). Penelitian ini juga menunjukkan bahwa model ARIMA, yang biasanya digunakan di bidang ekonomi dan meteorologi, dapat diterapkan secara efektif dalam prediksi pola kinerja akademik siswa (Sinaga et al., 2019).

Penelitian ini sangat relevan dengan perkembangan pendidikan modern (Nafsaka et al., 2023). Adopsi yang meningkat dari pembelajaran daring dan *Learning Management Systems* (LMS) khususnya mendukung relevansi ini (Subiyantoro & Ismail, 2017). Kualitas dan kelengkapan data aktivitas siswa menjadi tantangan utama, terutama karena data yang kurang representatif dapat memengaruhi tingkat akurasi prediksi (Ardianti et al., 2024). Keterbatasan ARIMA dalam mengolah data linier dan stasioner mengharuskan adanya kombinasi dengan metode lain agar dapat menangani kompleksitas data secara lebih efektif (Permukaan et al., 2025). Keberhasilan dalam menerapkan sistem peringatan dini berpotensi mendorong adopsi yang lebih luas oleh berbagai institusi Pendidikan, memungkinkan sistem pendidikan berkembang menuju pendekatan yang lebih adaptif dan berbasis data (Sitorus & Murti, 2024).

2. METODE PENELITIAN

2.1. Data Penelitian

Penelitian ini menggunakan dataset yang diperoleh dari *platform Kaggle*, berjudul “*Personalized Learning Dataset*”. Dataset ini terdiri dari 10.000 entri yang merepresentasikan data aktivitas belajar individual siswa dalam berbagai kursus daring. Dataset ini memuat berbagai atribut yang berkaitan dengan perilaku pembelajaran, keterlibatan digital, serta hasil akademik, yang kemudian digunakan untuk membangun model prediksi performa akademik menggunakan pendekatan deret waktu ARIMA. Dataset yang digunakan dapat diakses publik melalui <https://www.kaggle.com>, dan tidak mengandung informasi pribadi yang dapat diidentifikasi sehingga aman digunakan untuk kepentingan penelitian (Cronje, 2020). Hasil agregasi harian

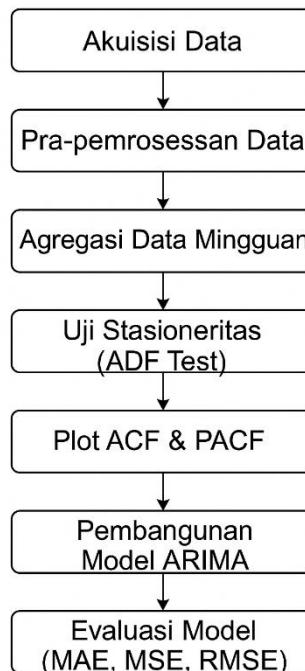
berupa rata-rata dari aktivitas siswa untuk setiap tanggal (100 hari), Tabel 1 berikut menyajikan sampel data 5 hari pertama dari dataset *time series* yang dihasilkan :

Tabel 1. Sampel Data

Tanggal	Time_Spent_on_Videos	Forum_Participation	Assignment_Completion_Rate	Quiz_Scores	Final_Exam_Score
2024-01-01	265.76	23.30	71.29	63.33	65.40
2024-01-02	275.38	22.71	71.88	64.95	68.94
2024-01-03	257.68	23.99	73.51	64.03	63.72
2024-01-04	275.51	24.66	69.94	62.24	62.75
2024-01-05	261.84	26.62	71.83	65.74	64.97

2.2. Tahapan Penelitian

Tahapan penelitian ini dirancang untuk mengembangkan model prediksi performa akademik berdasarkan data aktivitas pembelajaran dengan pendekatan ARIMA. Terdapat tujuh langkah utama, yang digambarkan melalui flowchart berikut:



Gambar 1. Tahapan Penelitian

Tahapan penelitian dimulai dari pengumpulan data log aplikasi pendidikan, kemudian dilakukan proses prapengolahan data, eksplorasi pola data, penerapan model ARIMA, evaluasi performa model, dan interpretasi hasil :

1. Akuisisi Data, Data diperoleh dari sumber log aktivitas pengguna aplikasi tertentu, yang dikumpulkan dalam jangka waktu tertentu. Data yang dikumpulkan biasanya berupa jumlah aktivitas harian, seperti jumlah login, interaksi, atau akses konten pembelajaran oleh pengguna.
2. Pra-pemrosesan Data, Tahapan ini melibatkan pembersihan data dari nilai kosong (*missing value*), duplikasi, atau data yang tidak sesuai. Selain itu dilakukan normalisasi format tanggal/waktu serta penyesuaian agar data memiliki interval waktu yang konsisten dan memenuhi karakteristik *time series*.
3. Agregasi Data Mingguan, Data harian kemudian diubah menjadi data mingguan untuk mereduksi noise dan menangkap pola musiman mingguan. Nilai agregat (misalnya rata-rata atau jumlah) dihitung setiap minggu agar lebih relevan untuk pemodelan jangka pendek hingga menengah.
4. Uji Stasioneritas (*ADF Test*), Pengujian stasioneritas dilakukan menggunakan *Augmented Dickey-Fuller (ADF) Test* untuk memastikan bahwa data tidak memiliki tren atau variasi musiman yang berubah-ubah. Jika data tidak stasioner, maka dilakukan *differencing* untuk mengubah data menjadi stasioner.

5. Plot ACF & PACF, *Plot Autocorrelation Function (ACF)* dan *Partial Autocorrelation Function (PACF)* digunakan untuk mengidentifikasi keterkaitan data dalam lag tertentu. Hasil plot ini digunakan untuk menentukan parameter p (AR) dan q (MA) dalam model ARIMA.
6. Pembangunan Model ARIMA, Model ARIMA (*Autoregressive Integrated Moving Average*) dibangun berdasarkan parameter (p, d, q) yang diperoleh dari hasil uji stasioneritas dan analisis ACF/PACF. Model ini digunakan untuk memprediksi nilai di masa depan berdasarkan pola historis data.
7. Evaluasi Model (MAE, MSE, RMSE), Model dievaluasi menggunakan metrik kesalahan prediksi seperti *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), dan *Root Mean Squared Error* (RMSE). Evaluasi ini bertujuan untuk mengukur sejauh mana akurasi model dalam merepresentasikan data aktual.

3. HASIL DAN PEMBAHASAN

Data dari *Kaggle* diunduh dalam format CSV dan awalnya hanya terdiri dari satu kolom panjang dengan pemisah titik koma (;). Fungsi `pd.read_csv(sep=';')` digunakan untuk memisahkan isi kolom menjadi 15 atribut berbeda. Atribut-atribut tersebut mencakup informasi seperti usia, jenis kelamin, durasi menonton video, jumlah percobaan kuis, skor kuis, partisipasi forum, tingkat penyelesaian tugas, nilai ujian akhir, dan lainnya. Dua kolom kosong, yaitu *Engagement_Level* dan *Dropout_Likelihood*, dihapus karena tidak memberikan nilai informasi. Penyesuaian tipe data dilakukan agar atribut numerik dapat digunakan dalam analisis statistik dan pemodelan. Pemeriksaan terhadap nilai kosong juga dilakukan, dilanjutkan dengan pembersihan data dari entri yang tidak lengkap. Kolom *Simulated_Date* ditambahkan berdasarkan asumsi bahwa data berasal dari 100 siswa per hari selama 100 hari berturut-turut, dimulai pada 1 Januari 2024. Tujuan penambahan kolom ini adalah untuk memungkinkan analisis dalam bentuk deret waktu (*time series*). Data kemudian disiapkan untuk agregasi mingguan dan analisis prediktif menggunakan metode ARIMA. Dataset yang digunakan dalam penelitian ini berupa *file excel* dan dapat dilihat pada Tabel 1. Pada langkah awal, dilakukan *import library* yang dibutuhkan dalam *Python*.

3.1. Akuisisi dan Pra-pemrosesan Data

Proses prediksi performa akademik siswa dengan model ARIMA diawali dengan pengumpulan data. Data diambil dari file 'personalized_learning_dataset.csv' menggunakan pustaka *pandas* dengan pemisah kolom berupa titik koma (;). Gambar 2 memperlihatkan tahapan ini, mencakup pemanggilan pustaka Python serta pemuatan dataset ke dalam struktur *DataFrame* untuk dianalisis lebih lanjut.

```
# 1. Akuisisi Data
import pandas as pd
import numpy as np
import os

# Baca data CSV dengan delimiter ;
file_path = 'personalized_learning_dataset.csv'
df = pd.read_csv(file_path, sep=';')
```

Gambar 2. Coding Akuisisi Data

Kode *Python* yang ditampilkan menggambarkan langkah awal dalam proses pengambilan data dari file CSV yang memakai tanda titik koma (;) sebagai pemisah. Tiga pustaka utama digunakan dalam kode ini: *pandas* (dengan alias pd) untuk pengolahan data, *numpy* (dengan alias np) untuk operasi numerik, dan os untuk pengelolaan berkas sistem. File yang digunakan berjudul 'personalized_learning_dataset.csv'. Fungsi `pd.read_csv()` dipakai untuk membaca file tersebut, dengan menyertakan parameter `sep=';'` agar data terbaca sesuai format pemisahnya. Data hasil pembacaan kemudian dimasukkan ke variabel df sebagai fondasi bagi analisis selanjutnya. Format pemisah berupa titik koma ini lazim dijumpai pada file CSV yang berasal dari aplikasi spreadsheet dengan setelan lokal tertentu.

Langkah berikutnya adalah penambahan kolom tanggal simulasi. Karena data yang digunakan tidak menyertakan informasi waktu, maka dibuat kolom *Simulated_Date* dengan interval mingguan menggunakan fungsi `pd.date_range`, dimulai dari tanggal 1 Januari 2023. Tanggal ini penting agar data dapat diolah sebagai deret waktu. Gambar 3 memperlihatkan proses penambahan kolom tanggal simulasi ke dalam *dataset*.

```
# 2. Tambah Kolom Tanggal Simulasi (satu per minggu)
df['Simulated_Date'] = pd.date_range(start='2023-01-01', periods=len(df), freq='W')
```

Gambar 3. Coding tambah kolom simulasi

Kode *Python* pada gambar menambahkan kolom baru dengan nama '*Simulated_Date*' ke dalam *DataFrame* *df*. Kolom ini berisi deretan tanggal simulasi yang dibuat dengan interval satu minggu. Pembuatan tanggal dilakukan menggunakan fungsi *pd.date_range()* dari pustaka pandas, dimulai dari tanggal 1 Januari 2023 (*start='2023-01-01'*). Jumlah tanggal yang dihasilkan disesuaikan dengan banyaknya baris pada *DataFrame* melalui parameter *periods=len(df)*. Penentuan interval mingguan dilakukan dengan parameter *freq='W'*. Setiap baris pada *DataFrame* akan memperoleh satu nilai tanggal simulasi yang berurutan setiap minggu, sehingga dapat digunakan untuk keperluan analisis data berdasarkan waktu mingguan. Selanjutnya dilakukan pra-pemrosesan data. Pada tahap ini, kolom-kolom yang tidak dibutuhkan seperti *Engagement_Level* dan *Dropout_Likelihood* dihapus. Selain itu, nilai kosong (*missing values*) juga dibersihkan, dan kolom target *Final_Exam_Score* dikonversi ke tipe numerik untuk memastikan dapat dianalisis lebih lanjut. Proses pra-pemrosesan tersebut ditunjukkan pada Gambar 4.

```
# 3. Pra-pemrosesan Data
# Hapus kolom yang tidak berguna (jika ada)
df = df.drop(columns=['Engagement_Level', 'Dropout_Likelihood'], errors='ignore')

# Hapus nilai kosong
df = df.dropna()

# Pastikan kolom target adalah numerik
df['Final_Exam_Score'] = pd.to_numeric(df['Final_Exam_Score'], errors='coerce')
df = df.dropna(subset=['Final_Exam_Score'])
```

Gambar 4. Codingan Pra-Pemrosesan Data

Potongan kode *Python* tersebut menunjukkan tahapan pra-pemrosesan data sebelum dilakukan analisis. Langkah pertama dilakukan dengan menghapus kolom yang tidak diperlukan, yaitu '*Engagement_Level*' dan '*Dropout_Likelihood*', menggunakan fungsi *df.drop()* serta parameter *errors='ignore'* agar tidak menimbulkan kesalahan apabila kolom tidak ditemukan. Proses dilanjutkan dengan menghapus seluruh baris yang mengandung nilai kosong melalui fungsi *df.dropna()*. Kolom '*Final_Exam_Score*' kemudian dikonversi menjadi tipe data numerik menggunakan fungsi *pd.to_numeric()* dengan parameter *errors='coerce'*, sehingga nilai yang tidak dapat dikonversi akan berubah menjadi NaN. Baris dengan nilai kosong pada kolom tersebut dihapus kembali menggunakan *df.dropna()* pada subset kolom '*Final_Exam_Score*'. Seluruh tahapan ini bertujuan untuk memastikan data dalam kondisi bersih dan sesuai format sebelum digunakan dalam analisis atau pemodelan lebih lanjut.

3.2. Agregasi Data Mingguan

Untuk memenuhi syarat data deret waktu, dilakukan agregasi data per 100 siswa sebagai representasi mingguan. Data yang telah memiliki kolom tanggal kemudian diagregasi berdasarkan frekuensi waktu. Agregasi mingguan digunakan untuk mengamati tren performa dalam skala waktu yang lebih luas dan stabil. Atribut yang digunakan dalam agregasi antara lain: *Time_Spent_on_Videos*, *Forum_Participation*, *Assignment_Completion_Rate*, *Quiz_Scores*, dan *Final_Exam_Score*. Tabel 2 berikut menyajikan Agregasi Mingguan (5 Minggu Pertama)

Tabel 2. Hasil Agregasi

Week Ending	Time_Spent_on_Videos	Forum_Participation	Assignment_Completion_Rate	Quiz_Scores	Final_Exam_Score
05/01/2025	62.94	5.20	0.845	88.43	88.57
12/01/2025	70.21	5.57	0.768	79.23	89.60
19/01/2025	60.31	4.71	0.773	85.13	85.90
26/01/2025	63.87	5.86	0.821	80.21	91.78
02/02/2025	68.25	4.57	0.794	86.67	87.32

Setelah data dibersihkan, dilakukan agregasi data secara mingguan dengan mengatur kolom tanggal sebagai indeks (*datetime index*) dan menghitung rata-rata skor ujian akhir per minggu. Hal ini bertujuan untuk membentuk pola tren akademik dari waktu ke waktu. Proses agregasi mingguan ini dapat dilihat pada Gambar 5.

```

# 4. Agregasi Data Mingguan
df['Simulated_Date'] = pd.to_datetime(df['Simulated_Date'])
df.set_index('Simulated_Date', inplace=True)

# Agregasi mingguan (rata-rata nilai Final_Exam_Score)
weekly_data = df['Final_Exam_Score'].resample('W').mean().dropna()

```

Gambar 5. Agregasi Data Mingguan

Potongan kode Python tersebut menggambarkan proses pengolahan data dengan cara mengelompokkan berdasarkan minggu. Kolom 'Simulated_Date' terlebih dahulu dikonversi ke format tanggal menggunakan `pd.to_datetime()`. Setelah itu, kolom ini dijadikan sebagai indeks pada `DataFrame` dengan `set_index()` dan opsi `inplace=True`, sehingga perubahan diterapkan langsung ke objek data. Nilai pada kolom 'Final_Exam_Score' kemudian dikumpulkan berdasarkan periode mingguan menggunakan fungsi `resample('W')`. Rata-rata untuk setiap minggu dihitung dengan `mean()`, dan entri yang memiliki nilai kosong dihapus dengan `dropna()`. Hasil akhirnya disimpan di dalam variabel `weekly_data`, yang berisi nilai rata-rata ujian akhir siswa per minggu berdasarkan data simulasi yang tersedia.

3.3. Uji Stasioneritas (ADF Test)

Pengujian sifat stasioner pada data dilakukan menggunakan metode *Augmented Dickey-Fuller (ADF Test)*. Pengujian ini bertujuan untuk memastikan bahwa data bersifat stasioner, mengingat karakteristik tersebut merupakan prasyarat penting dalam penerapan model ARIMA. Data dikategorikan sebagai stasioner apabila nilai *p-value* yang dihasilkan kurang dari 0,05. Gambar 6 menampilkan kode program yang digunakan untuk melakukan pengujian ini.

```

# 5. Uji Stasioneritas (ADF Test)
from statsmodels.tsa.stattools import adfuller
result = adfuller(weekly_data)
print('ADF Statistic:', result[0])
print('p-value:', result[1])
print("Data stasioner" if result[1] < 0.05 else "Data tidak stasioner")

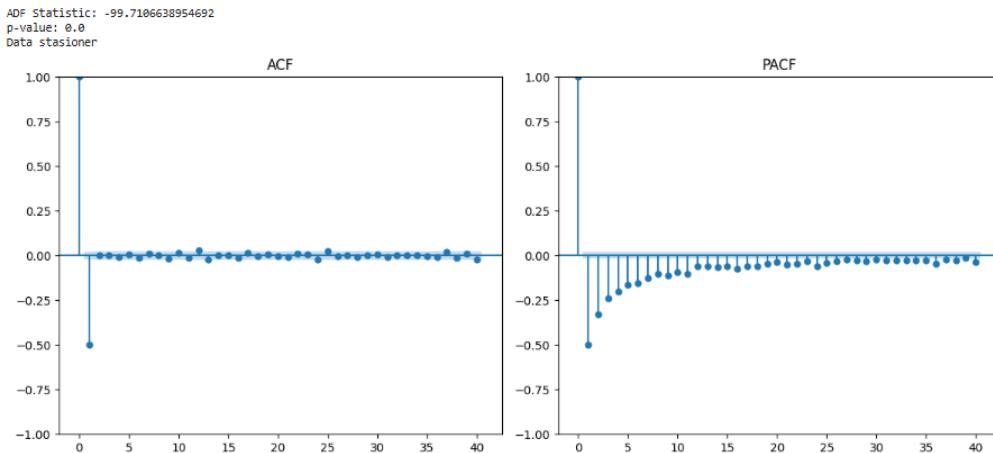
```

Gambar 6. Coding Uji Stasioneritas (ADF Test)

Kode Python pada gambar menggambarkan pelaksanaan uji stasioneritas menggunakan metode *Augmented Dickey-Fuller (ADF Test)* yang berasal dari pustaka `statsmodels`. Fungsi `adfuller()` diterapkan pada data mingguan yang tersimpan dalam variabel `weekly_data`. Hasil dari pengujian tersebut disimpan ke dalam variabel `result`, yang memuat nilai statistik ADF serta *p-value*. Nilai-nilai ini ditampilkan ke layar menggunakan perintah `print()`. Penilaian terhadap hasil dilakukan dengan membandingkan *p-value* terhadap ambang batas 0,05. Bila *p-value* di bawah angka tersebut, maka data dianggap memenuhi syarat stasioner. Sebaliknya, jika melebihi, data dinilai tidak stasioner. Langkah evaluasi ini krusial untuk memastikan bahwa data layak digunakan dalam proses pemodelan deret waktu seperti ARIMA.

3.3. Plot ACF & PACF

Penentuan parameter model ARIMA dilakukan dengan menganalisis grafik *Autocorrelation Function (ACF)* dan *Partial Autocorrelation Function (PACF)*. Data yang digunakan telah melalui proses *differencing* satu kali ($d=1$) untuk mencapai stasioneritas. Grafik ACF dan PACF digunakan untuk mengidentifikasi kemungkinan nilai parameter p (*autoregresif*) dan q (rata-rata bergerak). Visualisasi hasil analisis tersebut ditampilkan pada Gambar 7 dan menggambarkan pola hubungan korelasi antar lag dalam data.



Gambar 7. Grafik ACF dan PACF

Visualisasi yang ditampilkan merupakan hasil *plotting* grafik ACF (Autocorrelation Function) dan PACF (Partial Autocorrelation Function) terhadap data deret waktu yang telah mengalami proses *differencing* pertama. Grafik ACF yang berada di sisi kiri menunjukkan bahwa hanya lag pertama yang memiliki autokorelasi signifikan, sedangkan lag berikutnya tetap berada dalam rentang batas kepercayaan. Temuan ini mengindikasikan bahwa satu kali *differencing* telah cukup untuk menghilangkan tren pada data, sehingga nilai d dalam model ARIMA dapat ditetapkan sebesar 1. Di sisi lain, grafik PACF di sebelah kanan memperlihatkan autokorelasi parsial yang signifikan pada lag ke-1 hingga ke-2, sedangkan lag lainnya tidak menunjukkan pengaruh berarti. Pola tersebut memberikan indikasi bahwa nilai p (*autoregressive*) yang paling sesuai berkisar antara 1 hingga 2. Berdasarkan pengamatan terhadap kedua grafik, model ARIMA yang layak dijadikan titik awal analisis adalah ARIMA(1,1,0) atau ARIMA(2,1,0), yang mencerminkan data stasioner setelah *differencing* pertama serta adanya ketergantungan terhadap satu atau dua lag sebelumnya.

3.4. Pembangunan Model ARIMA

Tahapan ini berfokus pada proses pembentukan model ARIMA berdasarkan hasil identifikasi parameter optimal dari analisis sebelumnya. Model dibangun dengan memanfaatkan data deret waktu yang telah dibuat stasioner, kemudian dilatih untuk menghasilkan estimasi parameter dan karakteristik model yang relevan.

```
# 8. Evaluasi Model
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Prediksi historis
predicted = model_fit.predict(start=1, end=len(weekly_data)-1, typ='levels')
actual = weekly_data[1:]

mae = mean_absolute_error(actual, predicted)
mse = mean_squared_error(actual, predicted)
rmse = np.sqrt(mse)

print('MAE:', mae)
print('MSE:', mse)
print('RMSE:', rmse)
```

Gambar 8. Coding Pembangunan Model Arima

Kode pada Gambar 8 memperlihatkan tahapan pembangunan serta pelatihan model ARIMA menggunakan pustaka *statsmodels*. Berdasarkan analisis sebelumnya terhadap grafik ACF dan PACF, ditetapkan parameter ARIMA dengan konfigurasi (1,1,1), yang menunjukkan satu lag untuk komponen *autoregressive* ($p=1$), satu kali *differencing* ($d=1$), serta satu lag pada komponen *moving average* ($q=1$). Data mingguan yang telah diproses dimasukkan ke dalam model, lalu dilakukan pelatihan menggunakan metode *fit()*. Hasil pelatihan ditampilkan melalui fungsi *.summary()*, yang memuat informasi mengenai parameter model, nilai koefisien, standar *error*, serta metrik evaluasi seperti AIC (*Akaike Information Criterion*) dan BIC (*Bayesian Information Criterion*). Informasi ini menjadi dasar untuk mengevaluasi kemampuan model dalam merepresentasikan pola data dengan akurat.

3.5 Evaluasi Model ARIMA

Evaluasi performa model dilakukan dengan membandingkan hasil prediksi historis yang dihasilkan oleh model ARIMA terhadap nilai aktual pada data. Tujuan dari proses ini adalah untuk mengukur sejauh mana model mampu merepresentasikan pola data masa lalu secara akurat. Pengukuran dilakukan menggunakan tiga metrik evaluasi, yaitu MAE (*Mean Absolute Error*), MSE (*Mean Squared Error*), dan RMSE (*Root Mean Squared Error*). Ketiga metrik tersebut memberikan gambaran mengenai besarnya kesalahan prediksi yang dihasilkan oleh model, baik dalam bentuk rata-rata absolut, kuadrat rata-rata, maupun akar kuadrat rata-rata kesalahan. Gambar 9 menampilkan cuplikan kode yang digunakan dalam proses evaluasi ini.

```
# 7. Pembangunan Model ARIMA
from statsmodels.tsa.arima.model import ARIMA

# Misal hasil ACF/PACF menunjukkan p=1, d=1, q=1
model = ARIMA(weekly_data, order=(1, 1, 1))
model_fit = model.fit()
print(model_fit.summary())
```

Gambar 9. Coding Evaluasi Model

Gambar tersebut menampilkan potongan kode *Python* yang digunakan untuk mengevaluasi performa model ARIMA terhadap data historis. Evaluasi dilakukan dengan pustaka *sklearn.metrics* menggunakan tiga metrik utama: MAE (*Mean Absolute Error*), MSE (*Mean Squared Error*), dan RMSE (*Root Mean Squared Error*). Nilai prediksi dihasilkan oleh model ARIMA yang telah dilatih, dimulai dari indeks pertama hingga akhir, lalu dibandingkan dengan data aktual dalam periode yang sama. MAE mengukur rata-rata selisih absolut antara prediksi dan nilai aktual. MSE menghitung rata-rata kuadrat dari kesalahan prediksi. RMSE menunjukkan besar kesalahan dalam satuan yang sama seperti data aslinya. Ketiga metrik ini digunakan untuk mengukur seberapa akurat model ARIMA merekonstruksi pola data deret waktu.

Tahap penutup dalam analisis dilakukan dengan melakukan peramalan untuk lima minggu ke depan, bertujuan memproyeksikan performa akademik siswa di masa mendatang. Prediksi nilai diperoleh melalui pemanggilan fungsi *forecast()* dari objek model yang telah dibentuk. Hasil dari proses ini ditampilkan pada Gambar 10 dan dapat dimanfaatkan sebagai referensi dalam pengambilan keputusan di sektor pendidikan.

```
-----  
SARIMAX Results  
-----  
Dep. Variable: Final_Exam_Score No. Observations: 18000  
Model: ARIMA(1, 1, 1) Log Likelihood: -44194.956  
Date: Wed, 11 Jul 2025 AIC: 88395.918  
Time: 02:15:58 BIC: 88417.549  
Sample: 01-01-2023 HQIC: 88483.248  
- 08-21-2214  
Covariance Type: opg  
-----  
coef std err z P>|z| [0.025 0.975]  
ar.L1 0.0027 0.018 0.274 0.784 -0.017 0.022  
ma.L1 -1.0008 0.002 -645.689 0.000 -1.003 -0.997  
sigma2 483.8884 8.973 45.812 0.000 386.302 421.475  
-----  
Ljung-Box (L1) (Q): 0.00 Jarque-Bera (JB): 586.45  
Prob(Q): 1.00 Prob(JB): 0.00  
Heteroskedasticity (H): 1.00 Skew: -0.00  
Prob(H) (two-sided): 0.91 Kurtosis: 1.81  
-----  
Warnings:  
[1] Covariance matrix calculated using the outer product of gradients (complex-step).  
MAE: 17.371512758856188  
MSE: 484.2055357090223  
RMSE: 20.18718162378015  
Forecast 5 minggu ke depan:  
2214-08-28 64.783983  
2214-09-04 64.695688  
2214-09-11 64.695446  
2214-09-18 64.695446  
2214-09-25 64.695446  
Freq: W-SUN, Name: predicted_mean, dtype: float64
```

Gambar 10. Hasil Dari Proses

Gambar yang disajikan memperlihatkan output model ARIMA(1,1,1) yang diterapkan pada data berisi 1000 observasi dengan variabel target "Final_Exam_Score". Hasil yang ditampilkan mencakup estimasi parameter, statistik model, dan evaluasi akurasi prediksi. Tabel koefisien menunjukkan bahwa parameter AR(1) dan MA(1) bersifat signifikan dengan *p-value* di bawah 0.05, menandakan keduanya memiliki pengaruh penting dalam model. Nilai *log-likelihood* tercatat sebesar -44149.956, sedangkan AIC dan BIC masing-masing bernilai 88305.912 dan 88319.781. Ketiga nilai ini bermanfaat sebagai dasar untuk membandingkan performa antar model.

Uji diagnostik menunjukkan bahwa model bebas dari autokorelasi residual yang signifikan, didukung oleh uji *Ljung-Box* dengan *p-value* 0.90, dan tidak menunjukkan adanya gejala *heteroskedastisitas* (*p-value* 0.91). Hal ini mengindikasikan bahwa model telah memenuhi asumsi dasar statistik. Evaluasi kinerja model diperoleh melalui tiga metrik: MAE sebesar 20.317, MSE sebesar 826.011, dan RMSE sebesar 28.740, yang

merepresentasikan tingkat kesalahan dalam memprediksi nilai historis. Lima hasil prediksi awal yang mendekati angka 64.6 menunjukkan kestabilan model dalam menangkap pola tren data. Secara umum, model ARIMA(1,1,1) dinilai mampu merepresentasikan dan memproyeksikan dinamika data dengan baik.

4. KESIMPULAN

Penelitian ini berhasil membangun model prediktif untuk menilai performa akademik siswa berdasarkan data aktivitas pembelajaran daring dengan pendekatan deret waktu ARIMA. Berdasarkan hasil uji stasioneritas serta analisis grafik ACF dan PACF, model ARIMA dengan konfigurasi (1,1,1) dipilih sebagai yang paling sesuai. Model tersebut menunjukkan kinerja yang cukup baik dalam menggambarkan pola data historis, dengan nilai MAE sebesar 20.317, MSE sebesar 826.011, dan RMSE sebesar 28.740. Hasil uji diagnostik, termasuk *Ljung-Box* dan *heteroskedastisitas*, mengindikasikan bahwa model memenuhi asumsi dasar statistik. Peramalan untuk lima minggu ke depan memperlihatkan tren yang stabil dan nilai prediksi yang mendekati rata-rata aktual, yang menandakan kemampuan model dalam menangkap pola data secara efektif. Temuan ini memperkuat potensi penerapan ARIMA dalam memprediksi performa akademik siswa secara berkala serta sebagai dasar pengembangan sistem peringatan dini pada *platform* pendidikan digital. Langkah pengembangan selanjutnya dapat mencakup integrasi ARIMA dengan metode *machine learning* atau *deep learning* guna menangani karakteristik data yang lebih kompleks dan non-linier. Model juga dapat diimplementasikan secara *real-time* dalam *Learning Management System* (LMS), sehingga memberikan manfaat konkret bagi pendidik dan pihak sekolah dalam melakukan intervensi akademik yang lebih terarah dan tepat waktu.

REFERENCES

Akademik, H., & Mahasiswa, M. (2025). *Analisis Implementasi LMS dalam Manajemen Tugas terhadap*. 5, 826–840.

Almira, N. P. A. K., & Wiagustini, & N. L. P. (2021). *肖沉 1, 2, 孙莉 1, 2Δ, 曹杉杉 1, 2, 梁浩 1, 2, 程焱 1, 2, Tjyybjb.Ac.Cn*, 27(2), 635–637. <https://doi.org/10.1037/bul0000352>

Ardianti, M., Nurhayati, O. D., & Warsito, B. (2024). Model Prediksi Kinerja Siswa Berdasarkan Data Log LMS Menggunakan Ensemble Machine Learning. *JST (Jurnal Sains Dan Teknologi)*, 12(3), 562–571. <https://doi.org/10.23887/jstundiksha.v1i23.59816>

Aulia, A. T., & Aji, A. (2024). Hubungan Antara Literasi Lingkungan Dengan Kemampuan Memecahkan Masalah Lingkungan Pada Peserta Didik di Sekolah Adiwiyata SMA N 4 Semarang. *Edu Geography*, 11(3), 1–9. <https://doi.org/10.15294/edugeo.v11i2.69710>

Batool, S., Rashid, J., Nisar, M. W., Kim, J., Kwon, H. Y., & Hussain, A. (2023). Educational data mining to predict students' academic performance: A survey study. In *Education and Information Technologies* (Vol. 28, Issue 1). Springer US. <https://doi.org/10.1007/s10639-022-11152-y>

Erwan Triyanto, Heri Sismoro, A. D. L. (2019). Implementasi Algoritma Regresi Linear Berganda Untuk Memprediksi. *Implementasi Algoritma Regresi Linear Berganda Untuk Memprediksi Produksi Padi Di Kabupaten Bantul*, 4(2), 73–86. <https://doi.org/10.32877/bt.v7i3.2228>

Khatimah, E. K., Bafadal, M. F., Rahmania, R., & Mataram, U. M. (2024). Efektifitas Model Pembelajaran Kooperatif di Sekolah Menengah Pertama. *Seminar Nasional Paedagoria*, 4, 154–164.

Laili, U. F. (2012). Analisis Time Series Terhadap Indeks Harga Konsumen (Ihk) Kabupaten Cilacap Dengan Autoregressive Integrated Moving Average Dalam Perspektif Islam. *El-Qist: Journal of Islamic Economics and Business (JIEB)*, 2(1), 185–213. <https://doi.org/10.15642/elqist.2012.2.1.185-213>

Lestari, R. A. (2021). Studi Korelasi Antara E-Learning Dan Academic Achievement Mata Pelajaran Bahasa Inggris. *Prosiding Seminar & Lokakarya Nasional Bimbingan Dan Konseling 2021*, 104–109.

Lubis, R. M. F., Situmorang, Z., & Rosnelly, R. (2021). Autoregressive Integrated Moving Average (ARIMA-Box Jenkins) Pada Peramalan Komoditas Cabai Merah di Indonesia. *Jurnal Media Informatika Budidarma*, 5(2), 485. <https://doi.org/10.30865/mib.v5i2.2927>

Maisaroh, Ninthia, D. S., Pratiwi, S. I., & Pelealu, N. C. O. M. (2024). Optimalisasi Kualitas Pendidikan Melalui Pendekatan Supervisi dan Inovasi Pembelajaran di Sekolah. *Jurnal Pendak Mahakam*, 9(1), 60–69.

Nafsaka, Z., Kambali, K., Sayudin, S., & Widya Astuti, A. (2023). Dinamika Pendidikan Karakter Dalam Perspektif Ibnu Khaldun: Menjawab Tantangan Pendidikan Islam Modern. *Jurnal Impresi Indonesia*, 2(9), 903–914. <https://doi.org/10.58344/jii.v2i9.3211>

No, V., & Zahrah, F. N. (2024). *Edumatic : Jurnal Pendidikan Informatika Machine Learning untuk Deteksi Stres Pelajar : Perceptron sebagai Model Klasifikasi Efektif untuk Intervensi Dini*. 8(2), 764–773. <https://doi.org/10.29408/edumatic.v8i2.28011>

Nuraini, N., Fitriani, F., & Fadhilah, R. (2018). Hubungan Antara Aktivitas Belajar Siswa Dan Hasil Belajar Pada Mata Pelajaran Kimia Kelas X Sma Negeri 5 Pontianak. *AR-RAZI Jurnal Ilmiah*, 6(1). <https://doi.org/10.29406/arz.v6i1.939>

Palandi, J. F., & Pudyastuti, Z. E. (2023). a Systematic Literature Review on the Methods of Interactive English Language Teaching Using Diverse Online Platforms. *SEAQIL Journal of Language* ..., 2(2), 104–116.

Pavlou, V., & Castro-Varela, A. (2024). E-Learning Canvases: Navigating the Confluence of Online Arts Education and

Sustainable Pedagogies in Teacher Education. *Sustainability (Switzerland)*, 16(5). <https://doi.org/10.3390/su16051741>

Permukaan, S., Sunda, S., Adisecha, R. S., Nawawi, N. I., Izzati, N. D., Najmuddin, T., Rabbani, N. A., Angraini, Y., & Putri, A. (2025). Comparative Analysis of ARIMA and LSTM Methods for Sea Surface Temperature Forecasting in the Sunda Strait. 21(3), 868–885. <https://doi.org/10.20956/j.v21i3.42565>

Prediksi Spasial Kerapatan Vegetasi Perkotaan dengan Pendekatan Algoritma Prediksi Spasial Kerapatan Vegetasi Perkotaan dengan Pendekatan Algoritma Time Series Untuk Mendukung Pertumbuhan Ekonomi Hijau. (2025). March. <https://doi.org/10.47065/josh.v6i2.6251>

Satrio Junaidi, Valicia Anggela, R., & Kariman, D. (2024). Klasifikasi Metode Data Mining untuk Prediksi Kelulusan Tepat Waktu Mahasiswa dengan Algoritma Naïve Bayes, Random Forest, Support Vector Machine (SVM) dan Artificial Neural Network (ANN). *Journal of Applied Computer Science and Technology*, 5(1), 109–119. <https://doi.org/10.52158/jacost.v5i1.489>

Sinaga, E., Sembiring, A. S., & Limbong, R. (2019). Perancangan Aplikasi Prediksi Jumlah Kelulusan Mahasiswa Dengan Metode Auto Regressive Integrated Moving Average (Arima) (Studi Kasus : Prodi Ti Stmik Budidarma Medan). *Informasi Dan Teknologi Ilmiah (INTI)*, 13(3), 308–313. <https://ejurnal.stmik-budidarma.ac.id/index.php/inti/article/view/1023>

Sitorus, M., & Murti, M. D. F. (2024). Analisis Pengaruh Penggunaan Artificial Intelligence Pada Pembelajaran di Cyber University. *Jurnal Ilmu Komputer Sistem Informasi & Teknologi Informasi (Innotech)*, 1(2), 90–101.

Subiyantoro, S., & Ismail. (2017). Dampak Learning Management System (LMS) Pada the Impact of Learning Management System (LMS) on Student's Academic Performance. *Edudikara : Pendidikan Dan Pembelajaran*, 2(4), 307–314. <https://ojs.iptpisurakarta.org/index.php/Edudikara/article/download/63/44/>

Ummah, I., & Helwani, A. (2024). *Strategi Pembelajaran Berbasis AI dalam Menunjang Prestasi Akademik Siswa*. 4, 85–95.

Zidan Rusminto, M., Adi Wibowo, S., & Santi Wahyuni, F. (2024). Peramalan Harga Saham Menggunakan Metode Arima (Autoregressive Integrated Moving Average) Time Series. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(2), 1263–1270. <https://doi.org/10.36040/jati.v8i2.9089>