Bayu Widodo

17 February 2026

Daftar Isi

Daftar Gambar

> Tidak ada batasan bagi data untuk menceritakan kisahnya dan R adalah alat yang memungkinkan kita mendengar, melihat, dan memahaminya. Dalam usaha ini, eksplorasi, analisis, dan visualisasi menjadi fondasi penting yang mengubah angka-angka menjadi cerita bermakna.

4 Statistika Deskriptif dan Visualisasi Data

4.1 Capaian Pembelajaran bab 4

Setelah menyelesaikan Bab 4, mahasiswa mampu:

1. Menjelaskan konsep dan tujuan statistika deskriptif dalam menganalisis data, meliputi ukuran pemusatan, ukuran penyebaran, dan distribusi data.
2. Mengolah dan meringkas data menggunakan dplyr, termasuk melakukan seleksi data, penyaringan, pengelompokan, serta perhitungan statistik deskriptif secara efisien.
3. Menyajikan data dalam bentuk visualisasi yang informatif menggunakan ggplot2, seperti grafik batang, histogram, boxplot, dan grafik garis sesuai dengan karakteristik data.
4. Menginterpretasikan hasil statistika deskriptif dan visualisasi data, serta menarik kesimpulan yang relevan untuk mendukung analisis dan pengambilan keputusan berbasis data.

Kuliah / Teori

4.2 Paket dplyr

Dalam analisis data menggunakan R, meskipun base R telah menyediakan berbagai fungsi untuk manipulasi data dan visualisasi, sintaks yang digunakan sering kali relatif panjang, kurang konsisten, dan kurang intuitif, terutama ketika bekerja dengan dataset berukuran besar.

Untuk mengatasi keterbatasan tersebut, dikembangkan pendekatan modern melalui ekosistem tidyverse, salah satunya adalah paket dplyr, yang dirancang khusus untuk mempermudah proses manipulasi data secara efisien dan terstruktur.

Paket dplyr merupakan bagian inti dari lingkungan tidyverse yang menyediakan seperangkat fungsi (kata kerja) utama untuk manipulasi data, seperti memilih variabel, menyaring baris, membuat variabel baru, mengelompokkan data, dan melakukan agregasi. Seluruh fungsi dalam dplyr menggunakan sintaks yang ringkas, konsisten, dan mudah dibaca, serta terintegrasi dengan konsep tidy data.

Dengan dukungan operator pipe, alur analisis data dapat dituliskan secara berurutan dan logis, sehingga meningkatkan keterbacaan dan reprodusibilitas analisis. Melalui penggunaan dplyr dalam ekosistem tidyverse, proses analisis statistika deskriptif menjadi lebih cepat, rapi, dan sistematis dibandingkan dengan penggunaan base R semata.

4.2.1 Memasang Paket dplyr

Sebelum menggunakan paket dplyr dalam proses analisis data, pengguna perlu memastikan bahwa paket tersebut telah terpasang (installed) dan dimuat (loaded) ke dalam lingkungan kerja R.

Pemasangan paket dilakukan satu kali pada suatu sistem, sedangkan pemanggilan paket perlu dilakukan setiap kali sesi R dijalankan. Paket dplyr tersedia secara bebas melalui Comprehensive R Archive Network (CRAN) dan dapat dipasang menggunakan fungsi install.packages().

Untuk meningkatkan efisiensi dan mencegah pemasangan ulang yang tidak diperlukan, proses instalasi umumnya dikombinasikan dengan fungsi require() atau library(). Pendekatan ini memungkinkan R untuk terlebih dahulu memeriksa ketersediaan paket, dan hanya melakukan pemasangan jika paket tersebut belum terinstal pada sistem.

```r
if (!require(dplyr)) {
  install.packages("dplyr")
  library(dplyr)
}
```

Sintaks tersebut memastikan bahwa paket dplyr tersedia dan siap digunakan tanpa menimbulkan kesalahan akibat paket yang belum terpasang. Setelah paket berhasil dimuat, seluruh fungsi manipulasi data yang disediakan oleh dplyr dapat langsung digunakan dalam analisis.

Sebagai bagian dari ekosistem tidyverse, pemasangan dplyr juga dapat dilakukan secara tidak langsung dengan memasang paket tidyverse, yang akan mengikutsertakan dplyr beserta paket pendukung lainnya.

4.2.2 Kata Kerja dplyr

Paket dplyr menyediakan seperangkat fungsi inti yang dikenal sebagai kata kerja (verbs) untuk manipulasi data. Kata kerja ini dirancang agar sintaksnya ringkas, konsisten, dan mudah dibaca, sehingga sangat

mendukung alur analisis data yang terstruktur.

Tabel berikut menyajikan kata kerja utama (verbs) yang digunakan dalam paket dplyr. Setiap kata kerja merepresentasikan operasi dasar dalam manipulasi data, seperti pemilihan variabel, penyaringan observasi, transformasi data, pengelompokan, dan peringkasan statistik

Tabel 4.1: Kata Kerja Utama (*Verbs*) pada Paket `dplyr`

No.	Kata Kerja	Deskripsi	Contoh
1	`select()`	Digunakan untuk memilih satu atau beberapa kolom tertentu dari sebuah dataset. Fungsi ini memudahkan fokus analisis hanya pada variabel yang relevan tanpa mengubah struktur data lainnya.	`select(mtcars, mpg, hp)`
2	`filter()`	Digunakan untuk menyaring baris data berdasarkan kondisi logika tertentu. Hanya observasi yang memenuhi syarat yang akan dipertahankan dalam hasil akhir.	`filter(mtcars, cyl > 6)`
3	`mutate()`	Digunakan untuk membuat variabel baru atau memodifikasi variabel yang sudah ada berdasarkan operasi tertentu. Fungsi ini sangat berguna dalam proses transformasi data.	`mutate(mtcars, kpl = mpg * 0.425)`
4	`arrange()`	Digunakan untuk mengurutkan data berdasarkan satu atau lebih variabel, baik secara menaik (ascending) maupun menurun (descending).	`arrange(mtcars, desc(mpg))`
5	`group_by()`	Digunakan untuk mengelompokkan data berdasarkan variabel tertentu sebagai prasyarat untuk analisis agregasi atau peringkasan.	`group_by(mtcars, cyl)`
6	`summarise()`	Digunakan untuk menghasilkan ringkasan statistik dari data, seperti rata-rata, median, simpangan baku, dan ukuran statistik lainnya. Umumnya digunakan bersama `group_by()`.	`summarise(mtcars, mean_mpg = mean(mpg))`
7	`rename()`	Digunakan untuk mengganti nama kolom agar lebih deskriptif dan mudah dipahami, tanpa mengubah isi data.	`rename(mtcars, horsepower = hp)`

Kata kerja pada dplyr dirancang untuk digunakan bersama operator pipe (%>%), sehingga membentuk alur analisis data yang jelas dan berurutan, misalnya: seleksi data > pengelompokan > peringkasan. Pendekatan ini membuat kode lebih mudah dibaca, dipahami, dan dipelihara dibandingkan pendekatan base R yang sering kali lebih panjang dan kurang intuitif.

4.2.3 Operator pipe (%>%) dalam dplyr

Operator pipe (%>%) merupakan salah satu konsep kunci dalam penggunaan paket dplyr dan ekosistem tidyverse. Operator ini digunakan untuk mengalirkan (meneruskan) hasil suatu operasi ke operasi

berikutnya, sehingga kode R dapat ditulis secara berurutan dan mudah dibaca. Dengan pipe, hasil dari sebuah fungsi tidak perlu disimpan terlebih dahulu ke dalam objek antara, melainkan langsung digunakan sebagai input fungsi selanjutnya.

Secara konseptual, operator pipe dapat dibaca sebagai kata "kemudian" atau "lalu". Pendekatan ini sangat membantu mahasiswa dalam memahami alur analisis data karena mencerminkan cara berpikir manusia, yaitu melakukan langkah demi langkah secara sistematis.

Sebagai ilustrasi, perhatikan contoh berikut yang bertujuan untuk memilih beberapa kolom dari dataset dan kemudian menghitung ringkasan statistiknya.

```r
# tanpa menggunakan operator pipe
mean_mpg <- mean(mtcars$mpg)
mean_hp  <- mean(mtcars$hp)
```

```r
# menggunakan operator pipe
library(dplyr)

mtcars %>%
  summarise(
    mean_mpg = mean(mpg),
    mean_hp  = mean(hp)
  )
```

Pada contoh di atas, dataset mtcars dialirkan langsung ke fungsi summarise() tanpa perlu pemanggilan objek berulang. Hal ini membuat kode lebih ringkas dan mudah dipahami.

Berikut disajikan contoh penggunaan kata kerja (verbs) pada paket dplyr menggunakan data yang sederhana. Contoh ini bertujuan membantu mahasiswa memahami fungsi dan alur penggunaan setiap kata kerja dplyr secara bertahap, sehingga mahasiswa dapat dengan mudah menerapkannya pada dataset yang lebih besar dan kompleks dalam proses analisis data.

```r
library(dplyr)

data_nilai <- data.frame(
  nama  = c("Andi", "Budi", "Citra", "Dewi", "Eko"),
  nilai = c(80, 65, 90, 70, 85),
  kelas = c("A", "A", "B", "B", "A")
)
data_nilai
str(data_nilai)
```

1. select() digunakan untuk memilih variabel tertentu dari dataset. Pada contoh berikut hanya menampilkan hanya kolom nama dan nilai, tanpa kolom kelas.

```r
data_nilai %>%
  select(nama, nilai)
```

2. filter() — Menyaring Baris. Digunakan untuk menyaring data berdasarkan kondisi tertentu.

```r
# Menampilkan mahasiswa yang memperoleh nilai 80 atau lebih.
data_nilai %>%
  filter(nilai >= 80)
```

3. mutate() — Membuat Variabel Baru. Digunakan untuk menambah atau memodifikasi variabel.

```r
# Menambahkan variabel status yang menunjukkan kelulusan mahasiswa.
data_nilai %>%
  mutate(status = ifelse(nilai >= 75, "Lulus", "Tidak Lulus"))
```

4. arrange() — Mengurutkan Data. Digunakan untuk mengurutkan data berdasarkan suatu variabel.

```r
# Mengurutkan data mahasiswa berdasarkan nilai dari yang tertinggi ke terendah.
data_nilai %>%
  arrange(desc(nilai))
```

5. group_by() — Mengelompokkan Data. Digunakan untuk melakukan analisis berdasarkan kelompok tertentu.

```r
# Mengelompokkan data berdasarkan kelas (A dan B).
data_nilai %>%
  group_by(kelas)
```

6. summarise() — Ringkasan Statistik. Digunakan untuk menghitung ukuran statistik.

```r
# Menghitung nilai rata-rata dan nilai tertinggi dari seluruh mahasiswa.
data_nilai %>%
  summarise(
    rata_nilai = mean(nilai),
    nilai_maks = max(nilai)
  )
```

7. Kombinasi group_by() dan summarise().

```r
# Menghitung rata-rata nilai untuk setiap kelas.
data_nilai %>%
  group_by(kelas) %>%
  summarise(
    rata_nilai = mean(nilai)
  )
```

Catatan penting

Melalui contoh data sederhana ini, mahasiswa dapat melihat bahwa kata kerja dplyr membentuk alur analisis yang jelas dan logis, yaitu: data $\Rightarrow$ seleksi $\Rightarrow$ penyaringan $\Rightarrow$ transformasi $\Rightarrow$ pengelompokan $\Rightarrow$ ringkasan
Pendekatan ini menjadi dasar penting sebelum mahasiswa bekerja dengan dataset yang lebih besar dan kompleks, seperti mtcars atau iris, serta sebelum melanjutkan ke tahap visualisasi data menggunakan ggplot2.

4.3 Statistika Deskriptif

Statistik Deskriptif merupakan cabang dari ilmu statistik yang bertujuan untuk meringkas, mendeskripsikan, dan menyajikan serangkaian nilai atau kumpulan data. Statistik deskriptif

seringkali merupakan langkah pertama dan merupakan bagian penting dalam setiap analisis statistik. Statistik deskriptif memungkinkan untuk memeriksa kualitas data dan membantu untuk "memahami" data secara lengkap, jelas dan mendalam.

Serangkaian angka yang panjang tanpa persiapan atau ringkasan sering kali sulit memberikan informasi yang bermakna, karena pola atau tren dalam data tidak mudah dikenali. Sebagai contoh, berikut adalah data tinggi badan (dalam cm) dari 100 orang dewasa:

188.7, 169.4, 178.6, 181.3, 179, 173.9, 190.1, 174.1, 195.2, 174.4, 188, 197.9, 161.1, 172.2, 173.7, 181.4, 172.2, 148.4, 150.6, 188.2, 171.9, 157.2, 173.3, 187.1, 194, 170.7, 172.4, 157.4, 179.6, 168.6, 179.6, 182, 185.4, 168.9, 180, 157.8, 167.2, 166.5, 150.9, 175.4, 177.1, 171.4, 182.6, 167.7, 161.3, 179.3, 166.9, 189.4, 170.7, 181.6, 178.2, 167.2, 190.8, 181.4, 175.9, 177.8, 181.8, 175.9, 145.1, 177.8, 171.3, 176.9, 180.8, 189, 167.7, 188, 178.4, 185.4, 184.2, 182.2, 164.6, 174.1, 181.2, 165.5, 169.6, 180.8, 182.7, 179.6, 166.1, 164, 190.1, 177.6, 175.9, 173.8, 163.1, 181.1, 172.8, 173.2, 184.3, 183.2, 188.9, 170.2, 181.5, 188.9, 163.9, 166.4, 163.7, 160.4, 175.8, dan 181.5.

Melihat data mentah seperti ini, sulit bagi siapa pun untuk memahami pola atau informasi penting yang terkandung di dalamnya. Statistika deskriptif menjadi alat penting untuk merangkum data, sehingga memberikan gambaran yang lebih jelas. Memang, dengan merangkum data melalui ukuran tertentu, beberapa informasi detail mungkin hilang. Namun, dalam banyak kasus, kehilangan sedikit informasi tersebut lebih baik dibandingkan sulitnya memahami keseluruhan dataset.

Statistika deskriptif biasanya merupakan langkah awal yang krusial dalam setiap analisis statistik. Dengan menggunakan statistika deskriptif, kita dapat memeriksa kualitas data, mengenali pola, dan memperoleh pemahaman awal yang memadai sebelum melakukan analisis lebih lanjut.

Secara umum, ukuran-ukuran dalam statistika deskriptif dapat dikelompokkan menjadi dua jenis utama:

1. Ukuran Pemusatan, yang menggambarkan titik pusat atau nilai representatif dari data.
2. Ukuran Penyebaran, yang menggambarkan seberapa besar variasi atau penyimpangan data dari nilai pusatnya.

Ukuran pemusatan memberikan gambaran tentang nilai tengah atau kecenderungan sentral suatu dataset, sedangkan ukuran penyebaran (dispersion) menggambarkan sejauh mana data tersebar atau menyimpang dari nilai pusatnya. Pada bab ini, fokus akan diberikan pada implementasi statistika deskriptif yang paling umum di R, beserta visualisasi data yang relevan untuk mendukung pemahaman.

4.3.1 Dataset di R

R menyediakan berbagai dataset bawaan (built-in) yang siap digunakan untuk keperluan analisis data. Dataset ini mencakup beragam jenis data, mulai dari data numerik, kategorikal, hingga multivariat, yang berasal dari berbagai bidang seperti biologi, ekonomi, dan statistik.

Dataset bawaan ini memudahkan pengguna untuk menguji metode analisis, praktik visualisasi, atau eksplorasi statistika deskriptif, tanpa harus mengumpulkan data dari sumber eksternal. Selain itu, data-

set bawaan ini juga berguna sebagai bahan pembelajaran dan percobaan sebelum bekerja dengan data nyata yang lebih kompleks.

Pengguna dapat melihat daftar lengkap dataset bawaan menggunakan fungsi data(), dan sebagian dataset dapat dimuat secara eksplisit dengan menyebutkan nama dataset sebagai argumen fungsi tersebut.

```
data()
```

Sebagai contoh, salah satu dataset bawaan yang populer adalah dataset iris, yang berisi informasi mengenai panjang dan lebar sepal, panjang dan lebar petal, serta spesies dari 150 bunga iris. Untuk menggunakan dataset ini dalam analisis, dataset iris dapat disimpan ke dalam sebuah objek di R. Misalnya, kita dapat mengetikkan perintah berikut:

```
data_iris <- iris
```

Dengan perintah tersebut, dataset iris akan tersimpan dalam objek data_iris dan siap untuk dianalisis atau divisualisasikan. Menyimpan dataset ke dalam objek memungkinkan kita untuk melakukan manipulasi, eksplorasi, dan penerapan fungsi-fungsi dplyr maupun ggplot2 tanpa mengubah dataset bawaan asli. Pendekatan ini juga memudahkan pengelolaan data ketika bekerja dengan beberapa dataset sekaligus dalam satu sesi R.

Dataset iris berisi 150 observasi dan 5 variabel, yang merepresentasikan panjang dan lebar sepal, panjang dan lebar petal, serta spesies dari masing-masing bunga iris. Variabel panjang dan lebar sepal maupun petal merupakan variabel numerik, sedangkan variabel Species merupakan faktor dengan tiga level. Informasi ini dapat dilihat dengan perintah str(iris) di R, di mana kolom numerik ditunjukkan sebagai num, sedangkan kolom faktor ditunjukkan sebagai Factor w/ 3 levels.

```
?iris
str(data_iris)
head(data_iris,1)
```

Untuk memperkuat pemahaman mahasiswa, pada bagian ini disajikan perbandingan antara pemilihan kolom menggunakan teknik slicing pada base R dan pendekatan menggunakan paket dplyr. Melalui perbandingan ini, mahasiswa diharapkan mampu mengenali keunggulan dplyr, khususnya dari sisi keterbacaan sintaks, fleksibilitas dalam pemilihan variabel, serta kemudahan dalam proses analisis data.

Pendekatan pertama menggunakan dplyr memungkinkan pemilihan kolom numerik dilakukan secara otomatis berdasarkan tipe data, sebagaimana ditunjukkan pada kode berikut:

```
library(dplyr)

# memisahkan kolom numerik dari dataset iris
dat.num <- iris %>%
  select(where(is.numeric))

# menampilkan struktur dat.num
str(dat.num)
```

Sebagai pembanding, pemilihan kolom numerik juga dapat dilakukan menggunakan teknik slicing pada base R, dengan menentukan posisi kolom secara eksplisit, seperti pada contoh berikut:

```
# menyimpan dataset iris ke dalam variabel df
df <- iris
print(df)

```

```r
# memisahkan hanya kolom numerik menggunakan slicing
df.num <- iris[, 1:4]
str(df.num)
```

Dari kedua pendekatan tersebut dapat dilihat bahwa slicing pada base R bersifat lebih manual dan bergantung pada posisi kolom, sedangkan dplyr menawarkan pendekatan yang lebih adaptif dan mudah dipelihara, terutama ketika struktur dataset mengalami perubahan. Oleh karena itu, penggunaan dplyr lebih direkomendasikan dalam analisis data modern untuk mendukung alur kerja yang sistematis dan efisien.

4.3.2 Ringkasan Statistik Deskriptif: base R vs dplyr

1. Ringkasan Statistik Menggunakan base R
   Pada base R, ringkasan statistik biasanya dihitung satu per satu atau menggunakan beberapa fungsi yang terpisah.

```r
# memilih kolom numerik dengan slicing
df.num <- iris[, 1:4]

# ringkasan statistik dasar
mean(df.num$Sepal.Length)
sd(df.num$Sepal.Length)
min(df.num$Sepal.Length)
max(df.num$Sepal.Length)

# ringkasan seluruh kolom numerik
summary(df.num)
```

2. Ringkasan Statistik Menggunakan dplyr
   Dengan dplyr, ringkasan statistik dapat dilakukan secara ringkas dan terstruktur menggunakan satu alur perintah.

```r
library(dplyr) # cukup diload sekali

dat.num <- iris %>%
  select(where(is.numeric))

dat.num %>%
  summarise(
    mean_sepal_length  = mean(Sepal.Length),
    sd_sepal_length    = sd(Sepal.Length),
    min_sepal_length   = min(Sepal.Length),
    max_sepal_length   = max(Sepal.Length)
  )
```

3. Ringkasan Statistik Berdasarkan Kelompok (dplyr)
   Keunggulan utama dplyr terlihat ketika melakukan ringkasan statistik berdasarkan kelompok, misalnya berdasarkan spesies bunga.

```r
iris %>%
  group_by(Species) %>%
  summarise(
```

```
    mean_sepal_length = mean(Sepal.Length),
    mean_petal_length = mean(Petal.Length)
  )
```

Pendekatan ini jauh lebih sederhana dan mudah dibaca dibandingkan implementasi serupa menggunakan base R.

```
mean_sepal_length <- tapply(
  iris$Sepal.Length,
  iris$Species,
  mean
)

mean_petal_length <- tapply(
  iris$Petal.Length,
  iris$Species,
  mean
)
```

4.3.3 Ukuran Pemusatan

Ukuran pemusatan digunakan untuk menggambarkan nilai yang mewakili kecenderungan sentral dari suatu dataset. Ukuran pemusatan yang umum digunakan adalah mean (rata-rata), median, dan modus.

Sebagai contoh, kita akan menghitung ukuran pemusatan untuk variabel Income dan Life Exp di dataset State.X77. Dataset state.x77 merupakan dataset bawaan di R yang berisi data statistik dari 50 negara bagian di Amerika Serikat. Dataset ini terdiri dari beberapa variabel numerik, seperti jumlah penduduk (Population), tingkat kejahatan (Murder), pendapatan rata-rata (Income), tingkat buta huruf (Illiteracy), harapan hidup (Life Exp), dan beberapa indikator lainnya. Karena seluruh variabelnya bersifat numerik, dataset ini sangat sesuai untuk mempelajari statistika deskriptif, khususnya ukuran pemusatan dan ukuran penyebaran.

Untuk menggunakan dataset ini, kita dapat memuatnya ke dalam R sebagai berikut:

```
df_state <- state.x77

# mengubah menjadi data frame agar mudah dianalisis
df_state <- as.data.frame(state.x77)
str(df_state)
names(df_state)
# mean
mean(df_state$Income)

# median
median(df_state$Income)
```

Untuk menghitung ukuran pemusatan beberapa variabel sekaligus:

```
apply(
  df_state[, c("Income", "Life Exp")],
  2,
  mean
)
```

Penjelsan singkat:

- df_state[, c("Income", "Life Exp")], memilih dua kolom, yaitu Income dan Life Exp, dari dataset df_state.
- Angka 2 menunjukkan bahwa operasi dilakukan per kolom (bukan per baris, baris menggunakan angka 1).
- mean sebagai fungsi yang diterapkan, yaitu menghitung nilai rata-rata.

Hasil dari perintah ini adalah dua nilai mean: rata-rata Income dan rata-rata Life Exp, yang berfungsi sebagai ukuran pemusatan data untuk masing-masing variabel.

Jika menggunakan dplyr, sintaks untuk melakukan operasi yang sama menjadi lebih ringkas dan mudah dibaca. Padanan dari perintah apply() tersebut adalah sebagai berikut:

```r
library(dplyr)
df_state %>%
  summarise(
    mean_income   = mean(Income),
    mean_life_exp = mean(`Life Exp`)
  )
```

Penjelasan singkat:

- summarise() digunakan untuk menghitung ringkasan statistik.
- mean(Income) dan mean(`Life Exp`) menghitung nilai rata-rata masing-masing variabel.
- Tanda backtick (`) digunakan karena nama variabel Life Exp mengandung spasi.
- Hasilnya berupa satu baris data yang menampilkan rata-rata Income dan rata-rata Life Exp.

Alternatif yang lebih umum:

```r
library(dplyr)
df_state %>%
  summarise(
    across(
      c(Income, `Life Exp`),
      mean
    )
  )
```

4.3.4 Ukuran Penyebaran

Ukuran penyebaran digunakan untuk menggambarkan sejauh mana data menyebar atau bervariasi terhadap nilai pusatnya. Ukuran ini penting untuk melengkapi ukuran pemusatan, karena dua dataset dapat memiliki nilai rata-rata yang sama tetapi tingkat variasi yang berbeda. Ukuran penyebaran yang umum digunakan antara lain rentang (range), varians, dan simpangan baku (standard deviation).

1. Rentang (Range)
   Rentang menunjukkan selisih antara nilai maksimum dan minimum suatu variabel.

```r
df_state %>%
  summarise(
    range_income = max(Income) - min(Income),
    range_life_exp = max(`Life Exp`) - min(`Life Exp`)
  )
```

Nilai rentang memberikan gambaran seberapa jauh nilai tertinggi dan terendah dari Income dan Life Exp tersebar.

2. Varians
   Varians mengukur rata-rata kuadrat penyimpangan setiap data terhadap nilai rata-ratanya.

```
df_state %>%
  summarise(
    var_income = var(Income),
    var_life_exp = var(`Life Exp`)
  )
```

   Semakin besar nilai varians, semakin besar pula variasi data terhadap nilai rata-rata.

3. Simpangan Baku (Standard Deviation)
   Simpangan baku merupakan akar dari varians dan berada pada satuan yang sama dengan data asli.

```
df_state %>%
  summarise(
    sd_income = sd(Income),
    sd_life_exp = sd(`Life Exp`)
  )
```

   Simpangan baku menunjukkan tingkat penyimpangan data dari nilai rata-ratanya secara lebih mudah diinterpretasikan dibandingkan varians.

Melalui ukuran penyebaran, mahasiswa dapat memahami bahwa nilai rata-rata saja belum cukup untuk menggambarkan karakteristik data. Kombinasi antara ukuran pemusatan dan ukuran penyebaran memberikan gambaran yang lebih utuh mengenai distribusi dan variasi data, serta menjadi dasar penting sebelum melakukan visualisasi atau analisis statistik lanjutan.

4.4 Visualisasi Data

Analisis data sering kali tidak dapat dipisahkan dari proses visualisasi. Banyak pola, tren, maupun hubungan antarvariabel yang sulit dikenali hanya melalui angka atau tabel, namun dapat terlihat dengan lebih jelas ketika data disajikan dalam bentuk grafik. Oleh karena itu, visualisasi data menjadi langkah penting dalam memahami karakteristik data sekaligus membantu peneliti atau analis dalam menentukan model statistik yang sesuai untuk menarik kesimpulan dari data.

Bahasa pemrograman R menyediakan beberapa sistem visualisasi data yang umum digunakan, yaitu Base Plot, Lattice Plot, dan ggplot2. Masing-masing sistem memiliki karakteristik dan kegunaan tersendiri.

1. Base Plot
   Sistem ini memungkinkan pengguna membangun grafik secara bertahap, dimulai dari kanvas kosong kemudian menambahkan elemen-elemen visual seperti titik, garis, judul, dan anotasi. Base plot sering digunakan untuk eksplorasi data cepat (exploratory data analysis) karena sintaksnya relatif sederhana.

2. Lattice Plot
   Pada sistem ini, grafik dibuat dalam satu pemanggilan fungsi. Setelah grafik ditampilkan, strukturnya tidak dapat dimodifikasi kembali. Kelebihan utama lattice plot adalah kemampuannya dalam membuat subplot atau panel secara mudah dan konsisten.
3. ggplot2 Plot
   ggplot2 merupakan sistem visualisasi yang menggabungkan kelebihan dari base plot dan lattice plot. Sistem ini menyediakan pengaturan default yang baik, namun tetap sangat fleksibel untuk dikustomisasi. Dengan pendekatan grammar of graphics, ggplot2 memungkinkan pengguna membangun grafik secara sistematis berdasarkan data, estetika, dan lapisan (layers). ggplot2 juga didukung oleh komunitas yang besar serta dokumentasi dan contoh penggunaan yang sangat melimpah.

Dalam pembahasan ini, fokus akan diberikan pada ggplot2 karena kemampuannya menghasilkan visualisasi yang informatif, konsisten, dan siap digunakan untuk kebutuhan publikasi maupun laporan profesional. Selain itu, ggplot2 juga menyediakan fungsi qplot yang menawarkan sintaks lebih sederhana bagi pemula, sehingga memudahkan proses belajar sebelum memahami konsep visualisasi yang lebih mendalam.

4.4.1 Memuat (Load) Library ggplot2

Sebelum menggunakan ggplot2 untuk membuat visualisasi data, pengguna harus memastikan bahwa paket ggplot2 sudah terpasang dan dimuat ke dalam sesi kerja R. Library merupakan kumpulan fungsi dan fitur tambahan yang tidak selalu tersedia secara default pada instalasi R.

Apabila ggplot2 belum terinstal, proses instalasi dapat dilakukan satu kali menggunakan perintah:

```
1 if (!"ggplot2" %in% installed.packages()) {
2   install.packages("ggplot2")
3 }
4 library(ggplot2)
```

Perintah library(ggplot2) wajib dijalankan setiap kali memulai sesi kerja R atau RStudio sebelum menggunakan fungsi-fungsi visualisasi dari ggplot2. Jika library tidak dimuat, maka fungsi seperti ggplot(), geom_point(), atau geom_line() tidak akan dikenali oleh R dan akan menimbulkan pesan kesalahan (error).

4.4.2 Struktur Dasar ggplot2

Visualisasi data menggunakan ggplot2 dibangun berdasarkan konsep Grammar of Graphics, yaitu pendekatan sistematis dalam menyusun grafik dari beberapa komponen utama. Dengan pendekatan ini, grafik tidak dibuat sekaligus, tetapi dirangkai dari lapisan-lapisan (layers) yang saling melengkapi.

Secara umum, setiap grafik dalam ggplot2 tersusun atas beberapa komponen utama berikut:

1. Data (Data Set)
   Komponen pertama adalah data, yaitu kumpulan nilai yang akan divisualisasikan. Data biasanya berbentuk data frame atau tibble. Semua elemen grafik dalam ggplot2 selalu merujuk pada data ini sebagai sumber utama informasi.
2. Geometri (Geoms)
   Komponen kedua adalah geometri (geom), yaitu tanda visual yang digunakan untuk merepre-

sentasikan data pada grafik. Geom menentukan bagaimana data ditampilkan, misalnya sebagai titik, garis, atau batang. Beberapa contoh geom yang umum digunakan:
- geom_point(), titik (scatter plot)
- geom_line(), garis
- geom_bar(), batang
- geom_histogram(), histogram

Geom berfungsi sebagai penghubung antara data dan tampilan visual yang dihasilkan.

3. Sistem Koordinat (Coordinate System)
   Komponen ketiga adalah sistem koordinat, yang mengatur bagaimana posisi data ditampilkan pada ruang grafik. Sistem koordinat paling umum adalah sistem kartesius, tetapi ggplot2 juga mendukung sistem koordinat lain, seperti polar. Contohnya:
   - coord_cartesian(), koordinat kartesius
   - coord_polar(), koordinat polar
4. Komponen Tambahan (Additional Components)
   Selain tiga komponen utama tersebut, ggplot2 menyediakan beberapa komponen tambahan untuk penyempurnaan grafik, antara lain:
   - Statistical Transformation (stat)
     Digunakan untuk melakukan perhitungan statistik sebelum data divisualisasikan, misalnya perhitungan frekuensi atau rata-rata.
   - Scales
     Mengatur pemetaan nilai data ke aspek visual seperti warna, ukuran, dan skala sumbu.
   - Position Adjustment
     Mengatur posisi elemen visual agar tidak saling bertumpuk, misalnya dengan stack, dodge, atau jitter.
   - Faceting
     Digunakan untuk membagi satu grafik menjadi beberapa panel berdasarkan kategori tertentu, sehingga memudahkan perbandingan antar kelompok data.

4.4.3 Penyusunan Grafik dengan ggplot() dan Operator +

Pembuatan grafik dalam ggplot2 selalu diawali dengan pemanggilan fungsi ggplot(). Fungsi ini mendefinisikan data dan pengaturan awal grafik. Selanjutnya, grafik disusun dengan menambahkan lapisan (layers) menggunakan operator $+$.

Struktur umum ggplot2 dapat dituliskan sebagai berikut:

```
ggplot(data = <data>) +
  geom_<tipe>() +
  komponen_lain
```

Dengan pendekatan grammar of graphics, ggplot2 memungkinkan pengguna membangun berbagai jenis grafik dari komponen yang sama, yaitu:
Data + Geometri + Sistem Koordinat + Komponen Tambahan
Pendekatan ini menjadikan ggplot2 sebagai alat visualisasi yang kuat, konsisten, dan sangat sesuai untuk analisis data, penelitian, serta pelaporan profesional.

4.4.4 Contoh Penerapan ggplot2

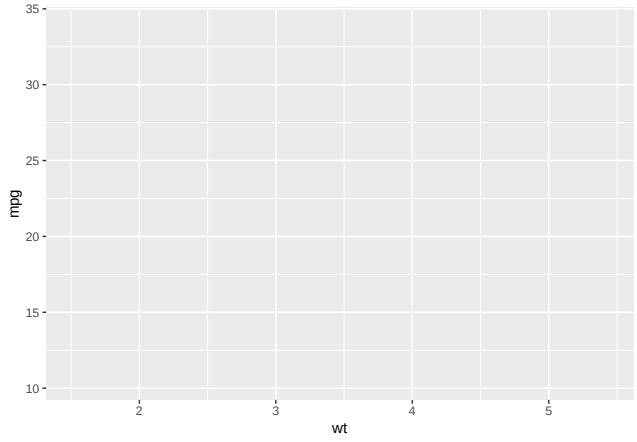
Pada contoh ini digunakan dataset bawaan R, yaitu mtcars, untuk memvisualisasikan hubungan antara berat mobil (wt) dan efisiensi bahan bakar (mpg).

1. Menentukan Data
   Langkah pertama adalah menentukan data yang akan digunakan sebagai sumber visualisasi. Pada tahap ini, grafik belum menampilkan apa pun karena belum ditentukan bagaimana data akan divisualisasikan.

```r
library(ggplot2)
df <- mtcars
ggplot(data = df)
```


2. Menentukan Aesthetic Mapping
   Selanjutnya, ditentukan pemetaan variabel ke dalam estetika visual menggunakan fungsi aes(). Grafik masih belum terlihat karena belum ditentukan bentuk visualnya.

```r
ggplot(data = mtcars, aes(x = wt, y = mpg))
```

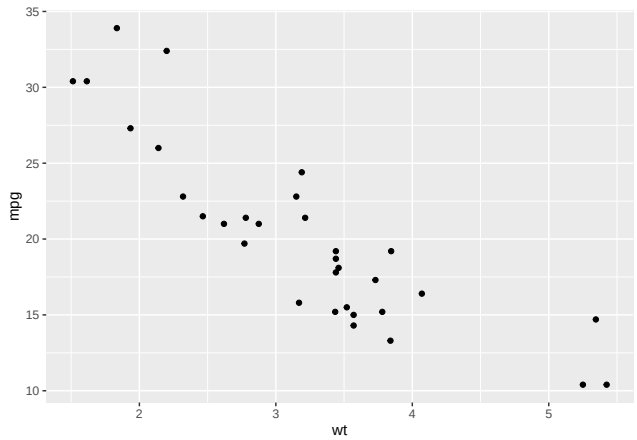


Artinya: wt dipetakan ke sumbu X dan mpg dipetakan ke sumbu Y

3. Menambahkan Geometri (Geom)
   Pada tahap ini ditambahkan geom untuk menentukan jenis grafik yang digunakan. Pada tahap

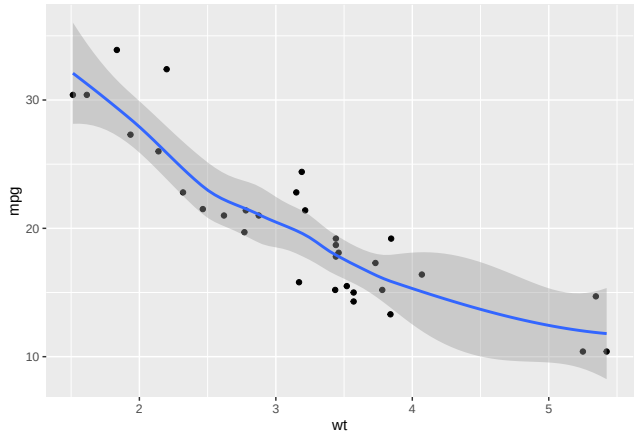
grafik mulai terlihat dalam bentuk scatter plot, di mana setiap titik merepresentasikan satu observasi.

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point()
```



4. Menambahkan Komponen Statistik (Opsional)
   Grammar of graphics memungkinkan penambahan statistical transformation untuk memperkaya informasi grafik.

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth()
```

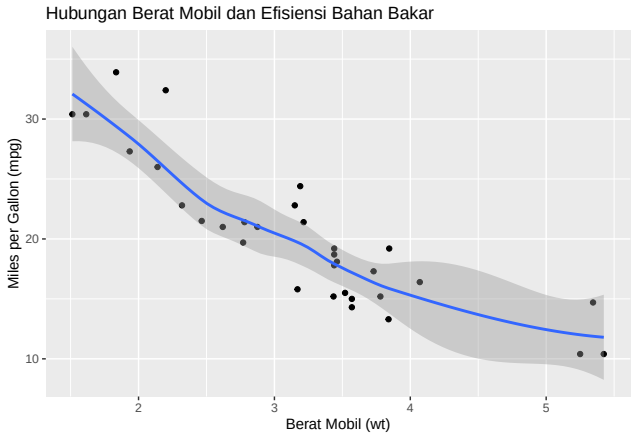


Lapisan geom_smooth() menambahkan garis tren untuk membantu melihat pola hubungan antara berat mobil dan efisiensi bahan bakar.

5. Menambahkan Skala dan Label
   Agar grafik lebih informatif, dapat ditambahkan judul dan label sumbu.

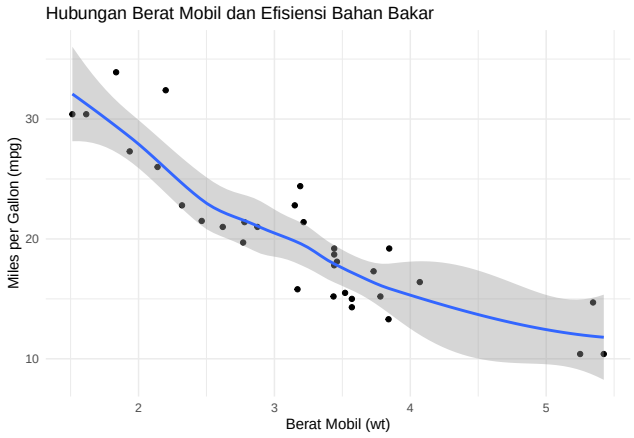
```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth() +
  labs(
    title = "Hubungan Berat Mobil dan Efisiensi Bahan Bakar",
```

```
    x = "Berat Mobil (wt)",
    y = "Miles per Gallon (mpg)"
  )
```



6. Menyesuaikan Tema (Opsional)
   Tema digunakan untuk mengatur tampilan visual grafik secara keseluruhan.

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth() +
  labs(
    title = "Hubungan Berat Mobil dan Efisiensi Bahan Bakar",
    x = "Berat Mobil (wt)",
    y = "Miles per Gallon (mpg)"
  ) +
  theme_minimal()
```



4.4.5 Faceting pada ggplot2

Faceting merupakan salah satu komponen penting dalam grammar of graphics yang digunakan untuk membagi satu grafik menjadi beberapa panel berdasarkan nilai suatu variabel kategorik. Dengan faceting, pengguna dapat membandingkan pola data antar kelompok secara lebih mudah dan sistematis.

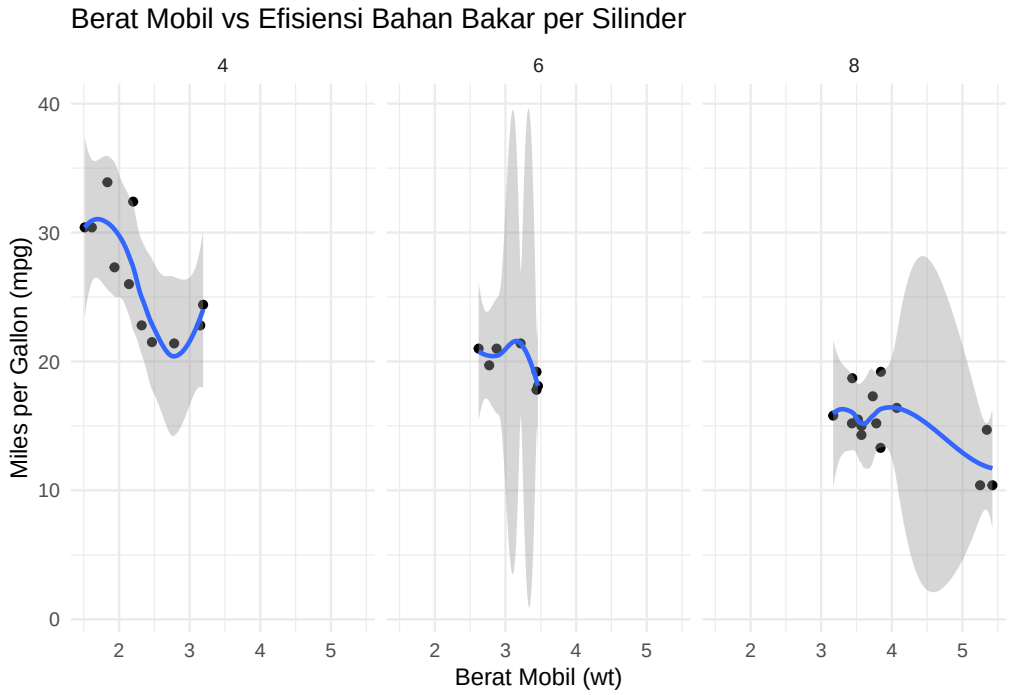
Dalam ggplot2, faceting memungkinkan visualisasi data yang sama ditampilkan dalam beberapa grafik

kecil (small multiples) dengan skala dan struktur yang konsisten.

Faceting bekerja dengan cara:
- menggunakan dataset yang sama, - memetakan variabel kategorik sebagai pembagi panel, - menampilkan satu grafik untuk setiap kategori.

Pendekatan ini sangat berguna untuk analisis perbandingan antar kelompok data. Berikut contoh, dataset mtcars digunakan untuk memvisualisasikan hubungan antara berat mobil (wt) dan efisiensi bahan bakar (mpg), yang dibagi berdasarkan jumlah silinder (cyl).

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~ cyl) +
  labs(
    title = "Berat Mobil vs Efisiensi Bahan Bakar per Silinder",
    x = "Berat Mobil (wt)",
    y = "Miles per Gallon (mpg)"
  ) +
  theme_minimal()
```

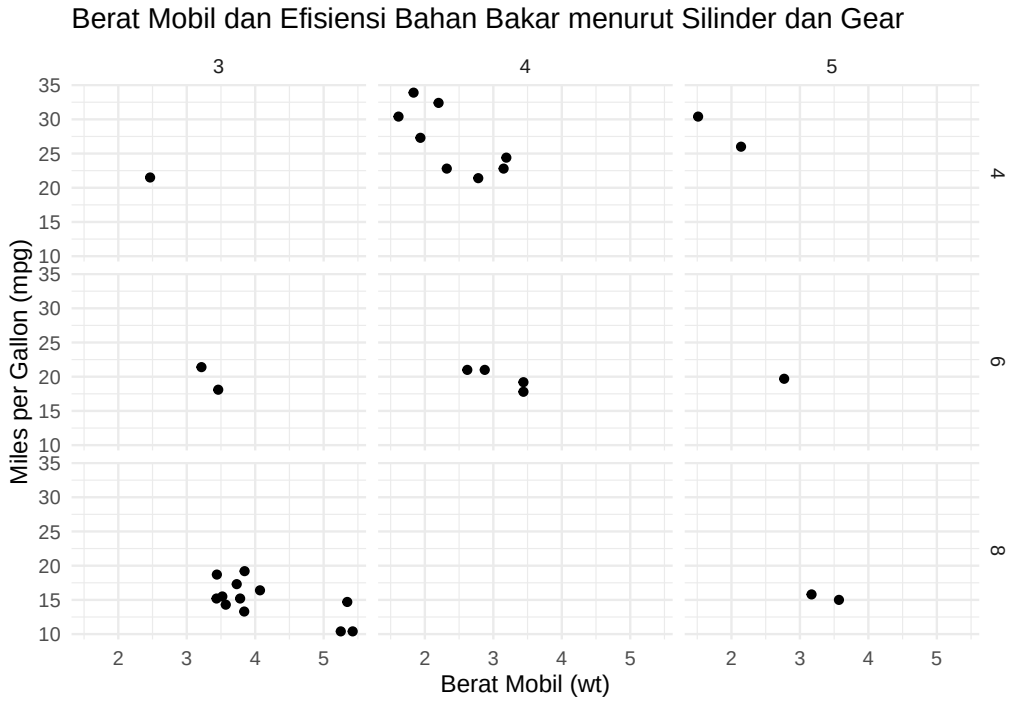


Pada grafik tersebut:

- facet_wrap(~ cyl) berarti: Buat beberapa grafik berdasarkan kategori cyl, di mana setiap panel menampilkan data untuk satu nilai cyl. Tanda ~ (tilde) Menunjukkan formula pemisah panel. Dalam konteks ini, tanda ~ dibaca sebagai "berdasarkan".
- Setiap panel merepresentasikan satu kategori jumlah silinder (cyl)
- Pola hubungan antara wt dan mpg dapat dibandingkan antar panel
- Skala dan struktur grafik tetap konsisten di setiap panel

Selain facet_wrap(), ggplot2 juga menyediakan facet_grid() untuk membuat faceting berdasarkan dua variabel.

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  facet_grid(cyl ~ gear) +
  labs(
    title = "Berat Mobil dan Efisiensi Bahan Bakar menurut Silinder dan Gear",
    x = "Berat Mobil (wt)",
    y = "Miles per Gallon (mpg)"
  ) +
  theme_minimal()
```



Faceting memungkinkan satu grafik ggplot2 diperluas menjadi beberapa panel dengan karakteristik yang sama, sehingga memudahkan analisis perbandingan. Dengan memanfaatkan facet_wrap() dan facet_grid(), pengguna dapat mengeksplorasi data secara lebih mendalam tanpa harus membuat grafik secara terpisah.

Praktikum

4.5 Praktikum

4.5.1 Praktikum 1

Berikut tabel latihan praktikum yang dirancang untuk membantu mahasiswa berlatih penggunaan kata kerja utama dplyr secara bertahap dan sistematis.

Tabel 4.2: Latihan Praktikum Manipulasi Data Menggunakan Paket `dplyr`

No.	Fungsi `dplyr`	Instruksi Latihan	Dataset
1	`select()`	Pilih hanya variabel `mpg`, `hp`, dan `cyl` dari dataset untuk analisis efisiensi mesin.	`mtcars`
2	`filter()`	Saring data mobil yang memiliki jumlah silinder lebih dari 6.	`mtcars`
3	`mutate()`	Buat variabel baru yang mengonversi konsumsi bahan bakar dari mil per galon (mpg) ke kilometer per liter.	`mtcars`
4	`arrange()`	Urutkan data berdasarkan nilai `mpg` dari yang tertinggi ke terendah.	`mtcars`
5	`group_by()`	Kelompokkan data berdasarkan jumlah silinder (`cyl`) untuk analisis lanjutan.	`mtcars`
6	`summarise()`	Hitung nilai rata-rata `mpg` untuk setiap kelompok jumlah silinder.	`mtcars`
7	`select()`	Pilih hanya variabel numerik yang berkaitan dengan ukuran bunga.	`iris`
8	`group_by()` + `summarise()`	Hitung rata-rata panjang sepal dan petal untuk setiap spesies bunga.	`iris`
9	`filter()`	Saring data bunga dengan panjang petal lebih dari 5 cm.	`iris`
10	`mutate()`	Tambahkan variabel baru berupa rasio antara panjang dan lebar sepal.	`iris`

4.5.2 Praktikum 2

Data yang digunakan berupa data tinggi badan, yang akan dianalisis untuk memperoleh statistik deskriptif serta divisualisasikan guna memahami pola dan sebaran data. Melalui praktikum ini, mahasiswa diharapkan mampu mengolah data secara sistematis, menginterpretasikan hasil analisis, dan menyajikannya dalam bentuk visual yang informatif.

```r
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(ggplot2)

tinggi <- c(
  188.7, 169.4, 178.6, 181.3, 179, 173.9, 190.1, 174.1, 195.2, 174.4,
  188, 197.9, 161.1, 172.2, 173.7, 181.4, 172.2, 148.4, 150.6, 188.2,
  171.9, 157.2, 173.3, 187.1, 194, 170.7, 172.4, 157.4, 179.6, 168.6,
  179.6, 182, 185.4, 168.9, 180, 157.8, 167.2, 166.5, 150.9, 175.4,
```

```
  177.1, 171.4, 182.6, 167.7, 161.3, 179.3, 166.9, 189.4, 170.7,
  181.6, 178.2, 167.2, 190.8, 181.4, 175.9, 177.8, 181.8, 175.9,
  145.1, 177.8, 171.3, 176.9, 180.8, 189, 167.7, 188, 178.4, 185.4,
  184.2, 182.2, 164.6, 174.1, 181.2, 165.5, 169.6, 180.8, 182.7,
  179.6, 166.1, 164, 190.1, 177.6, 175.9, 173.8, 163.1, 181.1,
  172.8, 173.2, 184.3, 183.2, 188.9, 170.2, 181.5, 188.9, 163.9,
  166.4, 163.7, 160.4, 175.8, 181.5
)

df_tinggi <- tibble(tinggi_badan = tinggi)
```

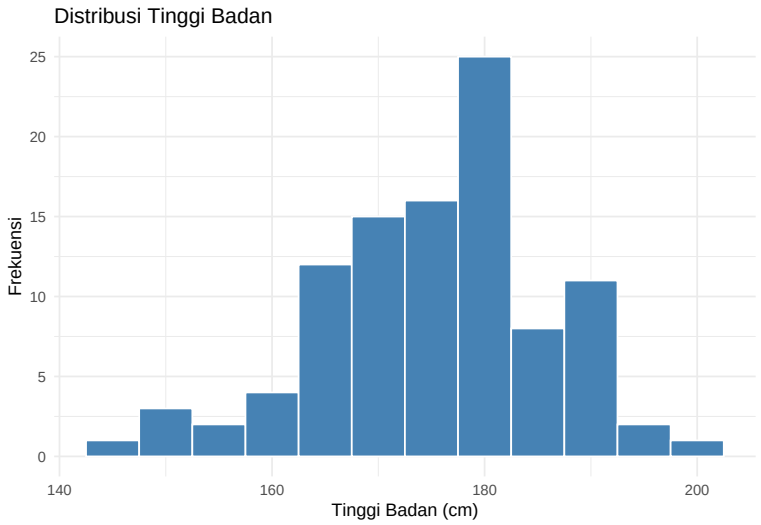
Maknanya:

- tinggi
  adalah vektor numerik yang berisi data tinggi badan (dalam cm).
- tibble()
  berfungsi membuat data frame modern (bagian dari tidyverse) yang lebih rapi saat ditampilkan dibanding data.frame().
- tinggi_badan = tinggi
  artinya:
  - membuat kolom bernama tinggi_badan
  - isinya diambil dari vektor tinggi
  - jadi df_tinggi adalah tabel satu kolom dengan nama kolom tinggi_badan.

```
str(df_tinggi)
```

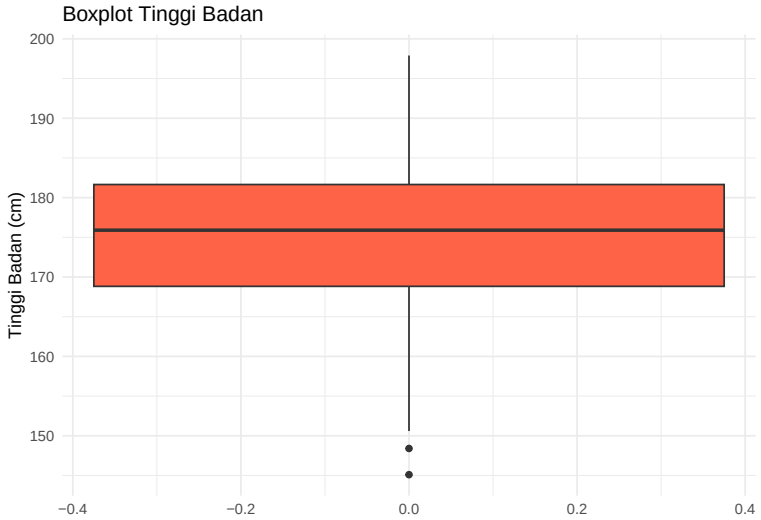
```
statistik <- df_tinggi %>%
  summarise(
    jumlah_data = n(),
    minimum = min(tinggi_badan),
    maksimum = max(tinggi_badan),
    rata_rata = mean(tinggi_badan),
    median = median(tinggi_badan),
    simpangan_baku = sd(tinggi_badan)
  )

statistik
statistik <- data.frame(statistik)
statistik
```

```
# Histogram
ggplot(df_tinggi, aes(x = tinggi_badan)) +
  geom_histogram(binwidth = 5, fill = "steelblue", color = "white") +
  labs(
    title = "Distribusi Tinggi Badan",
    x = "Tinggi Badan (cm)",
    y = "Frekuensi"
  ) +
  theme_minimal()
```



Histogram menunjukkan sebaran tinggi badan dan membantu melihat apakah data cenderung simetris atau miring.

```r
# Boxplot Tinggi Badan
ggplot(df_tinggi, aes(y = tinggi_badan)) +
  geom_boxplot(fill = "tomato") +
  labs(
    title = "Boxplot Tinggi Badan",
    y = "Tinggi Badan (cm)"
  ) +
  theme_minimal()
```



Boxplot memudahkan identifikasi: median, rentang interkuartil (IQR), kemungkinan outlier (misalnya tinggi sangat rendah atau sangat tinggi).

```r
# Klasifikasi tinggi badan
df_kategori <- df_tinggi %>%
  mutate(
    kategori = case_when(
      tinggi_badan < 160 ~ "Pendek",
      tinggi_badan < 180 ~ "Sedang",
```

```
      TRUE ~ "Tinggi"
    )
  )

# Menghitung jumlah data (frekuensi) pada setiap kategori di dalam kolom kategori
df_kategori %>%
  count(kategori)
```

Kesimpulan Umum :

- Data tinggi badan tersebar pada rentang yang cukup lebar.
- Nilai rata-rata dan median dapat digunakan sebagai gambaran tinggi badan tipikal.
- Boxplot dan histogram membantu melihat pola distribusi dan outlier.
- Pengelompokan kategori memudahkan analisis lanjutan (misalnya perbandingan kelompok).

4.5.3 Praktikum 3

Analisis Data Tinggi dan Berat Badan. Dataset women merupakan dataset bawaan R yang berisi data tinggi badan (height) dan berat badan (weight) dari 15 subjek perempuan dewasa. Dataset ini sering digunakan sebagai contoh hubungan antara dua variabel numerik.

```
library(dplyr)
library(ggplot2)

df_women <- as_tibble(women)
colnames(df_women)
df_women %>%
  summarise(
    n = n(),

    tinggi_min = min(height),
    tinggi_q1  = quantile(height, 0.25),
    tinggi_median = median(height),
    tinggi_mean   = mean(height),
    tinggi_q3  = quantile(height, 0.75),
    tinggi_max = max(height),
    tinggi_sd  = sd(height),

    berat_min = min(weight),
    berat_q1  = quantile(weight, 0.25),
    berat_median = median(weight),
    berat_mean   = mean(weight),
    berat_q3  = quantile(weight, 0.75),
    berat_max = max(weight),
    berat_sd  = sd(weight)
  ) %>%
  select(berat_max)
```

Pustaka

Intro to R Part 24: Hypothesis Testing, a. URL https://kaggle.com/code/hamelg/intro-to-r-part-24-hypothesis-testing.

Stats and R, b. URL https://statsandr.com/.

Jane M Horgan, editor. *Probability with R*. Wiley, 1 edition, January 2020. ISBN 978-1-119-53694-9 978-1-119-53696-3. doi: 10.1002/9781119536963. URL https://onlinelibrary.wiley.com/doi/book/10.1002/9781119536963.

Elinor Jones, Simon Harden, and Michael J Crawley. *The R book*. John Wiley & Sons, 2022.

Danielle Navarro. Introduction to probability. URL https://kpu.pressbooks.pub/learningstatistics/chapter/introduction-to-probability/. Book Title: Learning Statistics with R.

Måns Thulin. *Modern Statistics with R*. URL https://www-modernstatisticswithr-com.translate.goog/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc.

Mark PJ Van der Loo. *Learning RStudio for R statistical computing*. Packt Publishing Ltd, 2012.

Hadley Wickham, Garrett Grolemund, et al. *R for data science*, volume 2. O'Reilly Sebastopol, 2017.

Leland Wilkinson. ggplot2: elegant graphics for data analysis by wickham, h., 2011.