



# KMMI 2021

## Eksplorasi dan Visualisasi Data

Pertemuan 2:  
Pengantar R dan RStudio

# Pembahasan

1. Tata cara unduh dan install R dan RStudio
2. Menulis dan membaca data dengan Bahasa R
3. Struktur percabangan dan perulangan
4. Tipe data (vector, matrices, array, data frame, list, factor)
5. Operator aritmatika dan operator logika pada R
6. Pengecakan dan transformasi tipe data

# Bahasa R

- Bahasa dan lingkungan untuk komputasi statistik serta grafik
- Lingkungan yang cocok untuk *data science*
- Program R berupa ***interpreter*** yang ditulis secara baris per baris seperti program pada Command Line Interface (CLI) atau Shell
- Perangkat lunak khusus untuk manipulasi data, perhitungan matematis, dan tampilan grafis

# Fasilitas R

1. Mendukung fasilitas penanganan dan penyimpanan data yang efektif.
2. Menyediakan seperangkat operator untuk manipulasi array, khususnya matriks.
3. Efektif dalam melakukan pekerjaan terkait analisis data, visualisasi data, data science, dan machine learning.
4. Menyediakan banyak teknik statistik (seperti uji statistik, klasifikasi, pengelompokan dan reduksi data).
5. Sangat mudah untuk menggambar grafik menggunakan Bahasa R.
6. Dapat bekerja pada platform yang berbeda (Windows, Mac, Linux).
7. Bahasa pemrograman yang dikembangkan dengan baik, sederhana dan efektif yang mencakup kondisional, perulangan, fungsi rekursif, dan mendukung proses input dan output.

# Kelebihan R

1. Gratis dan ***Open-Source Software*** (OSS).
2. Ketersediaan ***package*** (perpustakaan fungsi/library) yang lengkap.
3. Fungsi-fungsi yang disediakan memudahkan pekerjaan yang berhubungan dengan **komputasi statistika, eksplorasi, dan visualisai data**.
4. Memudahkan dalam mempersiapkan data, analisis data, transformasi data, dan mendukung proses import/eksport dalam berbagai format
5. Menghasilkan grafik/visualisasi yang sangat bagus dengan dukungan paket-paket yang lengkap untuk visualisasi data.
6. Memudahkan dalam pembuatan laporan

# R Comment

Komentar di R menggunakan kode #.

```
# This is a comment  
"Hello World!"  
"Hello World!" # This is a comment  
# This is a comment  
# written in  
# more than just one line  
"Hello World!"
```

# Variabel R

1. R tidak memiliki perintah khusus untuk mendeklarasikan suatu variabel.
2. Variabel dibuat pada saat anda pertama kali menetapkan nilai.
3. Untuk menetapkan nilai ke variabel, anda dapat menggunakan simbol `<-`.
4. Untuk menampilkan atau mencetak nilai suatu variabel, cukup dengan mengetikkan nama variabelnya atau dengan perintah *print()*.

```
name <- "Agus"  
age <- 40  
name      # output "Agus"  
age       # output 40  
  
# Assign the same value to multiple variables in one line  
var1 <- var2 <- var3 <- "Orange"
```

# Variabel R

1. Nama variabel harus dimulai dengan huruf dan dapat berupa kombinasi huruf, angka, titik (.) dan garis bawah (\_).
2. Jika dimulai dengan titik (.) tidak boleh diikuti oleh angka.
3. Nama variabel tidak boleh diawali dengan angka atau garis bawah (\_)
4. Nama variabel peka terhadap huruf besar/kecil atau case sensitive (usia, Usia, dan AGE adalah tiga variabel berbeda)
5. Kata-kata *default* tidak dapat digunakan sebagai variabel contoh: TRUE, FALSE, NULL, if, dsb.



# Variabel R

```
# Legal variable names:
myvar <- "Agus"
my_var <- "Agus "
myVar <- "Agus"
MYVAR <- "Agus"
myvar2 <- "Agus"
.myvar <- Agus

# Illegal variable names:
2myvar <- "Agus"
my-var <- "Agus"
my var <- "Agus"
_my_var <- "Agus"
my_v@ar <- "Agus"
TRUE <- "Agus"
```

# Tipe Data R

- a. Numerik (10.5, 55, 787)
- b. Integer (1L, 55L, 100L, dimana huruf "L" menyatakan bilangan bulat)
- c. Kompleks ( $9 + 3i$ , dimana "i" adalah bagian imajiner)
- d. Karakter (string) diapit oleh "" ("k", "R menarik", "false", "11.5")
- e. Logis (boolean) (BENAR atau SALAH)

# R Math: Operasi Matematika

- a. Fungsi ***min()*** dan ***max()*** dapat digunakan untuk menentukan nilai terendah atau tertinggi.
- b. Fungsi ***sqrt()*** untuk mengembalikan akar kuadrat dari suatu angka.
- c. Fungsi ***abs()*** mengembalikan nilai absolut (positif) dari suatu angka.
- d. Fungsi ***ceiling()*** membulatkan angka ke atas bilangan bulat terdekat, dan fungsi ***floor()*** membulatkan angka ke bawah bilangan bulat terdekat.
- e. Dll..

# String R

- Karakter atau string digunakan untuk menyimpan suatu nilai teks.
- Sebuah string dikelilingi oleh tanda kutip tunggal atau tanda kutip ganda

```
str <- "Hello"
str <- "Lorem ipsum dolor sit amet,
consectetur adipiscing elit."

str # print the value of str
cat(str)
nchar(str) #jumlah karakter

str1 <- "Hello"
str2 <- "World"
paste(str1, str2) #menggabungkan dua string
```

# Operator

## A. Operator Matematika

| Operator | Nama                     | Contoh      |
|----------|--------------------------|-------------|
| +        | Penjumlahan              | $x + y$     |
| -        | Pengurangan              | $x - y$     |
| *        | Perkalian                | $x * y$     |
| /        | Pembagian                | $x / y$     |
| ^        | Eksponen (pangkat)       | $x ^ y$     |
| %%       | Modulus (sisa pembagian) | $x \% \% y$ |

# Operator

## B. Operator Perbandingan

| Operator           | Nama                    | Contoh                 |
|--------------------|-------------------------|------------------------|
| <code>==</code>    | Sama dengan             | <code>x == y</code>    |
| <code>!=</code>    | Tidak sama dengan       | <code>x != y</code>    |
| <code>&gt;</code>  | Lebih besar dari        | <code>x &gt; y</code>  |
| <code>&lt;</code>  | Lebih kecil dari        | <code>x &lt; y</code>  |
| <code>&gt;=</code> | Lebih besar sama dengan | <code>x &gt;= y</code> |
| <code>&lt;=</code> | Lebih kecil sama dengan | <code>x &lt;= y</code> |

# Operator

## C. Operator Logika

| Operator         | Deskripsi  |
|------------------|--|
| <b>&amp;</b>     | Operator AND mengembalikan nilai TRUE jika kedua elemen TRUE   |
| <b>  atau   </b> | Operator OR mengembalikan nilai TRUE jika salah satu pernyataannya BENAR   |
| <b>!</b>         | NOT mengembalikan nilai FALSE jika pernyataan awal bernilai TRUE dan mengembalikan nilai TRUE jika pernyataan awal bernilai FALSE (negasi) |

## D. Operator Penugasan (<-) untuk menetapkan nilai ke suatu variabel

# Percabangan (Kondisi)

- Percabangan adalah suatu perintah (pernyataan) yang memungkinkan suatu perintah (pernyataan) dieksekusi jika suatu kondisi terpenuhi atau tidak terpenuhi.
- Percabangan menggunakan operator kondisional yang akan menghasilkan nilai boolean (benar/true atau salah/false)
- Fungsi: program control, mengatur alur jalannya program sesuai dengan suatu kondisi yang terpenuhi

```
# contoh percabangan sederhana (if) saja
a <- 33
b <- 200

if (b > a) {
  print("b is greater than a")
}
```



# Percabangan (Kondisi)

```
# contoh percabangan (if else)
a <- 33
b <- 33
if (b > a) {
  print("b is greater than a")
} else if (a == b) {
  print("a and b are equal")
}

# percabangan bertingkat
x <- 41
if (x > 10) {
  print("Above ten")
  if (x > 20) {
    print("and also above 20!")
  } else {
    print("but not above 20.")
  }
} else {
  print("below 10.")
}
```

# Perulangan (Looping)

- Proses dalam pemrograman yang dilakukan secara berulang-ulang dalam batas yang telah ditentukan
- Fungsi:
  - Mengulang beberapa baris perintah
  - Melakukan suatu proses yang berulang-ulang, seperti mencetak angka dari 1 – 100
- R memiliki dua perintah loop:
  - While loop
  - For loop

# While Loop

- Dengan while loop kita dapat mengeksekusi satu set pernyataan selama kondisinya TRUE.

```
i <- 1

while (i < 6) {
  print(i)
  i <- i + 1
}
```

- Pada contoh di atas, loop akan terus menghasilkan angka mulai dari 1 sampai 5. Loop akan berhenti di angka 6 karena  $6 < 6$  adalah FALSE.
- Loop while membutuhkan variabel pengindeksan misalnya *i*, yang kita atur nilainya 1.

# For Loop

- Perulangan for digunakan untuk mengulangi suatu nilai dalam urutan (sequence).

```
for (x in 1:10) {  
    print(x)  
}
```

- Output:
  - 1
  - 2
  - 3
  - 4
  - ...
  - 10

# Struktur Data R

1. **Vector:** daftar item (lists) yang bertipe sama
2. **Lists:** dapat berisi banyak tipe data yang berbeda
3. **Matrices:** dataset dua dimensi dengan kolom dan baris.
4. **Arrays:** dapat memiliki lebih dari dua dimensi
5. **Data Frame:** data yang ditampilkan dalam format table dan dapat memiliki berbagai jenis data di dalamnya
6. **Factors:** digunakan untuk mengkategorikan suatu data

# Vector

```
# Vector of strings
fruits <- c("banana", "apple", "orange")

# Print fruits
fruits

# Vector length
length(fruits)
# Access the first item (banana)
fruits[1]

# Vector of numerical values
numbers <- c(1, 2, 3)

# Print numbers
numbers
numbers <- 1:10
numbers
sort(fruits) # Sort a string
sort(numbers) # Sort numbers
```

# List

```
# List of strings
thislist <- list("apple", "banana", "cherry")

# Print the list
thislist

# Access list
thislist[1]

# Update value
thislist[1] <- "blackcurrant"

# list length
length(thislist)

# check item lists
"apple" %in% thislist

# add item
append(thislist, "orange")
```

# Matrices

```
# Create a matrix
thismatrix <- matrix(c(1,2,3,4,5,6), nrow = 3, ncol = 2)

# Print the matrix
thismatrix

thismatrix <- matrix(c("apple", "banana", "cherry", "orange"), nrow = 2,
ncol = 2)

thismatrix

# access item
thismatrix[1, 2]

# matrix length
length(thismatrix)
```



# Arrays

```
# An array with one dimension with values ranging from 1 to 24
thisarray <- c(1:24)
thisarray

# An array with more than one dimension
multiarray <- array(thisarray, dim = c(4, 3, 2))
multiarray

# Access item array
thisarray <- c(1:24)
multiarray <- array(thisarray, dim = c(4, 3, 2))

multiarray[2, 3, 2]

# looping in array
thisarray <- c(1:24)
multiarray <- array(thisarray, dim = c(4, 3, 2))

for(x in multiarray){
  print(x)
}
```

# Data Frame

```
# Create a data frame
Data_Frame <- data.frame (
  Training = c("Strength", "Stamina", "Other"),
  Pulse = c(100, 150, 120),
  Duration = c(60, 30, 45)
)

# Print the data frame
Data_Frame

# Summarize data
summary(Data_Frame)

# Access item
Data_Frame[1]
Data_Frame[["Training"]]
```

# Factors

- Faktor digunakan untuk mengkategorikan suatu data. Contoh faktor adalah:
  1. Demografi: Pria/Wanita
  2. Musik: Rock, Pop, Klasik, Jazz
  3. Pelatihan: Kekuatan, Stamina
- Untuk membuat faktor, gunakan fungsi `factor()` dan tambahkan vektor sebagai argumennya.

```
# Create a factor
music_genre <- factor(c("Jazz", "Rock", "Classic", "Classic",
"Pop", "Jazz", "Rock", "Jazz"))

# Print the factor
music_genre
```

# Factors

```
# Print level
levels(music_genre)

# Factor length
length(music_genre)

# Access factor
music_genre[3]

# Change item value
music_genre[3] <- "Pop"
music_genre[3]
```



# Terima Kasih