



Penerapan Metode Time Series Dalam Forecasting Penjualan Pada “Nasi Goreng Bacot”

Bagas Adil Putrajaya ¹, Agung Brastama Putra ², Rizka Hadiwiyati ³

^{1,2,3} UPN "Veteran" Jawa Timur, Indonesia

Korespondensi penulis: bagas.adil10@gmail.com

Abstract, *The restaurant industry in Indonesia has experienced significant growth, driving the need for data-driven strategies to remain competitive. This study aims to apply and compare time series methods in forecasting sales at "Nasi Goreng Bacot" restaurant. The methods used are Simple Moving Average (SMA), Weighted Moving Average (WMA), and Single Exponential Smoothing (SES), with a focus on sales data from the year 2023. The research results indicate that SMA provides the most accurate predictions, with a Mean Absolute Error (MAE) value of 296.67, Mean Squared Error (MSE) of 129055.6, and Mean Absolute Percentage Error (MAPE) of 3.02%. WMA and SES, although useful in certain data conditions, show higher error rates in this case. This study confirms the effectiveness of SMA in the context of stable and less fluctuating restaurant sales data. With these results, restaurants can plan their inventory of raw materials and workforce more efficiently, reduce waste, and improve customer satisfaction.*

Keywords: *Forecasting, Time Series, Simple Moving Average, Weighted Moving Average, Single Exponential Smoothing, Restaurant Sales.*

Abstrak, Industri restoran di Indonesia mengalami pertumbuhan yang signifikan, mendorong kebutuhan akan strategi berbasis data untuk tetap kompetitif. Penelitian ini bertujuan untuk menerapkan dan membandingkan metode time series dalam forecasting penjualan pada restoran "Nasi Goreng Bacot". Metode yang digunakan adalah *Simple Moving Average (SMA)*, *Weighted Moving Average (WMA)*, dan *Single Exponential Smoothing (SES)*, dengan fokus pada data penjualan tahun 2023. Hasil penelitian menunjukkan bahwa SMA memberikan prediksi yang paling akurat dengan nilai *Mean Absolute Error (MAE)* sebesar 296.67, *Mean Squared Error (MSE)* sebesar 129055.6, dan *Mean Absolute Percentage Error (MAPE)* sebesar 3.02%. WMA dan SES, meskipun berguna dalam kondisi data tertentu, menunjukkan tingkat kesalahan yang lebih tinggi dalam kasus ini. Penelitian ini mengkonfirmasi keefektifan SMA dalam konteks data penjualan restoran yang stabil dan kurang fluktuatif. Dengan hasil ini, restoran dapat merencanakan stok bahan baku dan tenaga kerja dengan lebih efisien, mengurangi pemborosan, dan meningkatkan kepuasan pelanggan.

Kata Kunci : *Forecasting, Time Series, Simple Moving Average, Weighted Moving Average, Single Exponential Smoothing, Penjualan Restoran.*

1. LATAR BELAKANG

Tujuan utama dari forecasting di industri restoran adalah untuk mengestimasi jumlah pemesanan atau penjualan makanan dan minuman di masa depan. Dengan data peramalan yang tepat, manajemen restoran dapat merencanakan stok bahan baku dengan lebih cermat, mengatur tenaga kerja berdasarkan permintaan yang diharapkan, dan mengidentifikasi peluang untuk menarik pelanggan dengan promosi atau penawaran khusus (Apriliani dkk., 2020)

Metode forecasting yang digunakan untuk mengukurnya adalah dengan metode time series. Metode ini melibatkan analisis data historis untuk mengidentifikasi pola dan tren yang berulang dari waktu ke waktu. Dengan melihat data masa lalu, restoran dapat memprediksi pola yang mungkin muncul di masa depan dan mengambil tindakan yang sesuai. Tantangan utama dalam forecasting restoran melibatkan ketidakpastian akibat perubahan tren makanan,

peristiwa tak terduga, dan fluktuasi musiman yang dapat mempengaruhi permintaan. Peramalan (forecasting) penjualan umumnya menggunakan metode Exponential Smoothing, suatu tipe teknik peramalan rata-rata bergerak yang melakukan penimbangan terhadap data masa lalu dengan cara eksponensial sehingga data paling akhir mempunyai bobot atau timbangan lebih besar dalam rata-rata bergerak (Tarsisius Handoko, 2018). Forecasting (peramalan) menjadi alat penting dalam industri restoran untuk merencanakan dengan lebih baik, mengoptimalkan persediaan, dan menyesuaikan tenaga kerja sesuai permintaan. Dengan memanfaatkan berbagai metode forecasting yang sesuai, manajemen restoran dapat lebih siap menghadapi fluktuasi pasar dan memperkuat daya saing mereka dalam bisnis makanan dan minuman yang kompetitif (Lasek, A. dkk., 2019).

Banyaknya penjualan makanan maupun minuman pada Nasi Goreng menampilkan pola tren bersumber pada data historis penjualan. Exponential Smoothing dan Moving Average ialah dua tata cara dengan nilai error terendah yang digunakan meramalkan informasi kuantitatif (Liu dkk., 2020). Mereka pula sesuai buat memperkirakan informasi berpola tren. Tiap ditaksir tata cara tentu hendak menciptakan kesalahan (Tanizaki dkk., 2020). Mean absolute error (MAE) serta Mean squared error (MSE) ialah perlengkapan ukur yang digunakan buat menghitung kesalahan peramalan. Bila nilai MAE serta MSE kecil, peramalan dikatakan akurat (Frischa dkk., 2023). Hasil peramalan dijadikan acuan dalam efisiensi operasional serta menyiapkan perencanaan terhadap perkembangan pasar kedepan sehingga bisaenuhi kebutuhan bahan bersumber pada hasil peramalan.

2. TINJAUAN PUSTAKA

2.1 Profil Restoran Nasi Goreng “Bacot” Surabaya



Gambar 2.1 Profil Usaha: Nasi Goreng “Bacot” Surabaya

Nasi Goreng “Bacot” Surabaya adalah sebuah restoran yang didirikan oleh Ibu Susi pada tahun 2020 di kota Surabaya. Terletak strategis di Jl. Rungkut Madya No. 133, Rungkut, Surabaya, restoran ini menawarkan berbagai varian nasi goreng yang lezat dan inovatif.

Berbekal pengalaman panjang di dunia kuliner, Ibu Susi memulai usaha ini dengan tujuan menyajikan hidangan nasi goreng yang autentik namun memiliki sentuhan modern yang sesuai dengan selera masyarakat urban.

2.2 Forecasting

Metode forecasting adalah pendekatan atau teknik yang digunakan untuk meramalkan atau memprediksi nilai-nilai di masa depan berdasarkan data historis atau informasi yang tersedia. Tujuannya adalah untuk mengidentifikasi pola, tren, dan fluktuasi yang dapat mempengaruhi perubahan nilai di masa mendatang. Berbagai metode forecasting digunakan dalam berbagai konteks, termasuk bisnis, ekonomi, ilmu sosial, dan lain-lain (Sri Radina Putri NWP, H. S., 2018).

Exponential Smoothing

Exponential Smoothing merupakan metode peramalan dengan nilai rata-rata dengan teknik exponential. Data paling akhir pada metode ini memiliki bobot paling besar pada rata-rata bergerak (Heizer, J., and Render, B., 2020). Metode pemulusan eksponensial dapat memperoleh nilai parameter yang tidak diketahui dengan memperkirakannya dari data observasi. Nilai awal dan parameter yang tidak diketahui dapat diperkirakan dengan meminimalkan jumlah kesalahan kuadrat (SSE).

2.3 Single Exponential Smoothing

Analisis *Exponential Smoothing* merupakan salah satu analisis *time series*, dan merupakan metode peramalan dengan memberi nilai pembobot pada serangkaian pengamatan sebelumnya untuk memprediksi nilai masa depan (Heizer, J., and Render, B., 2020). Dalam konteks bisnis restoran, Peramalan Eksponensial dapat digunakan untuk memprediksi penjualan atau permintaan makanan di masa depan. Ini bisa sangat berguna untuk mengelola stok bahan makanan, mengatur jadwal karyawan, dan merencanakan promosi atau acara khusus.

2.4 Moving Average

Metode Moving Average (MA) adalah salah satu metode yang digunakan dalam analisis deret waktu untuk menghaluskan data dan mengidentifikasi pola tren atau fluktuasi dalam data. Metode ini melibatkan perhitungan rata-rata dari sejumlah periode waktu tertentu untuk menghasilkan titik data yang lebih halus.

2.5 Mean absolute error

Mean absolute error (MAE) adalah salah satu metode yang digunakan untuk mengukur tingkat keakuratan model peramalan. Nilai MAE menunjukkan rata – rata kesalahan (error) absolut antara hasil peramalan/prediksi dengan nilai riil (Subagyo, Pangestu, 2020).

2.6 Mean Absolute Percentage Error

Mean absolute persentase error (MAPE) adalah metode pengukuran kesalahan dalam metode peramalan dengan teknik kesalahan absolut di setiap periode dibagi dengan nilai pengamatan nyata untuk periode itu [4]. Selanjutnya hasilnya dihitung nilai rata-rata kesalahan persentase absolut. MAPE adalah tes kesalahan yang mencari nilai persentase perbedaan antara data aktual dan data perkiraan.

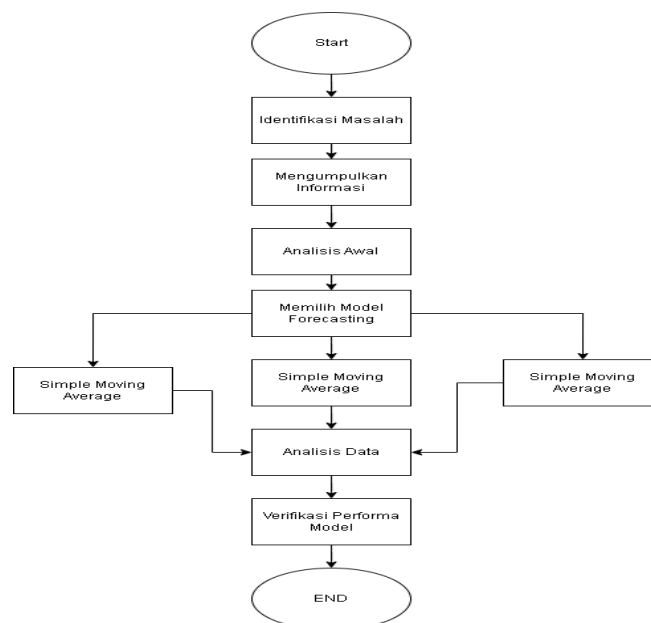
2.7 Python

Python adalah bahasa pemrograman yang sering digunakan dalam analisis data dan pengembangan model prediktif, termasuk untuk tujuan forecasting atau peramalan. Forecasting sendiri adalah proses perkiraan atau prediksi nilai masa depan berdasarkan data historis (Biri, R., Langi, Y. A. R., & Paendong, M. S., 2019).

3. METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian yang dilakukan pada penelitian ini digambarkan melaluidiagram yang ditunjukkan pada gambar berikut.



Gambar 3. 1 Langkah-langkah dalam melakukan penelitian

3.2 Identifikasi Masalah

1. Ketidakpastian dalam Permintaan:

Restoran "Nasi Goreng Bacot" menghadapi tantangan dalam memprediksi permintaan pelanggan. Ketidakpastian ini dapat menyebabkan kekurangan stok bahan baku atau

pemborosan bahan baku yang tidak terpakai, yang pada akhirnya dapat mengurangi kepuasan pelanggan dan meningkatkan biaya operasional.

2. Kekurangan Stok dan Pemborosan:

Beberapa menu makanan sering kali habis sebelum akhir hari operasional, menyebabkan ketidakpuasan pelanggan dan potensi kehilangan pendapatan. Sebaliknya, pemborosan bahan baku yang tidak terpakai juga menjadi masalah yang signifikan.

3. Kepuasan Pelanggan:

Ketidakmampuan restoran dalam memenuhi permintaan pelanggan dapat merusak reputasi restoran dalam jangka panjang. Pelanggan sering kali mengeluhkan bahwa menu favorit mereka tidak tersedia, yang tidak hanya menyebabkan kekecewaan tetapi juga membuat mereka berpikir dua kali untuk kembali.

4. Efisiensi Operasional:

Tanpa prediksi yang akurat, manajemen restoran kesulitan dalam merencanakan stok bahan baku dan tenaga kerja dengan efisien. Hal ini dapat menyebabkan ketidakefisienan operasional dan peningkatan biaya.

5. Pemilihan Metode Forecasting yang Tepat:

Penelitian ini bertujuan untuk menemukan metode time series yang paling sesuai untuk memprediksi penjualan di restoran "Nasi Goreng Bacot". Metode yang dipertimbangkan adalah Simple Moving Average (SMA), Weighted Moving Average (WMA), dan Single Exponential Smoothing (SES).

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Program Python menggunakan metode *Simple Moving Average*, *Weighted moving average* dan *Single Exponential Smoothing* (MAE, MSE) menggunakan Google Colab

Pada sub bab ini dijelaskan codingan pada google colab menghitung

4.1.1 Simple Moving Average

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Reading DataFrame from CSV file
df = pd.read_csv('/content/drive/MyDrive/dataset.csv')
```

Gambar 4. 1 Import Data Simple Moving Average

Pada gambar 4.1 dijelaskan bahwa pertama yaitu mengimpor dua perpustakaan penting. Baris pertama `import numpy as np` mengimpor perpustakaan NumPy, yang

menyediakan alat canggih untuk perhitungan numerik. Baris kedua `import pandas as pd` mengimpor perpustakaan Pandas, yang digunakan untuk analisis dan manipulasi data. Setelah itu, dibuat data berupa Seri Pandas yang berisi daftar nilai numerik, yang bisa mewakili angka penjualan dari waktu ke waktu. Kemudian, data tersebut diubah menjadi Pandas DataFrame bernama `df`, yang mengatur data ke dalam format tabel dengan kolom waktu (t) dan nilai (y).

```
# Calculating prediction using Simple Moving Average with window size of 3
df['pred'] = df['y'].rolling(window=3).mean().shift(1)
```

Gambar 4. 2 Prediksi *Simple Moving Average*

Pada gambar 4.2 digunakan untuk menghitung Rata-Rata Bergerak Sederhana (SMA), langkah pertama adalah menambahkan kolom baru 'pred' ke dalam DataFrame `df`, yang berisi nilai prediksi menggunakan SMA. Ini dilakukan dengan menerapkan jendela bergulir berukuran 3 ke kolom 'y', artinya setiap perhitungan mempertimbangkan 3 titik data terbaru untuk setiap langkah waktu. Kemudian, dihitung rata-rata nilai dalam jendela untuk setiap langkah waktu, memberikan prediksi SMA untuk langkah waktu berikutnya. Terakhir, nilai SMA yang dihitung digeser sebanyak 1 posisi ke depan untuk menyelaraskan prediksi dengan langkah waktu yang sesuai dalam data asli.

```
# Calculating error (e)
df['e'] = df['y'] - df['pred']

# Calculating square of error (e^2)
df['e^2'] = np.square(df['e'])
```

Gambar 4. 3 Coding mengukur error

Pada gambar 4.3 dijelaskan untuk menghitung kesalahan, langkah pertama adalah menambahkan kolom baru 'e' ke dalam DataFrame `df` yang berisi perhitungan kesalahan untuk setiap langkah waktu. Ini dilakukan dengan mengurangkan nilai prediksi ('pred') dari nilai sebenarnya ('y'), sehingga diperoleh perbedaan antara apa yang diprediksi oleh model dan apa yang sebenarnya terjadi. Selanjutnya, ditambahkan kolom 'e^2' yang berisi kesalahan kuadrat untuk setiap langkah waktu. Ini dilakukan dengan mengkuadratkan nilai kesalahan ('e'), sehingga kesalahan yang lebih besar mendapatkan bobot yang lebih signifikan.

```
# Calculating Mean Absolute Percentage Error (MAPE)
df['pe'] = df['e'].abs() / df['y'] * 100
df_filtered = df[df['pe'] != np.inf].dropna() # Filtering out null and infinite percentage errors
mape = df_filtered['pe'].mean()
```

Gambar 4.4 Mengukur menggunakan MAPE

Pada Gambar 4.4 dijelaskan bahwa DataFrame (df) yang berisi data prediksi dan nilai aktual diproses untuk menghitung kesalahan absolut persentase. Kesalahan absolut persentase dihitung dengan mengambil nilai absolut dari selisih antara nilai prediksi dan nilai aktual, kemudian dibagi dengan nilai aktual dan dikalikan dengan 100 untuk mengonversinya menjadi persentase. Setelah perhitungan kesalahan absolut persentase, ada langkah untuk memastikan bahwa hanya nilai yang valid dan dapat diandalkan yang digunakan untuk menghitung MAPE. Kode `df_filtered = df[df['pe'] != np.inf].dropna()` digunakan untuk menghilangkan nilai yang tidak valid dari DataFrame, seperti nilai infinit atau null. Hal ini dilakukan agar perhitungan MAPE hanya dilakukan dari data yang valid dan relevan. Setelah proses filtering, nilai rata-rata dari kesalahan absolut persentase yang valid dihitung menggunakan metode `.mean()`. Hasilnya disimpan dalam variabel `mape`, yang akan memberikan nilai MAPE dari prediksi yang telah dilakukan.

```
# Calculating Mean Absolute Error (MAE) and Mean Squared Error (MSE)
mae = df['e'].abs().mean()
mse = df['e^2'].mean()

# Displaying DataFrame
print(df)

print(f"MAE: {mae}")
print(f"MSE: {mse}")
print(f"MAPE: {mape}%")
```

Gambar 4. 5 Display Dataframe untuk *Simple Moving Average*

Pada gambar 4.5 dijelaskan untuk mengevaluasi dan memvisualisasikan hasil prediksi, pertama-tama ditampilkan konten DataFrame `df` yang kini mencakup data aktual ('y'), nilai prediksi ('pred'), kesalahan absolut ('|e|'), dan kesalahan kuadrat ('e^2'). Selanjutnya, dihitung Mean Absolute Error (MAE) dengan merata-ratakan nilai absolut dari kesalahan di kolom '|e|', yang mewakili besaran rata-rata kesalahan prediksi. Kemudian, dihitung Mean Squared Error (MSE) dengan merata-ratakan kesalahan kuadrat di kolom 'e^2', yang memberikan penekanan lebih besar pada kesalahan yang lebih besar dibandingkan MAE. Setelah perhitungan, hasil MAE dan MSE dicetak ke layar untuk memberikan informasi tentang besarnya kesalahan dalam prediksi. Nilai MAPE yang telah dihitung sebelumnya juga dicetak untuk memberikan informasi tentang kesalahan relatif dalam prediksi, diukur dalam bentuk persentase. Ini membantu dalam mengevaluasi kualitas prediksi secara keseluruhan dan memberikan pemahaman tentang seberapa dekat prediksi dengan nilai aktual.

```
# Plotting the actual data
plt.plot(df['t'], df['y'], label='Actual')

# Plotting the predicted data
plt.plot(df['t'], df['pred'], label='Predicted')

# Adding title and labels
plt.title('Simple Moving Average')
plt.xlabel('Time')
plt.ylabel('value')

# Adding legend
plt.legend()

# Displaying the plot
plt.show()
```

Gambar 4.6 Coding plot untuk *Simple Moving Average*

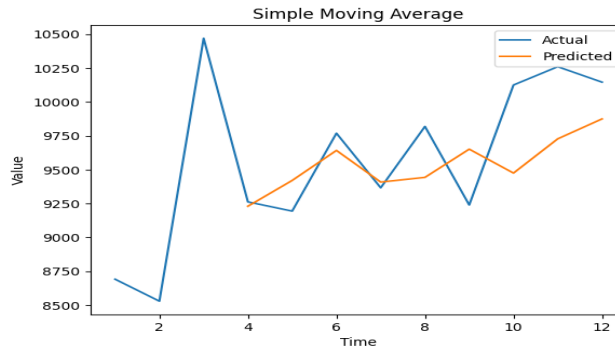
Pada gambar 4.6 dijelaskan dalam menggunakan matplotlib untuk membuat plot garis yang membandingkan data aktual ('y') dengan prediksi ('pred'). Plot pertama menampilkan data aktual dengan label "Sebenarnya", sedangkan plot kedua menampilkan nilai prediksi dengan label "Diprediksi". Plot ini memiliki judul "Simple Moving Average" dan sumbu x dilabeli sebagai "Waktu" sedangkan sumbu y dilabeli sebagai "Nilai". Dengan menambahkan legenda dapat membedakan antara data aktual dan prediksi. Hasil plot ini memperlihatkan visual perbandingan antara prediksi menggunakan Simple Moving Average (SMA) dengan data sebenarnya, memungkinkan untuk melihat seberapa baik model SMA dalam memprediksi nilai-nilai tersebut.

	t	y	pred	e	e ²	pe
0	1	8691	NaN	NaN	NaN	NaN
1	2	8529	NaN	NaN	NaN	NaN
2	3	10470	NaN	NaN	NaN	NaN
3	4	9262	9230.000000	32.000000	1024.000000	0.345498
4	5	9194	9420.333333	-226.333333	51226.777778	2.461750
5	6	9769	9642.000000	127.000000	16129.000000	1.300031
6	7	9366	9408.333333	-42.333333	1792.111111	0.451989
7	8	9819	9443.000000	376.000000	141376.000000	3.829311
8	9	9239	9651.333333	-412.333333	170018.777778	4.462965
9	10	10125	9474.666667	650.333333	422933.444444	6.423045
10	11	10260	9727.666667	532.333333	283378.777778	5.188434
11	12	10146	9874.666667	271.333333	73621.777778	2.674289

MAE : 296.666666666671
MSE : 129055.62962962993
MAPE : 3.0152568717588677%

Gambar 4.7 Hasil Perhitungan *Simple Moving Average*

Pada Gambar 4.7 dijelaskan bahwa hasil perhitungan *Simple Moving Average* berdasarkan data penjualan yang telah diperoleh maka MAE maupun MSE masih dianggap kecil. Nilai MAE yaitu 296.6, MSE yaitu 129055.6 dan MAPE yaitu 3.015%



Gambar 4. 8 Hasil Chart *Simple Moving Average*

Pada gambar 4.8 Secara umum, kode yang disajikan menggambarkan pendekatan fundamental dalam melakukan estimasi deret waktu dengan menggunakan metode *Simple Moving Average* (SMA). Kode tersebut melakukan perhitungan nilai prediksi berdasarkan rata-rata dari titik data terkini, mengkalkulasi kesalahan antara nilai prediksi dan nilai aktual, serta memvisualisasikan hasilnya dalam bentuk grafik.

4.1.2 Weighted moving average

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Reading DataFrame from CSV file
df = pd.read_csv('/content/drive/MyDrive/dataset.csv')

weights = np.array([0.2, 0.3, 0.5])
window_size = 3

wma = df['y'].rolling(window=window_size).apply(lambda x: np.average(x, weights=weights), raw=True)
wma = wma.dropna()

```

Gambar 4. 9 Coding Weighted Moving Average

Pada bagian ini menjelaskan bahwa memanggil library numPy dan Pandas. Library numPy menyediakan fungsi-fungsi untuk melakukan perhitungan numerik dan library Pandas menyediakan fungsi-fungsi untuk melakukan analisis dan manipulasi data. Kemudian memasukkan nilai-nilai pada variabel data yang dimiliki dari waktu ke waktu dan memberikan bobot pada array data. Selanjutnya, *Weighted moving average* atau WMA didapatkan dari mengembalikan nilai dari variabel data dengan memasukkan nilai `window_size` dan nilai bobot. Kemudian nilai `predicted_values` didapatkan dari hasil perhitungan variabel WMA yang dikembalikan.

```

# Create a DataFrame for the desired tabular format
df = pd.DataFrame([
    't': range(1, len(data)+1),
    'y': data,
    'pred': ['' for _ in range(window_size-1)] + list(predicted_values),
])

df['pred'] = df['pred'].shift(1)
df['y'] = pd.to_numeric(df['y'])
df['pred'] = pd.to_numeric(df['pred'])
df['|e|'] = (df['y'] - df['pred']).abs()
df['e^2'] = (df['y'] - df['pred'])**2
df['|e|'] = pd.to_numeric(df['|e|'])
df['e^2'] = pd.to_numeric(df['e^2'])
mae = df['|e|'].mean()
mse = df['e^2'].mean()

```

Gambar 4. 10 Coding Weighted moving average

Pada gambar 4.10 dijelaskan bahwa rumus untuk `df = pd.DataFrame({...})`: Blok kode ini membuat Pandas DataFrame bernama `df` untuk mengatur data dalam format tabel. DataFrame menyertakan kolom untuk `t` yaitu Langkah waktu (berkisar dari 1 hingga panjang data + 1), `y`: Poin data asli. Kemudian untuk `pred` adalah untuk Nilai prediksi menggunakan WMA (digeser 1 agar selaras dengan langkah waktu yang sesuai). nilai absolut `|e|` adalah Kesalahan absolut (perbedaan antara nilai aktual dan prediksi) sedangkan untuk `e^2` adalah Kesalahan kuadrat.

Pada “`df['y'] = pd.to_numeric`” `df['y']` berfungsi untuk mengonversi kolom 'y' (data asli) ke format numerik jika belum dilakukan. Kemudian “`df['pred'] = pd.to_numeric`” `df['pred']` berfungsi untuk mengonversi kolom 'pred' (nilai prediksi) ke format numerik. Mae didapatkan dari hasil dataframe rata-rata dari nilai `|e|` absolute dan MSE Mae didapatkan dari hasil dataframe rata-rata dari nilai `e^2` absolute.

```
# Calculating Mean Absolute Percentage Error (MAPE)
df['pe'] = df['|e|'] / df['y']
df_filtered = df[df['pe'] != np.inf].dropna() # Filtering out null and infinite percentage errors

mae = df['|e|'].mean()
mse = df['e^2'].mean()
mape = df_filtered['pe'].mean() * 100
```

Gambar 4. 11 Menghitung MAPE

Pada Gambar 4.11 dijelaskan bahwa Kode `df['pe'] = df['|e|'] / df['y']` digunakan untuk menghitung kesalahan persentase absolut untuk setiap observasi dalam DataFrame `df`. Ini dilakukan dengan membagi nilai absolut dari kesalahan (kolom '|e|') oleh nilai aktual (kolom 'y').Setelah perhitungan, kita harus memastikan bahwa hanya nilai yang valid dan dapat diandalkan yang digunakan untuk menghitung MAPE. Kode `df_filtered = df[df['pe'] != np.inf].dropna()` digunakan untuk memfilter DataFrame dan menghapus baris yang memiliki nilai kesalahan persentase yang tidak valid (infinite atau null). Setelah proses filtering, nilai rata-rata dari kesalahan persentase yang valid dihitung menggunakan metode `.mean()`.

Hasilnya kemudian dikalikan dengan 100 untuk mengonversinya menjadi persentase, karena MAPE diukur dalam bentuk persentase.

```
print(df)
print("\nMean Squared Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("Mean Absolute Percentage Error (MAPE):", mape, "%")
```

Gambar 4. 12 Print MSE , MAE, MAPE pada weighted moving average

Pada gambar 4.12 dapat digunakan untuk Mencetak DataFrame yang telah dimodifikasi beserta nilai MSE, MAE dan MAPE.

```
plt.figure(figsize=(10, 6))
plt.plot(df['t'], df['y'], label='Data Aktual (y)')
plt.plot(df['t'], df['pred'], label='Prediksi (pred)', linestyle='--')

plt.title('Weighted Moving Average')
plt.xlabel('Waktu (t)')
plt.ylabel('Nilai')
plt.legend()

plt.show()
```

Gambar 4.13 Coding plot pada *Weighted Moving Average*

Pada gambar 4.13 digunakan untuk membuat plot untuk visualisasi data actual dan prediksi. Kemudian menggunakan 'plt.plot' untuk menggambar garis data actual dan prediksi serta dapat menambahkan judul, label sumbu dan legenda untuk plot tersebut.

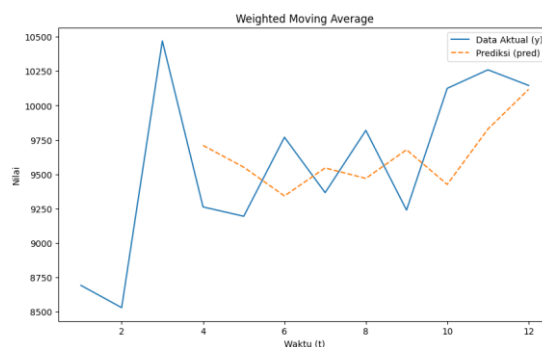
t	y	pred	e	e ²	pe	
0	1	8691	NaN	NaN	NaN	
1	2	8529	NaN	NaN	NaN	
2	3	10470	NaN	NaN	NaN	
3	4	9262	9531.9	269.9	72846.01	0.029141
4	5	9194	9477.8	283.8	80542.44	0.030868
5	6	9769	9469.6	299.4	89640.36	0.030648
6	7	9366	9495.1	129.1	16666.81	0.013784
7	8	9819	9452.5	366.5	134322.25	0.037326
8	9	9239	9673.1	434.1	188442.81	0.046986
9	10	10125	9438.4	686.6	471419.56	0.067812
10	11	10260	9798.0	462.0	213444.00	0.045029
11	12	10146	10015.3	130.7	17082.49	0.012882

Mean Squared Error (MSE): 14.2711.8588888889
Mean Absolute Error (MAE): 340.23333333333335
Mean Absolute Percentage Error (MAPE): 3.4941678457985765 %

Gambar 4. 14 Hasil perhitungan menggunakan *Weighted Moving Average*

Tabel dalam gambar 4.14 menunjukkan hasil perhitungan *Weighted Moving Average* (WMA) untuk suatu seri data. Kolom 't' tampaknya mewakili waktu atau indeks urutan, sementara 'y' tampaknya mewakili nilai aktual, dan 'pred' adalah nilai yang diprediksi, kemungkinan dari model WMA. Kolom '|e|' mewakili kesalahan absolut antara nilai aktual dan prediksi, dan kolom 'e²' mewakili kesalahan kuadrat.

Secara keseluruhan, tabel ini memberikan wawasan tentang bagaimana metode *Weighted Moving Average* dapat digunakan untuk membuat prediksi berdasarkan data historis dan seberapa baik prediksi tersebut dibandingkan dengan nilai aktual.



Gambar 4. 15 Grafik Hasil *Weighted Moving Average*

Pada gambar 4.15 dapat kita simpulkan bahwa grafik tersebut menunjukkan perbandingan antara data aktual dan prediksi menggunakan metode Weighted Moving Average. Ada dua garis pada grafik tersebut. Garis biru padat mewakili data aktual ('y'), dan garis oranye putus-putus mewakili data yang diprediksi ('pred'). Sumbu x, yang diberi label 'Waktu (t)', kemungkinan besar mewakili periode waktu dari 1 hingga 12. Sumbu y, yang diberi label 'Nilai', berkisar dari 8500 hingga 10500. Dari grafik tersebut, terlihat bahwa garis prediksi mengikuti tren data aktual tetapi dengan fluktuasi yang lebih sedikit, menunjukkan bahwa metode Weighted Moving Average telah digunakan untuk merata-ratakan data dan membuat prediksi.

4.1.3 Single Exponential Smoothing

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Reading DataFrame from CSV file
df = pd.read_csv('/content/drive/MyDrive/dataset.csv')
```

Gambar 4. 16 Import Dataset Single Exponential Smoothing

Kode yang diberikan pada gambar 4.16 merupakan bagian dari skrip Python yang digunakan untuk analisis data. Skrip ini dimulai dengan mengimpor library yang diperlukan, yaitu NumPy untuk operasi matematika dan array, Pandas untuk manipulasi dan analisis data, dan modul pyplot dari Matplotlib untuk visualisasi data. Setelah library diimpor, skrip membaca file CSV dari lokasi yang ditentukan dan menyimpan datanya dalam DataFrame Pandas. DataFrame adalah struktur data dua dimensi, seperti tabel, yang memungkinkan manipulasi dan analisis data yang efisien. Dengan demikian, dataset sekarang siap untuk dianalisis dan diproses lebih lanjut menggunakan berbagai metode dan teknik yang disediakan oleh library-library tersebut.

```
y = df['y'].values
pred = [y[0]] # Initial prediction

alpha = 0.7 # Alpha value

# Perform Single Exponential Smoothing
for t in range(1, len(y)):
    pred.append(alpha * y[t-1] + (1 - alpha) * pred[t-1])
```

Gambar 4. 17 Penggunaan Alpha dalam Single Exponential Smoothing

Kode pada gambar 4.17 tersebut melakukan operasi *Single Exponential Smoothing* pada data yang ada dalam kolom 'y' dari DataFrame. Nilai aktual diambil dari kolom 'y' dan disimpan dalam variabel y. Prediksi awal diatur sama dengan nilai pertama dari y. Nilai alpha, yang merupakan faktor pemulusan, diatur ke 0.7. Kemudian, untuk setiap titik data berikutnya

dalam y , prediksi baru dihitung sebagai kombinasi linier dari nilai aktual sebelumnya dan prediksi sebelumnya, dengan bobot α dan $1-\alpha$ masing-masing. Prediksi ini kemudian ditambahkan ke daftar prediksi. Dengan demikian, kode ini menghasilkan serangkaian prediksi yang telah dismoothering dengan metode *Single Exponential Smoothing*.

```
# Calculate absolute errors and squared errors
abs_errors = np.abs(y - np.array(pred))
squared_errors = abs_errors**2
```

Gambar 4. 18 Penggunaan MAE dan MSE pada *Single Exponential Smoothing*

Pada gambar 4.18 dijelaskan bahwa code tersebut menghitung kesalahan absolut dan kesalahan kuadrat dari prediksi yang telah dibuat. Kesalahan absolut dihitung dengan mengambil nilai absolut dari perbedaan antara nilai aktual y dan prediksi. Prediksi dikonversi ke array numpy sebelum pengurangan untuk memastikan operasi berjalan dengan baik. Kesalahan kuadrat kemudian dihitung dengan memangkatkan dua kesalahan absolut. Dengan demikian, kode ini menghasilkan dua array: satu berisi kesalahan absolut dan yang lain berisi kesalahan kuadrat, yang keduanya dapat digunakan untuk evaluasi lebih lanjut dari model prediksi. Kesalahan absolut memberikan ukuran kesalahan rata-rata, sementara kesalahan kuadrat memberikan penekanan lebih pada kesalahan yang lebih besar.

```
print('results table')
print(' t | y | pred | |e| | e^2 | e% |')
print('-----|-----|-----|-----|-----|-----|')
for t in range(len(y)):
    print(' %2d | %2d | %2d | %2d | %2d | %2d |' % (t+2, y[t], (pred[t]*.3f), (abs_errors[t]*.3f), (squared_errors[t]*.3f), (percent_errors[t]*.3f) ))
mse = np.mean(abs_errors)
mse = np.mean(squared_errors)
# calculate mape
mape = np.mean(percent_errors[np.nonzero(percent_errors)])
print('MAE: (mse:.3f)')
print('MSE: (mse:.3f)')
print('MAPE: (mape:.3f)')
```

Gambar 4. 19 Print hasil perhitungan

Pada gambar 4.19 Kode ini bertujuan untuk mencetak tabel hasil perhitungan dan statistik evaluasi dari data prediksi. Kode `'print("| t | y | pred | |e| | e^2 | e% |")'` digunakan untuk mencetak header tabel dengan kolom yang sesuai dengan data yang akan ditampilkan. Kode `'print("-----|-----|-----|-----|-----|-----|")'` digunakan untuk mencetak garis pembatas antara header dan isi tabel. Perulangan `'for'` digunakan untuk mencetak baris-baris tabel dengan menggunakan data yang telah dihitung sebelumnya, seperti nilai aktual (`'y'`), nilai prediksi (`'pred'`), kesalahan absolut (`'abs_errors'`), kesalahan kuadrat (`'squared_errors'`), dan kesalahan persentase (`'percent_errors'`). Setiap baris tabel mencakup nomor periode (`'t+2'`), nilai aktual, nilai prediksi, kesalahan absolut, kesalahan kuadrat, dan kesalahan persentase untuk setiap periode. Setelah mencetak tabel, kode tersebut juga

melakukan perhitungan statistik evaluasi, yaitu Mean Absolute Error (MAE), Mean Squared Error (MSE), dan Mean Absolute Percentage Error (MAPE).

- MAE dihitung sebagai rata-rata dari kesalahan absolut (`abs_errors`).
- MSE dihitung sebagai rata-rata dari kesalahan kuadrat (`squared_errors`).
- MAPE dihitung sebagai rata-rata dari kesalahan persentase (`percent_errors`) hanya dari nilai yang tidak nol.

Intinya, kode tersebut membantu dalam menganalisis, mengevaluasi, dan memvisualisasikan hasil prediksi, serta memberikan informasi yang berguna tentang seberapa baik model peramalan tersebut dalam memperkirakan nilai aktual.

```
# Plot the original data and predicted values
plt.plot(range(1, len(y) + 1), y, label='Original Data')
plt.plot(range(2, len(y) + 1), pred[1:], label='Predicted Values', linestyle='dashed', marker='o')
plt.xlabel('Time')
plt.ylabel('Values')
plt.title('Single Exponential Smoothing Prediction')
plt.legend()
plt.show()
```

Gambar 4. 20 Plot data dan prediksi perhitungan

Pada gambar 4.20 menjelaskan bahwa kode tersebut menciptakan plot yang menampilkan data asli dan nilai yang diprediksi. Plot pertama menunjukkan data asli, yang diambil dari array y, dan plot kedua menunjukkan nilai yang diprediksi, yang diambil dari array pred. Nilai-nilai ini diplot terhadap rentang waktu yang sama. Label ‘Original Data’ dan ‘Predicted Values’ ditambahkan ke plot untuk membedakan antara data asli dan prediksi. Plot prediksi dibuat dengan garis putus-putus dan marker lingkaran untuk membedakannya dari plot data asli. Label sumbu x dan y ditetapkan menjadi ‘Time’ dan ‘Values’ masing-masing, dan judul plot ditetapkan menjadi ‘Single Exponential Smoothing Prediction’. Akhirnya, legenda ditambahkan ke plot dan plot ditampilkan. Dengan demikian, kode ini memberikan visualisasi grafis yang jelas tentang bagaimana nilai yang diprediksi oleh model Single Exponential Smoothing dibandingkan dengan data asli.

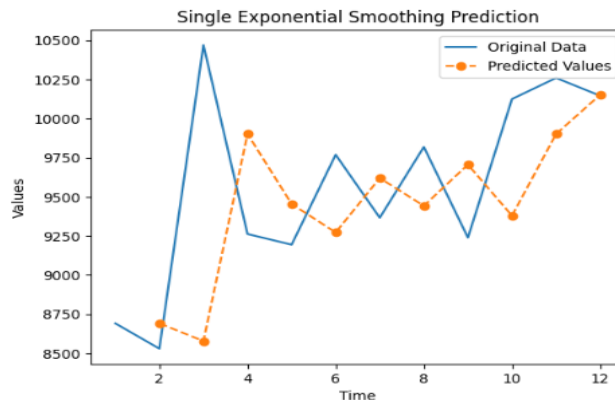
t	y	pred	e	e ²	e%
2	8691	8691.000	0.000	0.000	0.000
3	8529	8691.000	162.000	26244.000	1.899
4	10470	8577.600	1892.400	3581177.760	18.074
5	9262	9902.280	640.280	409958.478	6.913
6	9194	9454.084	260.084	67643.687	2.829
7	9769	9272.025	496.975	246983.952	5.087
8	9366	9619.908	253.908	64469.049	2.711
9	9819	9442.172	376.828	141999.140	3.838
10	9239	9705.952	466.952	218043.872	5.054
11	10125	9379.086	745.914	556388.435	7.367
12	10260	9901.226	358.774	128719.033	3.497
13	10146	10152.368	6.368	40.548	0.063

MAE : 471.707
MSE : 453472.329
MAPE : 5.212%

Gambar 4. 21 Hasil Perhitungan *Single Exponential Smoothing*

Pada gambar 4.21 dijelaskan bahwa table tersebut menunjukkan hasil dari metode *Single Exponential Smoothing* yang diterapkan pada suatu seri data. Kolom ‘t’ mewakili waktu atau indeks urutan, ‘y’ mewakili nilai aktual, dan ‘pred’ mewakili nilai yang diprediksi oleh

model. Kolom 'e' mewakili kesalahan, yaitu perbedaan antara nilai aktual dan prediksi, dan kolom 'e²' mewakili kesalahan kuadrat, yang merupakan kuadrat dari kesalahan tersebut.



Gambar 4. 22 Grafik hasil Single Exponential Smoothing

Pada gambar 4.22 menunjukkan hasil grafik dari metode *Single Exponential Smoothing* yang diterapkan pada suatu seri data. Ada dua garis pada grafik tersebut. Garis biru padat mewakili data asli, dan garis oranye putus-putus dengan marker lingkaran mewakili data yang diprediksi oleh model. Sumbu x, yang diberi label 'Time', mewakili periode waktu dari 1 hingga 12. Sumbu y, yang diberi label 'Values', berkisar dari sekitar 8500 hingga 10500. Dari grafik tersebut, terlihat bahwa garis prediksi mengikuti tren data asli tetapi dengan fluktuasi yang lebih sedikit, menunjukkan bahwa metode Single Exponential Smoothing telah digunakan untuk merata-ratakan data dan membuat prediksi. Ini sangat berguna dalam menganalisis tren dan membuat ramalan dalam berbagai bidang seperti keuangan atau manajemen persediaan. Grafik ini menunjukkan bagaimana metode tersebut dapat digunakan untuk membandingkan nilai aktual dengan nilai yang diprediksi.

4.1.4 Implementasi Program Phyton Menggunakan Flask

```

1 from flask import Flask, render_template, request, redirect, url_for
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import io
6 import base64
7 import os
8
9 # Set matplotlib to use the 'Agg' backend
10 plt.switch_backend('Agg')
11
12 app = Flask(__name__)
13 UPLOAD_FOLDER = 'dataset'
14 app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
15 ALLOWED_EXTENSIONS = {'csv'}
16

```

Gambar 4. 23Import Library

Pada Gambar 4.23 Coding awal digunakan untuk mengimpor berbagai library yang diperlukan untuk aplikasi Flask, termasuk Flask untuk pengembangan web, Pandas untuk manipulasi data, NumPy untuk operasi numerik, Matplotlib untuk visualisasi data, dan

beberapa modul untuk menangani file dan encoding gambar. Matplotlib Backend untuk Mengatur Matplotlib untuk menggunakan backend 'Agg', yang memungkinkan pembuatan grafik tanpa memerlukan antarmuka pengguna grafis. Konfigurasi Flask untuk menginisialisasi aplikasi Flask dan mengatur folder untuk mengunggah file serta ekstensi file yang diizinkan.

```
# Helper function to check allowed file extensions
def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

# Helper function to plot and return image
def create_plot(y, pred, title):
    plt.figure(figsize=(10, 6))
    plt.plot(range(1, len(y) + 1), y, label='Actual Data')
    plt.plot(range(1, len(pred) + 1), pred, label='Predicted Values', linestyle='dashed', marker='o')
    plt.xlabel('Time')
    plt.ylabel('Values')
    plt.title(title)
    plt.legend()
    img = io.BytesIO()
    plt.savefig(img, format='png')
    img.seek(0)
    plot_url = base64.b64encode(img.getvalue()).decode()
    plt.close()
    return plot_url
```

Gambar 4. 24 Coding menggunakan fungsi Pembantu

Pada Gambar 4.24 difungsikan untuk memeriksa apakah file yang diunggah memiliki ekstensi yang diizinkan (dalam hal ini, hanya file CSV yang diizinkan) serta membuat plot menggunakan Matplotlib dan mengembalikan gambar dalam format base64 yang dapat ditampilkan di halaman web. Fungsi ini menerima data aktual (y), data prediksi (pred), dan judul plot (title).

```
@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)
        file = request.files['file']
        if file.filename == '':
            return redirect(request.url)
        if file and allowed_file(file.filename):
            filename = 'dataset.csv'
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
            return redirect(url_for('home'))
    return render_template('home.html')
```

Gambar 4. 25 Route Home

Pada gambar 4.25 digunakan untuk menangani permintaan GET dan POST untuk halaman utama. Jika metode permintaan adalah POST, fungsi ini memeriksa apakah file diunggah dan apakah file tersebut memiliki ekstensi yang diizinkan. Jika ya, file disimpan di folder yang ditentukan dan halaman dimuat ulang. Jika tidak, halaman utama ditampilkan.

```
@app.route('/single_exponential')
def single_exponential():
    df = pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'], 'dataset.csv'))
    y = df['y'].values
    pred = y[0]
    alpha = 0.7

    for t in range(1, len(y)):
        pred = alpha * y[t-1] + (1 - alpha) * pred[t-1]

    abs_errors = np.abs(y - np.array(pred))
    squared_errors = abs_errors**2
    percent_errors = np.abs(y - np.array(pred)) / y * 100

    mae = np.mean(abs_errors)
    mse = np.mean(squared_errors)
    mape = np.mean(percent_errors[np.nonzero(percent_errors)])

    df_result = pd.DataFrame({
        't': range(1, len(y) + 1),
        'y': y,
        'pred': pred,
        'abs_errors': abs_errors,
        'squared_errors': squared_errors,
        'percent_errors': percent_errors
    })

    plot_url = create_plot(y, pred, 'Single Exponential Smoothing Prediction')
    return render_template('single_exponential.html', plot_url=plot_url, tables=df_result.to_html(classes='table table-striped', index=False), mse=mse, mae=mae, mape=mape)
```

Gambar 4. 26 Route Single Exponential Smoothing

Berdasarkan pada gambar 4.26 dapat digunakan untuk menghitung prediksi menggunakan metode Single Exponential Smoothing. Membaca data dari file CSV, menghitung prediksi, kesalahan absolut, kesalahan kuadrat, dan kesalahan persentase. Menghitung MAE, MSE, dan MAPE. Membuat plot dan menampilkan hasil dalam template HTML.

```

@app.route('/simple_moving')
def simple_moving():
    df = pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'], 'dataset.csv'))
    df['pred'] = df['y'].rolling(window=3).mean().shift(1)

    df['e'] = df['y'] - df['pred']
    df['e2'] = np.square(df['e'])
    df['pe'] = df['e'].abs() / df['y'] * 100
    df_filtered = df[df['pe'] != np.inf].dropna()

    mae = df['e'].abs().mean()
    mse = df['e2'].mean()
    mape = df_filtered['pe'].mean()

    plot_url = create_plot(df['y'], df['pred'], 'Simple Moving Average Prediction')
    return render_template('simple_moving.html', plot_url=plot_url, tables=df.to_html(classes='table table-striped', index=False), mae=mae, mse=mse, mape=mape)

```

Gambar 4. 27 Route Simple Moving Average

Pada Gambar 4.27 digunakan untuk menghitung prediksi menggunakan metode Simple Moving Average. Membaca data dari file CSV, menghitung prediksi dengan rolling window, menghitung kesalahan absolut, kesalahan kuadrat, dan kesalahan persentase. Menghitung MAE, MSE, dan MAPE. Membuat plot dan menampilkan hasil dalam template HTML.

```

@app.route('/weighted_moving')
def weighted_moving():
    df = pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'], 'dataset.csv'))
    weights = np.array([0.2, 0.3, 0.5])
    window_size = 3

    sma = df['y'].rolling(window=window_size).apply(lambda x: np.average(x, weights=weights), raw=True)
    sma = sma.dropna()

    df['pred'] = ['' for _ in range(window_size-1)] + list(sma.values)
    df['pred'] = df['pred'].shift(1)
    df['pred'] = pd.to_numeric(df['pred'], errors='coerce')

    df['e'] = (df['y'] - df['pred']).abs()
    df['e2'] = (df['y'] - df['pred'])**2
    df['pe'] = df['e'] / df['y']
    df_filtered = df[df['pe'] != np.inf].dropna()

    mae = df['e'].mean()
    mse = df['e2'].mean()
    mape = df_filtered['pe'].mean() * 100

    plot_url = create_plot(df['y'], df['pred'], 'Weighted Moving Average Prediction')
    return render_template('weighted_moving.html', plot_url=plot_url, tables=df.to_html(classes='table table-striped', index=False), mae=mae, mse=mse, mape=mape)

```

Gambar 4. 28 Route Weighted Moving Average

Pada gambar 4.28 digunakan untuk menghitung prediksi menggunakan metode Weighted Moving Average. Membaca data dari file CSV, menghitung prediksi dengan rolling window dan bobot tertentu, menghitung kesalahan absolut, kesalahan kuadrat, dan kesalahan persentase. Menghitung MAE, MSE, dan MAPE. Membuat plot dan menampilkan hasil dalam template HTML

```

@app.route('/summary')
def summary():
    # Read dataset
    df = pd.read_csv(os.path.join(app.config['UPLOAD_FOLDER'], 'dataset.csv'))

    # Single Exponential Smoothing
    y = df['y'].values
    pred_ses = [y[0]]
    alpha = 0.7
    for t in range(1, len(y)):
        pred_ses.append(alpha * y[t-1] + (1 - alpha) * pred_ses[t-1])
    mae_ses = np.mean(np.abs(y - np.array(pred_ses)))
    mse_ses = np.mean((y - np.array(pred_ses))**2)
    mape_ses = np.mean(np.abs((y - np.array(pred_ses)) / y) * 100)

    # Simple Moving Average
    df['pred_sma'] = df['y'].rolling(window=3).mean().shift(1)
    mae_sma = df['y'].sub(df['pred_sma']).abs().mean()
    mse_sma = df['y'].sub(df['pred_sma']).pow(2).mean()
    mape_sma = df['y'].sub(df['pred_sma']).abs().div(df['y']).mul(100).mean()

```

Gambar 4. 29 Route Summary SES dan SMA

Route ini mengumpulkan dan menyajikan hasil peramalan dari ketiga metode dalam satu tempat. Ini memudahkan pengguna untuk melihat dan membandingkan hasil dari berbagai metode tanpa harus mengunjungi beberapa halaman atau melakukan perhitungan manual

```
# Weighted Moving Average
weights = np.array([0.2, 0.3, 0.5])
wma = df['y'].rolling(window=3).apply(lambda x: np.average(x, weights=weights), raw=True).shift(1)
mse_wma = df['y'].sub(wma).abs().mean()
mse_sma = df['y'].sub(sma).pow(2).mean()
mape_wma = df['y'].sub(wma).abs().div(df['y']).mul(100).mean()

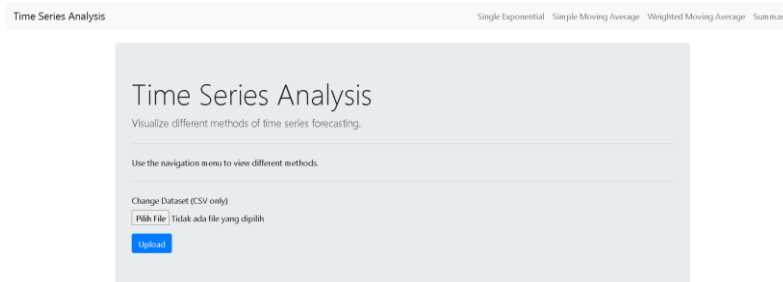
return render_template("summary.html", mae_ses=mae_ses, mse_ses=mse_ses, mape_ses=mape_ses,
                        mae_sma=mae_sma, mse_sma=mse_sma, mape_sma=mape_sma,
                        mae_wma=mae_wma, mse_wma=mse_wma, mape_wma=mape_wma)

if __name__ == '__main__':
    app.run(debug=False)
```

Gambar 4. 30 Route Summary WMA

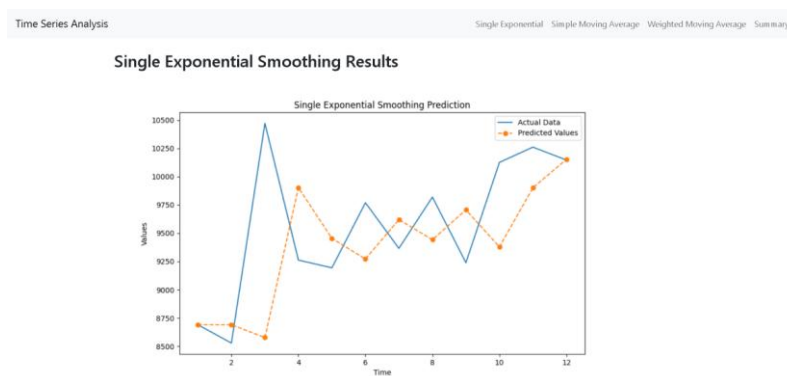
Route ini mengumpulkan dan menyajikan hasil peramalan dari ketiga metode dalam satu tempat. Ini memudahkan pengguna untuk melihat dan membandingkan hasil dari berbagai metode tanpa harus mengunjungi beberapa halaman atau melakukan perhitungan manual.

4.1.5 Hasil Implementasi Program Python



Gambar 4. 31 Dashboard Time Series Analysis

Pada gambar 4.31 diatas dapat dilihat bahwa pada menu dashboard awal membuka yaitu langsung terdapat menu *change dataset* dan *upload* file csv untuk memulai perhitungan forecasting menggunakan 3 metode yaitu *single exponential smoothing*, *simple moving average*, *weighted moving average* dan *summary* adalah untuk hasil akhir dari ketiga metode tersebut sehingga bisa membandingkan antara 3 metode tersebut mana yang lebih baik digunakan untuk perhitungan data tersebut.



Gambar 4. 32 Hasil Grafik Metode Single Exponential Smoothing

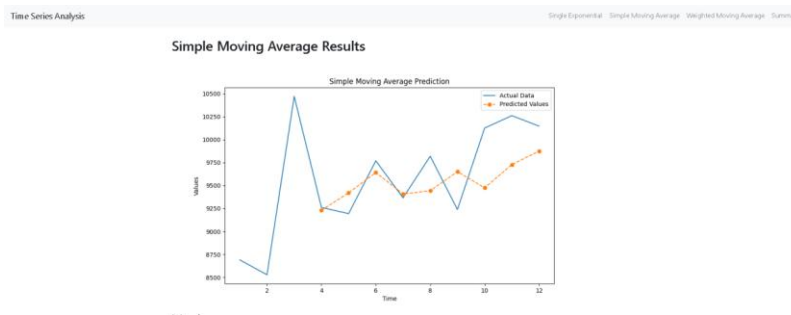
Pada metode perhitungan pertama yaitu menggunakan *single exponential smoothing* seperti pada gambar 4.32 diatas. Pada hasil perhitungan tersebut dapat menghasilkan grafik dari data actual dan data prediksi. Selanjutnya dapat dilihat tabel perhitungan yang ada di bawah grafik tersebut.

Metrics						
t	y	pred	e	e ²	pe	
1	8991	8991.000000	0.000000	0.000000e+00	0.000000	
2	8529	8991.000000	162.000000	2.624400e+04	1.899402	
3	10470	8577.800000	1892.400000	3.581170e+06	1.8074499	
4	9262	9992.200000	640.200000	4.099585e+05	6.912978	
5	9194	9454.084000	260.084000	6.784369e+04	2.828845	
6	9789	9272.025200	496.974800	2.469840e+05	3.087284	
7	9386	9819.907560	253.907560	6.448905e+04	2.719050	
8	9819	9442.172268	376.827732	1.419099e+05	3.837740	
9	9239	9705.351680	466.351680	2.180439e+05	3.054137	
10	10125	9379.085504	746.914496	5.583884e+05	7.367057	
11	10260	9991.225851	358.774149	1.287190e+05	3.498226	
12	10146	10152.387695	6.387695	4.054754e+01	0.092761	

MAE: 471.7068593678999
 MSE: 403472.3294823438
 MAPE: 5.212041573349176%

Gambar 4. 33 Hasil Perhitungan Single Exponential Smoothing

Pada gambar 4.33 diatas dapat dilihat bahwa pada tabel diperhitungkan menggunakan rumus *Single Exponential Smoothing* dan juga terdapat perhitungan error yaitu *Mean Absolute Error*, *Mean Squared Error* dan juga *Mean Absolute Percentage Error*.



Gambar 4. 34 Hasil Grafik Simple Moving Average

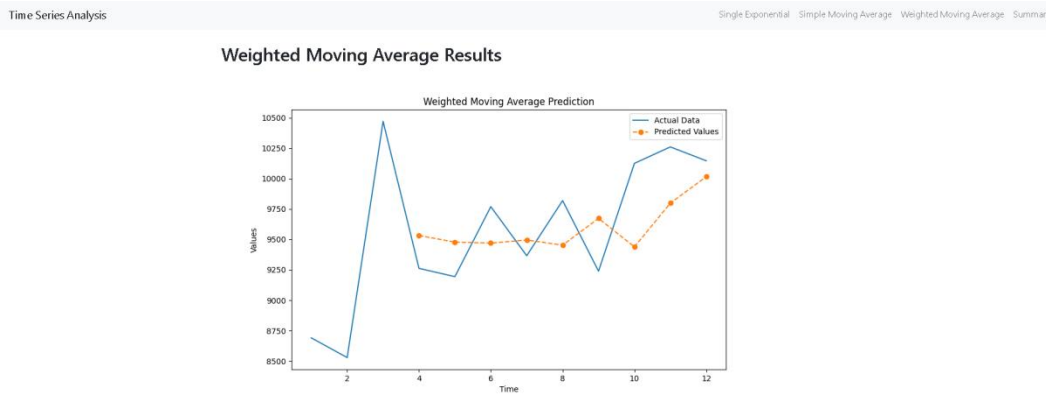
Pada metode perhitungan kedua yaitu menggunakan *Simple Moving Average* seperti pada gambar 4.34 diatas. Pada hasil perhitungan tersebut dapat menghasilkan grafik dari data actual dan data prediksi. Selanjutnya dapat dilihat tabel perhitungan yang ada di bawah grafik tersebut.

Metrics						
t	y	pred	e	e ²	pe	
1	8991	9491	-499	249001	-499	
2	8529	9491	-962	925444	-962	
3	10470	9491	979	958441	979	
4	9262	9230.000000	32.000000	1024.000000	3.345498	
5	9194	9420.333333	-226.333333	51226.777778	2.481750	
6	9789	9642.000000	127.000000	16129.000000	1.306031	
7	9386	9408.333333	-42.333333	1792.111111	0.451989	
8	9819	9443.000000	376.000000	141376.000000	3.825911	
9	9239	9651.333333	-412.333333	170038.777778	4.462985	
10	10125	9474.666667	650.333333	422933.444444	6.423940	
11	10260	9727.666667	532.333333	283376.777778	5.188454	
12	10146	9874.666667	271.333333	73621.777778	2.674289	

MAE: 298.6888888888887
 MSE: 129055.6296296293
 MAPE: 3.015254877588877%

Gambar 4. 15 Hasil Perhitungan Metode Simple Moving Average

Pada gambar 4.35 diatas dapat dilihat bahwa pada tabel diperhitungkan menggunakan rumus *Simple Moving Average* dan juga terdapat perhitungan error yaitu *Mean Absolute Error*, *Mean Squared Error* dan juga *Mean Absolute Percentage Error*.



Gambar 4. 36 Hasil Grafik Weighted Moving Average

Pada metode perhitungan ketiga yaitu menggunakan *Weighted Moving Average* seperti pada gambar 4.36 diatas. Pada hasil perhitungan tersebut dapat menghasilkan grafik dari data actual dan data prediksi. Selanjutnya dapat dilihat tabel perhitungan yang ada di bawah grafik tersebut.

t	y	pred	e	e ²	pe
1	8691	NaN	NaN	NaN	NaN
2	8529	NaN	NaN	NaN	NaN
3	10470	NaN	NaN	NaN	NaN
4	9262	9531.9	269.9	72866.01	0.029141
5	9194	9477.8	283.8	80542.44	0.030868
6	9769	9469.6	299.4	89640.36	0.030648
7	9366	9495.1	129.1	16666.81	0.013784
8	9819	9452.5	366.5	134322.25	0.037326
9	9239	9673.1	434.1	188442.81	0.046986
10	10125	9438.4	686.6	471419.56	0.067812
11	10260	9798.0	462.0	213444.00	0.045029
12	10146	10015.3	130.7	17082.49	0.012882

Metrics
 MAE: 340.2333333333335
 MSE: 142711.858888889
 MAPE: 3.4941678457985765%

Gambar 4. 37 Hasil Perhitungan Weighted Moving Average

Pada gambar 4.37 diatas dapat dilihat bahwa pada tabel diperhitungkan menggunakan rumus *Weighted Moving Average* dan juga terdapat perhitungan error yaitu *Mean Absolute Error*, *Mean Squared Error* dan juga *Mean Absolute Percentage Error*.

Method	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Mean Absolute Percentage Error (MAPE)
Single Exponential Smoothing	471.7068593678999	453472.3294823438	4.777704775570078%
Simple Moving Average	296.6666666666671	129055.62962962993	3.0152568717588673%
Weighted Moving Average	340.2333333333335	142711.858888889	3.4941678457985765%

Gambar 4. 38 Hasil Perbandingan dari Ketiga Metode Tersebut

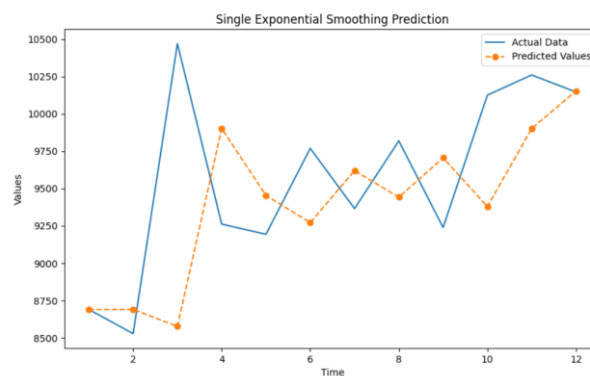
Pada gambar 4.38 yaitu perhitungan terakhir yaitu dapat menghitung apa saja hasil dari ketiga perhitungan diatas dan menunjukkan semua hasil error dari ketiga metode tersebut. Pada hasil akhir tersebut kita dapat melihat bahwa semakin kecil hasil dari perhitungan error maka semakin baik pula metode tersebut. Sehingga dihasilkan *Simple Moving Average* adalah metode yang paling baik untuk perhitungan penjualan dari data csv penjualan nasi goreng “Bacot”.

4.1.6 Testing Metode Menggunakan Simple Moving Average, Weighted Moving Average dan Exponential Smoothing

1. Testing Menggunakan Metode SES, SMA, dan WMA

Percobaan 1 :

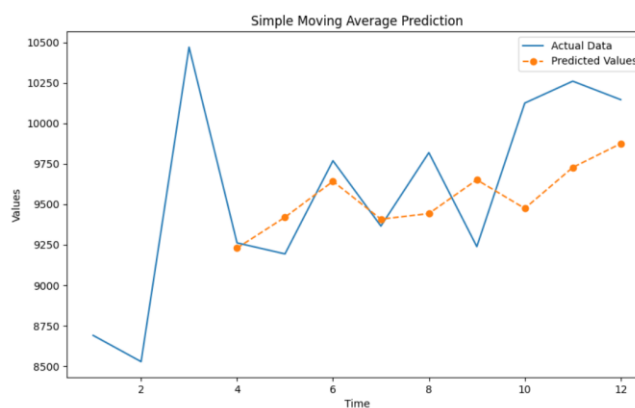
Single Exponential Smoothing Results



Gambar 4. 39 Grafik SES percobaan 1

Pada Gambar 4.39 yaitu dapat terlihat bahwa hasil percobaan 1 dengan menggunakan metode Single Exponential Smoothing dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

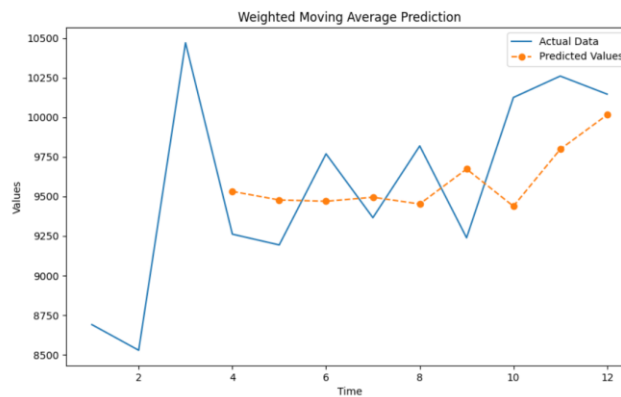
Simple Moving Average Results



Gambar 4. 40 Grafik SMA percobaan 1

Pada Gambar 4.40 yaitu dapat terlihat bahwa hasil percobaan 1 dengan menggunakan metode Simple Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Weighted Moving Average Results



Gambar 4. 41 Grafik WMA percobaan 1

Pada Gambar 4.41 yaitu dapat terlihat bahwa hasil percobaan 1 dengan menggunakan metode Weighted Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Summary of Forecasting Methods

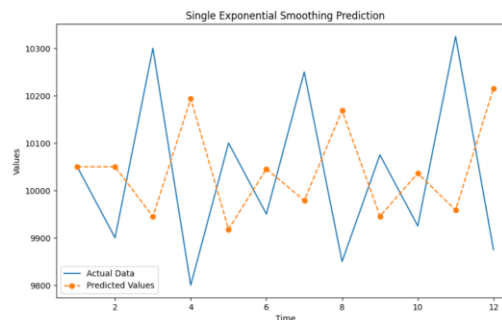
Method	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Mean Absolute Percentage Error (MAPE)
Single Exponential Smoothing	471.7068593678999	453472.3294823438	4.777704775570078%
Simple Moving Average	296.6666666666671	120055.62962062993	3.0152568717588673%
Weighted Moving Average	340.23333333333335	142711.8588888889	3.4941678457985765%

Gambar 4. 42 Hasil Perbandingan Percobaan 1

Pada Gambar 4.42 digunakan untuk melihat hasil dari ketiga metode tersebut dan membandingkan metode mana yang terbaik untuk forecasting penjualan. Hasil yang diperoleh yaitu metode Simple Moving Average adalah metode terbaik dalam perhitungan error tersebut dengan hasil MAPE terkecil.

Percobaan 2 :

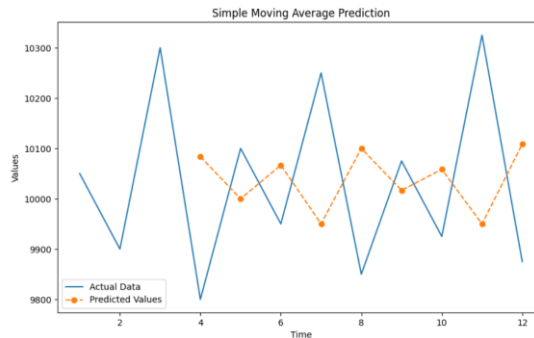
Single Exponential Smoothing Results



Gambar 4. 43 Grafik SES percobaan 2

Pada Gambar 4.43 yaitu dapat terlihat bahwa hasil percobaan 2 dengan menggunakan metode Single Exponential Smoothing dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

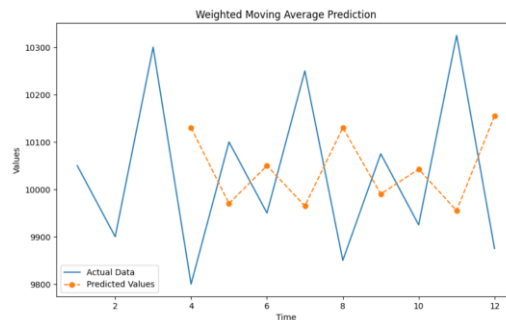
Simple Moving Average Results



Gambar 4. 44 Grafik SMA percobaan 2

Pada Gambar 4.44 yaitu dapat terlihat bahwa hasil percobaan 2 dengan menggunakan metode Simple Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Weighted Moving Average Results



Gambar 4. 45 Grafik WMA percobaan 2

Pada Gambar 4.45 yaitu dapat terlihat bahwa hasil percobaan 2 dengan menggunakan metode Weighted Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Summary of Forecasting Methods

Method	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Mean Absolute Percentage Error (MAPE)
Single Exponential Smoothing	226.08985046541633	66655.34991736042	2.2515286249034485%
Simple Moving Average	205.55555555555577	52515.432098765516	2.0484350421165614%
Weighted Moving Average	219.72222222222223	59084.02777777778	2.1921286525299792%

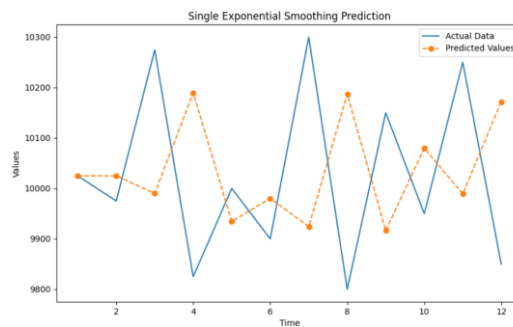
Gambar 4. 46 Hasil Perbandingan Percobaan 2

Pada Gambar 4.46 digunakan untuk melihat hasil dari ketiga metode tersebut dan membandingkan metode mana yang terbaik untuk forecasting penjualan. Hasil yang diperoleh yaitu metode Simple Moving Average adalah metode terbaik dalam perhitungan error tersebut

dengan hasil MAPE terkecil.

Percobaan 3:

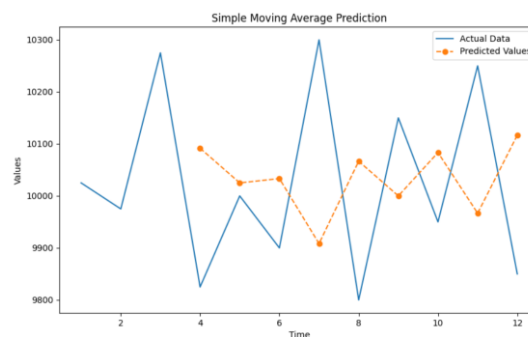
Single Exponential Smoothing Results



Gambar 4. 47 Grafik SES percobaan 3

Pada Gambar 4.47 yaitu dapat terlihat bahwa hasil percobaan 3 dengan menggunakan metode Single Exponential Smoothing dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

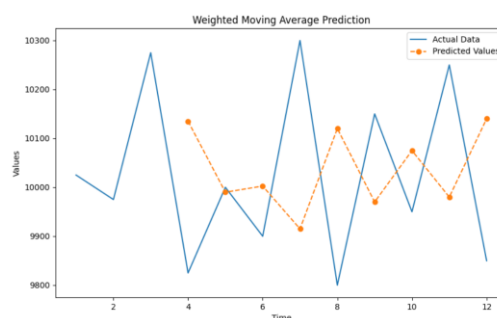
Simple Moving Average Results



Gambar 4. 48 Grafik SMA percobaan 3

Pada Gambar 4.48 yaitu dapat terlihat bahwa hasil percobaan 3 dengan menggunakan metode Simple Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Weighted Moving Average Results



Gambar 4. 49 Grafik WMA percobaan 3

Pada Gambar 4.49 yaitu dapat terlihat bahwa hasil percobaan 3 dengan menggunakan metode Weighted Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Summary of Forecasting Methods

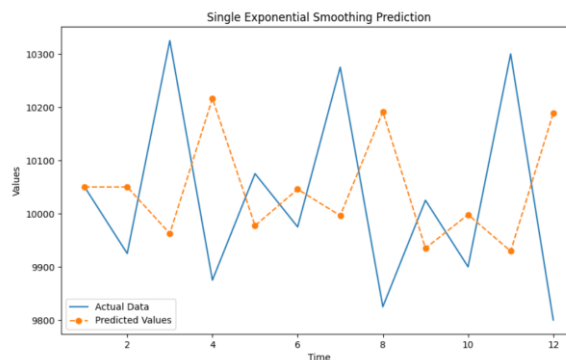
Method	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Mean Absolute Percentage Error (MAPE)
Single Exponential Smoothing	212.9170488912499	63480.121066727996	2.122906838061881%
Simple Moving Average	212.9629629629629	56188.27160493818	2.1248903093685083%
Weighted Moving Average	221.3888888888889	62484.02777777778	2.211303246698315%

Gambar 4.50 Hasil Perbandingan percobaan 3

Pada Gambar 4.50 digunakan untuk melihat hasil dari ketiga metode tersebut dan membandingkan metode mana yang terbaik untuk forecasting penjualan. Hasil yang diperoleh yaitu metode Single Exponential Smoothing adalah metode terbaik dalam perhitungan error tersebut dengan hasil MAPE terkecil.

Percobaan 4 :

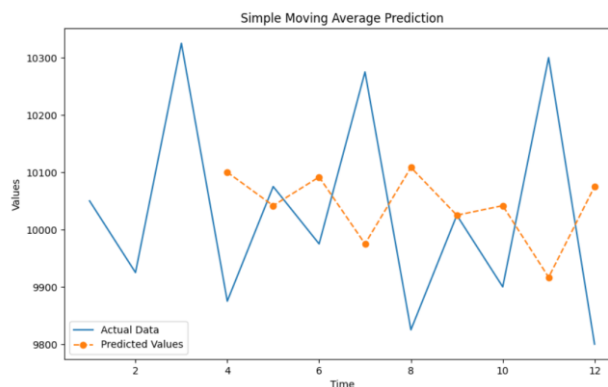
Single Exponential Smoothing Results



Gambar 4. 51 Grafik SES percobaan 4

Pada Gambar 4.51 yaitu dapat terlihat bahwa hasil percobaan 4 dengan menggunakan metode Single Exponential Smoothing dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

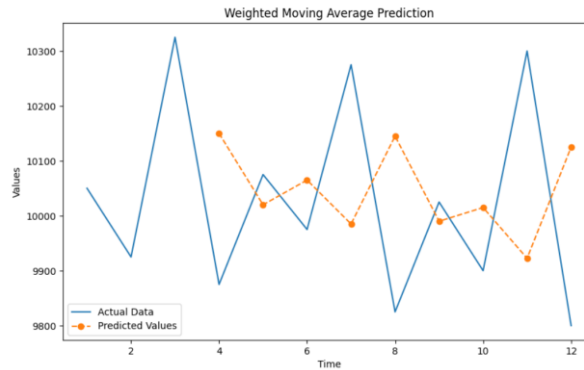
Simple Moving Average Results



Gambar 4. 52 Grafik SMA percobaan 4

Pada Gambar 4.52 yaitu dapat terlihat bahwa hasil percobaan 4 dengan menggunakan metode Simple Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Weighted Moving Average Results



Gambar 4.53 Grafik WMA percobaan 4

Pada Gambar 4.53 yaitu dapat terlihat bahwa hasil percobaan 4 dengan menggunakan metode Weighted Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Summary of Forecasting Methods

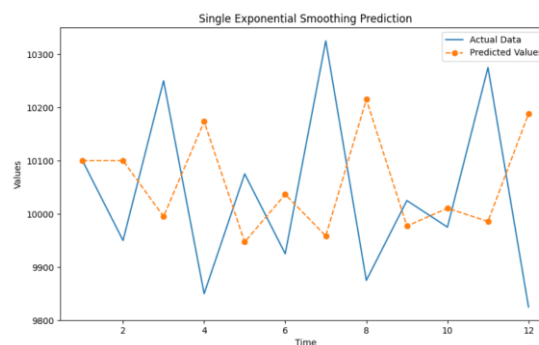
Method	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Mean Absolute Percentage Error (MAPE)
Single Exponential Smoothing	215.81059389895836	66350.22223619015	2.1500009519778134%
Simple Moving Average	195.37037037037044	53140.43209876549	1.9490236874558353%
Weighted Moving Average	209.16666666666666	59536.805555555555	2.0893857527259168%

Gambar 4. 54Hasil Perbandingan percobaan 4

Pada Gambar 4.54 digunakan untuk melihat hasil dari ketiga metode tersebut dan membandingkan metode mana yang terbaik untuk forecasting penjualan. Hasil yang diperoleh yaitu metode Simple Moving Average adalah metode terbaik dalam perhitungan error tersebut dengan hasil MAPE terkecil.

Percobaan 5 :

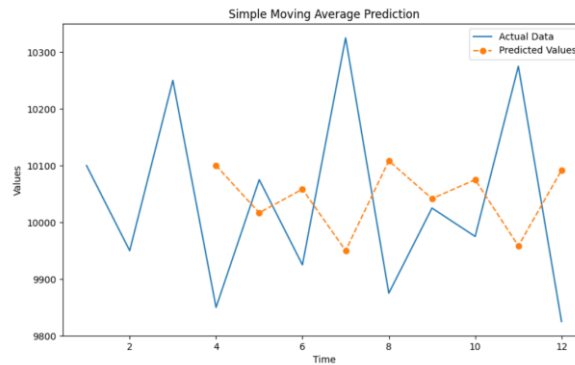
Single Exponential Smoothing Results



Gambar 4. 55 Grafik SES percobaan 5

Pada Gambar 4.55 yaitu dapat terlihat bahwa hasil percobaan 5 dengan menggunakan metode Single Exponential Smoothing dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

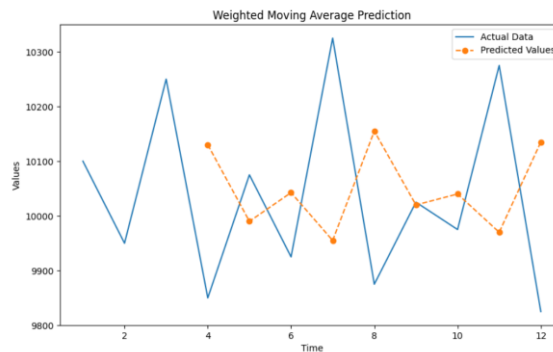
Simple Moving Average Results



Gambar 4. 56 Grafik SMA percobaan 5

Pada Gambar 4.56 yaitu dapat terlihat bahwa hasil percobaan 5 dengan menggunakan metode Simple Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Weighted Moving Average Results



Gambar 4. 57 Grafik WMA percobaan 5

Pada Gambar 4.57 yaitu dapat terlihat bahwa hasil percobaan 5 dengan menggunakan metode Weighted Moving Average dengan penjelasan bahwa warna biru merupakan garis data asli dan warna oren adalah garis data forecasting (perkiraan data)

Summary of Forecasting Methods

Method	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Mean Absolute Percentage Error (MAPE)
Single Exponential Smoothing	200.89495578208334	57513.660298616254	2.0012951445356317%
Simple Moving Average	194.44444444444446	51157.40740740739	1.9355708007572676%
Weighted Moving Average	201.94444444444446	56456.25	2.0126955321878004%

Gambar 4. 58 Hasil Perbandingan Percobaan 5

Pada Gambar 4.58 digunakan untuk melihat hasil dari ketiga metode tersebut dan membandingkan metode mana yang terbaik untuk forecasting penjualan. Hasil yang diperoleh yaitu metode Simple Moving Average adalah metode terbaik

1. 1.93%
2. 3.01%
3. 2.04%
4. 2.12%
5. 1.94%

Untuk menghitung rata-rata dari data MAPE tersebut dengan cara menjumlahkan semua nilai dan kemudian membaginya dengan jumlah data (5).

$$\text{Rumus rata-rata : Rata-rata} = \frac{\text{Jumlah Semua Nilai}}{\text{Jumlah Data}}$$

Menghitung jumlah semua nilai:

$$1.93 + 3.01 + 2.04 + 2.12 + 1.94 = 11.04$$

Membagi dengan jumlah data :

$$\text{Rata-rata} = \frac{11.04}{5} = 2.208$$

Jadi, rata-rata dari data MAPE adalah **2.208%**. dalam perhitungan error tersebut dengan hasil MAPE terkecil.

4.1.7 Kesimpulan Perhitungan Testing

Setelah melakukan lima kali percobaan untuk memprediksi penjualan di restoran "Nasi Goreng Bacot" menggunakan berbagai metode forecasting, hasil analisis menunjukkan bahwa metode Simple Moving Average (SMA) adalah yang paling efektif. Dari lima percobaan yang dilakukan, empat di antaranya menunjukkan bahwa Mean Absolute Percentage Error (MAPE) dari metode SMA lebih kecil dibandingkan dengan metode lainnya seperti Weighted Moving Average (WMA) dan Single Exponential Smoothing (SES). Hal ini menunjukkan bahwa SMA memberikan prediksi yang lebih akurat dan konsisten dalam konteks data penjualan restoran ini.

Dalam percobaan pertama, metode SMA menghasilkan MAPE yang lebih rendah dibandingkan dengan WMA dan SES. Hasil ini menunjukkan bahwa SMA mampu menangkap pola penjualan dengan lebih baik, meskipun data penjualan menunjukkan fluktuasi yang wajar. Percobaan kedua dan ketiga juga menunjukkan hasil yang serupa, di mana SMA terus-menerus memberikan prediksi yang lebih akurat dengan MAPE yang lebih kecil. Hal ini mengindikasikan bahwa SMA memiliki keunggulan dalam menangani data penjualan yang stabil dan konsisten, seperti yang terlihat pada data penjualan "Nasi Goreng Bacot".

Pada percobaan keempat, meskipun ada sedikit peningkatan dalam MAPE untuk metode SMA, hasilnya masih lebih baik dibandingkan dengan metode WMA dan SES. Ini menunjukkan bahwa SMA tetap menjadi metode yang lebih andal dalam memprediksi penjualan, bahkan ketika ada variasi dalam data. Percobaan kelima juga mengkonfirmasi keunggulan SMA, dengan MAPE yang lebih rendah dibandingkan dengan metode lainnya. Konsistensi hasil ini menunjukkan bahwa SMA adalah metode yang paling sesuai untuk digunakan dalam konteks data penjualan restoran ini.

Secara keseluruhan, hasil dari lima percobaan ini menunjukkan bahwa metode Simple Moving Average (SMA) adalah yang terbaik untuk memprediksi penjualan di restoran "Nasi Goreng Bacot". Dengan MAPE yang lebih rendah dalam empat dari lima percobaan, SMA terbukti memberikan prediksi yang lebih akurat dan dapat diandalkan. Oleh karena itu, manajemen restoran dapat menggunakan metode SMA untuk mengoptimalkan persediaan, meningkatkan kepuasan pelanggan, dan membuat keputusan strategis yang lebih baik.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan, dapat disimpulkan bahwa metode peramalan yang digunakan memiliki tingkat akurasi yang berbeda-beda. Metode Simple Moving Average (SMA) terbukti menjadi yang terbaik dalam hal peramalan data penjualan pada Restoran Nasi Goreng "Bacot" Surabaya. Hal ini ditunjukkan oleh nilai Mean Absolute Error (MAE), Mean Squared Error (MSE) dan Mean Absolute Percentage Error (MAPE) yang paling rendah dibandingkan dengan metode Weighted Moving Average (WMA) dan Single Exponential Smoothing (SES).

Metode SMA memberikan nilai MAE sebesar 296.67, nilai MSE sebesar 129,055.6, dan MAPE sebesar 3.015 % yang menunjukkan tingkat kesalahan yang lebih rendah dan lebih dapat diandalkan dalam peramalan data penjualan. Sedangkan metode Weighted Moving Average dan Single Exponential Smoothing memiliki nilai kesalahan yang lebih tinggi, sehingga kurang efektif dalam aplikasi peramalan untuk dataset ini.

5.2 Saran

1. Untuk penelitian selanjutnya, disarankan untuk mencoba metode peramalan lain seperti ARIMA atau metode berbasis machine learning untuk dibandingkan dengan metode yang telah digunakan dalam penelitian ini. Hal ini dapat memberikan wawasan lebih lanjut mengenai metode yang paling optimal untuk data penjualan yang serupa.

2. Disarankan untuk menggunakan dataset yang lebih besar dan lebih bervariasi untuk meningkatkan akurasi model peramalan. Data yang lebih lengkap dapat memberikan gambaran yang lebih akurat mengenai pola penjualan.
3. Selain menggunakan data penjualan, penelitian selanjutnya dapat mempertimbangkan faktor-faktor eksternal seperti musim, promosi, dan tren pasar yang dapat mempengaruhi penjualan. Analisis ini dapat memberikan prediksi yang lebih akurat dan juga bisa menggunakan metode *decompose* untuk melihat apakah data yang diteliti adalah termasuk data yang stasioner atau yang termasuk musim, tren, dll.

6. DAFTAR REFERENSI

- Apriliani, A., Zainuddin, H., Agussalim, A., & Hasanuddin, Z. (2020). Peramalan Tren Penjualan Menu Restoran Menggunakan Metode Single Moving Average. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 7(6).
- Biri, R., Langi, Y. A. R., & Paendong, M. S. (2019). Penggunaan metode smoothing EKSPONENSIAL Dalam meramal pergerakan inflasi Kota Palu. *JURNAL ILMIAH SAINS*, 13(1), 68.
- Frischa, L., Sari, R. F., & Husein, I. (2023). Comparison of ARIMA and Winters Methods on Sales Forecasting of Furniture Companies at UD Podomoro Asahan.
- Heizer, J., & Render, B. (2020). *Operations Management: Sustainability and Supply Chain Management*. Edinburgh Gate: Pearson.
- Lasek, A., Cercone, N., & Saunders, J. (2019). Sales and customer demand forecasting: Literature survey and categorization of methods. *Social-Informatics and Telecommunications Engineering, LNICST, Lecture Notes of the Institute for Computer Sciences*.
- Liu, L.-M., Bhattacharyya, S., Sclove, S. L., Chen, R., & Lattyak, W. J. (2020). Data mining on time series: An illustration using fast-food restaurant franchise data. *Computational Statistics & Data Analysis*, 37(4).
- Sri Radina Putri NWP, H. S. (2018). Usulan Penggunaan Metode Forecasting Untuk Permintaan Kopi Robusta Pada Pt. Xyz. *Industrial Engineering Online*.
- Subagyo, Pangestu. (2020). *Forecasting: Konsep dan Aplikasi*. BPPE UGM.
- Tanizaki, T., Hoshino, T., Shimmura, T., & Takenaka, T. (2020). Restaurants store management based on demand forecasting. *Procedia CIRP*, 88.
- Tarsisius Handoko. (2018). *Dasar-dasar Manajemen Produksi dan Operasi*.