

**PRAKTIKUM PEMODELAN STATISTIKA**  
**MODUL 10**



**Disusun oleh :**

**Nama : Fidelia Ping**

**NIM : 245410012**

**Kelas : Informatika 1**

**PROGRAM STUDI INFORMATIKA**  
**PROGRAM SARJANA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA**  
**YOGYAKARTA**  
**2025**

# MODUL 10

## FORECASTING DENGAN MODEL MOVING AVERAGE

### A. TUJUAN PRAKTIKUM

1. Memahami konsep dasar forecasting dengan model moving average.
2. Melakukan forecasting dengan model moving average menggunakan R

### B. PEMBAHASAN LISTING

#### PRAKTIK

#### Kasus 1 : Forecasting dg Simple Moving Average

Kasus: Meramalkan Jumlah Penumpang Pesawat Tahun 1960

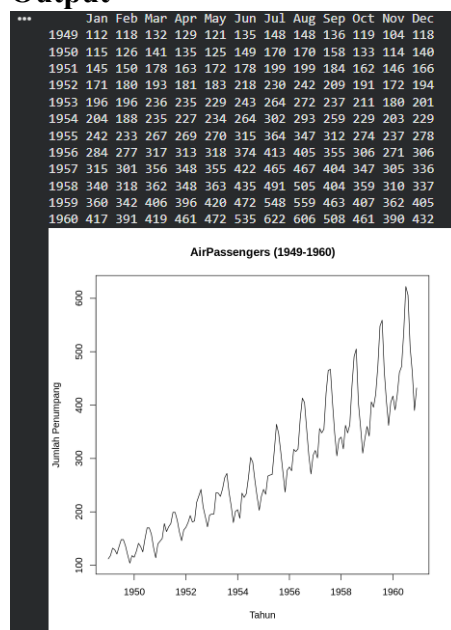
#### 1. DATA & EKSPLORASI AWAL

```
# Dataset bawaan R
data("AirPassengers")
AP <- AirPassengers

# Lihat ringkasan
print(AP)
plot(AP, main = "AirPassengers (1949-1960)", ylab = "Jumlah Penumpang", xlab = "Tahun")
```

**Pembahasan :** Kode di atas memanggil dataset bawaan R bernama AirPassengers, yaitu data jumlah penumpang pesawat internasional bulanan dari tahun 1949 hingga 1960. Baris data("AirPassengers") memuat dataset, kemudian disimpan ke dalam variabel AP. Perintah print(AP) digunakan untuk menampilkan seluruh nilai deret waktu (time series) tersebut dalam bentuk angka per bulan secara berurutan. Selanjutnya, fungsi plot() membuat grafik time series yang menunjukkan perubahan jumlah penumpang berdasarkan waktu, dengan judul "AirPassengers (1949–1960)" serta label sumbu x (Tahun) dan sumbu y (Jumlah Penumpang). Secara keseluruhan, kode ini bertujuan untuk menampilkan dan memvisualisasikan pola data sebelum dilakukan tahap forecasting atau analisis lanjutan.

#### Output



**Pembahasan Output :** Output pertama dari print(AP) berupa daftar angka yang merepresentasikan jumlah penumpang tiap bulan dari 1949 sampai 1960, ditampilkan dalam format deret waktu (ts) R dengan pembagian per tahun dan frekuensi 12 bulan. Data menunjukkan peningkatan penumpang dari sekitar 100 ribu ke lebih dari 600 ribu pada akhir periode. Output kedua berupa grafik time series yang menampilkan tren kenaikan jumlah penumpang dari tahun ke tahun, dengan pola musiman (seasonality) sangat jelas jumlah penumpang naik setiap pertengahan tahun dan turun di awal tahun. Grafik ini menggambarkan adanya pola tren meningkat dan musim yang konsisten sehingga cocok untuk analisis deret waktu dan metode forecasting seperti Moving Average atau ARIMA.

## 2. TRAIN-TEST SPLIT

```
# Misal: Training = sampai 1959, Testing = tahun 1960
train <- window(AP, end = c(1959, 12))
test  <- window(AP, start = c(1960, 1))
```

**Pembahasan :** Dengan menggunakan fungsi window() pada objek time series AP, kode train <- window(AP, end = c(1959, 12)) mengambil semua data dari awal (Januari 1949) sampai akhir Desember 1959 sebagai data pelatihan. Sedangkan test <- window(AP, start = c(1960, 1)) mengambil data mulai Januari 1960 hingga akhir dataset sebagai data pengujian/validasi. Teknik ini memisahkan data historis untuk membangun model (training) dengan data masa depan yang belum dipakai (testing), sehingga kita bisa mengevaluasi seberapa baik model memprediksi data “baru” (tahun 1960).

## 3. FORECASTING SIMPLE MOVING AVERAGE (SMA)

```
# Kita pakai SMA dengan orde = 12 bulan (1 tahun)
k <- 12 # panjang jendela moving average

# Untuk forecasting, kita akan membuat ramalan 1-step-ahead secara bergulir
train_test_all <- c(train, test) # gabungkan untuk memudahkan akses indeks

n_train <- length(train)
n_test  <- length(test)

forecast_sma <- rep(NA, n_test)

# Loop: untuk setiap titik di test, ramalkan dengan rata-rata 12 bulan terakhir
for (i in 1:n_test) {
  # indeks terakhir data aktual sebelum diramal
  idx_last_actual <- n_train + i - 1

  # ambil 12 data terakhir (jika sudah cukup)
  window_values <- train_test_all[(idx_last_actual - k + 1):idx_last_actual]

  forecast_sma[i] <- mean(window_values)
}

# Bandingkan forecast vs aktual
data.frame(
  Periode = time(test),
  Aktual  = as.numeric(test),
  Forecast = round(forecast_sma, 2)
)
```

**Pembahasan :** Kode ini melakukan peramalan menggunakan metode Simple Moving Average (SMA) dengan panjang jendela 12 bulan, artinya setiap nilai prediksi dihitung dari rata-rata penumpang selama 12 bulan terakhir sebelum periode yang diramalkan. Variabel `k <- 12` menentukan ukuran jendela SMA, sementara `train_test_all <- c(train, test)` menggabungkan semua data sehingga loop dapat mengakses data historis secara berurutan. Di dalam loop, setiap iterasi `i` menghitung posisi data aktual terakhir (`idx_last_actual`) dan mengambil 12 data sebelumnya sebagai `window_values`, kemudian menghitung rata-ratanya sebagai nilai forecast. Proses ini diulang sebanyak jumlah data testing (`n_test`, yaitu 12 bulan tahun 1960), sehingga menghasilkan prediksi 1-step-ahead untuk setiap bulan.

## Output

... A data.frame: 12 x 3

Periode	Aktual	Forecast
<ts>	<dbl>	<dbl>
1960.000	417	428.33
1960.083	391	433.08
1960.167	419	437.17
1960.250	461	438.25
1960.333	472	443.67
1960.417	535	448.00
1960.500	622	453.25
1960.583	606	459.42
1960.667	508	463.33
1960.750	461	467.08
1960.833	390	471.58
1960.917	432	473.92

**Pembahasan Output :** Output berupa sebuah data frame yang menampilkan periode (bulan 1960), nilai aktual, dan nilai hasil prediksi SMA (forecast) untuk setiap bulan. Kolom “Aktual” menampilkan jumlah penumpang yang benar-benar tercatat pada dataset *AirPassengers*, sedangkan kolom “Forecast” menampilkan nilai peramalan yang berasal dari rata-rata 12 bulan sebelumnya. Karena SMA menggunakan jendela tetap, nilai forecast cenderung lebih halus dan tidak mengikuti lonjakan musiman yang ada pada data asli. Dengan membandingkan kedua kolom tersebut, pengguna dapat melihat seberapa baik prediksi mengikuti pola aktual, dan nilai ini nantinya bisa digunakan untuk menghitung error (RMSE, MAE, MAPE) pada langkah evaluasi model.

#### 4. EVALUASI MODEL

```
aktual <- as.numeric(test)
prediksi <- forecast_sma

# RMSE (Root Mean Squared Error)
rmse <- sqrt(mean((aktual - prediksi)^2))

# MAE (Mean Absolute Error)
mae <- mean(abs(aktual - prediksi))

# MAPE (Mean Absolute Percentage Error) dalam persen
mape <- mean(abs((aktual - prediksi) / aktual)) * 100

cat("Evaluasi Model SMA (k = ", k, ")\n", sep = "")
cat("RMSE : ", rmse, "\n", sep = "")
cat("MAE : ", mae, "\n", sep = "")
cat("MAPE : ", mape, " %\n", sep = "")
```

**Pembahasan :** Kode ini melakukan evaluasi performa model peramalan Simple Moving Average (SMA) dengan membandingkan nilai aktual (aktual) dan nilai hasil prediksi (prediksi). Tiga metrik error digunakan: RMSE, MAE, dan MAPE. RMSE menghitung akar dari rata-rata kuadrat selisih antara aktual dan prediksi, sehingga lebih sensitif terhadap kesalahan besar. MAE menghitung rata-rata selisih absolut antara nilai aktual dan prediksi, memberikan gambaran seberapa besar kesalahan rata-rata tanpa memperbesar efek error yang besar. MAPE mengukur kesalahan dalam persentase terhadap nilai aktual, sehingga menunjukkan tingkat akurasi relatif model. Ketiga nilai ini kemudian dicetak dengan fungsi cat() agar tampil rapi dan informatif sesuai format yang digunakan pada modul praktikum.

##### Output

```
*** Evaluasi Model SMA (k = 12)
RMSE : 77.20483
MAE : 58.27083
MAPE : 11.4224 %
```

**Pembahasan Output :** Output berupa tiga nilai evaluasi: RMSE, MAE, dan MAPE. RMSE menunjukkan seberapa besar deviasi prediksi SMA dari nilai aktual dalam satuan jumlah penumpang—semakin kecil, semakin baik. MAE memperlihatkan rata-rata selisih absolut antara nilai prediksi dan nilai asli, sehingga memberikan ukuran kesalahan yang mudah dipahami. MAPE menampilkan tingkat kesalahan dalam bentuk persentase; misalnya jika MAPE bernilai 6%, berarti rata-rata kesalahan prediksi adalah 6% dari nilai aktual. Secara keseluruhan, output ini membantu menilai apakah SMA dengan jendela 12 sudah cukup akurat atau perlu diganti dengan metode lain seperti WMA atau model yang lebih adaptif.

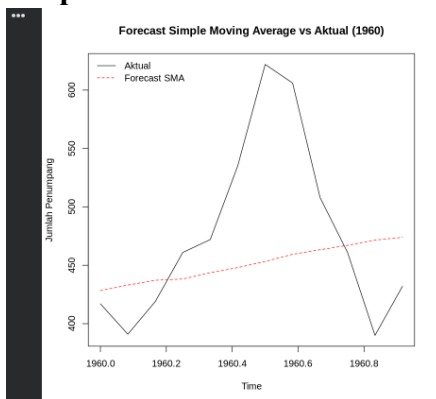
## 5. VISUALISASI: AKTUAL vs FORECAST

```
ts.plot(test, forecast_sma,
        col = c("black", "red"),
        lty = c(1,2),
        main = "Forecast Simple Moving Average vs Aktual (1960)",
        ylab = "Jumlah Penumpang")

legend("topleft",
      legend = c("Aktual", "Forecast SMA"),
      col     = c("black", "red"),
      lty     = c(1,2),
      bty     = "n")
```

**Pembahasan :** Kode ini menampilkan visualisasi hasil peramalan Simple Moving Average (SMA) dengan menggunakan fungsi `ts.plot()` untuk memplot dua deret waktu sekaligus, yaitu data aktual (`test`) dan data hasil perkiraan (`forecast_sma`) untuk periode tahun 1960. Argumen `col = c("black", "red")` membedakan warna garis, di mana warna hitam digunakan untuk data aktual dan merah untuk hasil prediksi. Parameter `lty = c(1,2)` menentukan tipe garis, dengan garis solid untuk data aktual dan garis putus-putus untuk forecast. Judul grafik (`main`) dan label sumbu (`ylab`) ditambahkan agar grafik lebih informatif. Selanjutnya, fungsi `legend()` digunakan untuk menambahkan keterangan pada bagian kiri atas grafik sehingga pengguna dapat dengan mudah membedakan antara garis aktual dan garis hasil peramalan.

### Output



**Pembahasan Output :** Output dari kode ini berupa sebuah grafik time series yang memperlihatkan perbandingan antara nilai aktual jumlah penumpang pesawat pada tahun 1960 dengan nilai prediksi yang dihasilkan oleh model SMA 12-periode. Pada grafik tersebut, garis hitam menunjukkan pola aktual yang cenderung mengalami kenaikan dan memiliki pola musiman bulanan yang kuat, sementara garis merah putus-putus menggambarkan nilai forecast yang lebih halus karena dihitung sebagai rata-rata dari 12 bulan sebelumnya. Perbedaan antara kedua garis menunjukkan seberapa baik model SMA mengikuti pola data asli—model biasanya tertinggal pada puncak dan lembah musiman karena sifat SMA yang meratakan fluktuasi. Grafik ini membantu menilai apakah SMA tepat digunakan atau perlu diganti dengan metode peramalan yang lebih responsif.

## Praktik 2: Forecasting Weighted Moving Average

Forecasting dengan Weighted Moving Average (WMA) menggunakan dataset bawaan R yaitu `nottem`, data suhu udara bulanan di Nottingham (Inggris), 1920–1939.

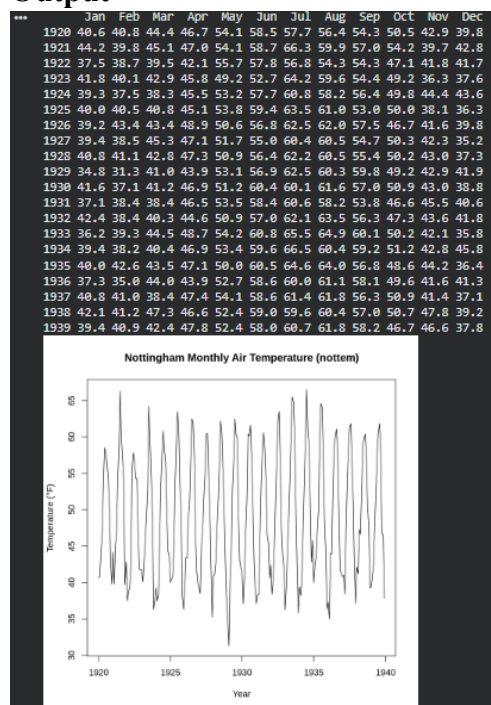
### 1. Install & load packages

```
# Dataset bawaan R
data("nottem")
ts_data <- nottem # time series bulanan

# Cek ringkasan dan plot
print(ts_data)
plot(ts_data,
      main = "Nottingham Monthly Air Temperature (nottem)",
      ylab = "Temperature (°F)",
      xlab = "Year")
```

**Pembahasan :** Kode ini memuat dataset bawaan R bernama `nottem`, yaitu data suhu udara bulanan di Nottingham, Inggris, dari tahun 1920 sampai 1939. Baris `data("nottem")` mengambil dataset tersebut, kemudian disimpan ke dalam variabel `ts_data` sebagai objek time series. Perintah `print(ts_data)` digunakan untuk menampilkan seluruh nilai suhu per bulan dalam format deret waktu lengkap, sedangkan fungsi `plot()` membuat grafik time series untuk menggambarkan perubahan suhu dari tahun ke tahun. Grafik diberi judul “Nottingham Monthly Air Temperature (nottem)” serta label sumbu x (Year) dan sumbu y (Temperature), sehingga membantu pengguna mengamati pola musiman maupun tren tahunan sebelum melakukan peramalan dengan Weighted Moving Average.

### Output



**Pembahasan Output :** Output pertama dari `print(ts_data)` berupa daftar angka yang menunjukkan nilai suhu udara bulanan dalam satuan derajat Fahrenheit selama 20 tahun (1920–1939). Data terlihat berulang mengikuti pola musiman: suhu naik pada pertengahan tahun dan turun pada awal serta akhir tahun. Output kedua berupa grafik time series yang menampilkan gelombang musiman yang jelas, di mana setiap tahun terjadi siklus naik-turun suhu secara teratur tanpa tren jangka panjang yang signifikan.

Grafik ini memperlihatkan bahwa data stabil dan memiliki pola bulanan yang kuat, sehingga cocok dianalisis menggunakan metode peramalan seperti Weighted Moving Average, yang lebih responsif terhadap perubahan terbaru dibanding SMA.

## 2. Split dataset

```
# Train: sampai 1937-12
train <- window(ts_data, end = c(1937, 12))

# Test: mulai 1938-01
test <- window(ts_data, start = c(1938, 1))

# 3. Forecasting dengan WMA
# Bobot untuk WMA (dari lama ke baru)
#Bulan paling lama → bobot 1
#Tengah → bobot 2
#Bulan terbaru → bobot 3
weights <- c(1, 2, 3)
k <- length(weights)

# Gabungkan train + test untuk memudahkan sliding window
all_values <- c(train, test)
n_train <- length(train)
n_test <- length(test)

# Vektor untuk menyimpan hasil forecast
forecast_wma <- rep(NA, n_test)

# Loop: 1-step-ahead forecast untuk setiap titik test
for (i in 1:n_test) {
  idx_last <- n_train + i - 1 # indeks observasi terakhir sebelum diramal
  window_data <- all_values[(idx_last - k + 1):idx_last] # 3 bulan terakhir
  forecast_wma[i] <- sum(window_data * weights) / sum(weights)
}

# Lihat perbandingan aktual vs forecast
comparison <- data.frame(
  Periode = time(test),
  Aktual = as.numeric(test),
  WMA_Forecast = round(forecast_wma, 2)
)
print(comparison)

actual <- as.numeric(test)
pred <- forecast_wma
```

**Pembahasan :** Kode ini melakukan proses forecasting menggunakan metode Weighted Moving Average (WMA) pada dataset nottem. Pertama, data dibagi menjadi dua bagian: train (hingga Desember 1937) dan test (mulai Januari 1938). Kemudian ditentukan bobot WMA sebanyak tiga nilai: 1 untuk bulan terlama, 2 untuk bulan tengah, dan 3 untuk bulan terbaru. Data train dan test digabung dalam satu vektor (all\_values) agar model mudah mengambil tiga data terakhir sebelum setiap peramalan dilakukan. Selanjutnya, sebuah loop dijalankan untuk menghasilkan forecast 1-step-ahead untuk tiap bulan pada periode test; pada setiap iterasi, model mengambil tiga bulan terakhir, mengalikan masing-masing dengan bobot, menjumlahkannya, dan membaginya dengan total bobot. Hasil peramalan tiap bulan disimpan dalam forecast\_wma, lalu dibandingkan dengan data aktual melalui data frame comparison.



## Output

***	Periode	Aktual	WMA_Forecast
1	1938.000	42.1	40.83
2	1938.083	41.2	40.32
3	1938.167	47.3	40.82
4	1938.250	46.6	44.40
5	1938.333	52.4	45.93
6	1938.417	59.0	49.62
7	1938.500	59.6	54.73
8	1938.583	60.4	58.20
9	1938.667	57.0	59.90
10	1938.750	50.7	58.57
11	1938.833	47.8	54.42
12	1938.917	39.2	50.30
13	1939.000	39.4	43.98
14	1939.083	40.9	40.73
15	1939.167	42.4	40.12
16	1939.250	47.8	41.40
17	1939.333	52.4	44.85
18	1939.417	58.0	49.20
19	1939.500	60.7	54.43
20	1939.583	61.8	58.42
21	1939.667	58.2	60.80
22	1939.750	46.7	59.82
23	1939.833	46.6	53.05
24	1939.917	37.8	48.57

**Pembahasan Output :** Output berupa sebuah tabel (comparison) yang menampilkan tiga kolom: Periode, Aktual, dan WMA\_Forecast. Kolom “Periode” menunjukkan bulan-bulan pada tahun 1938–1939 yang menjadi data testing, “Aktual” berisi nilai suhu udara sebenarnya pada setiap bulan tersebut, dan “WMA\_Forecast” berisi hasil prediksi yang dihitung menggunakan metode Weighted Moving Average berbobot (1, 2, 3). Melalui tabel ini, dapat terlihat bagaimana nilai hasil forecast cenderung lebih halus dibandingkan data aktual karena bobot terarah membuat peramalan lebih menekankan nilai terbaru. Tabel ini juga menjadi dasar untuk menghitung error pada langkah evaluasi dan memberikan gambaran awal mengenai akurasi model WMA dalam mengikuti pola musiman suhu udara di Nottingham.

## 4. Evaluasi

```
# Root Mean Squared Error
RMSE <- sqrt(mean((actual - pred)^2))

# Mean Absolute Error
MAE <- mean(abs(actual - pred))

# Mean Absolute Percentage Error
MAPE <- mean(abs((actual - pred) / actual)) * 100

cat("Evaluasi Model Weighted Moving Average (WMA)\n")
cat("RMSE :", RMSE, "\n")
cat("MAE :", MAE, "\n")
cat("MAPE :", MAPE, "%\n")
```

**Pembahasan :** Potongan kode tersebut digunakan untuk menghitung tiga metrik evaluasi utama pada model peramalan menggunakan Weighted Moving Average (WMA). Pertama, RMSE dihitung dengan rumus akar dari rata-rata kuadrat selisih antara data aktual dan hasil prediksi, sehingga menunjukkan seberapa besar error yang terjadi dengan memberi penalti lebih besar pada kesalahan yang besar. Kedua, MAE menghitung rata-rata nilai absolut dari selisih antara nilai aktual dan prediksi, memberikan gambaran umum tentang besar kesalahan tanpa memperhitungkan arah. Ketiga, MAPE mengukur persentase kesalahan absolut rata-rata terhadap nilai aktual sehingga mudah dipahami

dalam bentuk persentase. Setelah ketiga metrik dihitung, fungsi `cat()` digunakan untuk menampilkan hasil evaluasi tersebut secara rapi di konsol.

### Output

```
*** Evaluasi Model Weighted Moving Average (WMA)
RMSE : 6.56262
MAE : 5.608333
MAPE : 11.56121 %
```

**Pembahasan Output :** Output dari kode tersebut akan menampilkan tiga nilai evaluasi, yaitu RMSE, MAE, dan MAPE, yang masing-masing menggambarkan kualitas kinerja model WMA dalam memprediksi data. Jika nilai RMSE dan MAE rendah, hal ini menunjukkan bahwa model menghasilkan error yang kecil dan prediksi cukup mendekati nilai aktual. Sementara itu, nilai MAPE yang semakin kecil menunjukkan bahwa kesalahan prediksi relatif terhadap nilai aktual juga rendah, sehingga model dianggap lebih akurat secara persentase. Secara keseluruhan, ketiga metrik tersebut memberikan gambaran seberapa baik model WMA dalam melakukan peramalan, sehingga memudahkan pengguna untuk menilai apakah model sudah layak digunakan atau perlu ditingkatkan.

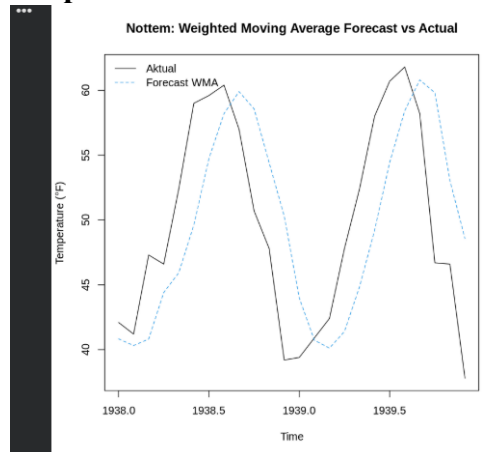
## 5. Visualisasi

```
# Plot time series test & forecast
ts.plot(test, forecast_wma,
        col = c(1, 4),          # 1=black, 4=blue
        lty = c(1, 2),
        main = "Nottem: Weighted Moving Average Forecast vs Actual",
        ylab = "Temperature (°F)")

legend("topleft",
      legend = c("Aktual", "Forecast WMA"),
      col     = c(1, 4),
      lty     = c(1, 2),
      bty     = "n")
```

**Pembahasan :** Kode tersebut digunakan untuk menampilkan visualisasi perbandingan antara data aktual (`test`) dan hasil peramalan model Weighted Moving Average (`forecast_wma`) dalam bentuk grafik time series. Fungsi `ts.plot()` memplot kedua deret waktu secara bersamaan dengan warna berbeda, di mana warna hitam (`col = 1`) merepresentasikan data aktual dan warna biru (`col = 4`) menunjukkan hasil peramalan, serta garis diberi gaya berbeda menggunakan `lty` untuk memudahkan pembedaan. Judul grafik dan label sumbu juga ditentukan agar plot lebih informatif. Selain itu, fungsi `legend()` ditempatkan di kiri atas untuk menampilkan keterangan garis, sehingga pembaca dapat memahami dengan jelas mana garis aktual dan mana garis hasil forecast.

## Output



**Pembahasan Output :** Output dari kode ini berupa grafik deret waktu yang memperlihatkan dua garis: garis hitam sebagai data aktual dan garis biru putus-putus sebagai prediksi WMA. Melalui visualisasi ini, pengguna dapat melihat seberapa dekat garis prediksi mengikuti pola data aktual pada periode pengujian. Jika garis forecast berada dekat dengan garis aktual, maka model dianggap memiliki performa yang baik, sedangkan jika garisnya sering menjauh atau tidak mengikuti pola musiman, berarti akurasi model kurang optimal. Grafik ini membantu mengevaluasi kualitas model secara visual, melengkapi evaluasi numerik seperti RMSE, MAE, dan MAPE.

### Kasus 3 : Forecasting dg Exponential Moving Average

Menggunakan dataset bawaan R *EuStockMarkets* (data indeks saham Eropa harian).

Dengan mengambil satu seri saja, misalnya DAX dari Jerman, lalu lakukan forecasting dengan Exponential Moving Average (EMA).

#### 1. Load Dataset bawaan

```
data("EuStockMarkets")
head(EuStockMarkets)

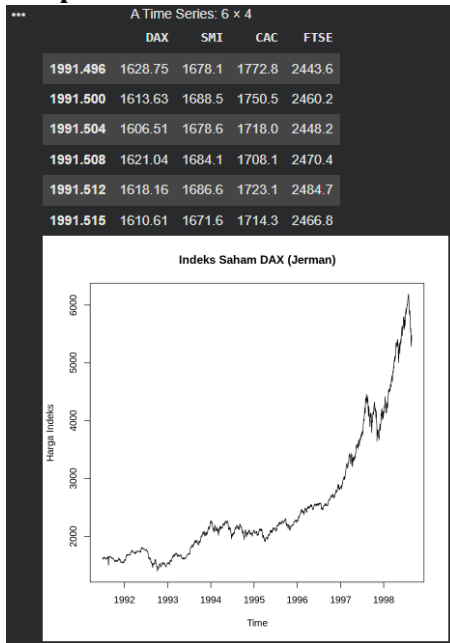
# Ambil satu indeks, misalkan DAX
dax <- EuStockMarkets[, "DAX"]

# Ubah ke time series
ts_dax <- ts(dax, start = c(1991, 130), frequency = 260) # 260 hari
kerja/tahun

plot(ts_dax,
      main = "Indeks Saham DAX (Jerman)",
      ylab = "Harga Indeks")
```

**Pembahasan :** Kode tersebut digunakan untuk mengambil dan memvisualisasikan data indeks saham Jerman (DAX) dari dataset bawaan R yaitu *EuStockMarkets*. Pertama, dataset dipanggil menggunakan `data("EuStockMarkets")`, kemudian `head()` digunakan untuk melihat beberapa baris awal sebagai gambaran struktur data. Selanjutnya, kolom indeks DAX diambil dari dataset dan disimpan ke dalam variabel `dax`. Data tersebut kemudian dikonversi menjadi objek *time series* melalui fungsi `ts()`, dengan penentuan waktu mulai (`start = c(1991, 130)`) dan frekuensi 260 yang merepresentasikan jumlah hari kerja bursa dalam setahun. Terakhir, fungsi `plot()` digunakan untuk menampilkan grafik time series harga indeks DAX agar pola pergerakannya dapat terlihat dengan jelas.

## Output



**Pembahasan Output :** Output kode tersebut berupa grafik deret waktu yang menampilkan pergerakan harga indeks saham DAX dari tahun 1991 pada hari ke-130 hingga tahun berikutnya sesuai panjang data. Grafik menunjukkan fluktuasi harga indeks dari waktu ke waktu, sehingga pengguna dapat melihat tren naik, penurunan, maupun volatilitas pasar saham Jerman pada periode tersebut. Visualisasi ini memberikan gambaran awal mengenai dinamika pasar dan membantu dalam analisis lanjutan seperti peramalan, deteksi pola, atau evaluasi stabilitas pasar.

## 2. Hitung Exponential Moving Average (EMA)

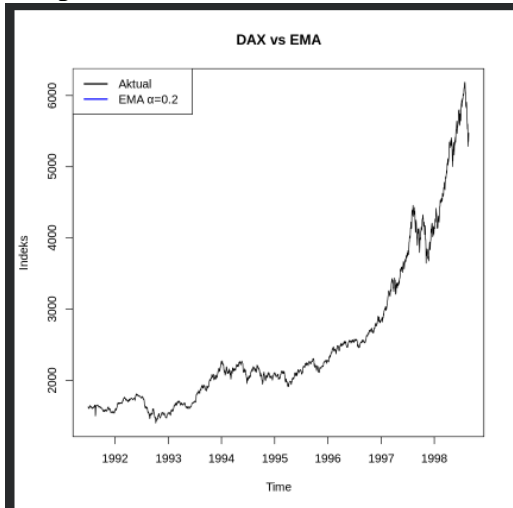
```
install.packages("TTR")
library(TTR)

alpha <- 0.2
ema_dax <- EMA(ts_dax, ratio = alpha)

# Plot bersama data aktual
plot(ts_dax, main = "DAX vs EMA", ylab = "Indeks")
lines(ema_dax, col = "blue", lwd = 2)
legend("topleft",
      legend = c("Aktual", "EMA  $\alpha=0.2$ "),
      col = c("black", "blue"),
      lty = 1, lwd = 2)
```

**Pembahasan :** Kode tersebut digunakan untuk melakukan perhitungan Exponential Moving Average (EMA) pada data indeks saham DAX dan memvisualisasikan hasilnya. Pertama, paket **TTR** di-install dan dimuat karena menyediakan fungsi teknikal analisis seperti EMA. Variabel  $\alpha$  ditetapkan sebesar 0.2 sebagai smoothing factor yang menentukan seberapa besar bobot nilai terbaru. Fungsi `EMA()` kemudian diterapkan pada data time series `ts_dax` untuk menghasilkan garis EMA yang lebih halus dibandingkan data asli. Setelah itu, grafik data aktual diplot menggunakan `plot()`, dan garis EMA ditambahkan dengan `lines()` berwarna biru agar mudah dibedakan. Terakhir, legend ditampilkan di sudut kiri atas untuk memberi keterangan mana garis aktual dan mana garis EMA.

## Output



**Pembahasan Output :** Output dari kode tersebut berupa sebuah grafik yang menampilkan dua garis: garis hitam yang menunjukkan data aktual indeks DAX, dan garis biru tebal yang menunjukkan nilai EMA dengan  $\alpha = 0.2$ . Garis EMA terlihat lebih halus dibandingkan data asli karena metode EMA mengurangi efek fluktuasi jangka pendek serta memberi bobot lebih besar pada data terbaru. Melalui grafik ini, pengguna dapat dengan mudah mengamati tren umum indeks saham serta membandingkan bagaimana EMA mengikuti pola pergerakan harga. Visualisasi ini sangat berguna untuk analisis teknikal, identifikasi tren, atau dasar untuk strategi trading berbasis indikator EMA.

## 3. Forecast 5 Hari ke Depan

```
forecast_step <- 5

forecast_value <- tail(ema_dax, 1)
forecast_ema <- rep(as.numeric(forecast_value), forecast_step)
cat("Forecast 5 Hari ke Depan\n")
forecast_ema
```

**Pembahasan :** Kode tersebut digunakan untuk membuat forecast sederhana menggunakan nilai terakhir dari Exponential Moving Average (EMA). Pertama, variabel `forecast_step` diisi dengan angka 5, yang berarti model akan memprediksi lima hari ke depan. Nilai EMA terakhir diambil menggunakan `tail(ema_dax, 1)` dan disimpan ke dalam `forecast_value`, karena pada metode EMA, nilai terbaru dianggap sebagai representasi terbaik dari tren saat ini. Kemudian, `rep()` digunakan untuk menggandakan nilai tersebut sebanyak jumlah langkah prediksi, menghasilkan vektor `forecast_ema` yang berisi prediksi untuk 5 hari mendatang. Terakhir, perintah `cat()` digunakan untuk menampilkan judul output di konsol, dan variabel `forecast_ema` ditampilkan sebagai hasil forecast.

## Output

```
... Forecast 5 Hari ke Depan
5506.13033425249 · 5506.13033425249 · 5506.13033425249 · 5506.13033425249 · 5506.13033425249
```

**Pembahasan Output :** Output dari kode tersebut berupa lima nilai prediksi yang semuanya sama, yaitu nilai EMA terakhir dari indeks DAX. Karena metode ini menggunakan pendekatan *naïve EMA forecast*, model berasumsi bahwa nilai beberapa hari ke depan akan mengikuti level EMA saat ini tanpa perubahan tren tambahan. Dengan



**Pembahasan Output :** Output yang dihasilkan terdiri dari dua bagian: pertama, tampilan deret waktu yang menunjukkan data penjualan X-Boost dalam format time series, di mana setiap baris merepresentasikan penjualan bulanan dari tahun 2023 hingga awal 2025. Kedua, grafik deret waktu yang memperlihatkan bagaimana penjualan berubah dari bulan ke bulan, termasuk fluktuasi, lonjakan, dan tren umum dalam periode tiga tahun tersebut. Melalui grafik ini, pengguna dapat melihat pola penjualan, apakah cenderung meningkat, stabil, atau mengalami variasi tertentu setiap tahun, sehingga membantu dalam analisis lebih lanjut seperti peramalan atau evaluasi performa bisnis.

```
train <- window(ts_xboost, end = c(2024,12))
test  <- window(ts_xboost, start = c(2025,1))
```

**Pembahasan :** kode tersebut digunakan untuk memisahkan data deret waktu penjualan X-Boost menjadi dua bagian, yaitu data *training* dan *testing*. Fungsi `window()` pertama mengambil data dari awal time series hingga akhir tahun 2024, sehingga variabel `train` berisi seluruh data penjualan tahun 2023 dan 2024. Kemudian, fungsi `window()` kedua mengambil data mulai Januari 2025, sehingga variabel `test` berisi data penjualan tahun 2025 yang digunakan sebagai pembandingan untuk evaluasi model. Pemisahan ini penting dalam analisis dan peramalan karena memungkinkan model belajar dari data historis, lalu diuji pada data yang tidak digunakan dalam proses pelatihan untuk mengukur akurasinya.

```
k <- 3 # jendela SMA sesuai latihan

all_values <- c(train, test)
n_train <- length(train)
n_test  <- length(test)

forecast_sma <- rep(NA, n_test)

for (i in 1:n_test) {
  idx_last <- n_train + i - 1
  window_data <- all_values[(idx_last - k + 1):idx_last]
  forecast_sma[i] <- mean(window_data)
}

# Tabel Hasil
data.frame(
  Bulan      = time(test),
  Aktual     = as.numeric(test),
  Forecast   = round(forecast_sma,2)
)
```

**Pembahasan :** kode tersebut digunakan untuk melakukan peramalan penjualan X-Boost menggunakan metode Simple Moving Average (SMA) dengan jendela  $k = 3$ . Pertama, seluruh data dari `train` dan `test` digabung menjadi satu deret waktu `all_values`, kemudian panjang masing-masing bagian dihitung. Variabel `forecast_sma` disiapkan sebagai penampung hasil prediksi dengan panjang yang sama seperti data `test`. Proses peramalan dilakukan melalui *loop*, di mana setiap iterasi mengambil tiga data terakhir sebelum titik prediksi (sesuai nilai  $k$ ) menggunakan indeks yang dihitung dari posisi `train`. Nilai rata-rata dari tiga data tersebut menjadi nilai `forecast` untuk bulan tersebut. Terakhir, hasil prediksi ditampilkan dalam bentuk tabel yang memuat waktu, nilai aktual dari data `test`, serta nilai hasil `forecast SMA` yang sudah dibulatkan dua desimal.

## Output

\*\*\* A data.frame: 3 x 3

Bulan	Aktual	Forecast
<ts>	<dbl>	<dbl>
2025.000	245	250.00
2025.083	230	248.33
2025.167	258	247.67

**Pembahasan Output :** Output yang dihasilkan berupa sebuah tabel yang menampilkan tiga kolom: bulan (time test), nilai aktual penjualan pada tiga bulan awal 2025, dan nilai prediksi yang dihitung menggunakan metode SMA dengan jendela tiga bulan. Tabel ini memudahkan perbandingan langsung antara penjualan sebenarnya dan hasil peramalan, sehingga pengguna dapat menilai apakah metode SMA mampu mengikuti pola data dengan baik. Jika nilai forecast mendekati nilai aktual, maka kinerja model dapat dikatakan cukup baik; sebaliknya, jika perbedaannya besar, model SMA mungkin kurang cocok atau jendela yang digunakan perlu disesuaikan.

```
actual <- as.numeric(test)
pred   <- forecast_sma

RMSE <- sqrt(mean((actual - pred)^2))
MAE  <- mean(abs(actual - pred))
MAPE <- mean(abs((actual - pred) / actual)) * 100

cat("Evaluasi Model SMA (k=3)\n")
cat("RMSE :", RMSE, "\n")
cat("MAE  :", MAE,  "\n")
cat("MAPE :", MAPE, "%\n")
```

**Pembahasan :** Kode tersebut digunakan untuk mengevaluasi akurasi model peramalan Simple Moving Average (SMA) dengan jendela  $k = 3$  berdasarkan tiga metrik umum: RMSE, MAE, dan MAPE. Nilai aktual dari data test disimpan dalam actual, sedangkan hasil prediksi yang sebelumnya dihitung disimpan dalam pred. RMSE dihitung dengan akar dari rata-rata kuadrat selisih antara actual dan prediksi, memberikan penalti untuk error besar. MAE menghitung rata-rata selisih absolut, menunjukkan besar error rata-rata tanpa memperhatikan arah. MAPE menghitung persentase kesalahan relatif terhadap nilai aktual sehingga mudah diinterpretasikan. Setelah ketiga metrik dihitung, cat() digunakan untuk mencetak hasil evaluasi model secara rapi di konsol.

## Output

\*\*\* Evaluasi Model SMA (k=3)

RMSE :	12.48851
MAE :	11.22222
MAPE :	4.672333 %

**Pembahasan Output :** output dari kode tersebut berupa tiga nilai evaluasi yaitu RMSE, MAE, dan MAPE yang menggambarkan kualitas performa model SMA dalam memprediksi penjualan X-Boost untuk tahun 2025. Jika nilai RMSE dan MAE rendah, berarti perbedaan antara hasil forecast dan nilai aktual kecil, menunjukkan model bekerja cukup baik. Nilai MAPE memberikan gambaran dalam bentuk persentase; semakin kecil



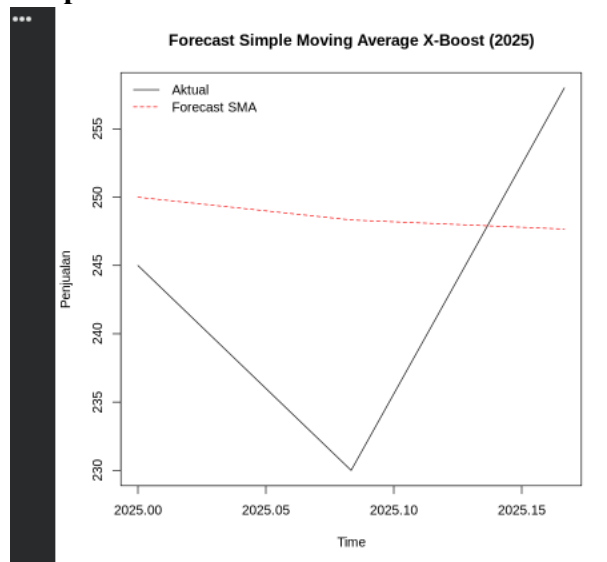
persentasenya, semakin akurat prediksi model terhadap data nyata. Secara keseluruhan, output evaluasi ini membantu menilai apakah SMA dengan jendela 3 sudah cocok digunakan atau perlu diperbaiki dengan mengubah parameter atau mengganti metode forecasting.

```
ts.plot(test, forecast_sma,
        col = c("black","red"),
        lty = c(1,2),
        main = "Forecast Simple Moving Average X-Boost (2025)",
        ylab = "Penjualan")

legend("topleft",
      legend = c("Aktual", "Forecast SMA"),
      col     = c("black","red"),
      lty     = c(1,2),
      bty     = "n")
```

**Pembahasan :** Kode tersebut digunakan untuk membuat visualisasi perbandingan antara data penjualan aktual tahun 2025 dan hasil peramalan menggunakan metode Simple Moving Average (SMA). Fungsi `ts.plot()` menampilkan kedua deret waktu secara bersamaan dengan garis berwarna hitam untuk data aktual (`test`) dan garis merah putus-putus untuk hasil forecast (`forecast_sma`). Parameter `lty` digunakan untuk membedakan gaya garis sehingga prediksi dan data aktual mudah dikenali. Judul grafik dan label sumbu ditambahkan agar plot lebih informatif. Selanjutnya, fungsi `legend()` ditempatkan di pojok kiri atas untuk memberikan keterangan garis, sehingga pengguna dapat mengenali mana garis aktual dan mana garis hasil peramalan SMA.

### Output



**Pembahasan Output :** Output dari kode tersebut berupa grafik deret waktu yang menampilkan dua garis: garis hitam yang menunjukkan nilai penjualan aktual pada tahun 2025 dan garis merah putus-putus yang menunjukkan hasil prediksi SMA dengan jendela 3 bulan. Melalui grafik ini, pengguna dapat melihat secara visual seberapa baik prediksi mengikuti pola nilai aktual. Jika garis merah mendekati garis hitam, maka model SMA dianggap cukup akurat. Sebaliknya, jika garisnya menyimpang jauh, berarti metode SMA kurang sesuai atau jendela perhitungan perlu diubah. Grafik ini memberikan cara cepat dan intuitif untuk menilai kualitas peramalan sebelum melakukan analisis lanjutan.

2. Lakukan peramalan penjualan smartphone menggunakan metode Weighted Moving Average untuk data penjualan smartphone merek “Zeta” (dalam ribu unit) – Bulanan Tahun 2022 s/d November 2025 sbb:

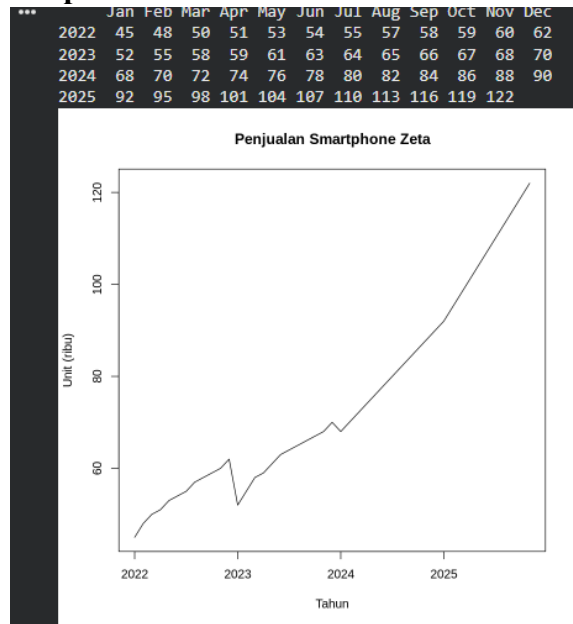
```
# Data penjualan smartphone Zeta (2022–Nov 2025)
zeta <- c(
  # 2022
  45,48,50,51,53,54,55,57,58,59,60,62,
  # 2023
  52,55,58,59,61,63,64,65,66,67,68,70,
  # 2024
  68,70,72,74,76,78,80,82,84,86,88,90,
  # 2025 (sampai November)
  92,95,98,101,104,107,110,113,116,119,122
)

ts_zeta <- ts(zeta, start = c(2022,1), frequency = 12)

print(ts_zeta)
plot(ts_zeta, main = "Penjualan Smartphone Zeta", ylab = "Unit (ribu)", xlab
     = "Tahun")
```

**Pembahasan :** Kode tersebut digunakan untuk membuat deret waktu penjualan smartphone Zeta dari tahun 2022 hingga November 2025, kemudian menampilkannya dalam bentuk grafik. Data penjualan bulanan pertama dimasukkan ke dalam vektor zeta, kemudian diubah menjadi *time series* menggunakan fungsi `ts()` dengan titik awal Januari 2022 dan frekuensi 12 untuk data bulanan. Objek `ts_zeta` kemudian dicetak menggunakan `print()` agar bentuk deret waktunya terlihat secara numerik. Setelah itu, fungsi `plot()` digunakan untuk menggambarkan pergerakan penjualan smartphone Zeta dari waktu ke waktu, lengkap dengan judul grafik serta label sumbu X dan Y sehingga memudahkan analisis visual terhadap tren penjualan.

### Output



**Pembahasan Output :** Output dari kode tersebut terdiri dari dua bagian, yaitu deret waktu numerik pada konsol dan grafik penjualan dalam bentuk kurva. Deret waktu yang dicetak menampilkan angka penjualan setiap bulan dari 2022 hingga November 2025, sehingga memudahkan pemeriksaan data secara langsung. Sementara itu, grafik yang

ditampilkan menunjukkan pola peningkatan penjualan smartphone Zeta yang cenderung stabil dan bertahap dari tahun ke tahun. Dengan visualisasi ini, pengguna dapat melihat adanya tren naik yang konsisten, mencerminkan pertumbuhan permintaan atau performa pasar yang positif terhadap produk Zeta sepanjang periode tersebut.

```
train <- window(ts_zeta, end = c(2025,11)) # sampai November 2025

# Create a placeholder time series for the forecast period (December 2025)
# This 'test' will contain NA values, as actual data is not available yet.
test_start_year <- 2025
test_start_month <- 12
test_length <- 1 # Forecasting for one month
test <- ts(rep(NA, test_length), start = c(test_start_year,
      test_start_month), frequency = 12)
```

**Pembahasan :** Kode tersebut digunakan untuk memisahkan data penjualan smartphone Zeta menjadi bagian *training* dan periode *forecast* yang belum memiliki data aktual. Bagian training diambil menggunakan window() hingga November 2025, sehingga model nantinya hanya akan belajar dari data historis yang tersedia. Karena penjualan bulan Desember 2025 belum diketahui, dibuatlah sebuah objek time series bernama test yang berisi nilai NA sebagai placeholder. Time series ini dibuat dengan fungsi ts(), dimulai pada Desember 2025 dengan frekuensi bulanan, dan panjangnya satu bulan karena hanya ingin memprediksi Desember saja. Struktur ini penting dalam forecasting karena memungkinkan pemodelan dan perhitungan prediksi meskipun data aktual untuk bulan yang akan diprediksi belum tersedia.

```
weights <- c(1,2,3)
k <- length(weights)

# Ambil 3 data terakhir untuk forecasting
last3 <- tail(train, 3)

forecast_wma <- sum(last3 * weights) / sum(weights)

cat("Forecast WMA Desember 2025 =", forecast_wma, "ribu unit\n")
```

**Pembahasan :** Kode tersebut digunakan untuk melakukan peramalan penjualan smartphone Zeta bulan Desember 2025 menggunakan metode Weighted Moving Average (WMA) dengan bobot tiga periode terakhir. Pertama, vektor weights berisi bobot 1, 2, dan 3, di mana bobot terbesar diberikan pada data terbaru agar pengaruhnya lebih dominan. Nilai k dihitung untuk mengetahui jumlah bobot yang digunakan. Kemudian, fungsi tail(train, 3) mengambil tiga data terakhir dari deret waktu penjualan sebagai dasar perhitungan. Nilai forecast dihitung menggunakan rumus WMA, yaitu hasil penjumlahan perkalian antara masing-masing data dan bobotnya dibagi dengan total bobot. Terakhir, hasil prediksi ditampilkan menggunakan cat() dalam satuan ribu unit.

## Output

```
... Forecast WMA Desember 2025 = 120 ribu unit
```

**Pembahasan Output :** Output dari kode ini berupa satu nilai forecast penjualan bulan Desember 2025 menggunakan metode WMA. Angka yang ditampilkan merupakan gabungan berbobot dari tiga bulan terakhir (September, Oktober, dan November 2025), dengan bobot terbesar diberikan pada bulan November sebagai data paling baru. Hasil ini memberikan estimasi penjualan Zeta untuk bulan Desember 2025 dalam ribuan unit, yang dapat digunakan sebagai dasar perencanaan produksi, stok, atau strategi pemasaran. Karena WMA lebih sensitif terhadap data terbaru, hasil forecast biasanya mengikuti tren terkini secara lebih akurat dibandingkan Simple Moving Average.

### C. PEMBAHASAN TUGAS

```
# --- LANGKAH 1: Instalasi & Memuat Library ---
if(!require(forecast)) install.packages("forecast")
if(!require(ggplot2)) install.packages("ggplot2")
if(!require(readr)) install.packages("readr")

library(forecast)
library(ggplot2)
library(readr)

# --- LANGKAH 2: Memuat Dataset ---
url <- "https://raw.githubusercontent.com/jbrownlee/Datasets/master/monthly-car-sales.csv"
data <- read_csv(url, show_col_types = FALSE)

# Menampilkan 5 data teratas untuk memastikan data masuk
print("Data Awal:")
head(data)

# --- LANGKAH 3: Konversi ke Time Series (ts) Object ---
ts_data <- ts(data$Sales, start = c(1960, 1), frequency = 12)

# --- LANGKAH 4: Forecasting dengan SES (Exponential Smoothing) --
# h = 12 artinya kita meramal 12 bulan ke depan.
fit_ses <- ses(ts_data, h = 12)

# Melihat ringkasan model (termasuk nilai alpha yang optimal)
print("Ringkasan Model SES:")
summary(fit_ses)

# LANGKAH 5: Visualisasi
# Membuat plot data aktual vs hasil forecasting
autoplot(fit_ses) +
  autolayer(fitted(fit_ses), series="Fitted (EMA)") +
  ggtitle("Forecasting Penjualan Mobil Bulanan (Metode SES/EMA)") +
  xlab("Tahun") +
  ylab("Penjualan") +
  theme_minimal()

# LANGKAH 6: Menampilkan Nilai Hasil Forecast
print("Hasil Forecasting 12 Bulan ke Depan:")
print(fit_ses)
```

## Output

```
*** [1] "Data Awal:"
      A tibble: 6 × 2
      Month Sales
      <chr>   <dbl>
1 1960-01   6550
2 1960-02   8728
3 1960-03  12026
4 1960-04  14395
5 1960-05  14587
6 1960-06  13791
```

Pembahasan : menampilkan enam baris pertama dataset untuk memastikan data telah dimuat dengan benar. Terlihat bahwa dataset memiliki dua kolom utama yaitu Month berisi informasi waktu dalam format teks dan Sales berisi angka penjualan mobil.

```
*** [1] "Ringkasan Model SES:"

Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
ses(y = ts_data, h = 12)

Smoothing parameters:
alpha = 0.9999

Initial states:
l = 6549.187

sigma: 3329.249

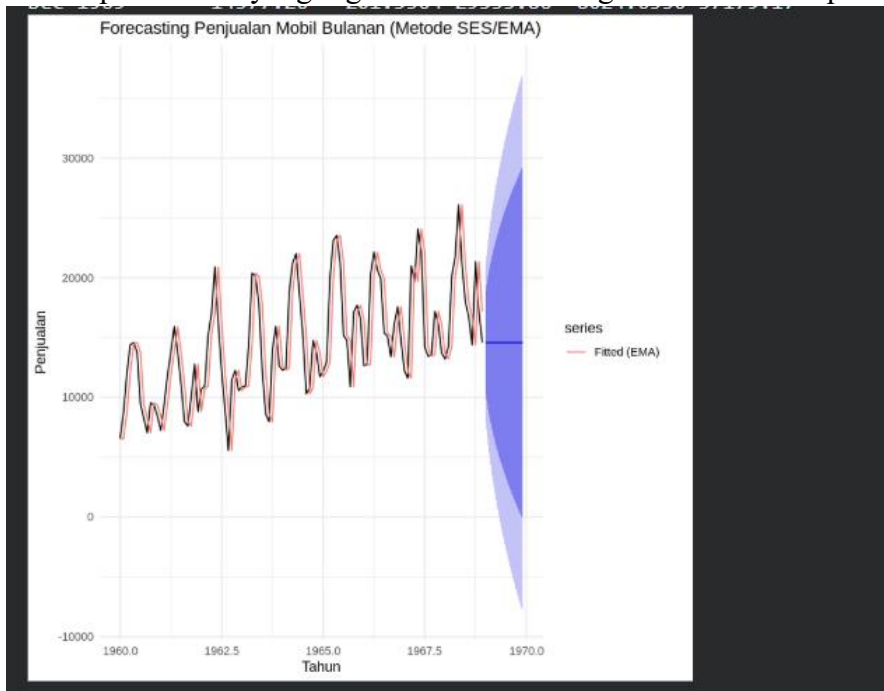
AIC      AICc      BIC
2261.520 2261.751 2269.566

Error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set 74.34145 3298.279 2605.381 -1.99231 18.48899 1.633446 0.1559908

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
Jan 1969      14577.26 10310.6558 18843.86 8052.0518 21102.47
Feb 1969      14577.26 8543.6721 20610.85 5349.6833 23804.84
Mar 1969      14577.26 7187.7772 21966.74 3276.0211 25878.50
Apr 1969      14577.26 6044.6913 23109.83 1527.8221 27626.70
May 1969      14577.26 5037.6059 24116.91 -12.3822 29166.90
Jun 1969      14577.26 4127.1273 25027.39 -1404.8390 30559.36
Jul 1969      14577.26 3289.8534 25864.67 -2685.3389 31839.86
Aug 1969      14577.26 2510.5363 26643.98 -3877.2015 33031.72
Sep 1969      14577.26 1778.5846 27375.94 -4996.6251 34151.15
Oct 1969      14577.26 1086.2865 28068.23 -6055.4036 35209.92
Nov 1969      14577.26 427.8205 28726.70 -7062.4405 36216.96
Dec 1969      14577.26 -201.3364 29355.86 -8024.6530 37179.17
[1] "Hasil Forecasting 12 Bulan ke Depan:"
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
Jan 1969      14577.26 10310.6558 18843.86 8052.0518 21102.47
Feb 1969      14577.26 8543.6721 20610.85 5349.6833 23804.84
Mar 1969      14577.26 7187.7772 21966.74 3276.0211 25878.50
Apr 1969      14577.26 6044.6913 23109.83 1527.8221 27626.70
May 1969      14577.26 5037.6059 24116.91 -12.3822 29166.90
Jun 1969      14577.26 4127.1273 25027.39 -1404.8390 30559.36
Jul 1969      14577.26 3289.8534 25864.67 -2685.3389 31839.86
Aug 1969      14577.26 2510.5363 26643.98 -3877.2015 33031.72
Sep 1969      14577.26 1778.5846 27375.94 -4996.6251 34151.15
Oct 1969      14577.26 1086.2865 28068.23 -6055.4036 35209.92
Nov 1969      14577.26 427.8205 28726.70 -7062.4405 36216.96
Dec 1969      14577.26 -201.3364 29355.86 -8024.6530 37179.17
```

Pembahasan : Output ini menampilkan ringkasan model Simple Exponential Smoothing di mana parameter smoothing alpha yang dihasilkan sangat tinggi yaitu 0.9999. Nilai alpha yang mendekati satu mengindikasikan bahwa model memberikan

bobot terbesar pada data pengamatan paling akhir sehingga hasil prediksi sangat dipengaruhi oleh fluktuasi terkini dibandingkan pola historis jangka panjang. Bagian error measures seperti RMSE dan MAPE menunjukkan tingkat kesalahan model terhadap data latih yang digunakan untuk mengevaluasi akurasi peramalan.



Pembahasan : Output ini menampilkan hasil prediksi numerik untuk tahun 1969 di mana nilai Point Forecast terlihat konstan pada angka 14577.26 untuk setiap bulannya. Pola data ini mengonfirmasi sifat metode Simple Exponential Smoothing yang menghasilkan ramalan garis lurus (flat) berdasarkan level data terakhir yang telah dihaluskan. Namun, kolom interval kepercayaan (Lo dan Hi) menunjukkan rentang yang semakin melebar dari Januari hingga Desember, yang mengindikasikan bahwa tingkat ketidakpastian prediksi semakin tinggi seiring bertambahnya jarak waktu peramalan.

#### D. KESIMPULAN

Forecasting menggunakan model Moving Average (MA), baik Simple Moving Average (SMA), Weighted Moving Average (WMA), maupun Exponential Moving Average (EMA), merupakan metode yang efektif untuk memprediksi data deret waktu yang memiliki pola relatif stabil atau tren yang tidak terlalu fluktuatif. SMA memberikan gambaran rata-rata sederhana dari beberapa periode sebelumnya, sementara WMA dan EMA memberikan penekanan lebih besar pada data terbaru sehingga lebih responsif terhadap perubahan tren. Melalui proses pemisahan data, perhitungan nilai rata-rata, dan evaluasi menggunakan metrik seperti RMSE, MAE, dan MAPE, model Moving Average mampu memberikan estimasi yang cukup baik untuk periode mendatang. Secara keseluruhan, metode ini mudah diterapkan, mudah dipahami, dan cocok digunakan sebagai pendekatan awal dalam analisis peramalan sebelum beralih ke model yang lebih kompleks.