

**PRAKTIKUM PEMODELAN STATISTIKA**  
**MODUL 9**



**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 9

### FORECASTING DENGAN MODEL NAIVE

#### A. TUJUAN PRAKTIKUM

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

#### B. PEMBAHASAN LISTING

##### PRAKTIK

##### Kasus 1 :

Forecasting dg Simple Naïve

Berikut data penjualan bulanan mie instan (dalam ribuan dus) tahun 2022–2024. Akan dilakukan prediksi terhadap penjualan

##### 1. Install & load package yang diperlukan

```
# 1. Install & load package yang diperlukan
install.packages("forecast")
library(forecast)
library(ggplot2)
library(tseries)
```

**Pembahasan :** Kode ini digunakan untuk menyiapkan lingkungan kerja R dengan memanggil paket-paket yang dibutuhkan dalam analisis runtun waktu. Perintah `install.packages("forecast")` berfungsi untuk menginstal paket **forecast** jika belum tersedia di sistem. Selanjutnya, `library(forecast)` digunakan untuk memuat paket tersebut yang menyediakan berbagai fungsi peramalan dan analisis deret waktu seperti ARIMA dan plot diagnostik. `library(ggplot2)` dipanggil untuk keperluan visualisasi data yang lebih fleksibel dan informatif, sedangkan `library(tseries)` digunakan untuk analisis statistik deret waktu, khususnya uji stasioneritas seperti ADF dan KPSS.

##### Output

-

##### Pembahasan Output :

##### 2. Data penjualan

```
# 2. Data penjualan
bulan <- c("Jan-22", "Feb-22", "Mar-22", "Apr-22", "Mei-22", "Jun-22",
           "Jul-22", "Agu-22", "Sep-22", "Okt-22", "Nov-22", "Des-22",
           "Jan-23", "Feb-23", "Mar-23", "Apr-23", "Mei-23", "Jun-23",
           "Jul-23", "Agu-23", "Sep-23", "Okt-23", "Nov-23", "Des-23",
           "Jan-24", "Feb-24", "Mar-24", "Apr-24", "Mei-24", "Jun-24",
           "Jul-24", "Agu-24")

penjualan <- c(120, 125, 130, 128, 135, 140, 138, 142, 145, 150, 148, 155,
              160, 158, 165, 170, 168, 172, 175, 180, 178, 185, 182, 190,
              195, 192, 200, 205, 208, 210, 215, 220)
```

**Pembahasan :** Kode ini digunakan untuk menyiapkan data penjualan dalam bentuk vektor yang akan dianalisis sebagai data runtun waktu. Variabel `bulan` berisi label waktu penjualan dalam satuan bulan, dimulai dari Januari 2022 hingga Agustus 2024. Sementara itu, variabel `penjualan` berisi data numerik yang merepresentasikan jumlah penjualan pada setiap bulan yang bersesuaian. Kedua vektor ini disusun dengan urutan yang sama

agar setiap nilai penjualan tepat mewakili periode bulan tertentu, sehingga dapat digunakan pada tahap analisis, visualisasi, atau pemodelan runtun waktu selanjutnya.

## Output

-

## Pembahasan Output :

### 3. Data dijadikan time series (ts)

```
# 3. Data dijadikan time series (ts)
data_ts <- ts(penjualan, start = c(2022, 1), frequency = 12)
print(data_ts)
```

**Pembahasan :** Kode `ts(penjualan, start = c(2022, 1), frequency = 12)` digunakan untuk mengubah data penjualan bulanan menjadi objek deret waktu (*time series*) di R. Argumen `start = c(2022, 1)` menunjukkan bahwa data dimulai pada Januari 2022, sedangkan `frequency = 12` menandakan bahwa data dicatat setiap bulan dalam satu tahun. Objek deret waktu ini kemudian disimpan dalam variabel `data_ts` dan ditampilkan ke layar menggunakan `print(data_ts)`.

## Output

```
***      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2022 120 125 130 128 135 140 138 142 145 150 148 155
2023 160 158 165 170 168 172 175 180 178 185 182 190
2024 195 192 200 205 208 210 215 220
```

**Pembahasan Output :** Output `print(data_ts)` menampilkan data penjualan dalam format deret waktu, di mana setiap nilai penjualan ditampilkan sesuai dengan periode waktunya (bulan dan tahun). Tampilan ini memudahkan pengguna untuk memastikan bahwa data telah berhasil dikonversi menjadi objek *time series* dengan struktur waktu yang benar, sehingga siap digunakan untuk visualisasi, analisis tren, uji stasioneritas, maupun peramalan penjualan pada tahap selanjutnya.

### 4. Forecasting dengan Metode Naive (random walk)

```
# 4. Forecasting dengan Metode Naive (random walk)
naive_forecast <- naive(data_ts, h = 12)    # ramal 12 bulan ke depan

# Lihat hasil ramalan
summary(naive_forecast)
```

**Pembahasan :** Kode ini digunakan untuk melakukan peramalan penjualan menggunakan metode **Naive**, yaitu metode peramalan sederhana yang mengasumsikan bahwa nilai pada periode terakhir akan menjadi prediksi untuk periode-periode selanjutnya (*random walk*). Fungsi `naive(data_ts, h = 12)` dari paket `forecast` digunakan untuk meramalkan data penjualan selama 12 bulan ke depan, di mana `h = 12` menyatakan horizon peramalan. Hasil peramalan kemudian disimpan dalam objek `naive_forecast`, dan fungsi `summary(naive_forecast)` digunakan untuk menampilkan ringkasan hasil peramalan tersebut.

## Output

```

*** Forecast method: Naive method

Model Information:
Call: naive(y = data_ts, h = 12)

Residual sd: 4.7655

Error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set 3.225806 4.765467 4.387097 1.912932 2.632537 0.1211905 -0.5157798

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
Sep 2024          220 213.8928 226.1072 210.6599 229.3401
Oct 2024          220 211.3631 228.6369 206.7910 233.2090
Nov 2024          220 209.4220 230.5780 203.8224 236.1776
Dec 2024          220 207.7856 232.2144 201.3197 238.6803
Jan 2025          220 206.3439 233.6561 199.1148 240.8852
Feb 2025          220 205.0405 234.9595 197.1214 242.8786
Mar 2025          220 203.8419 236.1581 195.2883 244.7117
Apr 2025          220 202.7263 237.2737 193.5821 246.4179
May 2025          220 201.6784 238.3216 191.9796 248.0204
Jun 2025          220 200.6874 239.3126 190.4639 249.5361
Jul 2025          220 199.7447 240.2553 189.0222 250.9778
Aug 2025          220 198.8441 241.1559 187.6448 252.3552

```

**Pembahasan Output :** Output dari `summary(naive_forecast)` menampilkan informasi penting mengenai hasil peramalan, seperti nilai ramalan (*forecast*), interval kepercayaan (biasanya 80% dan 95%), serta metode yang digunakan yaitu *Naive method*. Pada metode ini, semua nilai ramalan untuk 12 bulan ke depan umumnya sama dengan nilai penjualan terakhir pada data historis. Interval kepercayaan menunjukkan rentang ketidakpastian ramalan, yang semakin melebar seiring bertambahnya periode ramalan. Hasil ini memberikan gambaran awal yang sederhana mengenai proyeksi penjualan dan dapat dijadikan pembandingan dengan metode peramalan yang lebih kompleks.

## 5. Visualisasi Hasil Forecasting

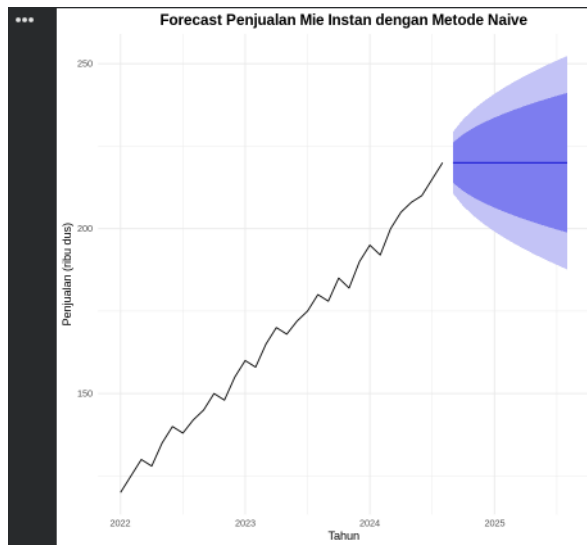
```

# 5. Visualisasi Hasil Forecasting
autoplot(naive_forecast) +
  ggtitle("Forecast Penjualan Mie Instan dengan Metode Naive") +
  xlab("Tahun") + ylab("Penjualan (ribu dus)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"))

```

**Pembahasan :** Kode ini digunakan untuk memvisualisasikan hasil peramalan penjualan mie instan menggunakan metode **Naive**. Fungsi `autoplot(naive_forecast)` secara otomatis membuat grafik deret waktu yang menampilkan data historis, nilai ramalan, serta interval kepercayaan dari objek hasil peramalan. Fungsi `ggtitle()`, `xlab()`, dan `ylab()` digunakan untuk menambahkan judul grafik serta label pada sumbu X dan Y. Selanjutnya, `theme_minimal()` diterapkan untuk memberikan tampilan grafik yang lebih sederhana dan bersih, dan pengaturan `theme()` digunakan untuk memusatkan judul grafik serta menyesuaikan ukuran dan ketebalan teks judul agar lebih informatif.

### Output



**Pembahasan Output :** Output yang dihasilkan berupa grafik peramalan penjualan mie instan. Grafik tersebut menampilkan data penjualan historis hingga periode terakhir, diikuti oleh garis ramalan selama 12 bulan ke depan yang bersifat datar, sesuai dengan karakteristik metode Naive yang menggunakan nilai terakhir sebagai prediksi. Selain itu, terlihat area bayangan di sekitar garis ramalan yang menunjukkan interval kepercayaan, di mana rentangnya semakin lebar seiring bertambahnya periode ramalan. Visualisasi ini membantu memahami pola ramalan secara intuitif serta tingkat ketidakpastian hasil peramalan yang dihasilkan oleh metode Naive.

## Praktik 2 :

Forecasting Naive with Drift

Berikut ini data harga saham PT Bank Central Asia Tbk (BBCA.JK) dari Januari 2022 sampai Juni 2025 (bulanan, closing price).

Harga saham biasanya mengandung tren + random walk → sangat cocok untuk Naive with Drift..

### 1. Install & load packages

```
# 1. Install & load packages
suppressMessages({
  install.packages(c("forecast", "ggplot2", "dplyr", "writexl", "lubridate",
"scales"))
  library(forecast)
  library(ggplot2)
  library(dplyr)
  library(writexl)
  library(lubridate)
  library(scales)
})
```

**Pembahasan :** Kode ini digunakan untuk menyiapkan lingkungan kerja R dengan menginstal dan memuat beberapa paket yang diperlukan untuk analisis dan visualisasi data runtun waktu. Fungsi `install.packages(c(...))` menginstal paket **forecast** (peramalan deret waktu), **ggplot2** (visualisasi), **dplyr** (manipulasi data), **writexl** (ekspor data ke Excel), **lubridate** (manipulasi tanggal/waktu), dan **scales** (format sumbu grafik). Fungsi `library()` memuat paket-paket tersebut agar fungsinya bisa digunakan. Seluruh proses

dibungkus dengan `suppressMessages({ ... })` untuk menonaktifkan pesan notifikasi atau peringatan saat instalasi dan pemanggilan library, sehingga output lebih rapi.

## Output

-

## Pembahasan Output :

### 2. Input data harga saham BBKA (akhir bulan)

```
# 2. Input data harga saham BBKA (akhir bulan)
harga_saham <- c(7225, 7775, 8050, 7525, 7800, 8350, 8575, 8800, 9125, 9450,
9275, 9600,
                9950, 9700, 10100, 10750, 10450, 10900, 11250, 11800, 11500,
12100,
                11950, 12500, 13000, 12850, 13400, 13950, 14200, 14750,
15200, 15800, 16200,
                16800, 16500, 17200, 17800, 17500, 18100, 18600, 18900,
19400)

# Jadikan time series (bulanan)
bbca_ts <- ts(harga_saham, start = c(2022, 1), frequency = 12)

# 3. Forecasting Naive with Drift (12 bulan ke depan)
forecast_drift <- rwf(bbca_ts, h = 12, drift = TRUE, level = c(80, 95))
summary(forecast_drift)
```

**Pembahasan :** Kode ini digunakan untuk melakukan peramalan harga saham BBKA menggunakan metode **Random Walk with Drift**. Pertama, vektor `harga_saham` berisi data harga saham BBKA pada akhir bulan selama beberapa tahun dimasukkan. Kemudian, fungsi `ts(harga_saham, start = c(2022, 1), frequency = 12)` mengubah data menjadi objek *time series* bulanan (`bbca_ts`). Selanjutnya, fungsi `rwf(bbca_ts, h = 12, drift = TRUE, level = c(80, 95))` digunakan untuk meramalkan 12 bulan ke depan dengan mempertimbangkan *drift*, yaitu rata-rata perubahan per periode, dan memberikan interval kepercayaan 80% dan 95%. Hasil peramalan disimpan dalam objek `forecast_drift`, dan fungsi `summary(forecast_drift)` menampilkan ringkasan nilai ramalan.

## Output

```
*** Forecast method: Random walk with drift

Model Information:
Call: rwf(y = bbca_ts, h = 12, drift = TRUE, level = c(80, 95))

Drift: 296.9512 (se 51.9747)
Residual sd: 332.8004

Error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 3.549122e-13 328.7168 261.3325 -0.2086509 2.198289 0.07364938
ACF1
Training set -0.3357349

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
Jul 2025    19696.95 19265.28 20128.62 19036.77 20357.13
Aug 2025    19993.90 19376.20 20611.60 19049.21 20938.59
Sep 2025    20290.85 19525.58 21056.12 19120.47 21461.23
Oct 2025    20587.80 19694.16 21481.45 19221.10 21954.51
Nov 2025    20884.76 19874.59 21894.92 19339.84 22429.67
Dec 2025    21181.71 20063.16 22300.25 19471.04 22892.37
Jan 2026    21478.66 20257.71 22699.61 19611.38 23345.94
Feb 2026    21775.61 20456.83 23094.39 19758.72 23792.50
Mar 2026    22072.56 20659.59 23485.54 19911.60 24233.52
Apr 2026    22369.51 20865.29 23873.74 20069.00 24670.03
May 2026    22666.46 21073.43 24259.50 20230.12 25102.80
Jun 2026    22963.41 21283.62 24643.21 20394.39 25532.44
```

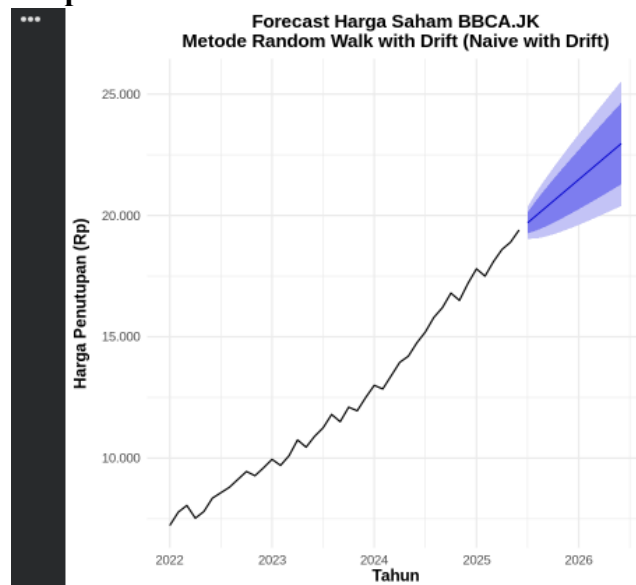
**Pembahasan Output :** Output dari summary(forecast\_drift) menampilkan nilai ramalan harga saham untuk 12 bulan ke depan beserta interval kepercayaan 80% dan 95%. Metode Random Walk with Drift membuat ramalan meningkat secara linear mengikuti rata-rata kenaikan bulanan pada data historis. Interval kepercayaan menunjukkan tingkat ketidakpastian ramalan, di mana rentang semakin melebar seiring bertambahnya periode ramalan. Ringkasan ini memberikan gambaran awal mengenai arah pergerakan harga saham BBKA dan risiko ketidakpastian yang terkait, serta dapat digunakan sebagai dasar pembandingan dengan metode peramalan lainnya.

#### 4. Visualisasi

```
# 4. Visualisasi
autoplot(forecast_drift) +
  ggtitle("Forecast Harga Saham BBKA.JK\nMetode Random Walk with Drift (Naive with Drift)") +
  xlab("Tahun") + ylab("Harga Penutupan (Rp)") +
  scale_y_continuous(labels = comma_format(big.mark = ".", decimal.mark = ",")) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
        axis.title = element_text(face = "bold"))
```

**Pembahasan :** Kode ini digunakan untuk memvisualisasikan hasil peramalan harga saham BBKA menggunakan metode **Random Walk with Drift**. Fungsi autoplot(forecast\_drift) membuat grafik otomatis yang menampilkan data historis, nilai ramalan, serta interval kepercayaan. Fungsi ggtitle(), xlab(), dan ylab() menambahkan judul dan label sumbu grafik. scale\_y\_continuous(labels = comma\_format(...)) memformat angka sumbu Y agar lebih mudah dibaca dengan tanda ribuan. theme\_minimal(base\_size = 14) memberikan tampilan yang bersih dan sederhana, sedangkan theme() menyesuaikan posisi, ukuran, dan ketebalan teks judul serta label sumbu agar grafik lebih informatif dan menarik.

#### Output



**Pembahasan Output :** Output berupa grafik runtun waktu harga saham BBKA beserta proyeksi 12 bulan ke depan. Garis utama menunjukkan data historis, sedangkan garis ramalan menunjukkan prediksi harga saham dengan tren linear mengikuti drift. Area bayangan di sekitar garis ramalan menunjukkan interval kepercayaan 80% dan 95%, yang semakin melebar seiring

bertambahnya periode ramalan, menggambarkan ketidakpastian prediksi. Visualisasi ini memudahkan pemahaman arah pergerakan harga saham dan tingkat risiko ketidakpastian dalam jangka waktu ramalan.

### Praktik 3 :

Forecasting Seasonal Naïve

Penjualan Ritel di Indonesia (2019–2025)

(Data terinspirasi dari BPS — ada lonjakan besar tiap Desember & Lebaran).

Akan dilakukan forecasting menggunakan metode Seasonal Naïve.

#### 1. Install & load packages

```
# 1. Install & load packages
suppressMessages({
  install.packages(c("forecast", "ggplot2", "dplyr", "writexl", "lubridate",
"scale"))
  library(forecast)
  library(ggplot2)
  library(dplyr)
  library(writexl)
  library(lubridate)
  library(scales)
})
```

**Pembahasan :** Kode ini menyiapkan lingkungan kerja R untuk analisis runtun waktu dan manipulasi data. Perintah `install.packages(c(...))` menginstal paket **forecast** (peramalan deret waktu), **ggplot2** (visualisasi), **dplyr** (manipulasi data), **writexl** (ekspor ke Excel), **lubridate** (manipulasi tanggal/waktu), dan **scales** (format sumbu grafik). Fungsi `library()` memuat paket-paket tersebut agar fungsi-fungsinya dapat digunakan. Seluruh proses dibungkus dalam `suppressMessages({ ... })` untuk menonaktifkan pesan notifikasi dan membuat output lebih rapi.

#### Output

-

#### Pembahasan Output :

#### 2. Data Indeks Penjualan Ritel (Jan 2019 – Jun 2025)

```
# 2. Data Indeks Penjualan Ritel (Jan 2019 – Jun 2025)
# Ada pola musiman kuat: naik tajam tiap Desember & saat Lebaran (Apr-Mei)
penjualan_ritel <- c(
  # 2019
  185, 188, 192, 205, 198, 195, 202, 200, 208, 215, 220, 285, # Des lonjakan
  # 2020
  178, 175, 180, 168, 172, 180, 188, 192, 198, 205, 210, 275,
  # 2021
  182, 185, 190, 195, 202, 198, 205, 208, 212, 218, 225, 292,
  # 2022
  195, 198, 202, 208, 215, 210, 218, 220, 225, 232, 240, 305,
  # 2023
  210, 212, 218, 225, 238, 232, 240, 245, 250, 258, 265, 325,
  # 2024
  225, 228, 235, 242, 255, 248, 258, 262, 268, 275, 282, 340,
  # 2025 (sampai Juni)
  240, 242, 248, 255, 268, 262
)
```



```
# Jadikan time series bulanan
ritel_ts <- ts(penjualan_ritel, start = c(2019, 1), frequency = 12)
```

**Pembahasan :** Kode ini digunakan untuk menyiapkan data indeks penjualan ritel dari Januari 2019 hingga Juni 2025 sebagai objek deret waktu (*time series*). Vektor penjualan\_ritel berisi nilai penjualan bulanan, di mana terlihat pola musiman yang kuat, seperti lonjakan pada Desember dan saat Lebaran (April–Mei). Fungsi ts() digunakan untuk mengubah data numerik ini menjadi objek deret waktu, dengan argumen start = c(2019, 1) menunjukkan periode awal Januari 2019 dan frequency = 12 menunjukkan bahwa data dicatat setiap bulan. Objek ini kemudian disimpan dalam variabel ritel\_ts untuk analisis lebih lanjut.

## Output

-

## Pembahasan Output :

### 3. Forecasting Seasonal Naïve (12 bulan ke depan: Jul 2025 – Jun 2026)

```
# 3. Forecasting Seasonal Naïve (12 bulan ke depan: Jul 2025 – Jun 2026)
seasonal_naive <- snaive(ritel_ts, h = 12)
summary(seasonal_naive)
```

**Pembahasan :** Kode ini digunakan untuk melakukan peramalan penjualan ritel menggunakan metode **Seasonal Naïve**, yang mengasumsikan bahwa setiap nilai pada bulan tertentu akan sama dengan nilai bulan yang sama pada tahun sebelumnya. Fungsi snaive(ritel\_ts, h = 12) meramalkan 12 bulan ke depan (Juli 2025 hingga Juni 2026) dengan mempertahankan pola musiman historis. Hasil peramalan disimpan dalam variabel seasonal\_naive, dan fungsi summary(seasonal\_naive) digunakan untuk menampilkan ringkasan hasil peramalan, termasuk nilai ramalan, interval kepercayaan, dan metode yang digunakan.

## Output

```
*** Forecast method: Seasonal naive method

Model Information:
Call: snaive(y = ritel_ts, h = 12)

Residual sd: 16.8788

Error measures:
      ME    RMSE    MAE    MPE    MAPE  MASE    ACF1
Training set 10.71212 16.8788 15.92424 4.279376 7.110938 1 0.9059832

Forecasts:
      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jul 2025      258 236.3689 279.6311 224.9182 291.0818
Aug 2025      262 240.3689 283.6311 228.9182 295.0818
Sep 2025      268 246.3689 289.6311 234.9182 301.0818
Oct 2025      275 253.3689 296.6311 241.9182 308.0818
Nov 2025      282 260.3689 303.6311 248.9182 315.0818
Dec 2025      340 318.3689 361.6311 306.9182 373.0818
Jan 2026      240 218.3689 261.6311 206.9182 273.0818
Feb 2026      242 220.3689 263.6311 208.9182 275.0818
Mar 2026      248 226.3689 269.6311 214.9182 281.0818
Apr 2026      255 233.3689 276.6311 221.9182 288.0818
May 2026      268 246.3689 289.6311 234.9182 301.0818
Jun 2026      262 240.3689 283.6311 228.9182 295.0818
```

**Pembahasan Output :** Output dari summary(seasonal\_naive) menampilkan nilai ramalan penjualan untuk 12 bulan mendatang, beserta interval kepercayaan 80% dan 95%. Nilai ramalan pada setiap bulan akan mengikuti pola musiman yang sama dengan bulan yang bersangkutan pada tahun sebelumnya. Interval kepercayaan menunjukkan tingkat ketidakpastian ramalan, yang cenderung melebar untuk bulan-bulan yang lebih jauh dari data historis. Hasil ini memberikan gambaran prediksi penjualan ritel sambil

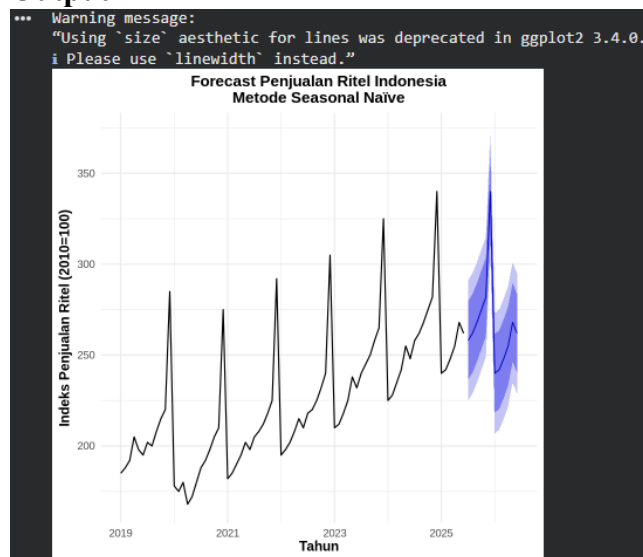
mempertahankan pola musiman yang sudah ada, sehingga dapat digunakan sebagai dasar perencanaan stok dan strategi pemasaran.

#### 4. Visualisasi pola musiman

```
# 4. Visualisasi pola musiman
autoplot(seasonal_naive) +
  ggtitle("Forecast Penjualan Ritel Indonesia\nMetode Seasonal Naïve") +
  xlab("Tahun") +
  ylab("Indeks Penjualan Ritel (2010=100)") +
  scale_y_continuous(labels = comma) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
        axis.title = element_text(face = "bold")) +
  geom_line(color = "steelblue", size = 1) +
  geom_point(color = "steelblue")
```

**Pembahasan :** Kode ini digunakan untuk memvisualisasikan hasil peramalan penjualan ritel Indonesia menggunakan metode **Seasonal Naïve**. Fungsi `autoplot(seasonal_naive)` menampilkan grafik data historis, ramalan, dan interval kepercayaan secara otomatis. Fungsi `ggtitle()`, `xlab()`, dan `ylab()` menambahkan judul dan label sumbu. `scale_y_continuous(labels = comma)` memformat angka sumbu Y agar lebih mudah dibaca. `theme_minimal()` memberikan tampilan bersih, sedangkan pengaturan `theme()` menyesuaikan posisi dan format teks judul dan label. `geom_line()` dan `geom_point()` menambahkan garis dan titik untuk menonjolkan pola data historis dan ramalan.

#### Output



**Pembahasan Output :** Output berupa grafik runtun waktu penjualan ritel beserta prediksi 12 bulan ke depan (Juli 2025–Juni 2026). Grafik menunjukkan data historis, garis ramalan yang mengikuti pola musiman tahunan, serta interval kepercayaan 80% dan 95%. Lonjakan musiman terlihat jelas, misalnya pada bulan Desember dan saat Lebaran, yang dipertahankan dalam ramalan. Visualisasi ini memudahkan pemahaman pola musiman, tren, dan ketidakpastian dalam prediksi, sehingga membantu perencanaan penjualan dan strategi bisnis.

## LATIHAN

### Kasus 1 :

Lakukan peramalan penjualan menggunakan metode naïve (simple naïve) untuk data penjualan produk minuman energi “X-Boost” (dalam karton) Tahun 2023–2025.

#### 1. Install & load package

```
suppressMessages({  
  if(!require(forecast)) install.packages("forecast")  
  if(!require(ggplot2)) install.packages("ggplot2")  
  library(forecast)  
  library(ggplot2)  
})
```

**Pembahasan :** Kode ini digunakan untuk menyiapkan lingkungan R agar siap melakukan analisis deret waktu dan visualisasi. Fungsi `if(!require(...)) install.packages(...)` memastikan bahwa paket **forecast** (untuk peramalan deret waktu) dan **ggplot2** (untuk visualisasi) sudah terinstal; jika belum, paket akan diinstal otomatis. Perintah `library()` memuat paket tersebut agar semua fungsi di dalamnya dapat digunakan. Seluruh perintah dibungkus dengan `suppressMessages({ ... })` agar pesan notifikasi atau peringatan tidak ditampilkan, sehingga output lebih bersih dan rapi.

#### Output

-

#### Pembahasan Output :

#### 2. Data Penjualan Minuman Energi X-Boost

```
# Data dari Jan 2023 s/d Mar 2025  
penjualan_xboost <- c(  
  # 2023 (Jan-Des)  
  210, 195, 220, 205, 218, 230, 225, 240, 235, 228, 215, 245,  
  # 2024 (Jan-Des)  
  228, 212, 240, 222, 235, 248, 242, 260, 255, 250, 232, 268,  
  # 2025 (Jan-Mar)  
  245, 230, 258  
)
```

**Pembahasan :** Kode ini digunakan untuk mendefinisikan data penjualan bulanan dari Januari 2023 hingga Maret 2025, yang nantinya bisa digunakan untuk analisis atau peramalan menggunakan metode XGBoost atau metode lainnya. Vektor `penjualan_xboost` berisi angka penjualan tiap bulan secara berurutan, dimulai dari Januari 2023 hingga Maret 2025, sehingga setiap nilai mewakili periode waktu yang konsisten. Data ini masih berbentuk vektor numerik sederhana dan bisa diubah menjadi objek *time series* jika ingin dianalisis sebagai runtun waktu.

#### Output

-

#### Pembahasan Output :

#### 3. Data dijadikan time series (ts)

```
xboost_ts <- ts(penjualan_xboost, start = c(2023, 1), frequency = 12)  
print(xboost_ts)
```

**Pembahasan :** Kode ini digunakan untuk mengubah vektor `penjualan_xboost` menjadi objek *time series* bulanan di R. Fungsi `ts(penjualan_xboost, start = c(2023, 1), frequency = 12)` mengonversi data numerik menjadi deret waktu, dengan `start = c(2023, 1)` menunjukkan periode awal Januari 2023, dan `frequency = 12` menunjukkan bahwa data

dicatat setiap bulan. Objek *time series* ini kemudian disimpan dalam variabel `xboost_ts` dan ditampilkan ke layar menggunakan `print(xboost_ts)` untuk memastikan struktur data sudah benar.

### Output

```
***
      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2023  210 195 220 205 218 230 225 240 235 228 215 245
2024  228 212 240 222 235 248 242 260 255 250 232 268
2025  245 230 258
```

**Pembahasan Output :** Output `print(xboost_ts)` menampilkan data penjualan dalam format deret waktu, di mana setiap nilai penjualan terkait dengan bulan dan tahun yang bersangkutan. Tampilan ini memudahkan pengguna untuk memastikan data sudah tersusun sesuai waktu dan siap digunakan untuk analisis tren, visualisasi, uji stasioneritas, atau peramalan dengan metode XGBoost maupun metode runtun waktu lainnya.

## 4. Forecasting dengan Metode Naive (12 bulan ke depan)

```
naive_forecast_xboost <- naive(xboost_ts, h = 12)

# Lihat hasil ramalan
summary(naive_forecast_xboost)
```

**Pembahasan :** Kode ini digunakan untuk melakukan peramalan penjualan bulanan `xboost_ts` menggunakan metode **Naive**, yang mengasumsikan bahwa nilai penjualan pada periode terakhir akan menjadi prediksi untuk 12 bulan berikutnya. Fungsi `naive(xboost_ts, h = 12)` menghasilkan peramalan untuk horizon 12 bulan ke depan, dan hasilnya disimpan dalam variabel `naive_forecast_xboost`. Fungsi `summary(naive_forecast_xboost)` menampilkan ringkasan peramalan, termasuk nilai ramalan dan interval kepercayaan (confidence interval) 80% dan 95%.

### Output

```
***
Forecast method: Naive method

Model Information:
Call: naive(y = xboost_ts, h = 12)

Residual sd: 17.91

Error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set 1.846154 17.91003 15.92308 0.4997217 6.783999 0.8560794 -0.5208851

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
Apr 2025          258 235.0474 280.9526 222.8970 293.1030
May 2025          258 225.5401 290.4599 208.3568 307.6432
Jun 2025          258 218.2449 297.7551 197.1998 318.8002
Jul 2025          258 212.0947 303.9053 187.7940 328.2060
Aug 2025          258 206.6764 309.3236 179.5073 336.4927
Sep 2025          258 201.7778 314.2222 172.0155 343.9845
Oct 2025          258 197.2731 318.7269 165.1261 350.8739
Nov 2025          258 193.0802 322.9198 158.7137 357.2863
Dec 2025          258 189.1421 326.8579 152.6909 363.3091
Jan 2026          258 185.4174 330.5826 146.9945 369.0055
Feb 2026          258 181.8747 334.1253 141.5765 374.4235
Mar 2026          258 178.4898 337.5102 136.3996 379.6004
```

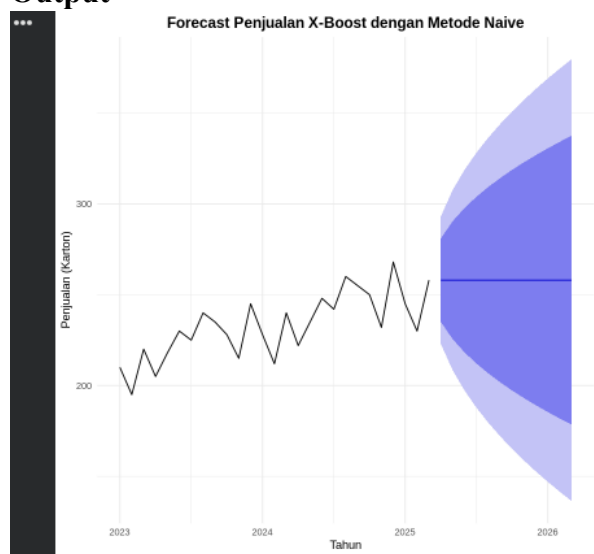
**Pembahasan Output :** Output `summary(naive_forecast_xboost)` menunjukkan nilai ramalan penjualan untuk 12 bulan ke depan, di mana semua nilai prediksi sama dengan nilai penjualan bulan terakhir pada data historis, sesuai karakteristik metode Naive. Interval kepercayaan 80% dan 95% ditampilkan untuk menunjukkan tingkat ketidakpastian ramalan, yang cenderung melebar seiring bertambahnya periode ramalan. Ringkasan ini memberikan gambaran awal tentang prediksi penjualan dan ketidakpastian terkait, yang dapat menjadi referensi pembandingan dengan metode peramalan lain.

## 5. Visualisasi Hasil Forecasting

```
autoplot(naive_forecast_xboost) +  
  ggtitle("Forecast Penjualan X-Boost dengan Metode Naive") +  
  xlab("Tahun") + ylab("Penjualan (Karton)") +  
  theme_minimal() +  
  theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"))
```

**Pembahasan :** Kode ini digunakan untuk memvisualisasikan hasil peramalan penjualan `xboost_ts` menggunakan metode **Naive**. Fungsi `autoplot(naive_forecast_xboost)` membuat grafik otomatis yang menampilkan data historis, nilai ramalan, dan interval kepercayaan. Fungsi `ggtitle()`, `xlab()`, dan `ylab()` menambahkan judul grafik serta label sumbu X dan Y. `theme_minimal()` memberikan tampilan yang bersih, sementara `theme()` menyesuaikan posisi, ukuran, dan ketebalan teks judul agar grafik lebih informatif dan menarik.

### Output



**Pembahasan Output :** Output berupa grafik runtun waktu penjualan X-Boost dengan prediksi 12 bulan ke depan. Garis utama menunjukkan data historis, sedangkan garis ramalan horizontal mengikuti nilai penjualan bulan terakhir karena metode Naive hanya menyalin nilai terakhir untuk periode selanjutnya. Area bayangan di sekitar garis ramalan menunjukkan interval kepercayaan 80% dan 95%, yang menggambarkan ketidakpastian prediksi. Visualisasi ini membantu memahami pola penjualan historis, arah prediksi, dan tingkat ketidakpastian dalam 12 bulan ke depan.

### Kasus 2 :

Lakukan peramalan penjualan smartphone menggunakan metode naïve with drift untuk data penjualan smartphone merek “Zeta” (dalam ribu unit) – Bulanan Tahun 2022 s/d November 2025.

#### 1. Install & load packages

```
suppressMessages({  
  library(forecast)  
  library(ggplot2)  
  library(scales)  
})
```

**Pembahasan :** Kode ini digunakan untuk memuat paket-paket R yang diperlukan untuk analisis runtun waktu dan visualisasi tanpa menampilkan pesan notifikasi. Fungsi `library(forecast)` memuat paket untuk peramalan deret waktu, `library(ggplot2)` untuk

visualisasi data, dan library(scales) untuk memformat sumbu grafik agar lebih mudah dibaca (misalnya menambahkan tanda ribuan atau persen). Seluruh perintah dibungkus dalam suppressMessages({ ... }) agar pesan notifikasi atau peringatan tidak muncul, sehingga output tetap rapi.

## Output

-

## Pembahasan Output :

### 2. Input data penjualan Smartphone Zeta

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

# Jadikan time series (bulanan)
zeta_ts <- ts(penjualan_zeta, start = c(2022, 1), frequency = 12)
```

**Pembahasan :** Kode ini digunakan untuk menyiapkan data penjualan produk Zeta dari Januari 2022 hingga November 2025 sebagai objek *time series* bulanan. Vektor penjualan\_zeta berisi angka penjualan per bulan secara berurutan, sementara fungsi ts(penjualan\_zeta, start = c(2022, 1), frequency = 12) mengubah data numerik menjadi objek *time series*, di mana start = c(2022, 1) menunjukkan periode awal Januari 2022 dan frequency = 12 menunjukkan data bulanan. Objek ini disimpan dalam variabel zeta\_ts untuk analisis tren, visualisasi, uji stasioneritas, atau peramalan.

## Output

-

## Pembahasan Output :

### 3. Forecasting Naive with Drift (12 bulan ke depan)

```
forecast_drift_zeta <- rwf(zeta_ts, h = 12, drift = TRUE, level = c(80, 95))
summary(forecast_drift_zeta)
```

**Pembahasan :** Kode ini digunakan untuk meramalkan penjualan produk Zeta selama 12 bulan ke depan menggunakan metode **Random Walk with Drift**. Fungsi rwf(zeta\_ts, h = 12, drift = TRUE, level = c(80, 95)) membuat prediksi berdasarkan nilai terakhir dari data historis, dengan tambahan *drift* yang mencerminkan rata-rata perubahan per periode. Argumen h = 12 menunjukkan horizon ramalan 12 bulan, dan level = c(80, 95) menambahkan interval kepercayaan 80% dan 95%. Hasil peramalan disimpan dalam variabel forecast\_drift\_zeta, dan summary() menampilkan ringkasan nilai ramalan dan interval kepercayaan.

## Output

```

***
Forecast method: Random walk with drift

Model Information:
Call: rwf(y = zeta_ts, h = 12, drift = TRUE, level = c(80, 95))

Drift: 1.6739 (se 0.295)
Residual sd: 2.0006

Error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE
Training set -1.699143e-15 1.978739 1.018904 -0.2785788 1.518144 0.05798638
ACF1
Training set -0.009160085

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
Dec 2025      123.6739 121.0823 126.2655 119.7104 127.6374
Jan 2026      125.3478 121.6440 129.0517 119.6833 131.0124
Feb 2026      127.0217 122.4385 131.6050 120.0122 134.0313
Mar 2026      128.6957 123.3496 134.0417 120.5196 136.8717
Apr 2026      130.3696 124.3330 136.4061 121.1375 139.6017
May 2026      132.0435 125.3663 138.7207 121.8316 142.2554
Jun 2026      133.7174 126.4362 140.9986 122.5817 144.8531
Jul 2026      135.3913 127.5342 143.2484 123.3750 147.4076
Aug 2026      137.0652 128.6547 145.4757 124.2025 149.9279
Sep 2026      138.7391 129.7935 147.6848 125.0579 152.4203
Oct 2026      140.4130 130.9474 149.8787 125.9365 154.8896
Nov 2026      142.0870 132.1140 152.0599 126.8347 157.3392

```

**Pembahasan Output :** Output summary(forecast\_drift\_zeta) menampilkan nilai ramalan penjualan Zeta untuk 12 bulan ke depan beserta interval kepercayaan 80% dan 95%. Nilai ramalan meningkat secara linear mengikuti tren rata-rata bulanan dari data historis karena adanya *drift*. Interval kepercayaan semakin melebar seiring bertambahnya periode ramalan, menunjukkan ketidakpastian yang meningkat. Ringkasan ini memberikan gambaran awal arah tren penjualan dan risiko ketidakpastian yang terkait, sehingga berguna untuk perencanaan produksi dan strategi bisnis.

#### 4. Visualisasi

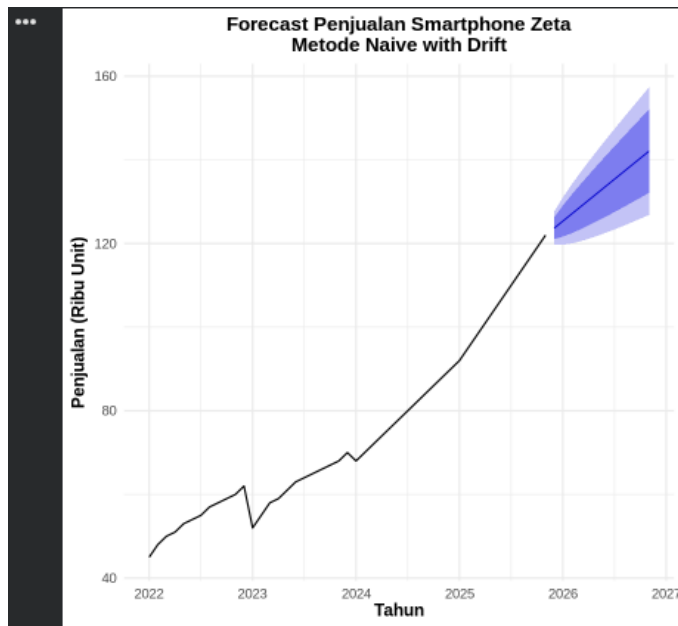
```

autoplot(forecast_drift_zeta) +
  ggtitle("Forecast Penjualan Smartphone Zeta\nMetode Naive with Drift") +
  xlab("Tahun") + ylab("Penjualan (Ribuan Unit)") +
  scale_y_continuous(labels = comma_format(big.mark = ".", decimal.mark =
",")) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
        axis.title = element_text(face = "bold"))

```

**Pembahasan :** Kode ini digunakan untuk memvisualisasikan hasil peramalan penjualan smartphone Zeta menggunakan metode **Naive with Drift**. Fungsi `autoplot(forecast_drift_zeta)` membuat grafik yang menampilkan data historis, nilai ramalan, dan interval kepercayaan. `ggtitle()`, `xlab()`, dan `ylab()` menambahkan judul dan label sumbu. `scale_y_continuous(labels = comma_format(...))` memformat angka sumbu Y agar lebih mudah dibaca dengan tanda ribuan. `theme_minimal()` memberikan tampilan bersih dan sederhana, sedangkan `theme()` menyesuaikan posisi, ukuran, dan ketebalan teks judul serta label sumbu untuk memperjelas visualisasi.

#### Output



**Pembahasan Output :** Output berupa grafik runtun waktu penjualan smartphone Zeta beserta prediksi 12 bulan ke depan. Garis utama menunjukkan data historis, sedangkan garis ramalan meningkat secara linear mengikuti *drift* rata-rata bulanan dari data historis. Area bayangan di sekitar garis ramalan menunjukkan interval kepercayaan 80% dan 95%, yang semakin melebar seiring bertambahnya periode ramalan, menggambarkan ketidakpastian prediksi. Visualisasi ini memudahkan pemahaman tren penjualan dan risiko ketidakpastian dalam perencanaan bisnis dan strategi produksi.

### C. PEMBAHASAN TUGAS

Lakukan peramalan penumpang kereta api kelas ekonomi rute Jakarta–Surabaya menggunakan metode Seasonal Naïve, dengan data jumlah penumpang (dalam ribu orang) – bulanan tahun 2021 s/d November 2025. (Terjadi lonjakan tinggi setiap Juni–Juli (libur sekolah), Desember–Januari (libur Natal & Tahun Baru), dan menurun tajam setiap Februari) .

#### 1. Install & load packages

```
suppressMessages({
  library(forecast)
  library(ggplot2)
  library(scales)
})
```

**Pembahasan :** Kode ini digunakan untuk memuat paket-paket R yang diperlukan untuk analisis runtun waktu dan visualisasi tanpa menampilkan pesan notifikasi. Fungsi `library(forecast)` memuat paket untuk peramalan deret waktu, `library(ggplot2)` untuk membuat grafik, dan `library(scales)` untuk memformat sumbu grafik agar lebih mudah dibaca, misalnya menambahkan tanda ribuan atau persen. Seluruh perintah dibungkus dalam `suppressMessages({ ... })` agar pesan notifikasi atau peringatan tidak muncul, sehingga output tetap rapi.

#### Output

-

**Pembahasan Output :**



## 2. Data Penumpang Kereta Api

```
# Data dari Jan 2021 s/d Nov 2025
penumpang_ka <- c(
  # 2021
  185, 138, 155, 168, 175, 245, 258, 210, 165, 160, 170, 235,
  # 2022
  192, 142, 160, 172, 180, 252, 265, 215, 170, 165, 175, 242,
  # 2023
  198, 145, 165, 178, 185, 260, 272, 220, 175, 170, 180, 250,
  # 2024
  205, 148, 170, 182, 190, 268, 280, 228, 180, 175, 185, 258,
  # 2025 (Sampai November)
  212, 152, 175, 188, 195, 275, 288, 235, 185, 180, 190
)

# Jadikan time series bulanan
ka_ts <- ts(penumpang_ka, start = c(2021, 1), frequency = 12)
```

**Pembahasan :** Kode ini digunakan untuk menyiapkan data jumlah penumpang kereta api dari Januari 2021 hingga November 2025 sebagai objek *time series* bulanan. Vektor `penumpang_ka` berisi data jumlah penumpang per bulan secara berurutan. Fungsi `ts(penumpang_ka, start = c(2021, 1), frequency = 12)` mengubah vektor ini menjadi objek *time series*, di mana `start = c(2021, 1)` menunjukkan periode awal Januari 2021 dan `frequency = 12` menunjukkan data bulanan. Objek ini disimpan dalam variabel `ka_ts` agar siap untuk analisis tren, visualisasi, uji stasioneritas, atau peramalan.

### Output

-

**Pembahasan Output :**

## 3. Forecasting Seasonal Naïve (12 bulan ke depan)

```
seasonal_naive_ka <- snaive(ka_ts, h = 12)
summary(seasonal_naive_ka)
```

**Pembahasan :** Kode ini digunakan untuk melakukan peramalan jumlah penumpang kereta api selama 12 bulan ke depan menggunakan metode **Seasonal Naïve**. Fungsi `snaive(ka_ts, h = 12)` membuat prediksi berdasarkan pola musiman historis, di mana nilai pada bulan tertentu diprediksi sama dengan nilai bulan yang sama pada tahun sebelumnya. Hasil peramalan disimpan dalam variabel `seasonal_naive_ka`, dan fungsi `summary(seasonal_naive_ka)` menampilkan ringkasan hasil peramalan, termasuk nilai ramalan, interval kepercayaan 80% dan 95%, serta metode yang digunakan.

### Output

```

***
Forecast method: Seasonal naive method

Model Information:
Call: snaive(y = ka_ts, h = 12)

Residual sd: 5.8872

Error measures:
      ME      RMSE      MAE      MPE      MAPE  MASE      ACF1
Training set 5.723404 5.887238 5.723404 2.830241 2.830241 1 0.2273689

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
Dec 2025           258 250.4552 265.5448 246.4612 269.5388
Jan 2026           212 204.4552 219.5448 200.4612 223.5388
Feb 2026           152 144.4552 159.5448 140.4612 163.5388
Mar 2026           175 167.4552 182.5448 163.4612 186.5388
Apr 2026           188 180.4552 195.5448 176.4612 199.5388
May 2026           195 187.4552 202.5448 183.4612 206.5388
Jun 2026           275 267.4552 282.5448 263.4612 286.5388
Jul 2026           288 280.4552 295.5448 276.4612 299.5388
Aug 2026           235 227.4552 242.5448 223.4612 246.5388
Sep 2026           185 177.4552 192.5448 173.4612 196.5388
Oct 2026           180 172.4552 187.5448 168.4612 191.5388
Nov 2026           190 182.4552 197.5448 178.4612 201.5388

```

**Pembahasan Output :** Output summary(seasonal\_naive\_ka) menampilkan nilai ramalan penumpang kereta untuk 12 bulan mendatang beserta interval kepercayaan 80% dan 95%. Nilai ramalan pada setiap bulan mengikuti pola musiman yang sama dengan bulan bersangkutan pada tahun sebelumnya. Interval kepercayaan menunjukkan ketidakpastian prediksi, yang biasanya melebar untuk bulan-bulan lebih jauh dari data historis. Hasil ini memberikan gambaran tentang perkiraan jumlah penumpang sambil mempertahankan pola musiman, sehingga berguna untuk perencanaan operasional dan strategi layanan kereta.

#### 4. Visualisasi pola musiman

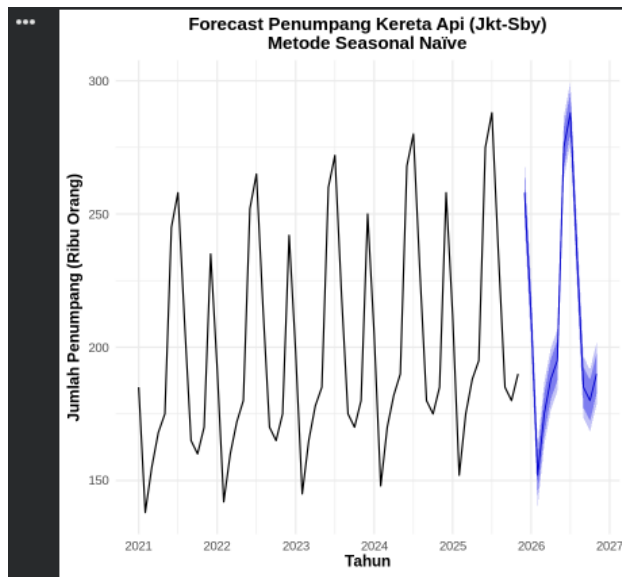
```

autoplot(seasonal_naive_ka) +
  ggtitle("Forecast Penumpang Kereta Api (Jkt-Sby)\nMetode Seasonal Naïve")
+
  xlab("Tahun") +
  ylab("Jumlah Penumpang (Ribuan Orang)") +
  scale_y_continuous(labels = comma) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
        axis.title = element_text(face = "bold")) +
  geom_line(color = "steelblue", size = 1) +
  geom_point(color = "steelblue")

```

**Pembahasan :** Kode ini digunakan untuk memvisualisasikan hasil peramalan jumlah penumpang kereta api rute Jakarta–Surabaya menggunakan metode **Seasonal Naïve**. Fungsi `autoplot(seasonal_naive_ka)` membuat grafik yang menampilkan data historis, nilai ramalan, dan interval kepercayaan. Fungsi `ggtitle()`, `xlab()`, dan `ylab()` menambahkan judul dan label sumbu. `scale_y_continuous(labels = comma)` memformat angka sumbu Y agar lebih mudah dibaca. `theme_minimal()` memberikan tampilan bersih, sementara `theme()` menyesuaikan posisi, ukuran, dan ketebalan teks. `geom_line()` dan `geom_point()` menambahkan garis dan titik untuk menonjolkan pola data historis dan ramalan.

**Output**



**Pembahasan Output :** Output berupa grafik runtun waktu jumlah penumpang kereta api beserta prediksi 12 bulan ke depan. Garis utama menunjukkan data historis, sedangkan garis ramalan mengikuti pola musiman tahunan dari data historis. Area bayangan di sekitar garis ramalan menunjukkan interval kepercayaan 80% dan 95%, yang semakin melebar untuk bulan-bulan lebih jauh, mencerminkan ketidakpastian prediksi. Visualisasi ini memudahkan pemahaman pola musiman, tren, dan tingkat risiko ketidakpastian dalam perencanaan operasional kereta.

#### D. KESIMPULAN

Berdasarkan hasil praktikum, metode **Naive** dan **Seasonal Naive** efektif digunakan untuk meramalkan data runtun waktu yang sederhana atau memiliki pola musiman. Metode Naive menghasilkan prediksi yang sama dengan nilai terakhir dari data historis, sehingga cocok untuk data tanpa tren atau musiman yang kuat. Sedangkan Seasonal Naive mempertahankan pola musiman dengan memprediksi nilai pada bulan tertentu sama seperti bulan yang bersangkutan pada tahun sebelumnya, sehingga lebih sesuai untuk data dengan musiman jelas, seperti penjualan ritel atau jumlah penumpang kereta api. Interval kepercayaan 80% dan 95% memberikan gambaran ketidakpastian prediksi, yang semakin melebar untuk periode ramalan yang lebih jauh. Secara keseluruhan, metode ini sederhana, mudah diterapkan, dan memberikan baseline prediksi yang berguna sebagai pembandingan dengan metode peramalan lain yang lebih kompleks.