Dasar Pemrograman R

dimas

2021-01-02

Contents

Chapter 1

Pengantar R

R adalah salah satu bahasa pemrograman yang khusus dirancang dalam menggunakan metode-metode statistika. Program R memberikan banyak kemudahan bagi pemula yang ingin memulai belajar pemrograman. Walaupun program R adalah open source, namun dengan library yang sangat lengkap membuat program R menjadi mampu digunakan untuk mengolah data dengan motode yang lengkap.

1.1 Pendefinisian Objek

Objek merupakan sebuah wadah untuk menyimpan informasi yang telah didefinisikan oleh pengguna R. Penyimpanan informasi ke dalam sebuah objek memberikan kemudahan bagi pengguna R untuk memanggil informasi yang sama berulang kali hanya dengan memanggil objeknya saja. Seringkali sebuah informasi memiliki banyak komponen, sehingga jika harus memanggilnya berulang kali akan menjadi sangat tidak efektif. Sehingga dengan mendefinisikannya ke dalam sebuah objek akan mempercepat proses pengerjaannya.

Terdapat beberapa cara dalam mendefinisikan objek pada R seperti yang tersajikan pada tabel berikut:

Contoh penggunaannya adalah sebagai berikut:

Table 1.1: Operator Pendefinisian Objek pada R

Operator	Operasi
<-, «-, =	Mendefinisikan objek sebelah kiri
->, -»	Mendefinisikan objek sebelah kanan

```
x <- 4
y <- c(1, 4, 6, 8, 5)
z <- c("Indonesia", "Raya")
```

```
x
```

```
## [1] 4
```

```
y
```

```
## [1] 1 4 6 8 5
```

```
z
```

```
## [1] "Indonesia" "Raya"
```

1.2 Paket (Package)

Chapter 2

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 2. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2020) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

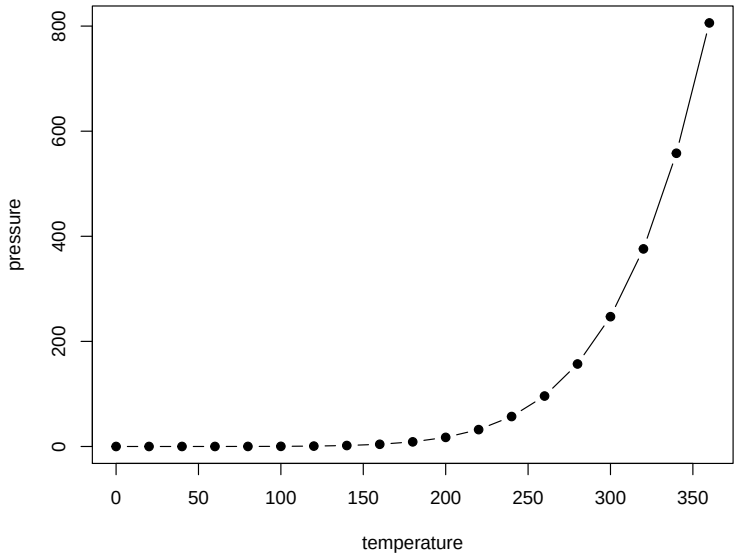


Figure 2.1: Here is a nice figure!

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 3

Tipe Data

Terdapat 5 tipe data pada bahasa pemrograman R. Setiap tipe data tersebut memiliki karakteristik sendiri sehingga tidak terjadi tumpang tindih dalam melakukan berbagai macam pengoperasiannya. Berikut adalah 6 tipe data pada bahasa pemrograman R:

3.1 Numerik (Numeric)

Tipe data numerik adalah tipe data yang berupa nilai/angka desimal. Tipe data ini merupakan tipe data yang dapat digunakan untuk melakukan operasi-operasi aritmatika seperti penjumlahan, pengurangan, perkailan, dsb. Jika kita definisikan objek `x` dengan suatu nilai/angka, maka tipe objek tersbut adalah `numeric`.

```
x <- 2.6
class(x)
```

```
## [1] "numeric"
```

Bahkan R akan mendefinisikan objek dengan tipe `numeric` jika berikan angka tanpa desimal.

```
x <- 5
class(x)
```

```
## [1] "numeric"
```

3.2 Bilangan Bulat (Integer)

Seperti yang kita tahu bahwa pendefinisian angka pada suatu objek akan secara otomatis membuat objek tersebut bertipe `numeric`. Sedangkan untuk mendefinisikan objek dengan tipe `integer` harus mendefinisikan secara khusus objek tersebut dengan perintah `as.integer`.

```
x <- as.integer(5)
class(x)
```

```
## [1] "integer"
```

Selain itu kita dapat mendefinisikan sebuah objek yang bertipe `integer` dengan menambahkan huruf `L` kapital pada akhir angka.

```
x <- 3L
class(x)
```

```
## [1] "integer"
```

Bagaimana jika kita berikan angka desimal pada objek yang kita definisikan sebagai `integer`?

```
x <- as.integer(3.76)
```

3.3 Bilangan Kompleks (Complex)

Bilangan kompleks dalam matematika adalah bilangan yang didefinisikan dengan $a + bi$, dengan a dan b adalah bilangan real. Sedangkan i adalah bilangan imajiner dan menyebabkan bi menjadi imajiner. Bilangan imajiner sendiri memiliki sifat $i^2 = 1$. Kita harus secara langsung mendefinisikan objek sebagai bilangan kompleks agar mendapatkan sebuah objek yang bertipe `complex`.

```
x <- 2 + 4i
class(x)
```

```
## [1] "complex"
```

3.4 Logika (Logical)

Objek dengan tipe logika adalah objek yang hanya memiliki 2 nilai saja yaitu `TRUE` dan `FALSE`.

```
x <- TRUE
class(x)
```

```
## [1] "logical"
```

3.5 Teks (Character)

Pendefinisian objek dengan tipe teks (`character`) merupakan hal yang cukup mudah. Kita perlu menambahkan tanda petik `"` pada awal dan akhir teks. Setelah itu objek akan terdefinisikan sebagai `character`.

```
x <- "Aplikasi Komputer"
class(x)
```

```
## [1] "character"
```

```
x
```

```
## [1] "Aplikasi Komputer"
```

Apakah tipe data dari objek yang didefinisikan dengan nilai `"2.4"`?

3.6 Faktor (Factor)

Faktor adalah tipe data pada bahasa pemrograman R yang digunakan untuk mendefinisikan sebuah objek menjadi sebuah objek dengan tipe data kategorik. Perintah yang digunakan untuk merubah sebuah objek menjadi sebuah faktor adalah `factor()`.

```
x <- factor(c(1, 2, 3))
y <- factor(c("SD", "SMP", "SMA", "SMA", "PT"))
x
```

```
## [1] 1 2 3
## Levels: 1 2 3
```

```
y
```

```
## [1] SD  SMP SMA SMA PT
## Levels: PT SD SMA SMP
```

Sebuah objek yang telah didefinisikan sebagaik faktor akan memiliki `levels` yang merupakan daftar kategori yang terdapat pada objek tersebut. Kita dapat menggunakan perintah `levels` untuk dapat memunculkan `levels` nya saja.

```
levels(x)
```

```
## [1] "1" "2" "3"
```

```
levels(y)
```

```
## [1] "PT"  "SD"  "SMA" "SMP"
```

- Apa perbedaan *output* dari objek yang bertipe `factor` dengan objek yang bertipe `numeric`?
- Apa perbedaan *output* dari objek yang bertipe `factor` dengan objek yang bertipe `character`?

Chapter 4

Struktur Data

Struktur data pada bahasa pemrograman R adalah sebuah konsep yang digunakan untuk menyimpan data berdasarkan kebutuhannya. Berdasarkan struktur datanya, terdapat 4 struktur data yang dapat kita definisikan pada program R:

- Vektor (`vector`)
- Matriks (`matrix`)
- Pendafaran (`list`)
- Kerangka Data (`data.frame`)

4.1 Vektor (Vector)

Vektor adalah struktur data yang paling sederhana pada bahasa pemrograman R. Vektor memuat barisan data dengan tipe data yang sama. Pendefinisian vektor dilakukan dengan menggunakan perintah `c()`. Semua data ditulis didalam tanda kurung dan setiap data dipisahkan dengan tanda koma `,`.

```
x <- c(1:5)
x
```

```
## [1] 1 2 3 4 5
```

```
y <- c("Hipertensi", "Diabetes", "Asam Urat")
y
```

```
## [1] "Hipertensi" "Diabetes"   "Asam Urat"
```

```r
length(x)
```

```
## [1] 5
```

```r
length(y)
```

```
## [1] 3
```

Apa yang terjadi apabila sebuah vektor diisi dengan 2 tipe data? misalnya `c(12, 4, TRUE)`

Selanjutnya kita dapat memanggil anggota dari sebuah objek `vector` dengan menggunakan tanda `[x]` setelah objek dengan `x` adalah nilai yang menyatakan *data ke-x*.

```r
x[2]
```

```
## [1] 2
```

```r
y[3]
```

```
## [1] "Asam Urat"
```

4.2 Matriks (Matrix)

Sama halnya dengan vektor, matriks merupakan struktur data yang hanya dapat menyimpan 1 tipe data saja. Perbedaan antara struktur data 'matrix' dengan 'vector' berada pada dimensi datanya. Vektor merupakan struktur data berdimensi 1 (hanya memiliki panjang data). Sedangkan matriks adalah struktur data yang berdimensi 2 (memiliki dimensi dan panjang data).

Pendefinisian matriks dilakukan dengan menggunakan perintah `matrix` dengan `syntax: matrix(data, nrow, ncol, byrow=FALSE)`.

```r
x <- matrix(
  c(1:8), 2, 4
)
x
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    5    7
## [2,]    2    4    6    8
```

```
dim(x)
```

```
## [1] 2 4
```

dengan:

- `data` adalah data yang akan kita gunakan
- `nrow` adalah jumlah baris
- `ncol` adalah jumlah kolom
- `byrow` adalah perintah opsional untuk memilih agar data yang kita miliki didaftarkan berdasarkan baris atau kolom

Apakah yang akan terjadi apabila kita menggunakan `byrow=FALSE` dan `byrow=TRUE`?

Berbeda dengan objek yang berupa `vector`, pemanggilan data pada objek `matrix` menggunakan tanda `[x,y]` dimana `x` adalah *data baris ke-x* dan `y` adalah *data kolom ke-y*. Apabila kita menggunakan tanda `[x,]` saja, maka kita memanggil semua data pada *baris ke-x*. Sedangkan apabila kita menggunakan tanda `[,y]` saja, maka kita memanggil semua data pada *kolom ke-y*.

```
x[2,3]
```

```
## [1] 6
```

```
x[2,]
```

```
## [1] 2 4 6 8
```

```
x[,4]
```

```
## [1] 7 8
```

```
x[6]
```

```
## [1] 6
```

Catatan: `x` dan `y` pada `[x,y]` dapat berupa `vector`.

4.3 Daftar (List)

Struktur data `list` dalam R adalah struktur data yang dapat mendaftarkan beberapa objek sekaligus tanpa perlu khawatir dengan tipe data yang berbeda. Struktur data 'list' dapat juga dikatakan sebagai vektor yang dapat menyimpan berbagai macam objek. Pendefinisian list dilakukan dengan menggunakan perintah `list()`.

```
a <- c(1, 3, 6, 2)
b <- c("Apa", "kabar", "anda", "hari", "ini", "?")
c <- c(TRUE, FALSE, FALSE, FALSE, TRUE)
x <- list(a=a, b=b, c=c)
x
```

```
## $a
## [1] 1 3 6 2
##
## $b
## [1] "Apa"   "kabar" "anda"  "hari"  "ini"   "?"
##
## $c
## [1]  TRUE FALSE FALSE FALSE  TRUE
```

Selanjutnya kita dapat memanggil masing-masing objek pada sebuah list dengan `syntax: ListObject$ObjekInList` atau dengan menggunakan `[[x]]` dimana `x` adalah sebuah angka yang merujuk pada **daftar ke berapa**.

```
x$a
```

```
## [1] 1 3 6 2
```

```
x[[3]]
```

```
## [1]  TRUE FALSE FALSE FALSE  TRUE
```

4.4 Kerangka Data (Data Frame)

Struktur data yang berupa *data frame* (`data.frame`) merupakan struktur data yang akan paling sering kita gunakan dalam pengolahan data. Struktur data ini digunakan untuk mendefinisikan sebuah tabel data yang mana setiap kolom adalah nama-nama objek/variabel pada *data frame* tersebut. Setiap objek/variabel dalam `data.frame` merupakan sebuah `vector`. Artinya setiap

Table 4.1: Data Mahasiswa 2020

nama	jk	tb	bb
Ahmad	Laki-laki	170	70
Ganjar	Laki-laki	169	67
Lusi	Perempuan	160	45
Andina	Perempuan	154	40
Elok	Perempuan	163	52

objek/variabel dalam `data.frame` hanya dapat memiliki 1 tipe data saja. Selain itu setiap objek/variabel yang berada dalam `data.frame` harus memiliki jumlah data (`length()`) yang sama.

Perintah `data.frame()` adalah perintah yang digunakan untuk medefinisikan sebuah objek sebagai sebuah data frame.

```
nama <- c("Ahmad", "Ganjar", "Lusi", "Andina", "Elok")
jk   <- factor(c("Laki-laki", "Laki-laki", "Perempuan", "Perempuan", "Perempuan"))
tb   <- c(170, 169, 160, 154, 163)
bb   <- c(70, 67, 45, 40, 52)
x    <- data.frame(nama, jk, tb, bb)
x
```

```
##     nama        jk  tb bb
## 1  Ahmad Laki-laki 170 70
## 2 Ganjar Laki-laki 169 67
## 3   Lusi Perempuan 160 45
## 4 Andina Perempuan 154 40
## 5   Elok Perempuan 163 52
```

Selanjutnya untuk memanggil objek/variabel pada sebuah `data.frame` mirip dengan memanggil data pada sebuah `matrix`. Selain itu kita dapat pula memanggil dengan cara menyebutkan nama objek/variabel pada `data.frame` tersebut.

```
x[,2]
```

```
## [1] Laki-laki Laki-laki Perempuan Perempuan Perempuan
## Levels: Laki-laki Perempuan
```

```
x[4,]
```

```
##     nama        jk  tb bb
## 4 Andina Perempuan 154 40
```

```
x$nama
```

```
## [1] "Ahmad"  "Ganjar" "Lusi"   "Andina" "Elok"
```

```
x[,c('jk', 'nama')]
```

```
##          jk   nama
## 1 Laki-laki  Ahmad
## 2 Laki-laki Ganjar
## 3 Perempuan   Lusi
## 4 Perempuan Andina
## 5 Perempuan   Elok
```

Chapter 5

Operator - operator pada R

Operator - operator pada R adalah perintah untuk melakukan suatu operasi seperti operasi artimatika, logika dan relasi. Secara umum terdapat 4 tipe operator pada R, yaitu:

- Operator Aritmatika
- Operator Logika
- Operator Relasi

5.1 Operator Aritmatika

Operator artimatika adalah operator yang digunakan untuk melakukan komputasi dengan menggunakan operasi - operasi matematika seperti penjumlahan, pengurangan, dll. Berikut adalah daftar operator aritmatika pada R:

Operator umumnya digunakan untuk pengoperasian pada objek bertipe `numeric` atau `integer` dengan struktur data berupa `vector` dan `matrix`.

Table 5.1: Operator Aritmatika pada R

Operator	Operasi
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
^	Perpangkatan
%%	Modulo

Table 5.2: Hasil Penggunaan Operator pada R

Operator	Operasi	Hasil
+	x + y	8.0
-	x - y	-2.0
*	x * y	15.0
/	x / y	0.6
^	x ^ y	243.0
%%	x %% y	3.0

Table 5.3: Operator Logika pada R

Operator	Operasi
!	Bukan/Ingkaran
&	Logika DAN pada masing-masing anggota objek
&&	Logika DAN
|	Logika ATAU
||	Logika ATAU pada masing-masing anggota objek

Berikut adalah contoh penggunaannya:

```
x <- 3
y <- 5
```

5.2 Operator Logika

Operator logika akan menghasilkan nilai `TRUE` dan `FALSE`. Seperti halnya logika matematika, terdapat 2 operator logika yaitu *dan* (`AND`), *atau* (`OR`) dan ingkaran (`NOT`). Berikut adalah daftar operator logika pada R:

Berikut adalah contoh hasil penggunaannya:

```
x <- c(TRUE, FALSE, 5)
y <- c(TRUE, FALSE, FALSE)
z <- c(FALSE, TRUE, TRUE)

!x
```

```
## [1] FALSE  TRUE FALSE
```

Table 5.4: Hasil Penggunaan Operator Logika

P1	P2	NegasiP1	NegasiP2	DAN	ATAU
TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
FALSE	FALSE	TRUE	TRUE	FALSE	FALSE

```
x&y
```

```
## [1]  TRUE FALSE FALSE
```

```
x&&z
```

```
## [1] FALSE
```

```
x|y
```

```
## [1]  TRUE FALSE  TRUE
```

```
x||z
```

```
## [1] TRUE
```

5.3 Operator Relasi

Operator relasi adalah operator yang digunakan untuk membandingkan 2 buah nilai/objek. Hasil *output* dari operator relasi ini adalah `TRUE` dan `FALSE`. Berikut adalah daftar operator relasi:

Operator relasi dapat dioperasikan layaknya operator logika. Berikut adalah contoh penggunaannya:

```
x <- 3
y <- 7
z <- 7
x<y
```

```
## [1] TRUE
```

Table 5.5: Operasi Relasi pada R

Operator	Operasi
<	Kurang dari
<=	Kurang dari sama dengan
>	Lebih dari
>=	Lebih dari sama dengan
==	Sama dengan
!=	Tidak sama dengan

```
x>y
```

```
## [1] FALSE
```

```
x<=y
```

```
## [1] TRUE
```

```
y<=z
```

```
## [1] TRUE
```

```
x!=y
```

```
## [1] TRUE
```

```
y==z
```

```
## [1] TRUE
```

Chapter 6

Plot dengan ggplot2

Plot adalah salah satu hal mendasar yang dapat digunakan untuk menggambarkan distribusi data. Program R pun memiliki perintah - perintah untuk membuat sebuah plot. Namun perintah - perintah tersebut masih merupakan perintah yang sederhana. Hasilnya, gambar kurang menarik untuk disajikan pada khalayak umum.

`ggplot2` merupakan sebuah paket yang dapat digunakan pada program R dengan cara melakukan instalasi terlebih dahulu menggunakan perintah `install.package("ggplot2")`. Paket `ggplot2` memberikan fasilitas bagi penggunanya dalam membuat sebuah plot yang menarik untuk dilihat. Setelah anda memahami jenis-jenis plot beserta cara penggunaannya, kita dapat menggunakan paket `ggpubr` yang telah disusun khusus untuk kebutuhan publikasi/penerbitan.

Setelah instalasi paket `ggplot2` selesai, selanjutnya kita dapat memanggil paket tersebut dengan perintah

```
library(ggplot2)
```

6.1 Bar Plot

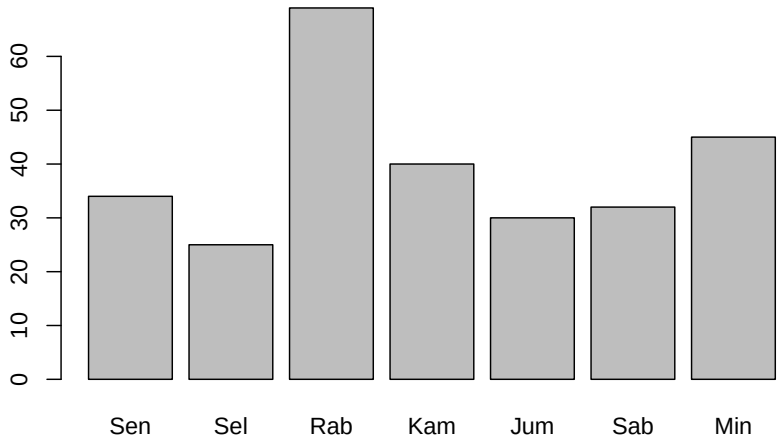
Grafik batang merupakan grafik yang digunakan untuk menunjukkan frekuensi dari dara. Grafik batang lebih sering digunakan untuk menunjukkan perbedaan jumlah dari data yang bertipe kategorik seperti jenis kelamin, asal daerah, status perniahan, dsb. Selanjutnya kita akan membuat sebuah grafik batang dengan menggunakan data berikut:

```
jpc <- c(34, 25, 69, 40, 30, 32, 45)
hr  <- c("Sen", "Sel", "Rab", "Kam", "Jum", "Sab", "Min")
x   <- data.frame(Hari = hr, Pasien = jpc)
x
```

```
##   Hari Pasien
## 1  Sen     34
## 2  Sel     25
## 3  Rab     69
## 4  Kam     40
## 5  Jum     30
## 6  Sab     32
## 7  Min     45
```

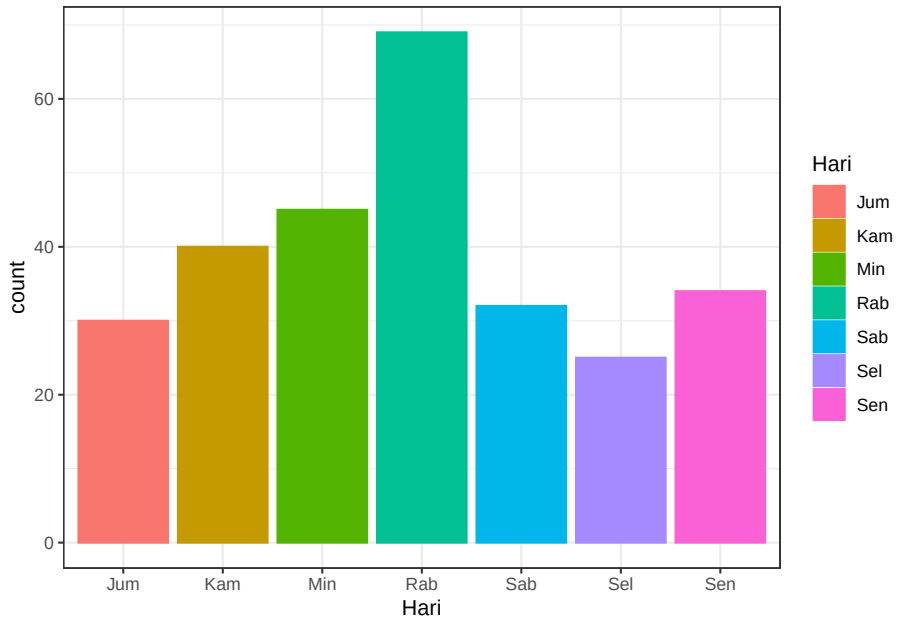
Terdapat perintah `barplot` dalam `R` yang merupakan perintah yang sudah tersedia saat melakukan instalasi `R`. Walapun perintah `barplot` dapat kita gunakan, namun hasil yang diberikan sangat sederhana dan lebih cocok digunakan untuk sebuah laporan. Berikut adalah cara menggunakan perintah `barplot`:

```
barplot(x$Pasien,
        names.arg = x$Hari)
```



Selanjutnya kita akan menggunakan paket `ggplot2` untuk menyajikan grafik batang yang lebih menarik. Berikut adalah perintah untuk membuat grafik batang dengan menggunakan paket `ggplot2`:

```
p <- ggplot(x, aes(Hari)) +
  geom_bar(aes(weight=Pasien, fill=Hari, colour=Hari)) +
  theme_bw()
p
```



penjelasan:

- `ggplot(x, aes(Hari))` adalah perintah untuk membuat sebuah objek `ggplot` dari variabel `Hari` pada data `x`
- `geom_bar(aes(weight=Pasien, fill=Hari, colour=Hari))`
  - `geom_bar()` adalah perintah untuk membuat grafik batang menggunakan `ggplot`
  - `weight` adalah banyak datanya (dalam kasus yang kita kerjakan: banyaknya Pasien setiap hari)
  - `fill` bertujuan untuk memberi warna batang (harus sama dengan `aes(Hari)` pada `ggplot` agar setiap batang memiliki warna yang berbeda)
  - `colour` bertujuan untuk memberi warna garis (harus sama dengan `aes(Hari)` pada `ggplot` agar setiap batang memiliki warna yang berbeda)
- `theme_bw()` bertujuan untuk menentukan tema `black and white` pada grafik

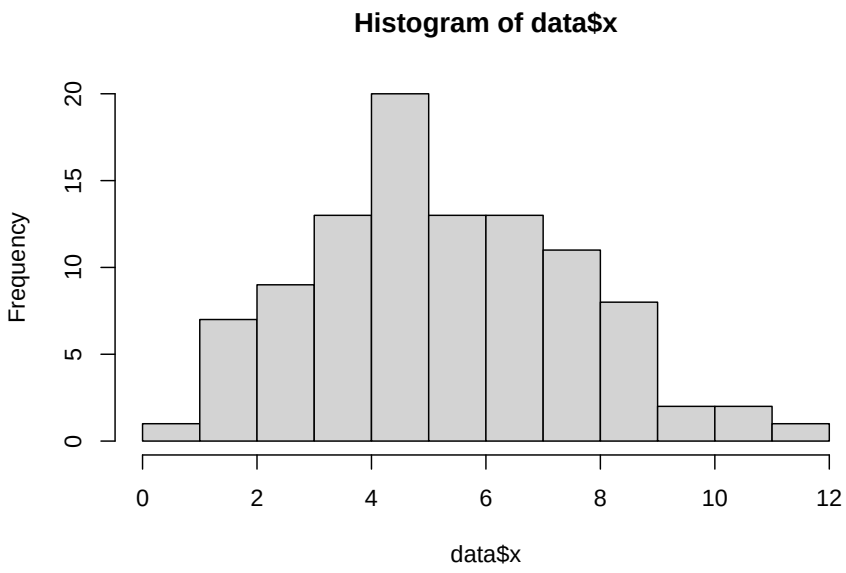
6.2 Grafik Histogram

Histogram merupakan grafik batang yang dapat menunjukkan seberapa sering suatu nilai yang berbeda terjadi. Histogram lebih sering digunakan untuk melihat distribusi dari suatu data. Berbeda dengan grafik batang, kita perlu menggunakan data numerik dalam membuat sebuah histogram. Berikut adalah data acak yang dibangkitkan dengan perintah `rnorm`:

```
data <- data.frame(x = rnorm(100,5,2))
```

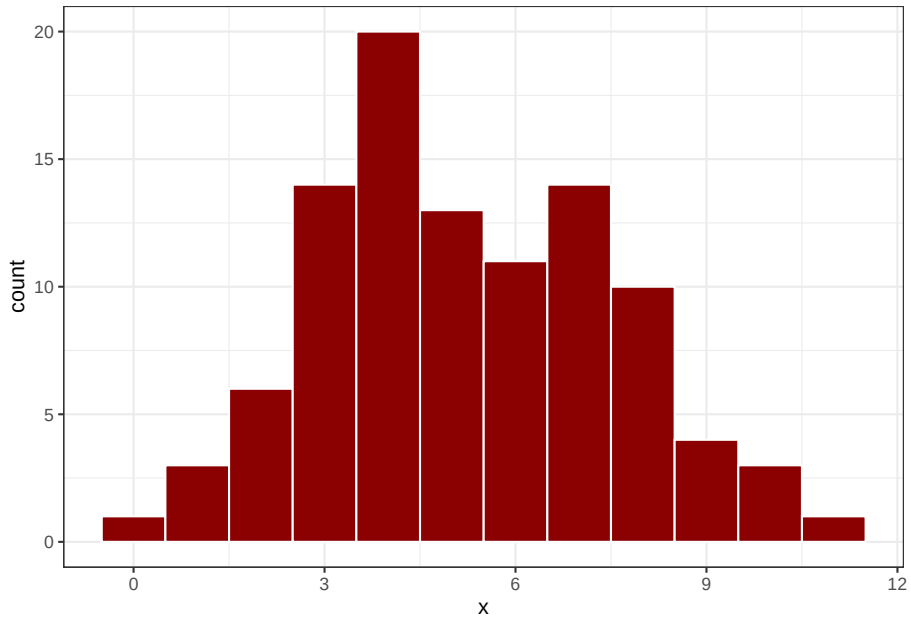
Kita dapat menggunakan perintah `hist` yang telah tersedia saat melakukan instalasi `R`.

```
hist(data$x)
```



Selanjutnya kita akan menggunakan perintah yang tersedia pada paket `ggplot2`.

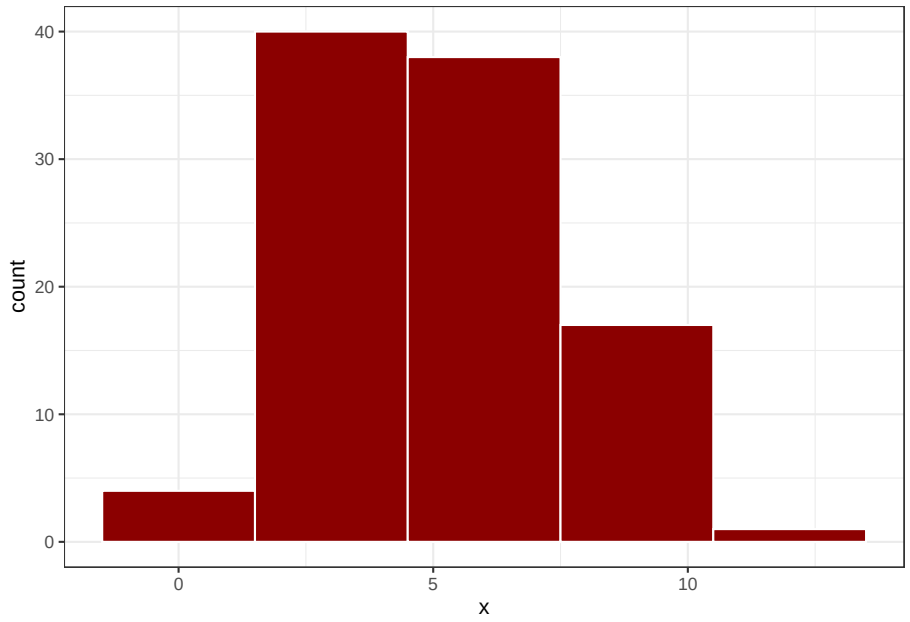
```
p <- ggplot(data, aes(x)) +
  geom_histogram(binwidth = 1,
                 color = "white",
                 fill= "darkred") +
  theme_bw()
p
```



penjelasan: * `ggplot(data, aes(x))` adalah perintah untuk membuat sebuah objek `ggplot` dari objek `x` pada data `data` * `geom_histogram(binwidth = 1, color = "white", fill= "darkred")` - `geom_histogram()` adalah perintah untuk membuat histogram menggunakan `ggplot` - `bandwidth` adalah lebar dari masing-masing batang - `fill` bertujuan untuk memberi warna batang (dalam kasus ini kita akan berikan warna yang sama untuk semua batang) - `colour` bertujuan untuk memberi warna garis (dalam kasus ini kita akan berikan warna yang sama untuk semua garis) * `theme_bw()` bertujuan untuk menentukan tema `black and white` pada plot

Selanjutnya bandingkan dengan histogram yang memiliki `bandwith` berbeda.

```
p <- ggplot(data, aes(x)) +
  geom_histogram(binwidth = 3,
                 color = "white",
                 fill= "darkred") +
  theme_bw()
p
```

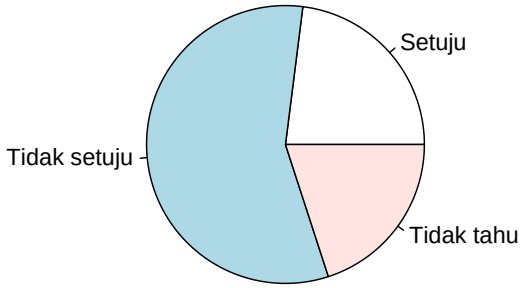


6.3 Pie Plot

```
jumlah <- c(23, 57, 20)
label <- c("Setuju", "Tidak setuju", "Tidak tahu")
x <- data.frame(label, jumlah)
```

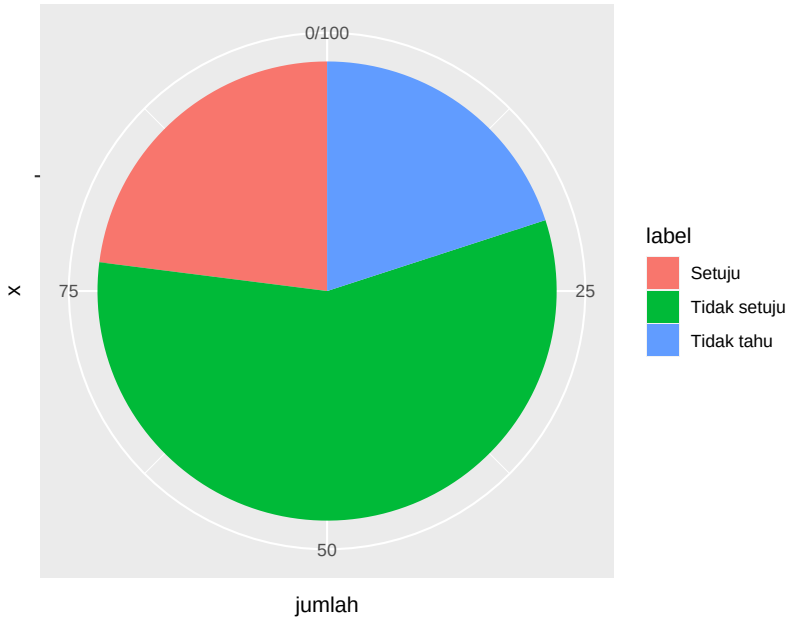
Menggunakan perintah `pie`

```
pie(x$jumlah, labels = x$label)
```



Menggunakan paket `ggplot2`

```
p <- ggplot(x, aes(x="", y=jumlah, fill=label)) +
  geom_bar(width = 1, stat = "identity")
p + coord_polar("y", start = 0)
```



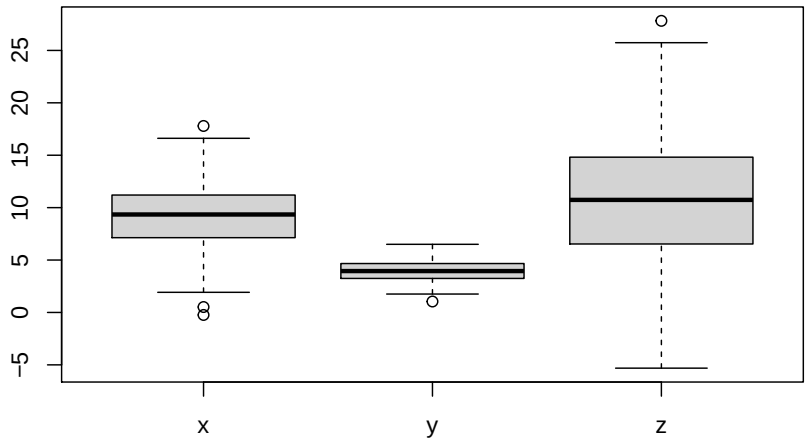
penjelasan:

6.4 Box Plot

```
x <- rnorm(250, 9, 3)
y <- rnorm(250, 4, 1)
z <- rnorm(250, 11, 6)
```

Menggunakan perintah `boxplot`

```
boxplot(x, y, z,
        names = c("x", "y", "z"))
```

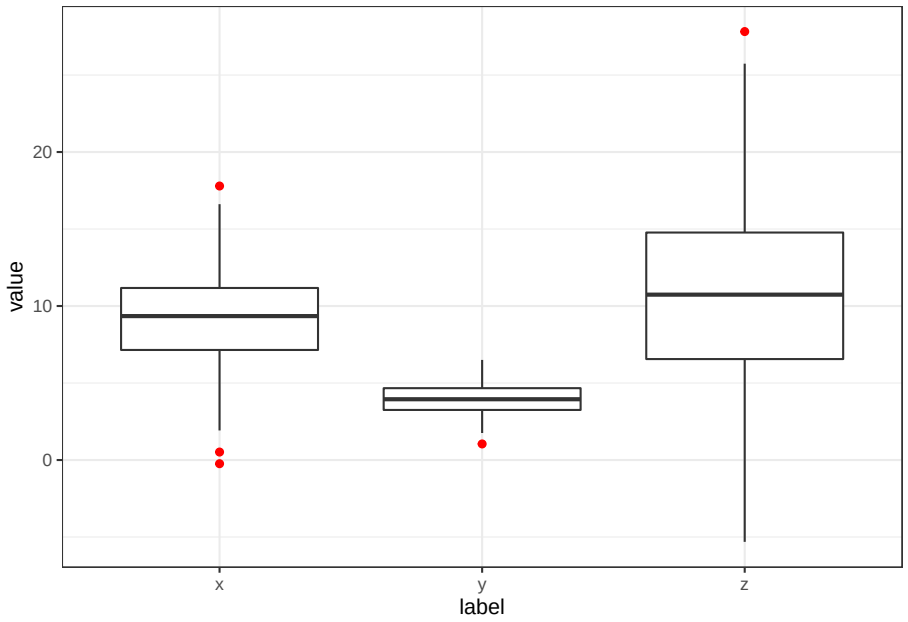


Menggunakan paket `ggplot2`

```
data <- data.frame(label = c(rep(c("x","y","z"),
                                 each=250)),
                   value = c(x, y, z))
head(data)
```

```
##   label     value
## 1     x  7.300475
## 2     x 10.703271
## 3     x  6.215125
## 4     x  5.082916
## 5     x 14.788104
## 6     x  9.592218
```

```
p <- ggplot(data, aes(x=label, y=value)) +
  geom_boxplot(outlier.colour = "red") +
  theme_bw()
p
```

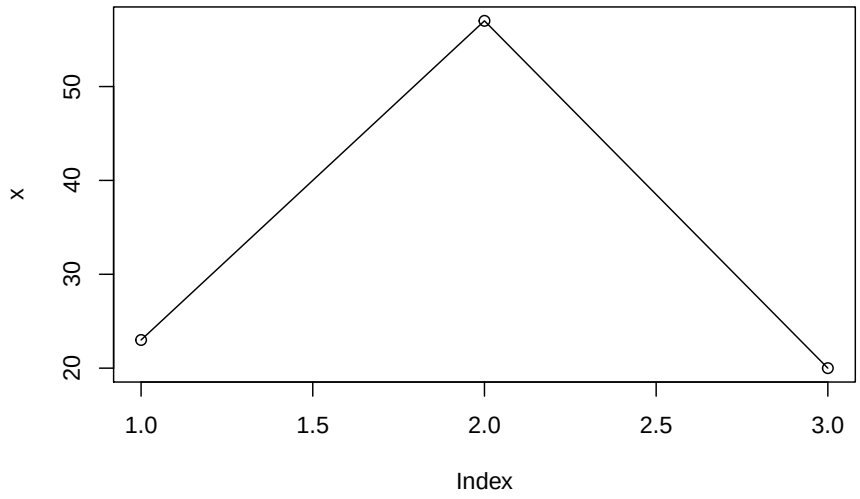


6.5 Line Plot

```
x <- c(23, 57, 20)
names(x) <- c("Setuju", "Tidak setuju", "Tidak tahu")
```

Menggunakan perintah `pie`

```
plot(x, type = "o")
```



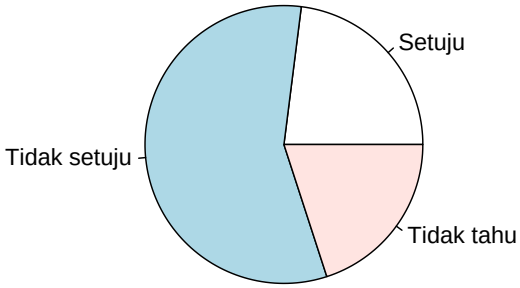
Menggunakan paket `ggplot2`

6.6 Scatter Plot

```
x <- c(23, 57, 20)
names(x) <- c("Setuju", "Tidak setuju", "Tidak tahu")
```

Menggunakan perintah `pie`

```
pie(x, labels = names(x))
```



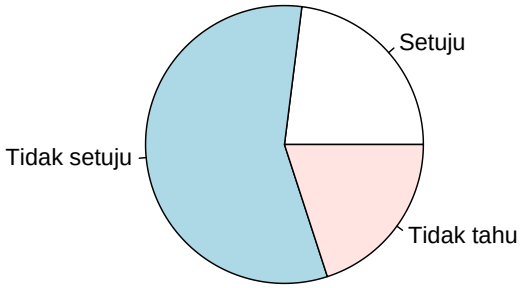
Menggunakan paket `ggplot2`

6.7 Area Plot

```
x <- c(23, 57, 20)
names(x) <- c("Setuju", "Tidak setuju", "Tidak tahu")
```

Menggunakan perintah `pie`

```
pie(x, labels = names(x))
```



Menggunakan paket `ggplot2`

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.21.6.