

## 2-Vektor-Matriks-DataFrame

Ashari Ramadhan

11/19/2020

### Data

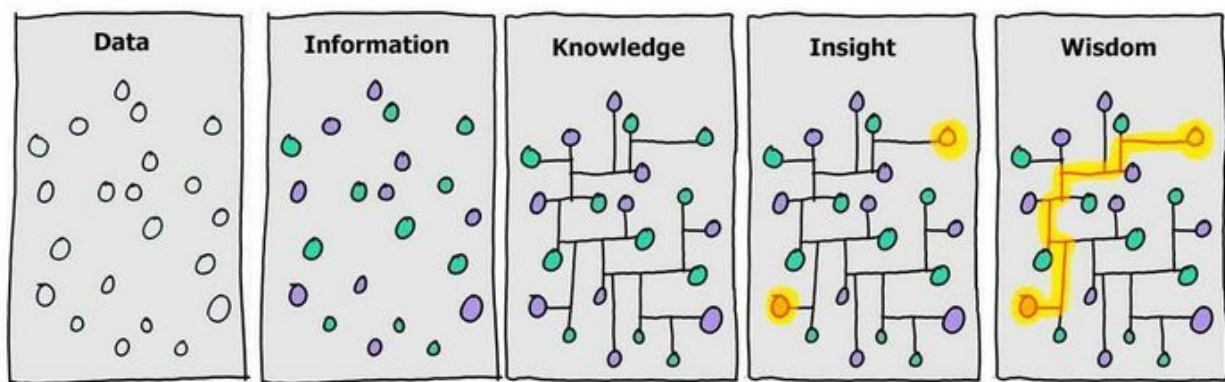


Figure 1: Data, Information, Knowledge, Wisdom

#### Contoh

Data = air jatuh dari langit Informasi = Jakarta hujan Knowledge = Jika Jakarta hujan, maka akan macet dan banjir Wisdom = Jangan keluar rumah jika tidak mendesak

Pada bahasa R, data adalah objek yang memiliki suatu atribut dan berbagai fungsionalitas. Sifat data ditentukan oleh type data dan mode data. Ada berbagai type data yang dikenal oleh R, antara lain vektor, matriks, list, data frame, array, dan factor. Pada kesempatan kali ini akan dibahas beberapa type data

### Vektor

Vektor dibuat dengan menggunakan fungsi `c()` (concatenate)

```
#membuat vektor numerik
vektor_1 <- c(15,21,32,43,51)
vektor_1
```

```
## [1] 15 21 32 43 51
```

```
#membuat vektor karakter
vektor_2 <- c('terima kasih', 'thank you', 'gracias', 'vielen dank')
vektor_2
```

```
## [1] "terima kasih" "thank you"      "gracias"      "vielen dank"
```

Kita juga dapat memberi nama untuk setiap nilai pada vektor

```
vektor_3 <- c('idn' = 'terima kasih', 'eng' = 'thank you',  
             'spn' = 'gracias', 'ger' = 'vielden dank')  
vektor_3
```

```
##           idn           eng           spn           ger  
## "terima kasih"    "thank you"    "gracias" "vielden dank"
```

Bisa juga ditulis seperti ini

```
vektor_4 <- c('terima kasih', 'thank you', 'gracias', 'vielen dank')  
names(vektor_4) <- c('idn', 'eng', 'spn', 'ger')  
vektor_4
```

```
##           idn           eng           spn           ger  
## "terima kasih"    "thank you"    "gracias" "vielen dank"
```

## Slicing

Slicing merupakan teknik memilih data dari sebuah set data. Sebelumnya kita memiliki vektor\_1 yang berisi nilai numerik. Setiap nilai pada vektor memiliki urutan / index yang di mulai dari 1 hingga sesuai banyaknya isi vektor. Untuk melakukan slicing kita menggunakan [] lalu mengisi index vektor.

```
vektor_1
```

```
## [1] 15 21 32 43 51
```

```
# memilih data ke 3 pada vektor_1  
vektor_1[3]
```

```
## [1] 32
```

```
# memilih data ke 3 sampai ke 5 pada vektor_1  
vektor_1[3:5]
```

```
## [1] 32 43 51
```

```
# memilih semua data pada vektor_1 kecuali data ke 2  
vektor_1[-2]
```

```
## [1] 15 32 43 51
```

```
# memilih semua data pada vektor_1 kecuali data ke 2  
vektor_1[6]
```

```
## [1] NA
```

Hasil terakhir NA karena memang tidak ada nilai pada index ke 6 sebab panjang vektor hanya sampai 5

## Merubah isi vektor

Sebelumnya kita memiliki vektor\_2 yang merupakan ucapak terima kasih pada beberapa bahasa.

```
vektor_2

## [1] "terima kasih" "thank you"      "gracias"      "vielen dank"

# mengganti nilai ke 2 pada vektor_2

vektor_2[2] <- 'Arigatogozaimashita'
vektor_2
```

```
## [1] "terima kasih"      "Arigatogozaimashita" "gracias"
## [4] "vielen dank"
```

Dari output di atas dapat dilihat “thank you” berubah menjadi “Arigatogozaimashita”

## menambah panjang vektor

vektor\_2 memiliki panjang 4

```
length(vektor_2)
```

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

Misal kita ingin menambah panjang vektor\_2 menjadi 7

```
vektor_2[5:7] <- c('kkhxbkhun', 'spasibo', 'kia ora')
vektor_2
```

```
## [1] "terima kasih"      "Arigatogozaimashita" "gracias"
## [4] "vielen dank"      "kkhxbkhun"          "spasibo"
## [7] "kia ora"
```

```
length(vektor_2)
```

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

Perlu di ingat jika vektor hanya bisa menyimpan data dalam satu tipe, numerik semua atau karakter semua

```
vektor_5 <- c(1, 2, 3, "a")
vektor_5
```

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

Pada output di atas kita lihat angka 1, 2 dan 3 otomatis di ubah menjadi karakter.

## Perhitungan pada vektor

Kita bisa melakukan perhitungan atau operasi aritmatika pada vektor dengan skalar ataupun dengan vektor pula.

```
vektor_6 <- c(0,8,4,6)
vektor_6
```

```
## [1] 0 8 4 6
```

```
vektor_6 + 4
```

```
## [1] 4 12 8 10
```

Untuk operasi aritmatika pada vektor dengan vektor, pastikan vektor-vektor yang digunakan memiliki panjang yang sama. Nilai pada vektor akan dioperasikan berdasarkan kesamaan indeks.

```
vektor_6 <- c(0,8,4,6)
vektor_7 <- c(6,8,1,2)
vektor_7 * vektor_6
```

```
## [1] 0 64 4 12
```

## Matriks

Sederhananya matriks adalah kumpulan vektor yang kemudian memiliki beberapa baris dan kolom.

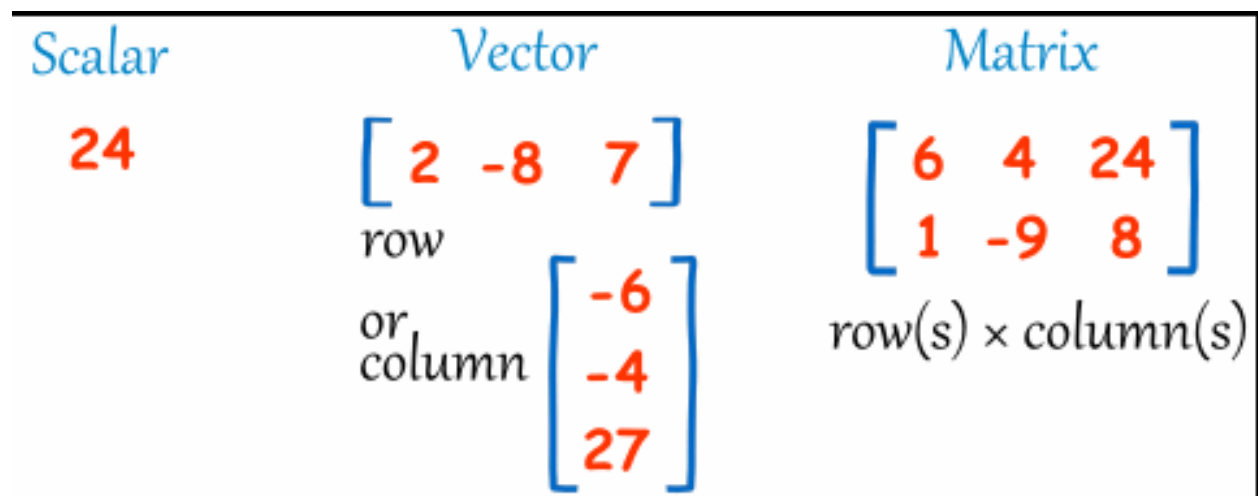


Figure 2: Data, Information, Knowledge, Wisdom

## Membuat matriks dengan menggabung vektor

```
vektor_1 <- c(14,24,81,42)
vektor_2 <- c(40,76,71,23)
vektor_3 <- c(0,1,0,1)
```

Menggabung berdasarkan kolom

```
matriks_1 <- cbind(vektor_1, vektor_2, vektor_3)
matriks_1
```

```
##      vektor_1 vektor_2 vektor_3
## [1,]      14      40        0
## [2,]      24      76        1
## [3,]      81      71        0
## [4,]      42      23        1
```

Menggabung berdasarkan baris

```
matriks_2 <- rbind(vektor_1, vektor_2, vektor_3)
matriks_2
```

```
##      [,1] [,2] [,3] [,4]
## vektor_1  14  24  81  42
## vektor_2  40  76  71  23
## vektor_3   0   1   0   1
```

Cek kelas matriks

```
class(matriks_1)
```

```
## [1] "matrix" "array"
```

Selain dengan menggunakan fungsi cbind atau rbind. Kita juga dapat membuat matriks dengan fungsi matrix()

```
matriks_3 <- matrix(
  data = c(1,2,3, 11,12,13, 4,66,77, 12,23,34),
  nrow = 3, byrow = TRUE
)
matriks_3
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3   11
## [2,]   12   13    4   66
## [3,]   77   12   23   34
```

```
class(matriks_3)
```

```
## [1] "matrix" "array"
```

Menentukan nama kolom dan baris

```

matriks_4 <- matrix(
  data = c(1,2,3, 11,12,13),
  nrow = 2, byrow = TRUE,
  dimnames = list(c("R1", "R2"),
                  c("C.1", "C.2", "C.3"))
)
matriks_4

```

```

##      C.1 C.2 C.3
## R1    1   2   3
## R2   11  12  13

```

Catatan:

- data: vektor data opsional
- nrow, ncol: jumlah baris dan kolom yang diinginkan, masing-masing.
- byrow: nilai logis. Jika FALSE (default) matriks diisi oleh kolom, jika tidak, matriks diisi oleh baris.
- dimnames: Daftar dua vektor yang memberikan nama baris dan kolom masing-masing.

Untuk mengetahui dimensi dari suatu matriks, kita dapat menggunakan fungsi `ncol()` untuk mengetahui jumlah kolom matriks dan `nrow()` untuk mengetahui jumlah baris pada matriks.

```
ncol(matriks_4)
```

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

```
nrow(matriks_4)
```

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

```
dim(matriks_4)
```

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

## Slicing matriks

Seperti dengan vektor, kita juga dapat memilih data pada matriks sesuai indeksnya. `Matriks[row, col]`

```

# memilih nilai pada baris ke 2 kolom ke 3
matriks_4[2,3]

```

```
## [1] 13
```

```

# memilih nilai semua baris kolom ke 2
matriks_4[,2]

```

```

## R1 R2
##  2 12

```

```
# memilih nilai baris ke 1 semua kolom
matriks_4[1,]
```

```
## C.1 C.2 C.3
## 1 2 3
```

## Perhitungan pada matriks

```
vektor_9 <- c(33,11,22)
vektor_9
```

```
## [1] 33 11 22
```

```
matriks_5 <- matrix(
  data = c(1,2,3, 4,5,6, 7,8,9),
  nrow = 3,
  byrow = F
)
matriks_5
```

```
##      [,1] [,2] [,3]
## [1,] 1    4    7
## [2,] 2    5    8
## [3,] 3    6    9
```

```
matriks_6 <- matrix(
  data = c(1,2,3, 4,5,6, 7,8,9),
  nrow = 3,
  byrow = T
)
matriks_6
```

```
##      [,1] [,2] [,3]
## [1,] 1    2    3
## [2,] 4    5    6
## [3,] 7    8    9
```

Matriks dengan vektor

```
matriks_5 + vektor_9
```

```
##      [,1] [,2] [,3]
## [1,] 34   37   40
## [2,] 13   16   19
## [3,] 25   28   31
```

Terlihat semua nilai pada baris 1 matriks\_5 di jumlahkan dengan nilai ke 1 vektor\_9

```
matriks_5 + matriks_6
```

```
##      [,1] [,2] [,3]
## [1,]    2    6   10
## [2,]    6   10   14
## [3,]   10   14   18
```

```
matriks_5 * matriks_6
```

```
##      [,1] [,2] [,3]
## [1,]    1    8   21
## [2,]    8   25   48
## [3,]   21   48   81
```

Jika dilakukan perkalian secara langsung pada kedua matriks tersebut, R hanya akan mengalikan matriks yang berada pada index yang sama. Perkalian matriks yang sebenarnya menggunakan `%*%`.

```
matriks_5 %*% matriks_6
```

```
##      [,1] [,2] [,3]
## [1,]   66   78   90
## [2,]   78   93  108
## [3,]   90  108  126
```

Beberapa perhitungan lainnya

```
# Total pada tiap kolom
colSums(matriks_6)
```

```
## [1] 12 15 18
```

```
# Rata-rata tiap baris
rowMeans(matriks_6)
```

```
## [1] 2 5 8
```

Sama halnya dengan vektor, matriks hanya bisa berisi satu jenis tipe data.

```
matriks7 <- matrix(
  data = c(1,"2",3, 4,5,6, 7,8,9),
  nrow = 3,
  byrow = T
)
matriks7
```

```
##      [,1] [,2] [,3]
## [1,] "1"  "2"  "3"
## [2,] "4"  "5"  "6"
## [3,] "7"  "8"  "9"
```

Semua angka akan dikonversi menjadi karakter.



## Latihan

Buat matriks berikut

$$\begin{bmatrix} 7 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

Matriks diagonal 3 x 3

Figure 3: latihan matriks

## Data Frame

Data frame ini sendiri merupakan kerangka data yang berisi variable yang memiliki karakteristik seperti matriks. Jadi data frame ini berbentuk seperti tabel. Berbeda dari matriks dataframe dapat menyimpan berbagai tipe data.

Dataframe juga dapat dibuat dengan mengabungkan beberapa vektor.

```
nama <- c('ri', 'batara', 'hanif', 'inda', 'indah', 'putri', 'rohis', 'wawan', 'yuli',  
          'yusuf')  
tahun_lahir <- c(1998, 2000, 2000, 2000, 2001, 2001, 2000, 2000, 2001, 2000)  
biodata <- data.frame(nama, tahun_lahir)  
biodata
```

```
##      nama tahun_lahir  
## 1      ri      1998  
## 2 batara      2000  
## 3 hanif      2000  
## 4  inda      2000  
## 5 indah      2001  
## 6 putri      2001  
## 7 rohis      2000  
## 8 wawan      2000  
## 9  yuli      2001  
## 10 yusuf      2000
```

```
class(biodata)
```

```
## [1] "data.frame"
```

## Ekspor dan Impor Data

Sebelum masuk bahasan ekspor dan import data, perlu mengetahui konsep working directory. Setiap pengguna akan bekerja pada tempat khusus yang disebut sebagai working directory. Working directory merupakan sebuah folder dimana R akan membaca dan menyimpan file kerja kita. Pada pengguna windows, working directory secara default pada saat pertama kali menginstall R terletak pada folder `c:\\Document`

Untuk mendapatkan working directory ketikkan kode berikut

```
getwd()
```

Untuk mengganti working directory ketikkan kode berikut. Contoh ingin mengganti working directory ke folder `~/Documents/R`.

```
setwd("~/Documents/R")
```

## Ekspor Data

Ekspor data adalah menyimpan data yang ada di R ke penyimpanan lain, misal laptop/komputer. Sebelumnya sudah di buat dataframe dengan nama `biodata`

```
biodata
```

```
##      nama tahun_lahir
## 1      ri           1998
## 2 batara           2000
## 3  hanif           2000
## 4   inda           2000
## 5  indah           2001
## 6  putri           2001
## 7  rohis           2000
## 8  wawan           2000
## 9   yuli           2001
## 10 yusuf           2000
```

Kita dapat mengeksport ke dalam beberapa format, sebagai contoh dataframe `biodata` akan disimpan dalam format `csv` dengan nama `biodata.csv`

```
write.csv(biodata, "data/biodata.csv")
```

Maka file `biodata.csv` akan tersimpan di working directory pada folder `data`

## Import data

Import data adalah memuat data dari penyimpanan lain, misal laptop/komputer ke penyimpanan R. Sebagai contoh mengimpor file csv ke R

```
nama_dataframe <- read.csv('path/nama_file.csv')
```

```
biodata <- read.csv('data/biodata.csv')
biodata
```

```
##      X   nama tahun_lahir
## 1    1     ri           1998
## 2    2 batara           2000
## 3    3 hanif            2000
## 4    4  inda            2000
## 5    5 indah            2001
## 6    6 putri            2001
## 7    7 rohis            2000
## 8    8 wawan            2000
## 9    9 yuli             2001
## 10 10 yusuf             2000
```

## Slicing Dataframe

Sama seperti matriks, slicing dataframe menggunakan dataframe[baris, kolom]. Namun ada tambahan tanda \$ untuk mengambil kolom dan bisa dengan menulis nama kolom juga.

```
#mengambil baris 1 - 3 kolom 2
biodata[1:3, 2]
```

```
## [1] "ri"      "batara" "hanif"
```

```
#mengambil variabel nama
biodata$nama
```

```
## [1] "ri"      "batara" "hanif"  "inda"   "indah"  "putri"  "rohis"  "wawan"
## [9] "yuli"    "yusuf"
```

```
#mengambil baris 1,3, 4, 6, 8 dan kolom nama dan tahun
biodata[c(1,3, 4, 6, 8), c('nama', 'tahun_lahir')]
```

```
##      nama tahun_lahir
## 1     ri           1998
## 3 hanif            2000
## 4  inda            2000
## 6 putri            2001
## 8 wawan            2000
```