

# Modul 01 - Introduction to R

Roni Yunis

3/12/2022

## Instalasi R dan RStudio

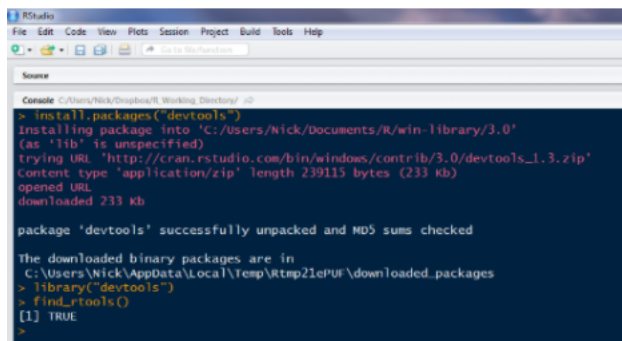
*Ada 3 hal yang harus di install:*

1. R: <https://cran.us.r-project.org/>
  2. R Studio: <https://posit.co/downloads/>
  3. instalasi Rtools untuk memudahkan proses instalasi package dengan mengikuti langkah berikut:
- Silahkan unduh terlebih dahulu instalasi Rtools pada <https://cran.r-project.org/bin/windows/Rtools/>  
Note: pastikan versi yang diunduh sesuai dengan versi R Anda.
  - Lakukan instalasi Rtools
  - Setelah selesai, buka RStudio dan lakukan instalasi package "devtools" dengan menjalankan code berikut pada console: `install.packages("devtools")`

Lakukan verifikasi terhadap instalasi Rtools sebagai berikut:

Panggil package \*devtools\* dengan menjalankan code berikut pada console:

- `library("devtools")`
- Kemudian ketik code berikut pada console: `find_rtools()`
- Output yang dihasilkan adalah "TRUE" jika instalasi Rtools anda sudah bekerja dengan baik



```
RStudio
File Edit Code View Plots Session Project Build Tools Help
Source
Console C:/Users/Nick/Desktop/R/Working Directory/ >
> install.packages("devtools")
Installing package into 'C:/Users/Nick/Documents/R/win-library/3.0'
(as 'lib' is unspecified)
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.0/devtools_1.3.zip'
content type 'application/zip' length 239113 bytes (233 Kb)
opened URL
downloaded 233 kb
package 'devtools' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
C:/Users/Nick/AppData/Local/Temp/Rtmp21ePUF/downloaded.packages
> library("devtools")
> find_rtools()
[1] TRUE
```

## Alternatif R Editor/Platform

- rstudio.cloud : <https://posit.cloud>
- kaggle.com : <https://kaggle.com>
- yupiter (anaconda) : <https://www.anaconda.com/products/individual> (harus di install pada laptop)

# Dasar Bahasa Pemrograman R

## R dan RStudio

R merupakan bahasa pemrograman dimana seperangkat instruksi akan diterjemahkan kedalam bahasa komputer, sedangkan RStudio merupakan aplikasi tambahan yang dapat membantu pengguna R melakukan pekerjaannya (editor R). Mengapa mempelajari R, berikut beberapa alasannya:

### 1. Dibangun oleh ahli statistik, untuk ahli statistik.

R adalah bahasa pemrograman statistik yang dibuat oleh Ross Ihaka dan Robert Gentleman di Departemen Statistik, di University of Auckland (Selandia Baru). R dibuat untuk analisis data, dan dengan demikian, berbeda dari bahasa pemrograman tradisional. R bukan hanya bahasa pemrograman statistik, R juga environment yang lengkap untuk analisis data dan perangkat lunak analisis data yang paling banyak digunakan saat ini.

### 2. Memiliki banyak Library

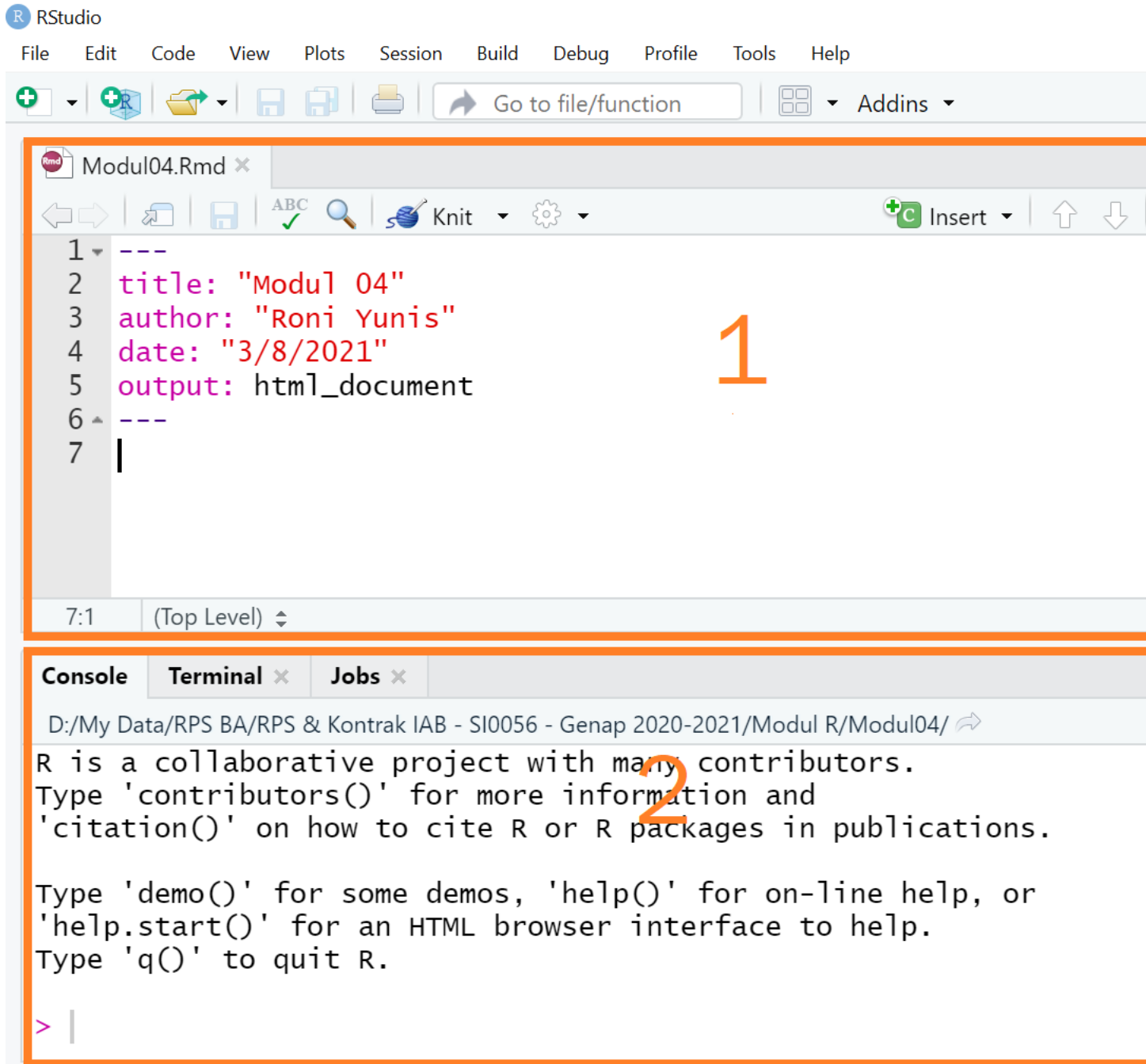
R menyediakan banyak packages tambahan yang menambahkan fungsionalitas out-of-the-box untuk berbagai kegunaan: uji statistik, analisis deret waktu, visualisasi yang indah, dan berbagai tugas machine learning seperti algoritme regresi, algoritme klasifikasi, dan algoritme clustering. Komunitas R terkenal karena kontribusinya yang aktif dalam hal packages.

### 3. Sumber Terbuka

Bagian dari alasan komunitasnya yang aktif dan berkembang pesat adalah sifat sumber terbuka (open-source) dari R. Pengguna dapat berkontribusi dalam pembuatan packages, banyak tools statistik dan template kustomisasi untuk visualisasi yang tidak ditemukan dalam aplikasi statistik lain.

## Navigasi dalam RStudio

RStudio memiliki 4 buah panel, yaitu:



1. **Panel Source** : Panel ini merupakan fitur utama dari RStudio, panel ini menampilkan file yang sedang dibuka pada RStudio.
2. **Panel Console** : Panel ini menampilkan console asli dari R yang digunakan untuk berkomunikasi dengan R session. Terdapat beberapa tab lain seperti Terminal yang dapat digunakan untuk mengakses komputer Anda melalui Command Line Interface (CLI).
3. **Panel Environment / History** : Bagian ini menampilkan seluruh object R yang sudah dibuat selama session yang sama. Terdapat tab History yang berfungsi untuk melihat history dari kode yang sudah dijalankan sebelumnya.
4. **Panel Files/Plot/Packages/Help** :

- Tab Files : Daftar dari berkas (file) yang berada dalam working directory.
- Tab Plot : Menampilkan visualisasi yang terbentuk
- Tab Packages : Berisi daftar packages yang sudah terinstall
- Tab Help : Menampilkan dokumentasi resmi dari setiap fungsi

## R Markdown

R markdown dapat digunakan untuk membuat laporan dari analisa dengan standar yang tinggi dan sangat baik. Dengan RMarkdown kita bisa menghasilkan laporan analisis yang kita buat kedalam bentuk HTML dan PDF. Untuk menghasil kedua file tersebut kita bisa menggunakan fungsi `knit` dari packages `rmarkdown`.

Pada R markdown Anda dapat memasukkan narasi atau interpretasi dari analisis yang dibuat serta memasukkan kode program dari analisis. Adapun tempat untuk memasukkan kode program pada R markdown disebut *chunk*. Terdapat 2 cara untuk membuat chunk yaitu :

1. menggunakan shortcut `ctrl + alt + i`
2. menggunakan tombol insert yang berada pada pojok kanan atas dari panel source kemudian pilih R

Ketika bekerja menggunakan RStudio penting bagi kita untuk mengetahui lokasi dari *working directory* saat ini. Fungsi pada R untuk mengetahui directory Anda saat ini adalah dengan membuat chunk lalu menjalankan fungsi `getwd()` seperti di bawah ini:

```
getwd()
```

```
## [1] "D:/All Doc/My Data/SI2110 - Ganjil 2324/Modul R/IAB_SI2110_2324"
```

Untuk menjalankan code diatas maka bisa melakukan dengan meletak cursor pada code tersebut, kemudian tekan “Ctrl + Enter” pada keyboard, atau dengan cara mengklik tombol *Run Current Chunk* pada sudut kanan chunk yang aktif.

Untuk memberikan keterangan dari code yang kita buat, bisa kita buat dengan tanda #. Tanda # bisa kita letakkan di dalam *chunk*. Contohnya seperti code berikut.

```
#melihat direktory aktif
getwd()
```

```
## [1] "D:/All Doc/My Data/SI2110 - Ganjil 2324/Modul R/IAB_SI2110_2324"
```

## Membuat objek di R

R menyimpan data ke dalam objek R, objek tersebut disimpan dalam memori dari setiap sesi di R, berikut merupakan contoh dari pembuatan objek di R:

```
inteligensi <- "Analitika Bisnis"
```

kata `inteligensi` di atas disebut dengan **objek** yang menampung nilai yaitu “Analitika Bisnis” atau disebut juga dengan assign (`<-`). (`<-`) bisa dibuat dengan cepat menggunakan kombinasi “Alt + -”. Jika Anda menjalankan chunk di atas, maka sekarang akan muncul objek `inteligensi` pada panel inveroiment. Untuk memanggil objek yang sudah dibuat kita bisa menjalankan chunk dibawah ini

```
inteligensi
```

```
## [1] "Analitika Bisnis"
```

Perlu diperhatikan bahwa R bahasa pemrograman yang *case sensitive* sehingga “Analitika Bisnis” akan berbeda dengan “analitika bisnis”. Untuk memeriksa apakah nilai pada sisi kiri sama dengan sisi kanannya, bisa menggunakan operator `==` atau persamaan. Jika hasilnya adalah **FALSE** artinya kedua nilai tersebut tidak sama atau berbeda, karena nilai yang disimpan pada objek `inteligensi` adalah Analitika Bisnis bukan analitika bisnis.

```
inteligensi == "analitika bisnis"
```

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

*Latihan* Cobalah Anda buat sebuah objek **Nama** dan isi dengan nama Anda sendiri.

```
#your code
```

## Tipe-tipe data di R

Pada umumnya struktur data dari objek di R adalah Vector. Vector merupakan kumpulan beberapa nilai yang memiliki tipe data yang sama dan tersusun dalam satu dimensi. Ada 5 buah type data di R yaitu: character, complex, numeric, integer, dan logical. Coba Anda jalankan code berikut ini:

```
# character
a_char <- c("STMIK", "Mikroskil", "Sistem Informasi", "Medan", "Roni Yunis")
# complex
a_comp <- c(1+3i, (1+3i)*2)
# numeric
a_num <- c(-1, 1, 2, 3/4, 0.5)
# integer
an_int <- c(1L, 2L)
# logical
a_log <- c(TRUE, TRUE, FALSE)
```

Keterangan: `c()` adalah fungsi untuk membuat vector

Sebagai contoh Anda diminta untuk membuat sebuah objek **orang** yang berisi 3 nama teman Anda. Maka code nya bisa dibuat seperti ini:

```
orang <- c("Roni", "Joosten", "Yunis")
```

Cara mengetahui type data dari suatu objek, Anda dapat menggunakan fungsi `class()`. Codenya seperti ini:

```
class(orang)
```

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

maka hasilnya kalau dijalankan maka objek orang tersebut bertipe character.

*Latihan* Buatlah sebuah objek **harga**, masukkan 3 nilai ini kedalamnya 2500, 3000, 4500. kemudian lihatlah type data apa dari objek tersebut dengan fungsi `class()`

```
#your code
```

bagaimana kalau dalam sebuah vector terdiri dari beberapa type data yang berbeda seperti chunk dibawah ini, apa yang terjadi? coba Anda periksa.

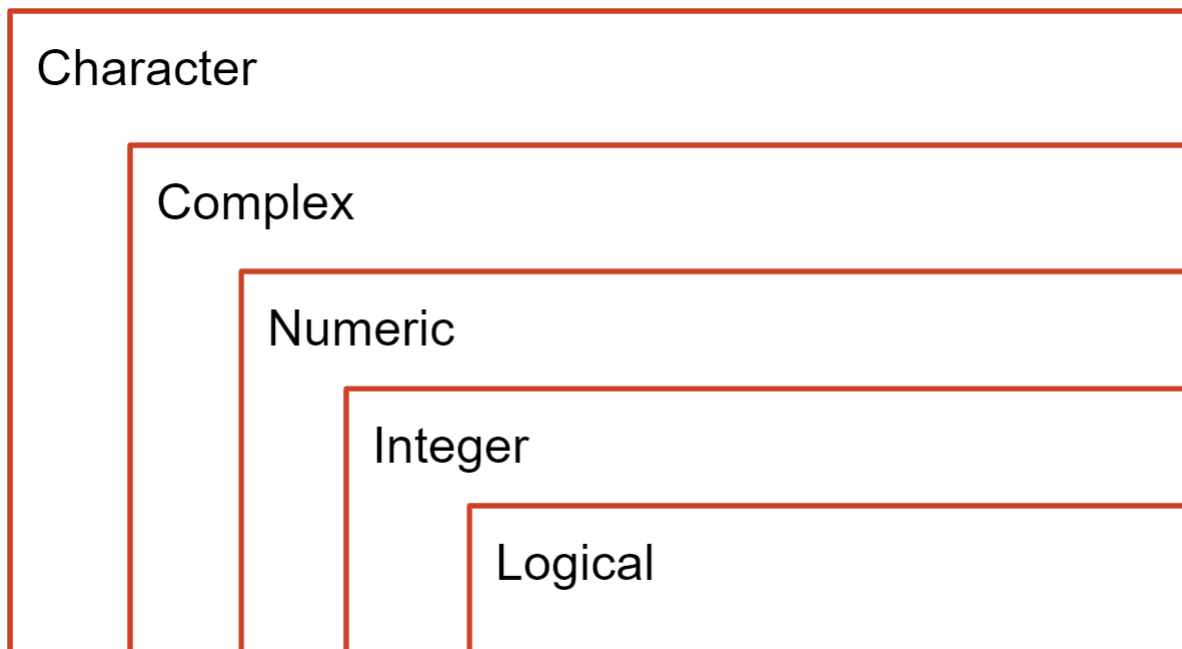
```
campur <- c("Aisyah", 2020, 46L, TRUE)
campur
```

```
## [1] "Aisyah" "2020"   "46"     "TRUE"
```

```
class(campur)
```

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

Bila diperhatikan setiap nilai pada vector `campur` memiliki tanda **petik dua**, artinya nilai tersebut merupakan sebuah objek dengan tipe character. Jadi semua nilai yang ada didalam objek `campur` tersebut dipaksa menjadi type character. Proses ini disebut juga dengan **implicit coercion**



Pada gambar hirarki diatas, bisa dijelaskan bahwa type data yang paling spesifik adalah logical, dan type yang paling umum adalah type character. Vector `campur` diubah menjadi type character karena terdapat elemen “Aisyah” yang bertipe character.

*Latihan* Tentukan type data dari vector-vector di bawah ini?

- `c(TRUE, 1L, 1/2)`
- `c("satu", 12, 33.3)`
- `c(1,2,3,4,5L)`

## Struktur Data di R

Setelah Anda mengetahui bagaimana cara menyimpan suatu objek ke dalam struktur data vector dan bagaimana sifat dari suatu vector, di R, terdapat pula beberapa cara untuk menyimpan elemen-elemen ke dalam objek selain vector, yakni ada pula struktur data list, matrix dan data frame. Selanjutnya, akan dibahas masing-masing struktur data tersebut.

### Matrix

Struktur data `matrix` merupakan salah satu cara di R untuk menyimpan data dalam bentuk objek. Matrix memiliki sifat yang sama dengan vector yaitu hanya dapat menyimpan satu tipe data saja. Adapun cara untuk membuat sebuah `matrix`, Anda dapat menggunakan fungsi `matrix()`:

Vector : 1 dimensi dan harus tipe data yang sama

Matrix : 2 dimensi dan harus tipe data yang sama

```
mat <- matrix(11:16, nrow = 3, ncol = 2)
mat
```

```
##      [,1] [,2]
## [1,]  11  14
## [2,]  12  15
## [3,]  13  16
```

Perhatikan bagaimana nilai dalam sebuah `matrix` terisi; `matrix` akan terisi secara baris terlebih dahulu karena parameter `byrow` yang bernilai `FALSE` secara *default*. Nilai dalam sebuah `matrix` akan terisi dari indeks [1, 1] dapat dibaca sebagai [row index, column index] yang merupakan nilai paling kiri atas.

### List

Dalam `vector` dan `matrix`, kita hanya bisa menggunakan elemen-elemen dengan tipe data yang sama. Jika kita ingin menyimpan beberapa data dengan tipe data yang berbeda, kita bisa menggunakan struktur data `list`:

```
our_list <- list(c(TRUE, FALSE), "TRUE", c(1, 6, 12), 1L)
our_list
```

```
## [[1]]
## [1]  TRUE FALSE
##
## [[2]]
## [1] "TRUE"
##
## [[3]]
## [1]  1   6  12
##
## [[4]]
## [1] 1
```

ketika melakukan subset pada list menggunakan `[]` itu berarti kita mengakses *nilai dan index nya*. `[[[]]` yang kita akses adalah *nilainya saja*

```
class(our_list[[3]])
```

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

*latihan:* coba akses `our_list` pada index ke 4 dan cek tipe datanya menggunakan fungsi `class()`

```
our_list[[4]]
```

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

```
class(our_list[[4]])
```

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

Seperti yang dapat dilihat di atas, sebuah `list` dapat menyimpan objek dengan tipe yang berbeda, dan juga memperbolehkan masing-masing objek di dalamnya untuk memiliki panjang yang berbeda juga. Seperti `vector`, sebuah `list` hanya memiliki 1 dimensi; namun jika Anda ingin mengakses elemen-elemen data di dalamnya, Anda dapat menggunakan `[]`. Meskipun kita sudah menggunakan `[]`, kita masih akan mendapatkan objek dengan struktur data berupa `list`:

```
class(our_list[3])
```

```
## [1] "list"
```

Untuk mengakses `vector` yang ada di dalamnya, maka kita bisa menggunakan `[[ ]]`:

```
class(our_list[[3]])
```

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

## Data Frame

Dalam pengolahan data, seringkali kita dihadapkan dengan data berbentuk tabel; seperti hasil rekapitulasi penjualan dari masing-masing toko. Umumnya, data dengan bentuk seperti ini akan memiliki 2 dimensi, yaitu baris yang menggambarkan setiap observasi dan kolom yang menggambarkan setiap variabel dari data.

Representasi data seperti penjelasan di atas, disebut dengan struktur data `data.frame`. Adapun cara untuk membuat sebuah data frame kita bisa menggunakan fungsi `data.frame()` seperti di bawah ini:

```
daftar_buah <- data.frame(  
  buah = c("Apel", "Semangka", "Jambu"),  
  ukuran = c(22, 76, 12),  
  stringsAsFactors = FALSE)
```

```
daftar_buah
```

```
##      buah ukuran  
## 1    Apel     22  
## 2 Semangka    76  
## 3   Jambu     12
```

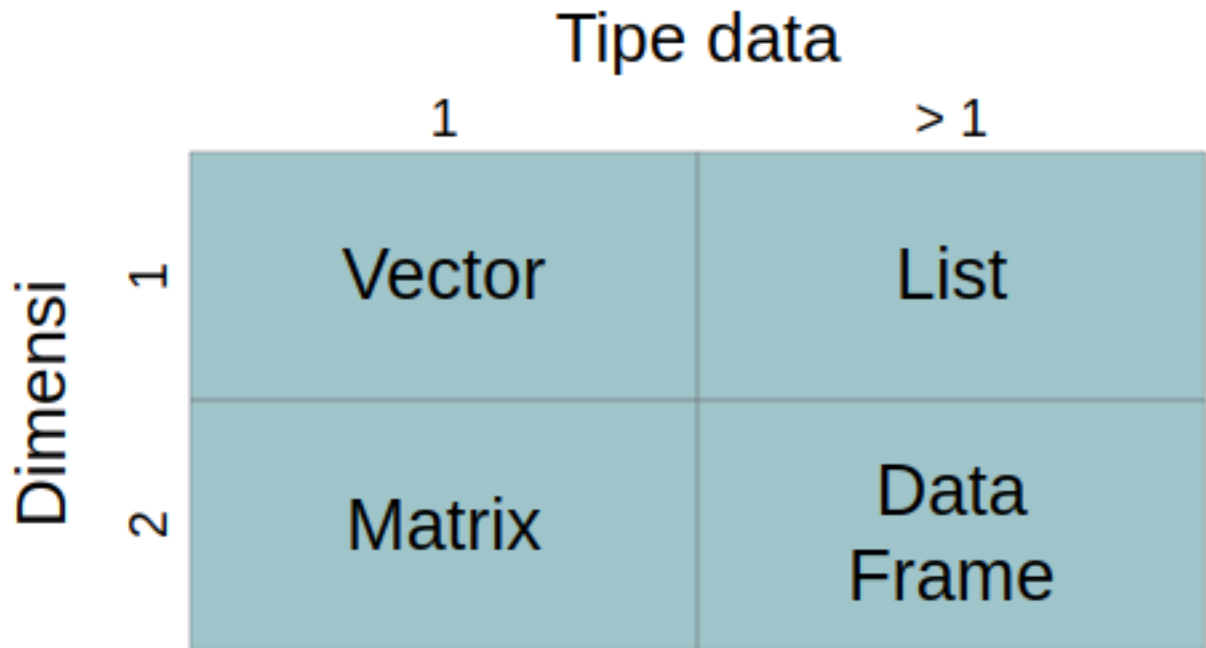


*latihan:* buat dataframe yang berisi 3 kolom dan simpan kedalam object `karyawan`

Nama : “Roni”, “Aisyah”, “Rasford” Usia : 46,21,25 female : FALSE, TRUE, FALSE

```
#your code
```

Gambaran dari rangkuman keempat struktur data yang sudah kita bahas di atas, dapat diilustrasikan seperti gambar di bawah ini:



## Penutup

Demikian pembahasan kita tentang Pengantar Bahasa Pemrograman R, semoga materi ini bisa Anda jadikan sebagai dasar untuk mengikuti materi-materi selanjutnya. *Selamat belajar dan salam sehat selalu*