PRISMA 9 (2026): 1-23
PRISMA, Prosiding Seminar Nasional Matematika
https://proceeding.unnes.ac.id/prisma/index
ISSN 2613-9189

Analisis Peramalan Jumlah Penumpang Kereta Api Bulanan Menggunakan Metode Smoothing, Regresi Tren, ARIMA, dan SARIMA

Abelia Damayanti[1]
[1] *Statistika, Universitas Negeri Semarang, Semarang, 50229, Indonesia*
[1]*abeliadamayanti@students.unnes.ac.id*

Abstrak

Peramalan jumlah penumpang kereta api merupakan hal penting dalam mendukung perencanaan dan evaluasi layanan transportasi publik. Penelitian ini bertujuan untuk memodelkan dan membandingkan beberapa metode peramalan runtun waktu stokastik, yaitu metode smoothing, regresi tren, ARIMA, dan SARIMA, menggunakan data jumlah penumpang kereta api bulanan periode Januari 2010 hingga Desember 2024. Data yang digunakan merupakan data sekunder yang diperoleh dari Badan Pusat Statistik dan PT Kereta Api Indonesia.

Analisis dilakukan menggunakan perangkat lunak R sebagai software utama, Python (Google Colab) sebagai pembanding implementasi model, serta Minitab untuk analisis eksploratif menggunakan metode smoothing sederhana. Tahapan analisis meliputi visualisasi data, uji stasioneritas, pemodelan runtun waktu, serta evaluasi kinerja model menggunakan Root Mean Square Error (RMSE) dan Mean Absolute Percentage Error (MAPE).

Hasil analisis menunjukkan bahwa metode smoothing dan regresi tren menghasilkan peramalan yang relatif sederhana dan kurang mampu menangkap pola musiman yang kuat pada data. Model ARIMA non-musiman memberikan hasil yang lebih baik, namun model SARIMA menunjukkan kinerja peramalan terbaik dengan nilai RMSE dan MAPE paling kecil. Model SARIMA mampu menangkap tren dan pola musiman tahunan pada data jumlah penumpang kereta api, sehingga direkomendasikan sebagai model terbaik dalam penelitian ini.

Kata kunci:

Peramalan, runtun waktu, kereta api, ARIMA, SARIMA

© 2026 Dipublikasikan oleh Jurusan Matematika, Universitas Negeri Semarang

I. PENDAHULUAN

Transportasi kereta api merupakan salah satu moda transportasi utama di Indonesia yang berperan penting dalam menunjang mobilitas masyarakat dan aktivitas ekonomi. Perkembangan jumlah penumpang kereta api dari waktu ke waktu mencerminkan dinamika kebutuhan transportasi serta pengaruh berbagai faktor, seperti pertumbuhan penduduk, kebijakan transportasi, dan kondisi eksternal tertentu. Oleh karena itu, informasi mengenai pola dan kecenderungan jumlah penumpang kereta api menjadi penting dalam proses perencanaan dan evaluasi layanan transportasi publik.

Data jumlah penumpang kereta api yang diamati secara berkala membentuk data runtun waktu yang memiliki karakteristik khusus, seperti adanya tren jangka panjang dan pola musiman. Penelitian sebelumnya menunjukkan bahwa data penumpang kereta api umumnya memiliki pola musiman tahunan yang kuat, sehingga memerlukan metode peramalan yang mampu menangkap karakteristik tersebut secara akurat (Utomo & Fanani, 2020). Selain itu, kejadian ekstrem seperti pandemi COVID-19 juga menyebabkan perubahan pola yang signifikan pada data jumlah penumpang kereta api, sehingga meningkatkan kompleksitas dalam proses peramalan.

Metode peramalan runtun waktu stokastik merupakan pendekatan yang banyak digunakan untuk memodelkan data dengan pola historis yang kompleks. Beberapa metode yang umum digunakan antara lain metode smoothing, regresi tren, Autoregressive Integrated Moving Average (ARIMA), serta Seasonal Autoregressive Integrated Moving Average (SARIMA). Metode smoothing dan regresi tren relatif sederhana dan mudah diterapkan, namun memiliki keterbatasan dalam menangkap pola musiman yang kuat. Sebaliknya, model ARIMA dan SARIMA dirancang untuk menangani ketergantungan data terhadap waktu, dengan model SARIMA secara khusus mampu menangkap pola musiman dalam data runtun waktu (Nurjanah, Ruhiat, & Andiani, 2018).

Selain pemilihan metode peramalan, penggunaan perangkat lunak statistik juga memegang peranan penting dalam proses analisis data runtun waktu. Perangkat lunak seperti R,

To cite this article:
Pertama, P., Kedua, P., & Ketiga, P. (2026). Klik di sini untuk menulis judul anda. *PRISMA, Prosiding Seminar Nasional Matematika 9*, 1-23

Python, dan Minitab memiliki karakteristik, kelebihan, serta keterbatasan masing-masing dalam implementasi metode peramalan. Beberapa penelitian menunjukkan bahwa penggunaan lebih dari satu perangkat lunak dapat memberikan gambaran yang lebih komprehensif terhadap hasil peramalan dan kemudahan implementasi metode (Katabba, 2021).

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menganalisis dan membandingkan hasil peramalan jumlah penumpang kereta api bulanan menggunakan metode smoothing, regresi tren, ARIMA, dan SARIMA. Selain itu, penelitian ini juga membandingkan hasil implementasi model menggunakan perangkat lunak R, Python (Google Colab), dan Minitab, sehingga diperoleh metode dan perangkat lunak yang paling sesuai untuk peramalan jumlah penumpang kereta api.

II. TINJAUAN PUSTAKAN

2.1 Data Runtun Waktu

Data runtun waktu adalah data yang dikumpulkan berdasarkan urutan waktu tertentu, seperti harian, bulanan, atau tahunan. Data runtun waktu memiliki ketergantungan antar periode, sehingga nilai pada waktu tertentu dipengaruhi oleh nilai pada waktu sebelumnya. Komponen utama dalam data runtun waktu meliputi tren, musiman, dan galat acak (Katabba, 2021).

Data jumlah penumpang kereta api yang diamati secara bulanan termasuk dalam data runtun waktu musiman karena memiliki pola yang berulang setiap tahun.

2.2 Stasioneritas Data

Stasioneritas merupakan salah satu asumsi penting dalam analisis runtun waktu. Suatu data dikatakan stasioner apabila nilai rata-rata dan variansnya konstan dari waktu ke waktu. Data runtun waktu yang tidak stasioner dapat menyebabkan model peramalan menghasilkan estimasi yang tidak akurat.

Uji stasioneritas yang umum digunakan adalah Augmented Dickey-Fuller (ADF) test. Apabila hasil uji menunjukkan data tidak stasioner, maka dilakukan proses differencing untuk menghilangkan tren hingga data menjadi stasioner (Nurjanah et al., 2018).

2.3 Metode Single Exponential Smoothing

Single Exponential Smoothing merupakan metode peramalan yang digunakan untuk data runtun waktu tanpa tren dan tanpa pola musiman. Metode ini memberikan bobot lebih besar pada data yang lebih baru.

Model Single Exponential Smoothing dirumuskan sebagai berikut:

$$\hat{y}_{t+1} = \alpha y_1 + (1-\alpha)\hat{y}_t \quad (1)$$

dengan:

- $\hat{y}_{t+1}$ adalah nilai ramalan periode ke- $t+1$
- y_t adalah nilai aktual periode ke-t
- α adalah parameter smoothing dengan $0 < \alpha < 1$.

2.4 Regresi Trend

Regresi tren digunakan untuk memodelkan kecenderungan jangka panjang data runtun waktu. Regresi tren linear dirumuskan sebagai:

$$y_t = \beta_0 + \beta_1 t + \varepsilon_t \quad (2)$$

Sedangkan regresi tren kuadratik dirumuskan sebagai:

$$y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \varepsilon_t \quad (3)$$

Dengan ε_t merupakan galat acak yang diasumsikan berdistribusi normal dengan rata-rata nol.

2.5 Model ARIMA

Model Autoregressive Integrated Moving Average (ARIMA) merupakan metode peramalan runtun waktu yang digunakan untuk data non-musiman yang telah stasioner. Model ARIMA dinyatakan dalam notasi ARIMA(p,d,q), di mana p menunjukkan orde autoregressive, d menunjukkan orde differencing, dan q menunjukkan orde moving average.

Secara umum, model ARIMA dirumuskan sebagai:

$$\phi(B)(1-B)^2 y_t = \theta(B)\varepsilon_t \quad (4)$$

dengan B merupakan operator backshift.

2.6 Model SARIMA

Model Seasonal Autoregressive Integrated Moving Average (SARIMA) merupakan pengembangan dari model ARIMA untuk data runtun waktu musiman. Model SARIMA dinyatakan dalam notasi ARIMA(p,d,q)(P,D,Q)^s, dengan *s* menyatakan periode musiman.

Model SARIMA dirumuskan sebagai:

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D y_t = \theta(B)\Theta B^s \varepsilon_t \quad (5)$$

Model SARIMA sesuai digunakan pada data jumlah penumpang kereta api yang memiliki pola musiman tahunan.

2.7 Ukuran Ketepatan Peramalan

Ketepatan hasil peramalan dievaluasi menggunakan ukuran kesalahan peramalan, yaitu Root Mean Square Error (RMSE) dan Mean Absolute Percentage Error (MAPE).

$$RMSE = \sqrt{\frac{1}{2}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad (6)$$

$$MAPE = \frac{100\%}{N}\sum_{t=1}^{n}\left|\frac{(y_t - \hat{y}_t)}{y_t}\right| \quad (7)$$

III. METODE PENELITIAN

3.1 Jenis dan Pendekatan Penelitian

Penelitian ini merupakan penelitian kuantitatif dengan pendekatan analisis runtun waktu (time series). Pendekatan ini digunakan karena data yang dianalisis tersusun secara kronologis berdasarkan waktu dan memiliki ketergantungan antar periode. Fokus penelitian tidak hanya pada pembentukan model peramalan, tetapi juga pada perbandingan kinerja metode peramalan serta perbedaan hasil implementasi antar perangkat lunak statistik.

3.2 Data dan Sumber Data

Data yang digunakan dalam penelitian ini adalah data sekunder berupa jumlah penumpang kereta api bulanan yang diperoleh dari sumber resmi. Data disusun dalam bentuk runtun waktu bulanan dengan jumlah observasi sebanyak 180 periode.

Karakteristik utama data meliputi:

1. Tren jangka panjang yang cenderung meningkat.
2. Pola musiman tahunan.
3. Fluktuasi ekstrem pada periode tertentu.

Karakteristik tersebut menjadikan data sesuai untuk dianalisis menggunakan metode smoothing, regresi tren, ARIMA, dan SARIMA.

3.3 Perangkat Lunak Analisis

Untuk menjaga objektivitas perbandingan, penelitian menggunakan tiga perangkat lunak dengan peran yang jelas:

1. R Studio

   Digunakan sebagai perangkat utama untuk membangun seluruh model: ETS/smoothing, regresi tren, ARIMA, dan SARIMA, termasuk evaluasi RMSE dan MAPE.

2. Google Colab (Python)

   Digunakan untuk validasi implementasi dan visualisasi hasil model (khususnya ARIMA/SARIMA beserta interval kepercayaan). Colab dipakai agar hasil tidak bergantung pada satu ekosistem software saja.

3. Minitab

   Digunakan untuk pemodelan smoothing berbasis GUI (Single Exponential Smoothing) sebagai pembanding perangkat lunak statistik yang umum digunakan di lingkungan pembelajaran dan industri.

Penggunaan lebih dari satu perangkat lunak bertujuan untuk membandingkan konsistensi hasil peramalan antar software dan mengevaluasi kemudahan implementasi masing-masing metode.

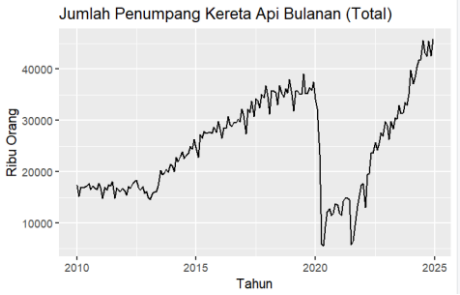
3.4 Tahapan Analisis Data

Tahapan analisis data dalam penelitian ini dilakukan secara sistematis sebagai berikut.

3.4.1 Visualisasi Data Awal

Tahap pertama adalah visualisasi data jumlah penumpang kereta api bulanan untuk memperoleh gambaran awal mengenai pola data, seperti tren dan fluktuasi musiman.

***Gambar 1**. Grafik runtun waktu jumlah penumpang kereta api bulanan*



3.4.2 Pembagian Data Latih dan Data Uji

Data dibagi menjadi dua bagian, yaitu:

- Data latih (training data): digunakan untuk membangun model peramalan.
- Data uji (testing data): digunakan untuk mengevaluasi kinerja model.

Pembagian dilakukan berdasarkan urutan waktu, di mana data hingga periode tertentu digunakan sebagai data latih, sedangkan periode berikutnya digunakan sebagai data uji.

3.4.3 Uji Stasioneritas Data

Uji stasioneritas dilakukan menggunakan Augmented Dickey-Fuller (ADF) test terhadap data latih. Uji ini bertujuan untuk

mengetahui apakah data telah memenuhi asumsi stasioneritas yang diperlukan dalam pemodelan ARIMA dan SARIMA.

Hasil uji menunjukkan bahwa data asli belum bersifat stasioner, sehingga diperlukan proses transformasi melalui differencing.

3.4.4 Differencing Orde Satu

Differencing orde satu dilakukan untuk menghilangkan komponen tren pada data.

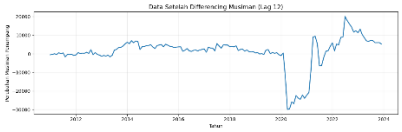
Gambar 2. *Data setelah differencing orde satu*



3.4.5 Differencing Musiman

Karena data memiliki pola musiman tahunan, dilakukan differencing musiman dengan lag 12. Selanjutnya, data hasil differencing orde satu dan differencing musiman digunakan untuk analisis lanjutan.

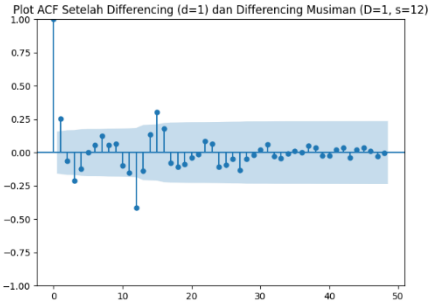
Gambar 3*. Data setelah differencing orde satu dan differencing musiman (lag 12)*



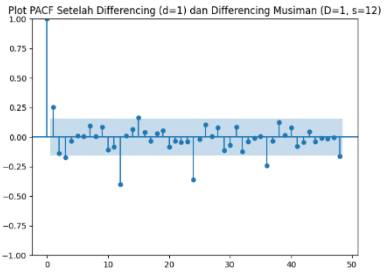
3.4.6 Identifikasi Model ARIMA dan SARIMA

Identifikasi model dilakukan menggunakan plot Autocorrelation Function (ACF) dan Partial Autocorrelation Function (PACF) dari data yang telah stasioner. Plot ini digunakan untuk menentukan orde parameter autoregressive dan moving average, baik non-musiman maupun musiman.

Gambar 4*. Plot ACF data setelah differencing*



Gambar 5*. Plot PACF data setelah differencing*



3.4.7 Pembentukan Model Peramalan

Berdasarkan hasil identifikasi, dibentuk beberapa model peramalan, yaitu:

1. Single Exponential Smoothing
2. Regresi Tren (linear dan kuadratik)
3. ARIMA non-musiman
4. SARIMA musiman

Pemilihan model terbaik dilakukan dengan membandingkan nilai kriteria informasi dan ukuran ketepatan peramalan.

3.4.8 Evaluasi Kinerja Model

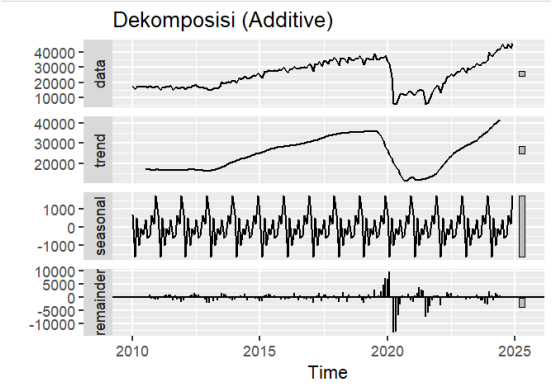
Evaluasi kinerja model dilakukan dengan membandingkan hasil peramalan dengan data aktual pada periode pengujian. Ukuran kesalahan peramalan yang digunakan adalah:

- Root Mean Square Error (RMSE)
- Mean Absolute Percentage Error (MAPE)

3.5 Dekomposisi Data

Untuk mengidentifikasi komponen pembentuk data runtun waktu secara lebih rinci, dilakukan dekomposisi aditif terhadap data jumlah penumpang kereta api. Dekomposisi ini memisahkan data menjadi tiga komponen utama, yaitu tren, musiman, dan residual.

Gambar 6*. Dekomposisi aditif data jumlah penumpang kereta api bulanan*

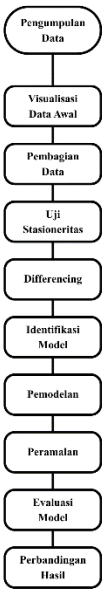


Hasil dekomposisi menunjukkan bahwa komponen tren mengalami peningkatan dalam jangka panjang, sementara komponen musiman memperlihatkan pola yang berulang setiap tahun. Komponen residual relatif berfluktuasi di sekitar nol, yang menunjukkan bahwa sebagian besar variasi data telah dijelaskan oleh komponen tren dan musiman. Temuan ini memperkuat dugaan bahwa data memiliki pola musiman tahunan yang cukup kuat.

3.6 Alur Penelitian

Untuk memperjelas tahapan penelitian, disusun diagram alur penelitian yang menggambarkan proses analisis secara keseluruhan.

Gambar 7. *Diagram alur penelitian peramalan jumlah penumpang kereta api*



IV. HASIL DAN PEMBAHASAN

4.1 Analisis Deskriptif Data

Analisis deskriptif dilakukan untuk memahami pola dasar data jumlah penumpang kereta api bulanan sebelum dilakukan pemodelan peramalan. Berdasarkan visualisasi data historis, terlihat bahwa jumlah penumpang kereta api menunjukkan tren meningkat dalam jangka panjang. Namun demikian, terdapat penurunan yang sangat tajam pada periode tertentu, yang menyebabkan pola data menjadi tidak stabil.

Selain tren, data juga memperlihatkan pola fluktuasi yang berulang setiap tahun, yang mengindikasikan adanya pola musiman tahunan. Pola musiman ini ditandai dengan kenaikan dan penurunan jumlah penumpang pada bulan-bulan tertentu yang relatif konsisten dari tahun ke tahun.

Gambar 8*. Grafik data historis jumlah penumpang kereta api bulanan*



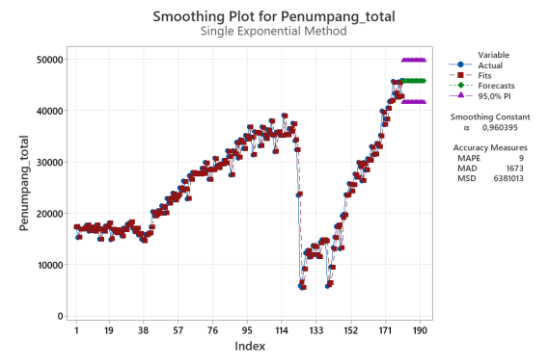
4.2 Hasil Peramalan Menggunakan Metode Smoothing

Metode smoothing yang digunakan dalam penelitian ini adalah Single Exponential Smoothing, yang diterapkan menggunakan perangkat lunak Minitab. Metode ini memberikan bobot lebih besar pada data terbaru, sehingga cocok untuk data tanpa tren dan tanpa pola musiman yang kuat.

Hasil peramalan menunjukkan bahwa metode smoothing menghasilkan nilai ramalan yang relatif konstan pada periode mendatang. Nilai parameter smoothing (α) yang mendekati satu menunjukkan bahwa model sangat responsif terhadap perubahan data terbaru. Namun, metode ini tidak mampu menangkap pola musiman dan perubahan ekstrem pada data jumlah penumpang kereta api.

Evaluasi kinerja model smoothing menunjukkan bahwa nilai kesalahan peramalan relatif lebih besar dibandingkan metode lain yang mempertimbangkan tren dan musiman..

***Gambar 9**. Hasil peramalan menggunakan Single Exponential Smoothing (Minitab)*



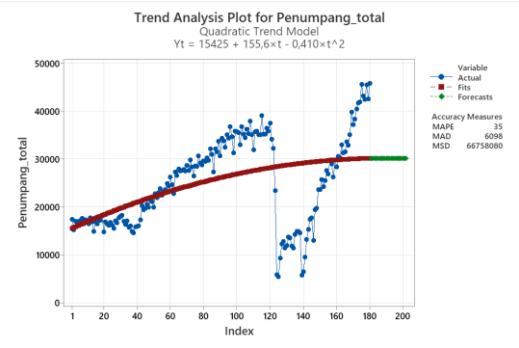
4.3 Hasil Peramalan Menggunakan Metode Regresi Tren

Regresi tren digunakan untuk memodelkan kecenderungan jangka panjang data. Dalam penelitian ini, digunakan regresi tren kuadratik karena mampu menangkap perubahan pola tren yang tidak linier.

Hasil estimasi menunjukkan bahwa model regresi tren mampu mengikuti arah umum peningkatan jumlah penumpang. Namun, model ini tidak mampu menangkap fluktuasi musiman dan perubahan ekstrem pada data. Hal ini tercermin dari pola residual yang masih menunjukkan struktur tertentu dan tidak bersifat acak sepenuhnya.

Nilai kesalahan peramalan yang dihasilkan oleh regresi tren lebih besar dibandingkan model ARIMA dan SARIMA, sehingga metode ini kurang optimal untuk data dengan pola musiman yang kuat.

***Gambar 10**. Hasil peramalan menggunakan regresi tren kuadratik (Minitab)*



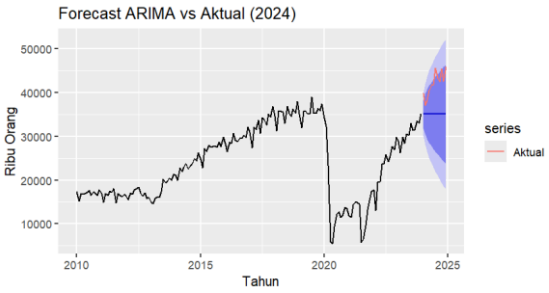
4.4 Hasil Peramalan dengan Model ARIMA

Model ARIMA non-musiman dibangun menggunakan perangkat lunak R dan divalidasi menggunakan Google Colab (Python). Model ini mampu menangkap ketergantungan data terhadap waktu melalui komponen autoregressive dan moving average.

Hasil peramalan menunjukkan bahwa model ARIMA mampu mengikuti pola data historis dengan lebih baik dibandingkan metode smoothing dan regresi tren. Namun, karena model ARIMA tidak secara eksplisit memasukkan komponen musiman, hasil peramalan pada periode tertentu masih menunjukkan deviasi dari data aktual.

Evaluasi residual menunjukkan bahwa residual model relatif acak, meskipun masih terdapat indikasi pola musiman yang belum sepenuhnya terakomodasi.

***Gambar 11**. Perbandingan hasil peramalan ARIMA dengan data aktual*



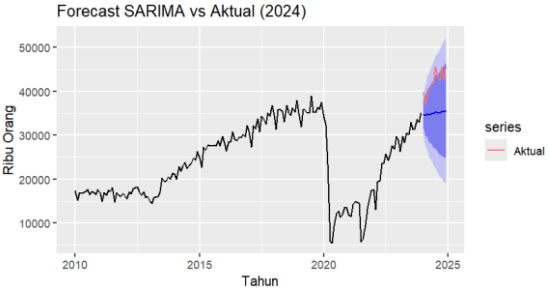
4.5 Hasil Peramalan dengan Model SARIMA

Model SARIMA dibangun untuk mengakomodasi tren dan pola musiman tahunan pada data jumlah penumpang kereta api. Model ini diimplementasikan menggunakan R Studio dan Google Colab, sehingga memungkinkan perbandingan hasil antar perangkat lunak.

Hasil peramalan menunjukkan bahwa model SARIMA mampu mengikuti pola data historis dengan lebih baik dibandingkan metode lain. Model ini berhasil menangkap fluktuasi musiman serta perubahan tren pada data.

Interval kepercayaan hasil peramalan menunjukkan rentang ketidakpastian yang wajar dan mencakup nilai aktual pada periode pengujian. Hal ini menunjukkan bahwa model SARIMA memberikan estimasi peramalan yang stabil dan reliabel.

***Gambar 12**. Hasil peramalan SARIMA beserta interval kepercayaan*



4.6 Perbandingan Kinerja Metode dan Perangkat Lunak

Perbandingan kinerja seluruh metode peramalan dilakukan menggunakan nilai RMSE dan MAPE pada data uji tahun 2024. Hasil perbandingan disajikan pada ***Tabel 1***.

Tabel 1. *Perbandingan Kinerja Metode Peramalan*

Metode	RMSE	MAPE(%)
Smoothing (ETS)	9226,28	21,03
Regresi Trend Linear	13641,25	31,62
Regresi Tren Kuadratik	22541,01	52,71
ARIMA	7488,57	16,29
SARIMA	7385,49	16,31

Berdasarkan ***Tabel 1***, metode regresi tren memiliki tingkat kesalahan paling besar karena tidak mampu menangkap pola musiman. Metode ARIMA dan SARIMA memberikan hasil peramalan yang jauh lebih baik. Model SARIMA menunjukkan performa terbaik berdasarkan nilai RMSE terendah, sehingga direkomendasikan sebagai metode peramalan yang paling sesuai untuk data jumlah penumpang kereta api bulanan.

4.7 Perbandingan Implementasi Perangkat Lunak

Selain membandingkan metode peramalan, penelitian ini juga membandingkan implementasi model menggunakan tiga perangkat lunak statistik, yaitu R Studio, Google Colab (Python), dan Minitab.

R Studio memberikan fasilitas analisis runtun waktu yang paling lengkap, termasuk dekomposisi, uji stasioneritas, identifikasi ACF-PACF, serta evaluasi residual. Google Colab digunakan sebagai pembanding implementasi berbasis Python dan menghasilkan pola peramalan yang konsisten dengan hasil dari R, meskipun proses identifikasi model lebih terbatas. Minitab digunakan untuk analisis smoothing dan tren sederhana berbasis antarmuka grafis, namun tidak mendukung pemodelan ARIMA dan SARIMA secara lengkap.

Secara keseluruhan, R Studio dinilai sebagai perangkat lunak yang paling sesuai untuk analisis peramalan runtun waktu yang komprehensif, sedangkan Google Colab dan Minitab berperan sebagai pembanding dan pendukung hasil analisis.

4.8 Pembahasan

Secara keseluruhan, hasil penelitian menunjukkan bahwa pemilihan metode peramalan harus disesuaikan dengan karakteristik data. Metode sederhana seperti smoothing dan regresi tren relatif mudah diterapkan, namun kurang efektif untuk data dengan pola musiman yang kuat. Model ARIMA memberikan peningkatan kinerja peramalan, namun masih memiliki keterbatasan dalam menangkap pola musiman.

Model SARIMA memberikan hasil terbaik karena mampu mengakomodasi komponen tren dan musiman secara simultan. Temuan ini konsisten dengan hasil analisis awal pada Bab III serta mendukung penggunaan model SARIMA untuk peramalan jumlah penumpang kereta api bulanan.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

1. Data jumlah penumpang kereta api bulanan memiliki karakteristik utama berupa tren jangka panjang dan pola musiman tahunan yang kuat. Hal ini telah dikonfirmasi melalui analisis awal data, dekomposisi runtun waktu, serta pengujian stasioneritas.
2. Metode peramalan smoothing (ETS) dan regresi tren relatif mudah diterapkan, namun memiliki keterbatasan dalam menangkap pola musiman pada data. Akibatnya, kedua metode tersebut menghasilkan tingkat kesalahan peramalan yang lebih besar dibandingkan metode berbasis model runtun waktu stokastik.
3. Model ARIMA non-musiman memberikan hasil peramalan yang lebih baik dibandingkan metode smoothing dan regresi tren, namun masih kurang optimal dalam merepresentasikan pola musiman tahunan karena tidak memasukkan komponen musiman secara eksplisit.
4. Model SARIMA menunjukkan kinerja peramalan terbaik dengan nilai RMSE dan MAPE paling kecil pada periode pengujian. Hal ini menunjukkan bahwa model SARIMA paling mampu menangkap karakteristik data

yang mengandung tren dan musiman secara simultan.

5. Perbandingan implementasi model menggunakan R Studio, Google Colab (Python), dan Minitab menunjukkan bahwa hasil peramalan ARIMA dan SARIMA pada R Studio dan Google Colab bersifat konsisten. Perbedaan perangkat lunak tidak memberikan perbedaan yang signifikan terhadap hasil peramalan, selama metode dan parameter yang digunakan sama. Minitab lebih sesuai digunakan untuk analisis eksploratif dan metode peramalan sederhana.

Secara keseluruhan, penelitian ini menunjukkan bahwa model SARIMA merupakan metode yang paling sesuai untuk peramalan jumlah penumpang kereta api bulanan.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran yang dapat diberikan adalah sebagai berikut:

1. Penelitian selanjutnya dapat mempertimbangkan penggunaan model peramalan yang lebih kompleks, seperti SARIMAX atau model berbasis machine learning, dengan memasukkan variabel eksternal yang berpotensi memengaruhi jumlah penumpang kereta api.
2. Periode data dapat diperluas atau dianalisis secara terpisah pada kondisi tertentu untuk melihat pengaruh kejadian ekstrem terhadap pola peramalan secara lebih mendalam.
3. Perbandingan perangkat lunak dapat dikembangkan dengan menambahkan platform statistik lain untuk memberikan gambaran yang lebih komprehensif terkait efisiensi dan fleksibilitas implementasi metode peramalan.

DAFTAR PUSTAKA

Katabba, Y. I. (2021). Metode Seasonal Autoregressive Integrated Moving Average (SARIMA) untuk Memprediksi Jumlah Penumpang Kereta Api di Pulau Sumatera.

Nurjanah, I. S., Ruhiat, D., & Andiani, D. (2018). Implementasi model autoregressive integrated moving average (arima) untuk peramalan jumlah penumpang kereta api di pulau sumatera 1,2,3, *3*(2), 145–156.

R. Heksantoro, "Penumpang KA Selama Libur Akhir Tahun di Daop 5 PurwokwertoNaik 5%," 1 1 2019. [Online]. Available: https://news.detik.com/berita-jawatengah/d-4366951/penumpang-ka-selama-libur-akhir-tahun-di-daop-5-purwokertonaik-5. [Accessed 13 10 2019]

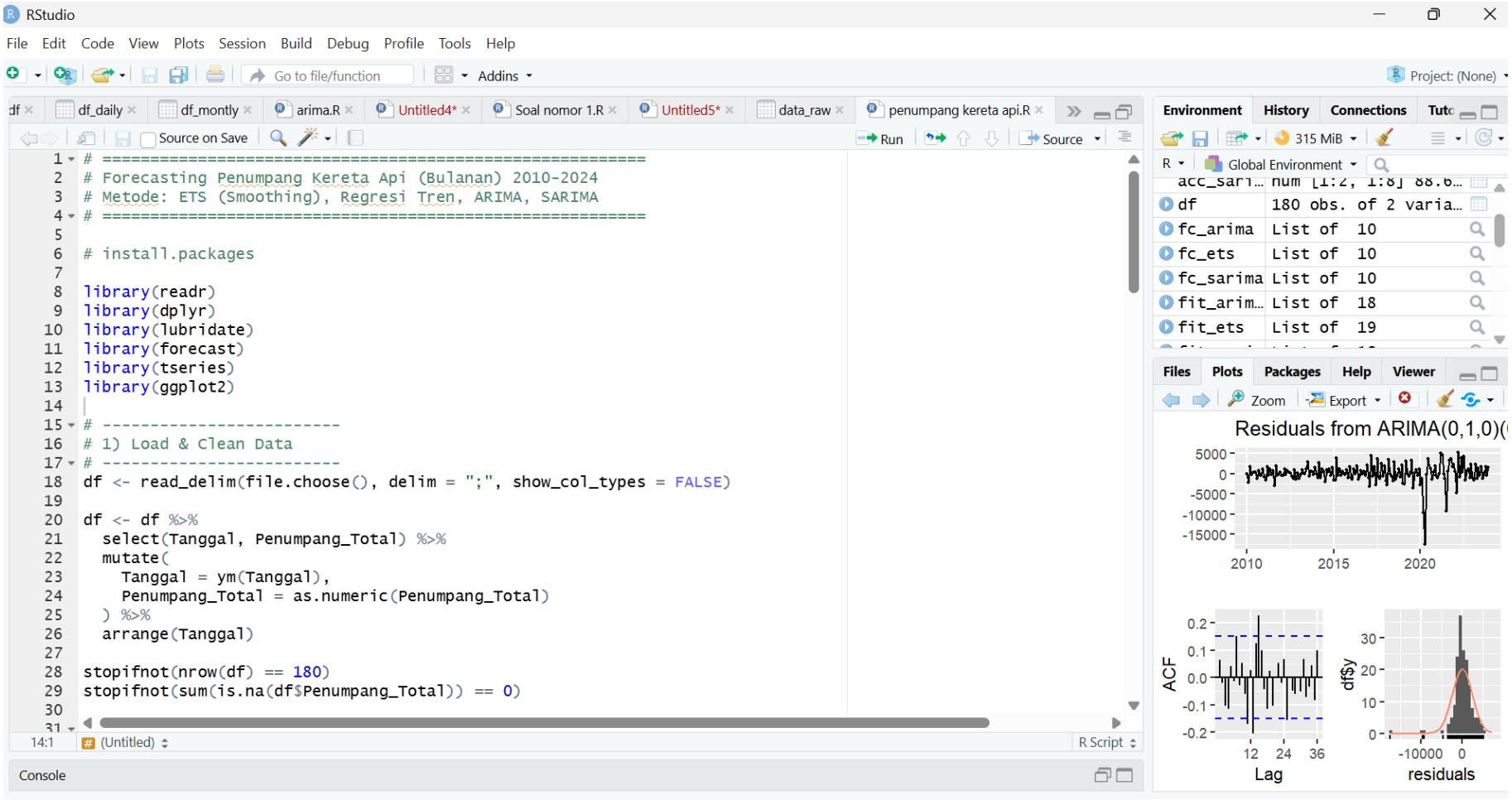
Utomo, P., & Fanani, A. (2020). Peramalan Jumlah Penumpang Kereta Api di Indonesia Menggunakan Metode Seasonal Autoregressive Integrated Moving Average (SARIMA), 169–178.

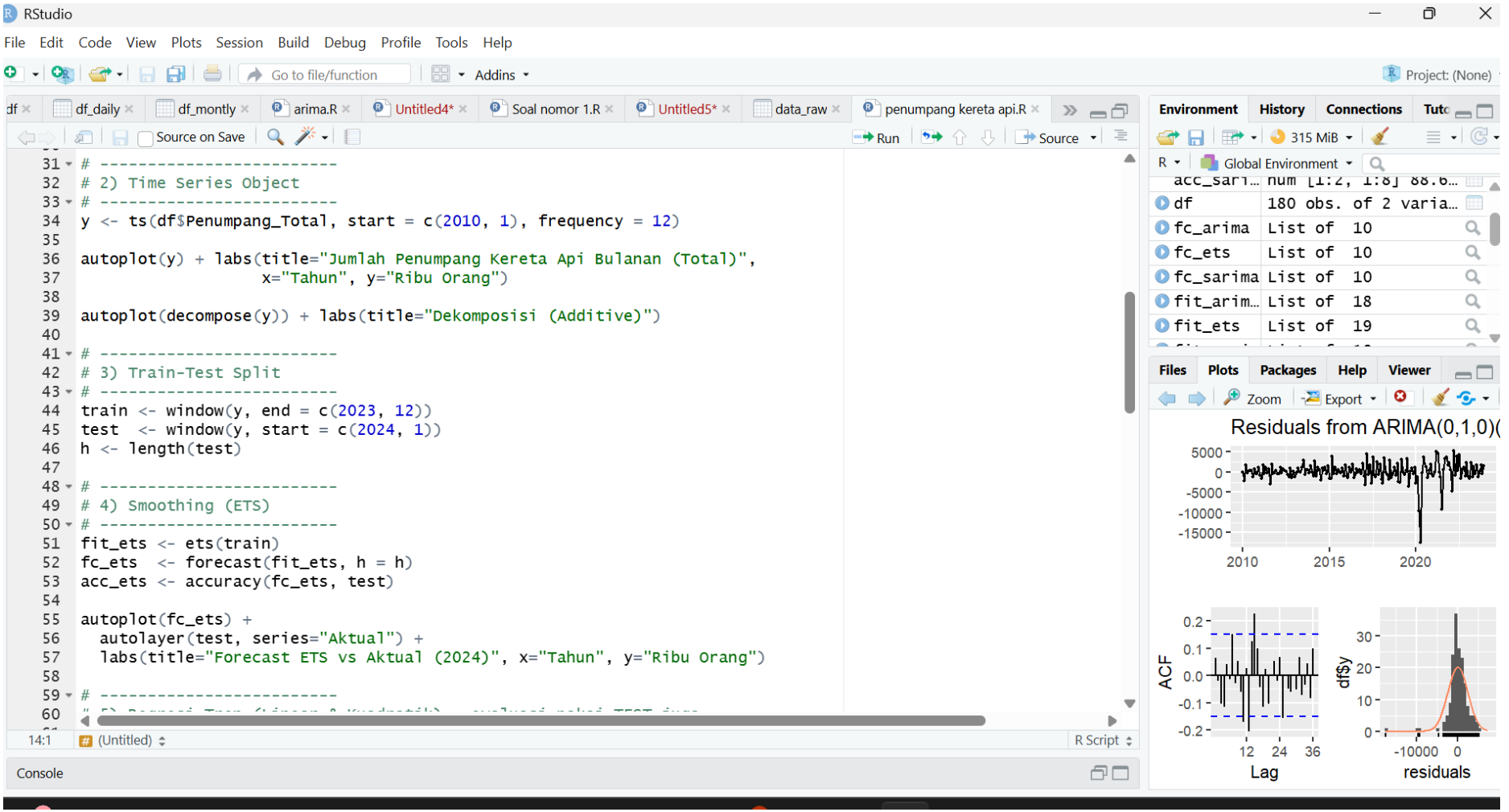
W. Y. Nasir, Peramalan Jumlah Penumpang Dari Pelayaran Dalam Negeri di Pelabuhan Kota Makasar Menggunakan SARIMA, Makassar: Universitas Islam Negeri Alaudin Makassar, 2015.

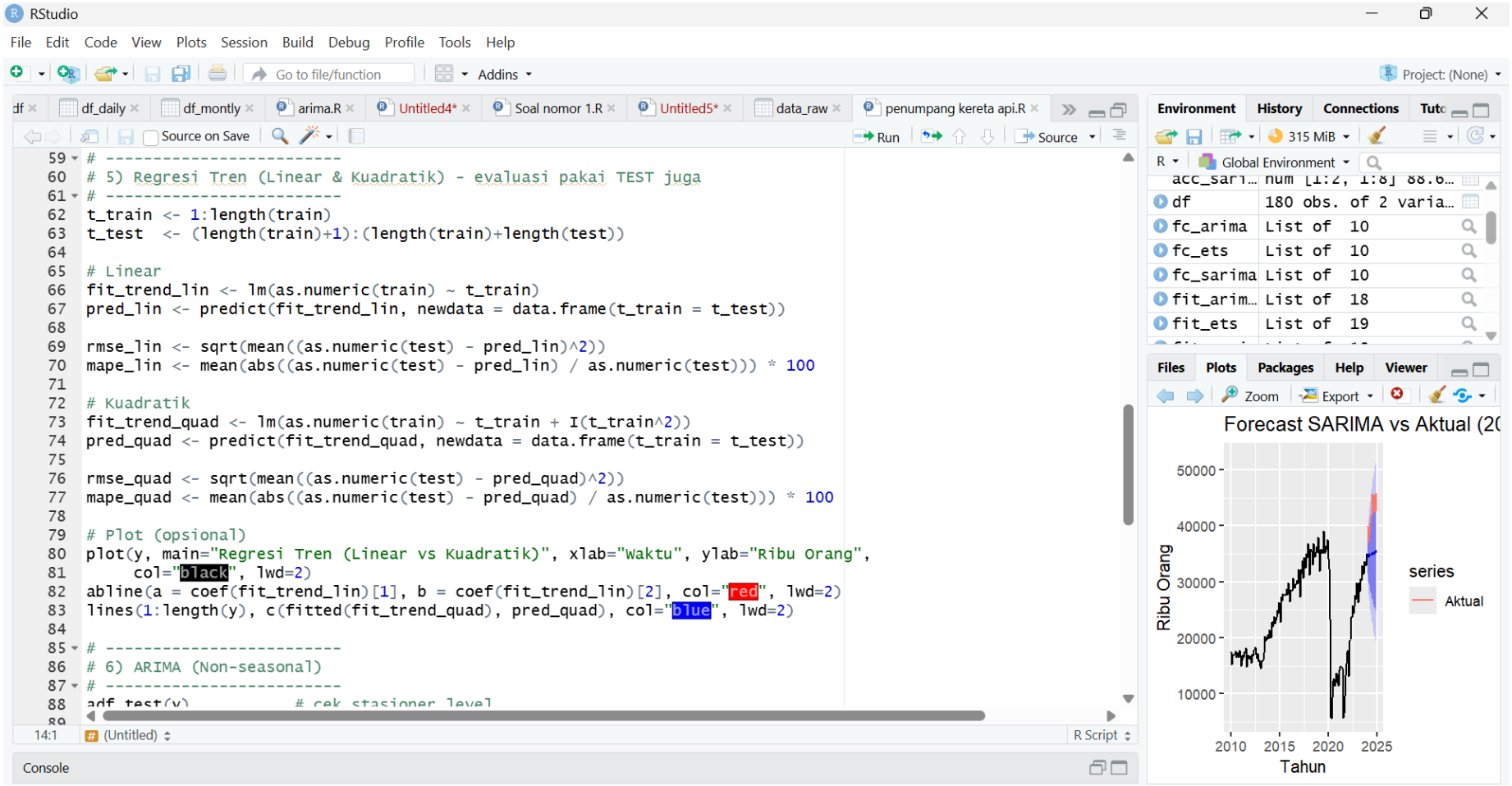
Lampiran

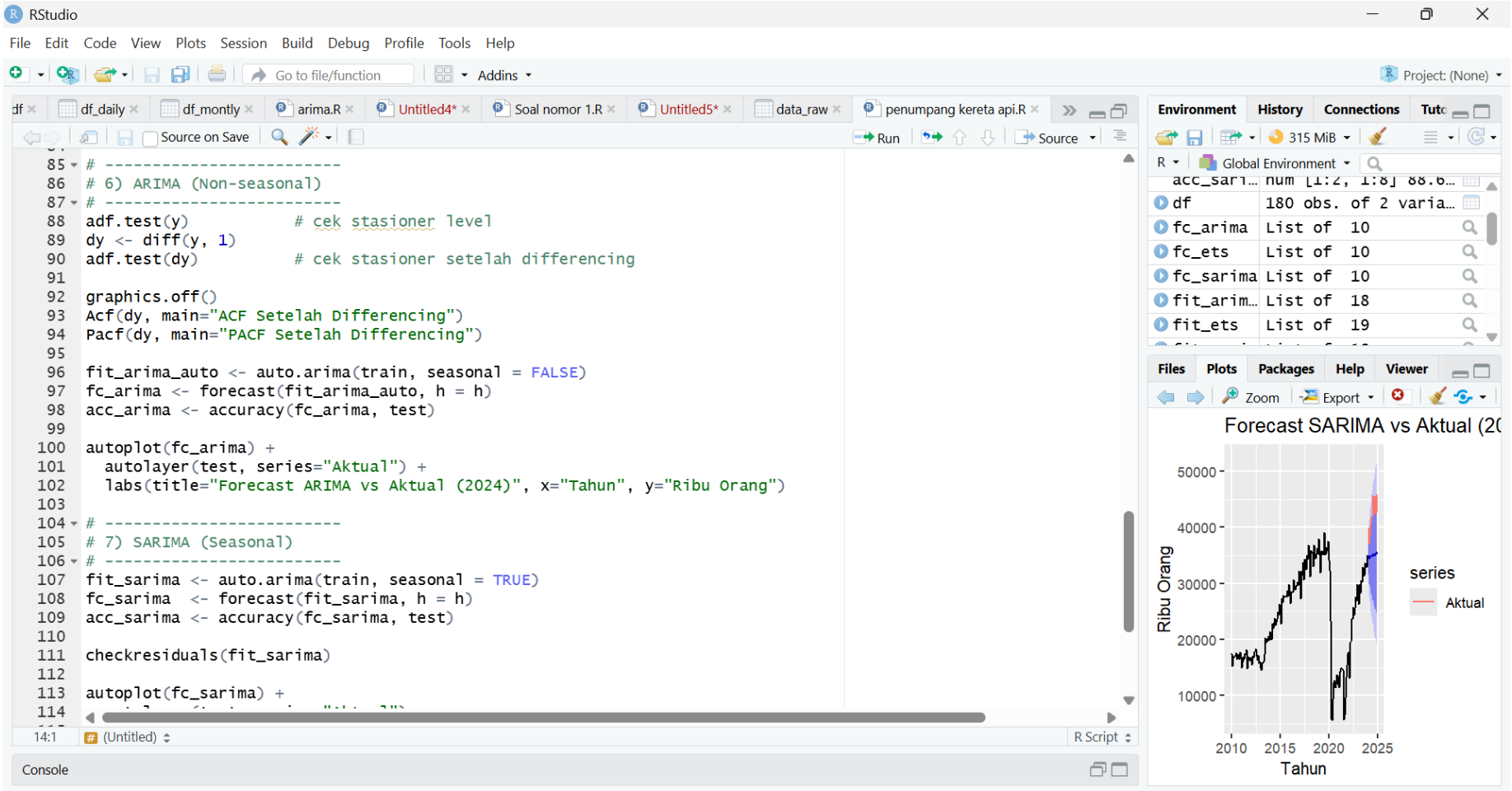
JUMLAH PENUMPANG KERETA API (2010-2024)															
2010-01	17424	2012-01	16283	2014-01	21092	2016-01	28358	2018-01	34717	2020-01	34130	2022-01	17737	2024-01	39860
2010-02	15207	2012-02	15490	2014-02	19998	2016-02	26510	2018-02	31278	2020-02	32283	2022-02	13067	2024-02	37166
2010-03	16992	2012-03	17090	2014-03	22836	2016-03	28617	2018-03	35875	2020-03	23425	2022-03	19438	2024-03	38335
2010-04	16832	2012-04	16746	2014-04	21908	2016-04	28435	2018-04	35754	2020-04	5898	2022-04	19713	2024-04	40459
2010-05	16988	2012-05	17771	2014-05	22988	2016-05	30703	2018-05	35482	2020-05	5484	2022-05	23594	2024-05	41704
2010-06	17259	2012-06	18062	2014-06	23840	2016-06	29159	2018-06	33030	2020-06	9290	2022-06	23612	2024-06	41888
2010-07	17680	2012-07	18309	2014-07	22500	2016-07	28831	2018-07	36800	2020-07	12238	2022-07	25741	2024-07	45605
2010-08	16477	2012-08	17056	2014-08	23199	2016-08	29588	2018-08	35190	2020-08	12774	2022-08	24288	2024-08	43230
2010-09	17301	2012-09	16368	2014-09	23593	2016-09	29516	2018-09	34504	2020-09	11429	2022-09	25567	2024-09	42481
2010-10	16908	2012-10	17127	2014-10	24923	2016-10	30263	2018-10	36236	2020-10	11937	2022-10	27610	2024-10	45499
2010-11	16469	2012-11	15773	2014-11	24356	2016-11	29690	2018-11	35298	2020-11	13722	2022-11	26924	2024-11	42590
2010-12	17733	2012-12	16104	2014-12	26275	2016-12	32150	2018-12	37965	2020-12	13515	2022-12	29824	2024-12	45829
2011-01	16891	2013-01	14900	2015-01	24676	2017-01	30949	2019-01	35122	2021-01	11901	2023-01	29017		
2011-02	14890	2013-02	14594	2015-02	22790	2017-02	27342	2019-02	31899	2021-02	11479	2023-02	26259		
2011-03	16978	2013-03	15826	2015-03	27267	2017-03	32170	2019-03	35751	2021-03	14284	2023-03	29820		
2011-04	16441	2013-04	16000	2015-04	26565	2017-04	31502	2019-04	35809	2021-04	14864	2023-04	28325		
2011-05	17522	2013-05	16113	2015-05	27910	2017-05	33745	2019-05	35102	2021-05	14857	2023-05	30534		
2011-06	17265	2013-06	17301	2015-06	27562	2017-06	30723	2019-06	35090	2021-06	14556	2023-06	30237		
2011-07	18132	2013-07	20245	2015-07	27612	2017-07	34310	2019-07	39035	2021-07	5747	2023-07	32948		
2011-08	14846	2013-08	19423	2015-08	27796	2017-08	33791	2019-08	35189	2021-08	6515	2023-08	31323		
2011-09	16921	2013-09	19738	2015-09	27549	2017-09	32498	2019-09	35221	2021-09	9566	2023-09	31518		
2011-10	16461	2013-10	20534	2015-10	28718	2017-10	35070	2019-10	36448	2021-10	13250	2023-10	33588		
2011-11	16179	2013-11	19919	2015-11	27669	2017-11	34361	2019-11	35877	2021-11	15317	2023-11	32910		
2011-12	16811	2013-12	21417	2015-12	29831	2017-12	36807	2019-12	37463	2021-12	17427	2023-12	35059		

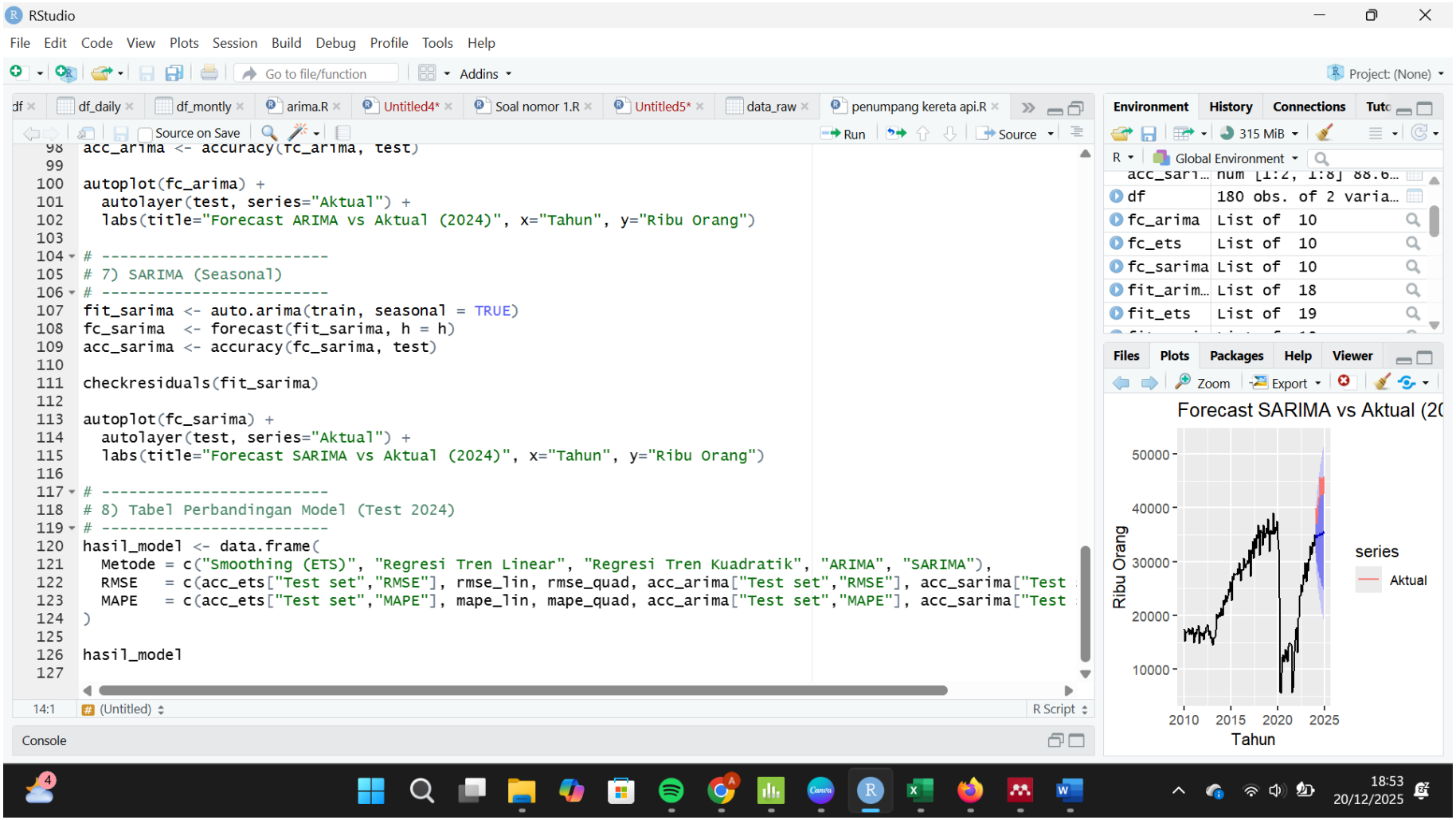
Tampilan R


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
df_daily
df_montly
arima.R
Untitled4*
Soal nomor 1.R
Untitled5*
data_raw
penumpang kereta api.R
Source on Save
Run
Source
# ==========
# Forecasting Penumpang Kereta Api (Bulanan) 2010-2024
# Metode: ETS (Smoothing), Regresi Tren, ARIMA, SARIMA
# ==========
# install.packages
library(readr)
library(dplyr)
library(lubridate)
library(forecast)
library(tseries)
library(ggplot2)
# ----------
# 1) Load & Clean Data
# ----------
df <- read_delim(file.choose(), delim = ";", show_col_types = FALSE)
df <- df %>%
select(Tanggal, Penumpang_Total) %>%
mutate(
Tanggal = ym(Tanggal),
Penumpang_Total = as.numeric(Penumpang_Total)
) %>%
arrange(Tanggal)
stopifnot(nrow(df) == 180)
stopifnot(sum(is.na(df$Penumpang_Total)) == 0)
14:1
(Untitled)
R Script
Console
Environment
History
Connections
315 MiB
Global Environment
df 180 obs. of 2 varia...
fc_arima List of 10
fc_ets List of 10
fc_sarima List of 10
fit_arim... List of 18
fit_ets List of 19
Files
Plots
Packages
Help
Viewer
Zoom
Export
Residuals from ARIMA(0,1,0)(
ACF
Lag
df$y
residuals

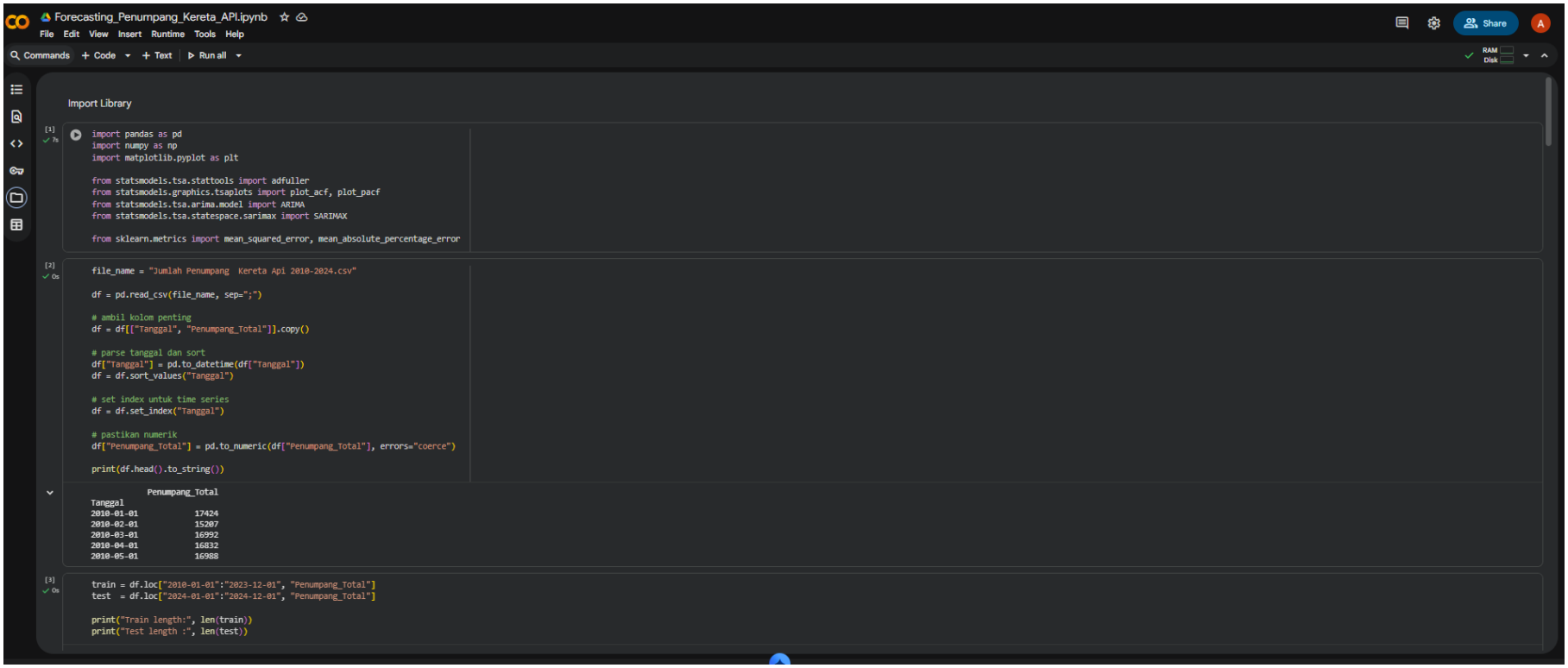

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
df
df_daily
df_montly
arima.R
Untitled4*
Soal nomor 1.R
Untitled5*
data_raw
penumpang kereta api.R
Source on Save
Run
Source
# ------------------------
# 2) Time Series Object
# ------------------------
y <- ts(df$Penumpang_Total, start = c(2010, 1), frequency = 12)
autoplot(y) + labs(title="Jumlah Penumpang Kereta Api Bulanan (Total)",
x="Tahun", y="Ribu Orang")
autoplot(decompose(y)) + labs(title="Dekomposisi (Additive)")
# ------------------------
# 3) Train-Test Split
# ------------------------
train <- window(y, end = c(2023, 12))
test <- window(y, start = c(2024, 1))
h <- length(test)
# ------------------------
# 4) Smoothing (ETS)
# ------------------------
fit_ets <- ets(train)
fc_ets <- forecast(fit_ets, h = h)
acc_ets <- accuracy(fc_ets, test)
autoplot(fc_ets) +
autolayer(test, series="Aktual") +
labs(title="Forecast ETS vs Aktual (2024)", x="Tahun", y="Ribu Orang")
# ------------------------
14:1
(Untitled)
R Script
Console
Environment
History
Connections
315 MiB
R
Global Environment
df 180 obs. of 2 varia...
fc_arima List of 10
fc_ets List of 10
fc_sarima List of 10
fit_arim... List of 18
fit_ets List of 19
Files
Plots
Packages
Help
Viewer
Zoom
Export
Residuals from ARIMA(0,1,0)(
5000
0
-5000
-10000
-15000
2010
2015
2020
ACF
0.2
0.1
0.0
-0.1
-0.2
12
24
36
Lag
df$y
30
20
10
0
-10000
0
residuals

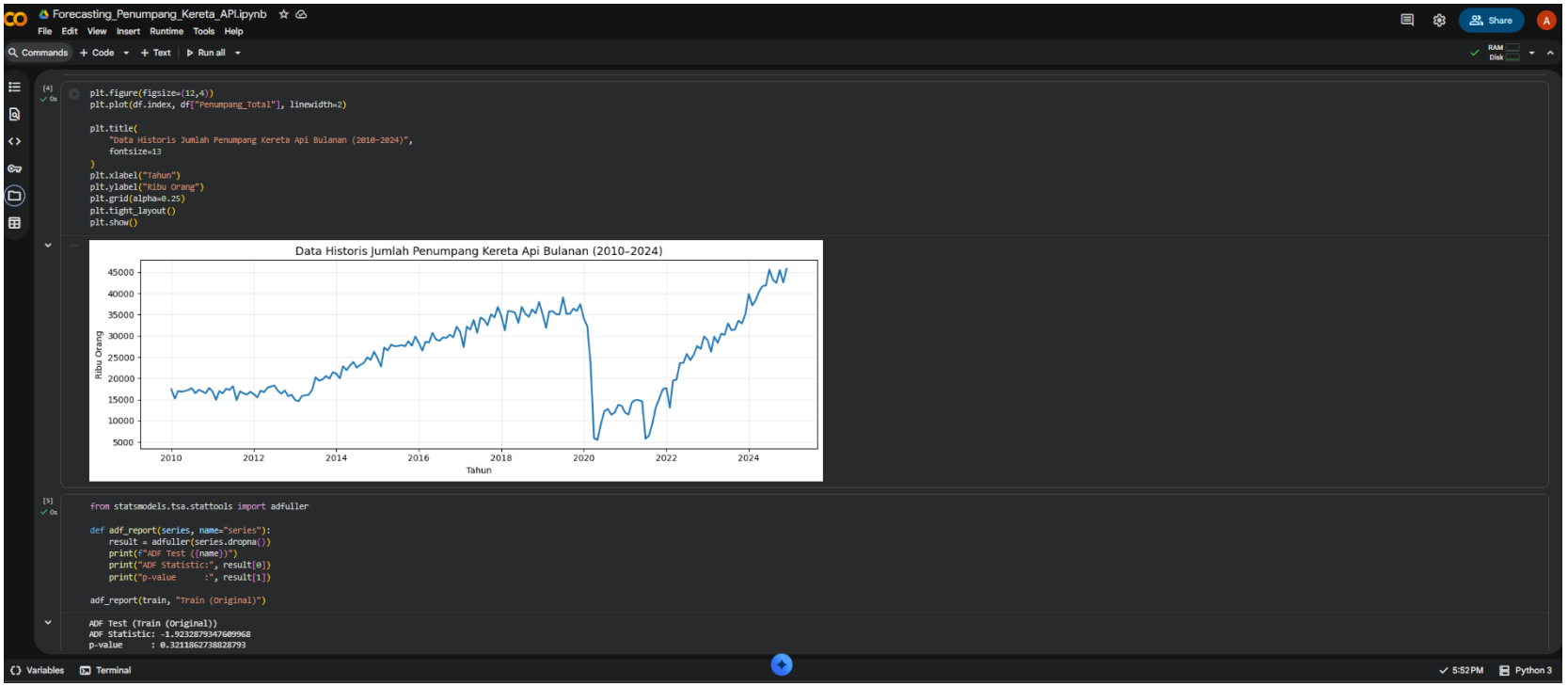

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
df
df_daily
df_montly
arima.R
Untitled4*
Soal nomor 1.R
Untitled5*
data_raw
penumpang kereta api.R
Source on Save
Run
Source
# -------------------------
# 5) Regresi Tren (Linear & Kuadratik) - evaluasi pakai TEST juga
# -------------------------
t_train <- 1:length(train)
t_test <- (length(train)+1):(length(train)+length(test))
# Linear
fit_trend_lin <- lm(as.numeric(train) ~ t_train)
pred_lin <- predict(fit_trend_lin, newdata = data.frame(t_train = t_test))
rmse_lin <- sqrt(mean((as.numeric(test) - pred_lin)^2))
mape_lin <- mean(abs((as.numeric(test) - pred_lin) / as.numeric(test))) * 100
# Kuadratik
fit_trend_quad <- lm(as.numeric(train) ~ t_train + I(t_train^2))
pred_quad <- predict(fit_trend_quad, newdata = data.frame(t_train = t_test))
rmse_quad <- sqrt(mean((as.numeric(test) - pred_quad)^2))
mape_quad <- mean(abs((as.numeric(test) - pred_quad) / as.numeric(test))) * 100
# Plot (opsional)
plot(y, main="Regresi Tren (Linear vs Kuadratik)", xlab="Waktu", ylab="Ribu Orang",
col="black", lwd=2)
abline(a = coef(fit_trend_lin)[1], b = coef(fit_trend_lin)[2], col="red", lwd=2)
lines(1:length(y), c(fitted(fit_trend_quad), pred_quad), col="blue", lwd=2)
# -------------------------
# 6) ARIMA (Non-seasonal)
# -------------------------
adf.test(y) # cek stasioner level
14:1
(Untitled)
R Script
Console
Environment
History
Connections
315 MiB
Global Environment
df 180 obs. of 2 varia...
fc_arima List of 10
fc_ets List of 10
fc_sarima List of 10
fit_arim... List of 18
fit_ets List of 19
Files
Plots
Packages
Help
Viewer
Zoom
Export
Forecast SARIMA vs Aktual (20
Ribu Orang
50000
40000
30000
20000
10000
2010
2015
2020
2025
Tahun
series
Aktual

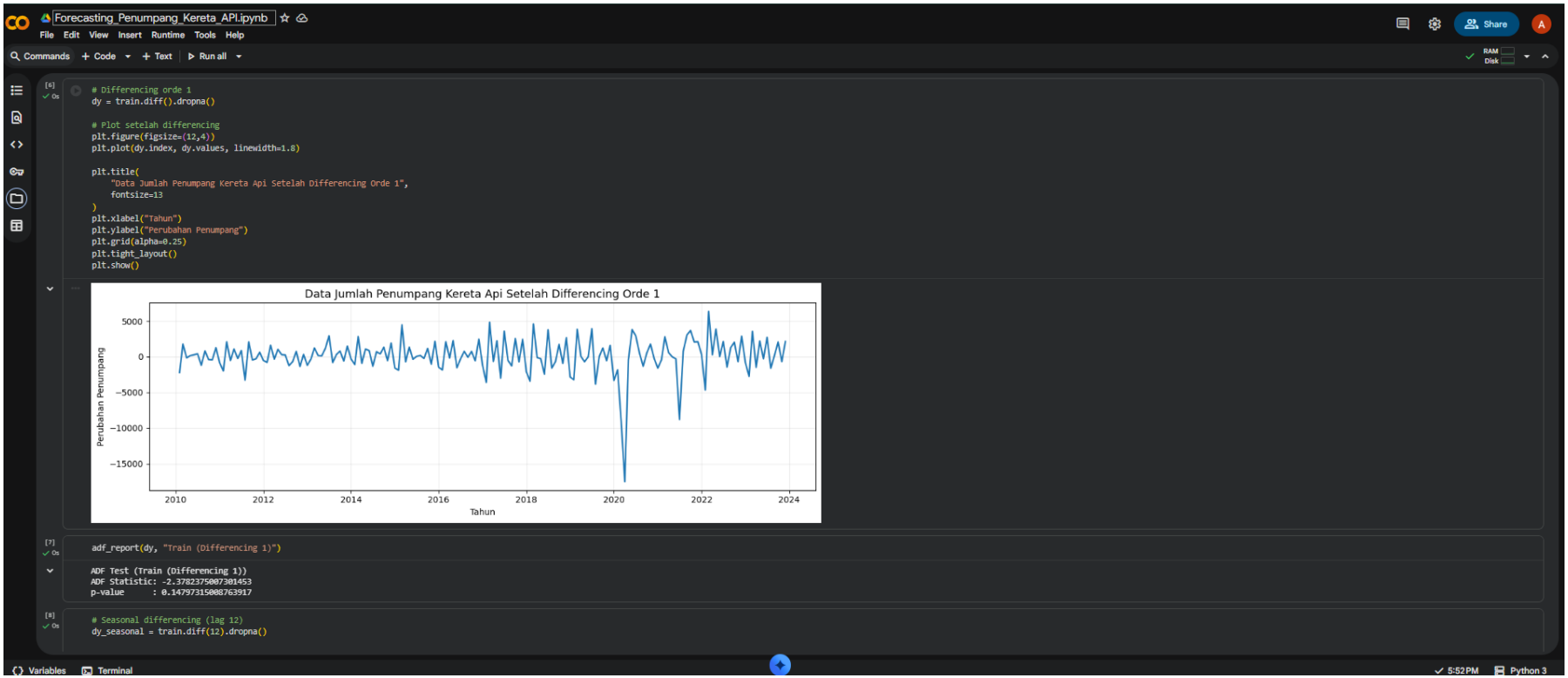

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
df
df_daily
df_montly
arima.R
Untitled4*
Soal nomor 1.R
Untitled5*
data_raw
penumpang kereta api.R
Source on Save
Run
Source
----
6) ARIMA (Non-seasonal)
----
adf.test(y) # cek stasioner level
dy <- diff(y, 1)
adf.test(dy) # cek stasioner setelah differencing
graphics.off()
Acf(dy, main="ACF Setelah Differencing")
Pacf(dy, main="PACF Setelah Differencing")
fit_arima_auto <- auto.arima(train, seasonal = FALSE)
fc_arima <- forecast(fit_arima_auto, h = h)
acc_arima <- accuracy(fc_arima, test)
autoplot(fc_arima) +
autolayer(test, series="Aktual") +
labs(title="Forecast ARIMA vs Aktual (2024)", x="Tahun", y="Ribu Orang")
# ------------------------
# 7) SARIMA (Seasonal)
# ------------------------
fit_sarima <- auto.arima(train, seasonal = TRUE)
fc_sarima <- forecast(fit_sarima, h = h)
acc_sarima <- accuracy(fc_sarima, test)
checkresiduals(fit_sarima)
autoplot(fc_sarima) +
14:1
(Untitled)
R Script
Console
Environment
History
Connections
315 MiB
Global Environment
df 180 obs. of 2 varia...
fc_arima List of 10
fc_ets List of 10
fc_sarima List of 10
fit_arim... List of 18
fit_ets List of 19
Files
Plots
Packages
Help
Viewer
Zoom
Export
Forecast SARIMA vs Aktual (20
Ribu Orang
50000
40000
30000
20000
10000
2010
2015
2020
2025
Tahun
series
Aktual

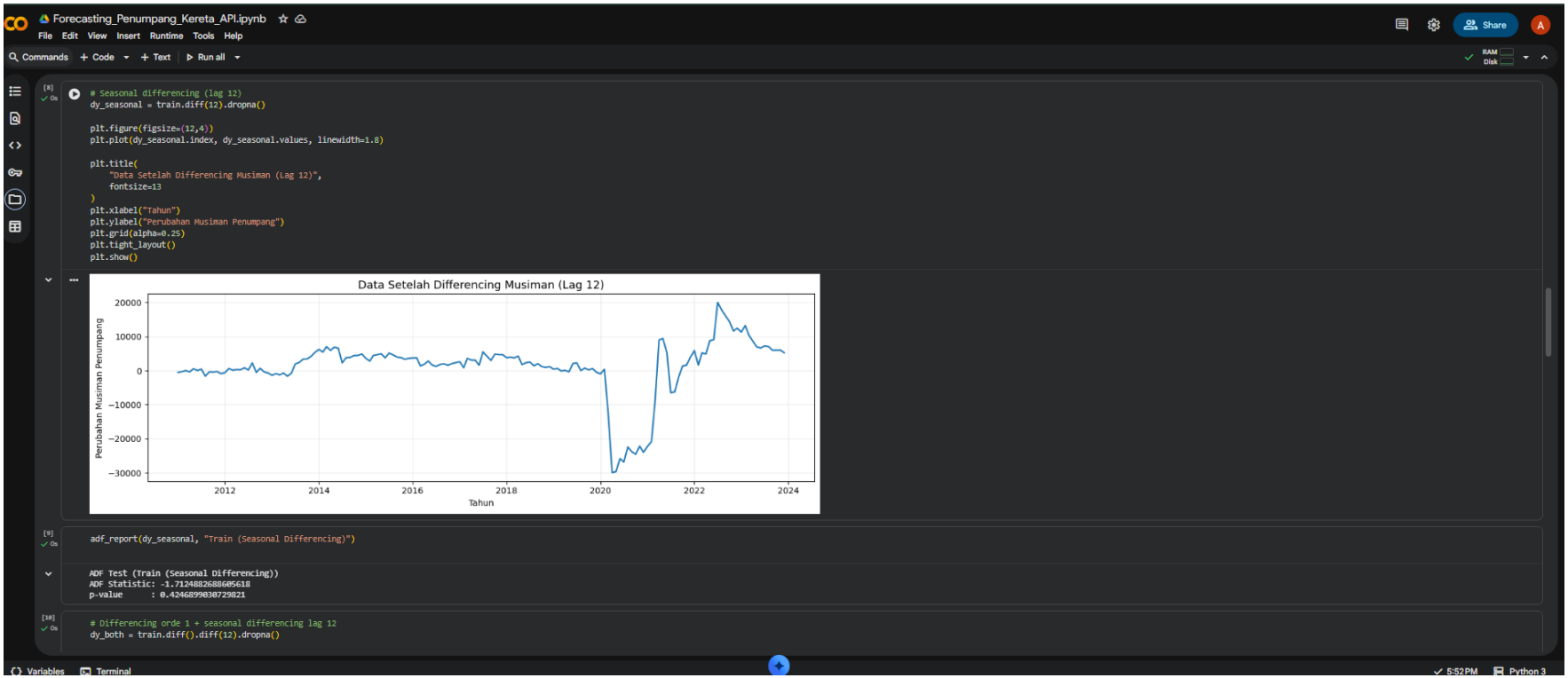

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
df
df_daily
df_montly
arima.R
Untitled4*
Soal nomor 1.R
Untitled5*
data_raw
penumpang kereta api.R
Source on Save
Run
Source
acc_arima <- accuracy(fc_arima, test)
autoplot(fc_arima) +
autolayer(test, series="Aktual") +
labs(title="Forecast ARIMA vs Aktual (2024)", x="Tahun", y="Ribu Orang")
# -------------------------
# 7) SARIMA (Seasonal)
# -------------------------
fit_sarima <- auto.arima(train, seasonal = TRUE)
fc_sarima <- forecast(fit_sarima, h = h)
acc_sarima <- accuracy(fc_sarima, test)
checkresiduals(fit_sarima)
autoplot(fc_sarima) +
autolayer(test, series="Aktual") +
labs(title="Forecast SARIMA vs Aktual (2024)", x="Tahun", y="Ribu Orang")
# -------------------------
# 8) Tabel Perbandingan Model (Test 2024)
# -------------------------
hasil_model <- data.frame(
Metode = c("Smoothing (ETS)", "Regresi Tren Linear", "Regresi Tren Kuadratik", "ARIMA", "SARIMA"),
RMSE = c(acc_ets["Test set","RMSE"], rmse_lin, rmse_quad, acc_arima["Test set","RMSE"], acc_sarima["Test
MAPE = c(acc_ets["Test set","MAPE"], mape_lin, mape_quad, acc_arima["Test set","MAPE"], acc_sarima["Test
)
hasil_model
14:1
(Untitled)
R Script
Console
Environment
History
Connections
315 MiB
Global Environment
df 180 obs. of 2 varia...
fc_arima List of 10
fc_ets List of 10
fc_sarima List of 10
fit_arim... List of 18
fit_ets List of 19
Files
Plots
Packages
Help
Viewer
Zoom
Export
Forecast SARIMA vs Aktual (20
Ribu Orang
50000
40000
30000
20000
10000
2010
2015
2020
2025
Tahun
series
Aktual
18:53
20/12/2025

Tampilan Google Colab




Forecasting_Penumpang_Kereta_API.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
[4]
plt.figure(figsize=(12,4))
plt.plot(df.index, df["Penumpang_Total"], linewidth=2)
plt.title(
"Data Historis Jumlah Penumpang Kereta Api Bulanan (2010–2024)",
fontsize=13
)
plt.xlabel("Tahun")
plt.ylabel("Ribu Orang")
plt.grid(alpha=0.25)
plt.tight_layout()
plt.show()
Data Historis Jumlah Penumpang Kereta Api Bulanan (2010–2024)
Ribu Orang
45000
40000
35000
30000
25000
20000
15000
10000
5000
2010
2012
2014
2016
2018
2020
2022
2024
Tahun
[5]
from statsmodels.tsa.stattools import adfuller
def adf_report(series, name="series"):
result = adfuller(series.dropna())
print(f"ADF Test ({name})")
print("ADF Statistic:", result[0])
print("p-value :", result[1])
adf_report(train, "Train (Original)")
ADF Test (Train (Original))
ADF Statistic: -1.9232879347609968
p-value : 0.3211862738828793
Variables Terminal
5:52 PM Python 3


Forecasting_Penumpang_Kereta_API.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM
Disk
[6]
0s
# Differencing orde 1
dy = train.diff().dropna()
# Plot setelah differencing
plt.figure(figsize=(12,4))
plt.plot(dy.index, dy.values, linewidth=1.8)
plt.title(
"Data Jumlah Penumpang Kereta Api Setelah Differencing Orde 1",
fontsize=13
)
plt.xlabel("Tahun")
plt.ylabel("Perubahan Penumpang")
plt.grid(alpha=0.25)
plt.tight_layout()
plt.show()
Data Jumlah Penumpang Kereta Api Setelah Differencing Orde 1
Perubahan Penumpang
5000
0
−5000
−10000
−15000
2010
2012
2014
2016
2018
2020
2022
2024
Tahun
[7]
0s
adf_report(dy, "Train (Differencing 1)")
ADF Test (Train (Differencing 1))
ADF Statistic: -2.3782375007301453
p-value : 0.14797315008763917
[8]
0s
# Seasonal differencing (lag 12)
dy_seasonal = train.diff(12).dropna()
Variables
Terminal
5:52PM
Python 3


Forecasting_Penumpang_Kereta_API.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
# Seasonal differencing (lag 12)
dy_seasonal = train.diff(12).dropna()
plt.figure(figsize=(12,4))
plt.plot(dy_seasonal.index, dy_seasonal.values, linewidth=1.8)
plt.title(
"Data Setelah Differencing Musiman (Lag 12)",
fontsize=13
)
plt.xlabel("Tahun")
plt.ylabel("Perubahan Musiman Penumpang")
plt.grid(alpha=0.25)
plt.tight_layout()
plt.show()
Data Setelah Differencing Musiman (Lag 12)
Perubahan Musiman Penumpang
20000
10000
0
−10000
−20000
−30000
2012
2014
2016
2018
2020
2022
2024
Tahun
adf_report(dy_seasonal, "Train (Seasonal Differencing)")
ADF Test (Train (Seasonal Differencing))
ADF Statistic: -1.712488268805618
p-value : 0.424689030729821
# Differencing orde 1 + seasonal differencing lag 12
dy_both = train.diff().diff(12).dropna()
Variables Terminal
5:52PM Python 3

Forecasting_Penumpang_Kereta_API.ipynb

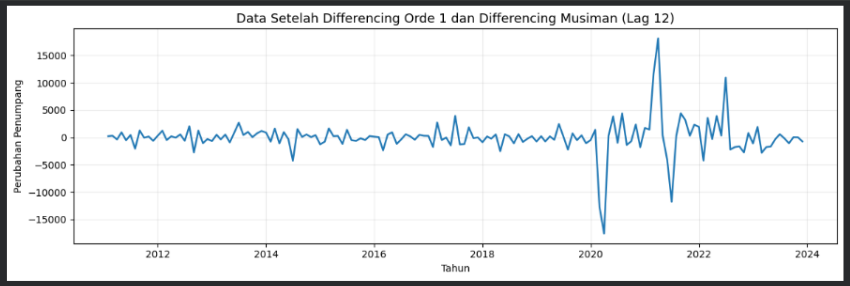
File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
# Differencing orde 1 + seasonal differencing lag 12
dy_both = train.diff().diff(12).dropna()

# Plot
plt.figure(figsize=(12,4))
plt.plot(dy_both.index, dy_both.values, linewidth=1.8)

plt.title(
    "Data Setelah Differencing Orde 1 dan Differencing Musiman (Lag 12)",
    fontsize=13
)
plt.xlabel("Tahun")
plt.ylabel("Perubahan Penumpang")
plt.grid(alpha=0.25)
plt.tight_layout()
plt.show()
```



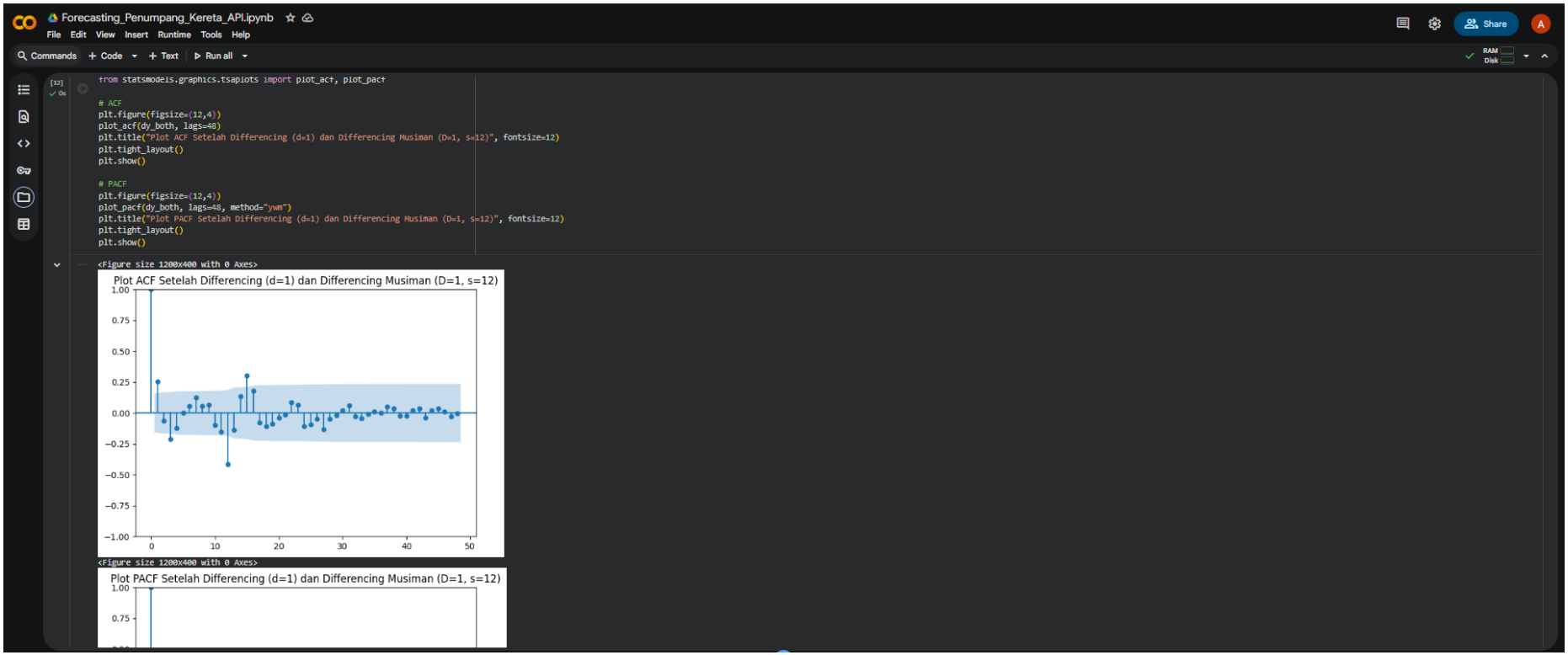

```
adf_report(dy_both, "Train (Diff 1 + Seasonal Diff 12)")
```

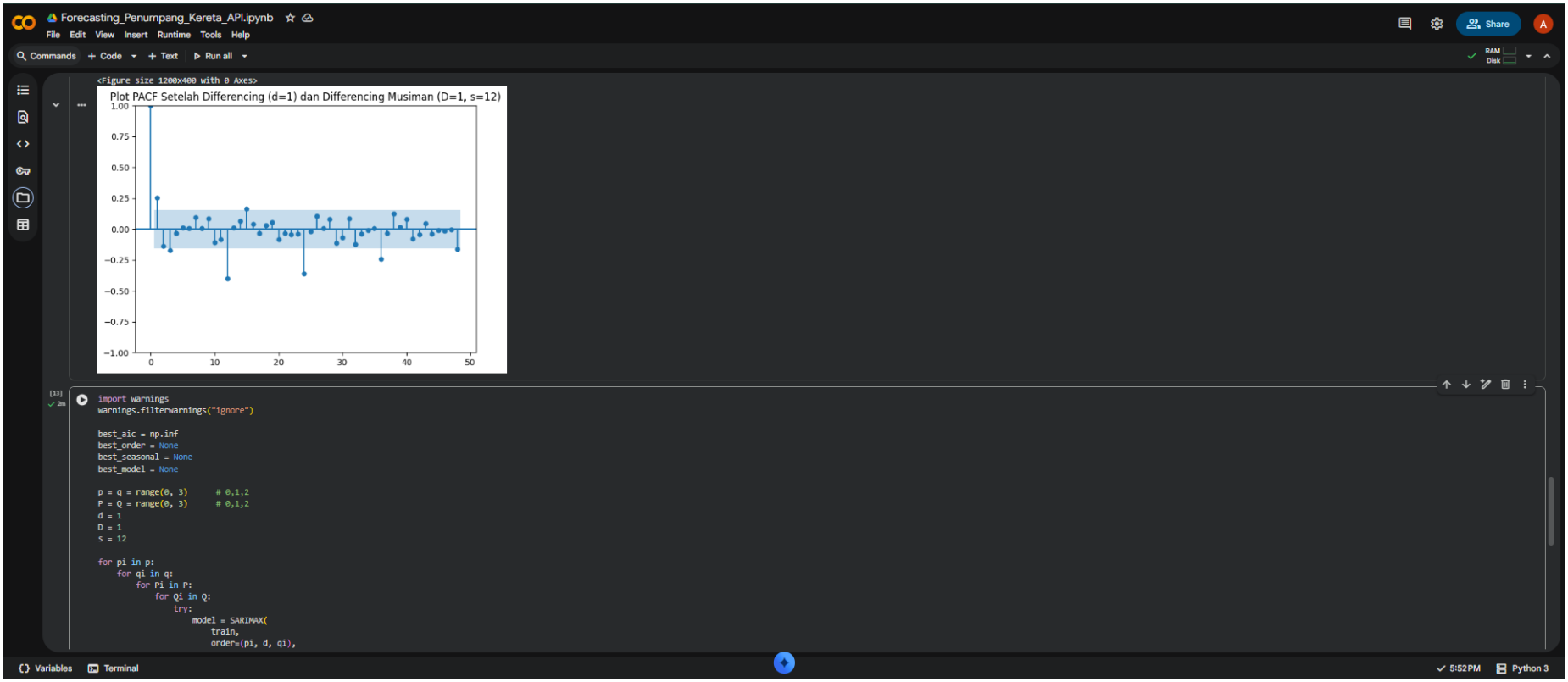
```
ADF Test (Train (Diff 1 + Seasonal Diff 12))
ADF Statistic: -5.471667763282614
p-value      : 2.3910563366804634e-06
```

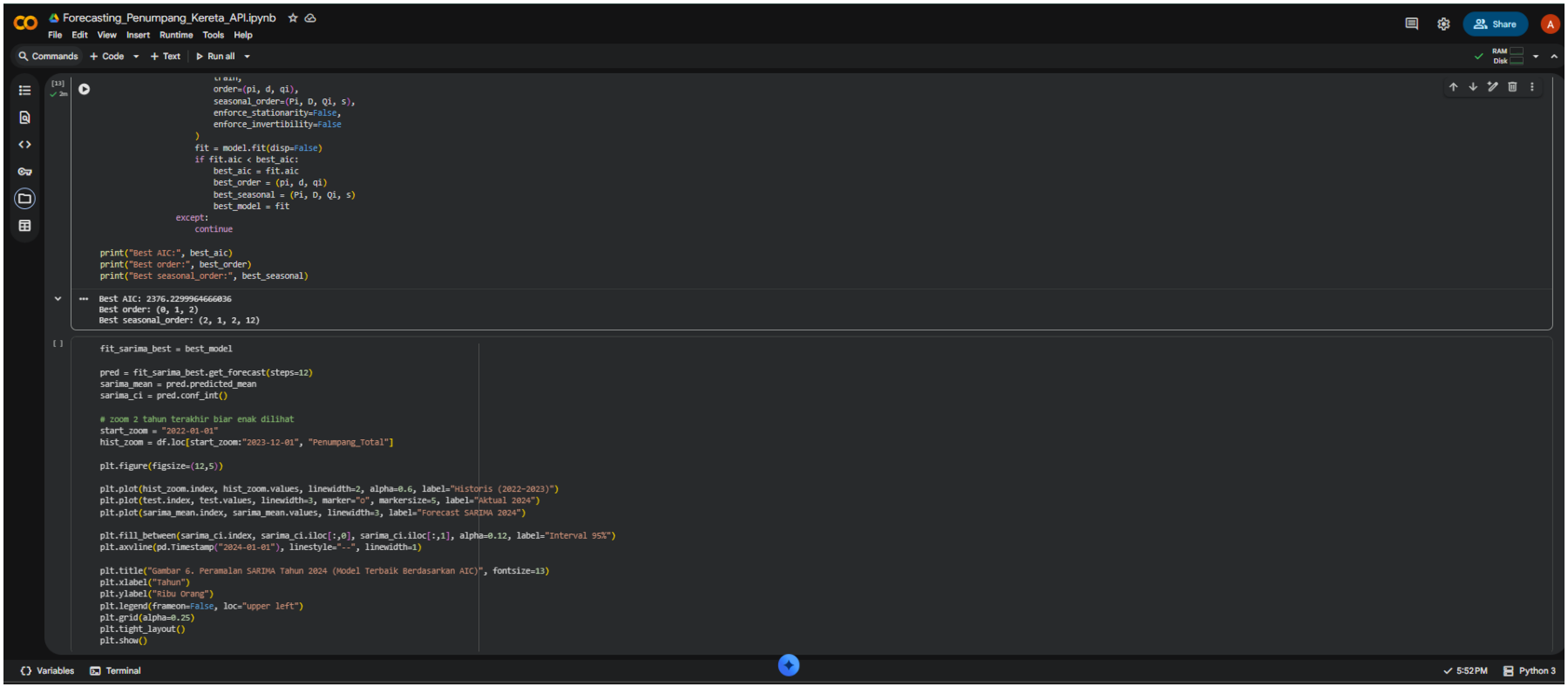
```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# ACF
```

Variables Terminal 5:52PM Python 3


Forecasting_Penumpang_Kereta_API.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM
Disk
[12]
0s
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
# ACF
plt.figure(figsize=(12,4))
plot_acf(dy_both, lags=48)
plt.title("Plot ACF Setelah Differencing (d=1) dan Differencing Musiman (D=1, s=12)", fontsize=12)
plt.tight_layout()
plt.show()
# PACF
plt.figure(figsize=(12,4))
plot_pacf(dy_both, lags=48, method="ywm")
plt.title("Plot PACF Setelah Differencing (d=1) dan Differencing Musiman (D=1, s=12)", fontsize=12)
plt.tight_layout()
plt.show()
<Figure size 1200x400 with 0 Axes>
Plot ACF Setelah Differencing (d=1) dan Differencing Musiman (D=1, s=12)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
<Figure size 1200x400 with 0 Axes>
Plot PACF Setelah Differencing (d=1) dan Differencing Musiman (D=1, s=12)
1.00
0.75


Forecasting_Penumpang_Kereta_API.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM
Disk
<Figure size 1200x400 with 0 Axes>
Plot PACF Setelah Differencing (d=1) dan Differencing Musiman (D=1, s=12)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
[33]
3m
import warnings
warnings.filterwarnings("ignore")
best_aic = np.inf
best_order = None
best_seasonal = None
best_model = None
p = q = range(0, 3) # 0,1,2
P = Q = range(0, 3) # 0,1,2
d = 1
D = 1
s = 12
for pi in p:
for qi in q:
for Pi in P:
for Qi in Q:
try:
model = SARIMAX(
train,
order=(pi, d, qi),
Variables
Terminal
5:52PM
Python 3



https://colab.research.google.com/drive/1WTAkVBIK6d2lg3Tue9Yg16HzquAUKRQI?usp=sharing