

# Pengenalan bahasa pemrograman R: untuk non-programmer

*Dasapta Erwin Irawan dan Prana Ugi*

*August 23, 2015*

## Contents

<b>1</b>	<b>Data punya peran sentral</b>	<b>1</b>
<b>2</b>	<b>Jenis data</b>	<b>5</b>
2.1	<i>Numeric</i> . . . . .	6
2.2	<i>Integer</i> . . . . .	6
2.3	<i>Complex</i> . . . . .	6
2.4	<i>Logical</i> . . . . .	7
2.5	<i>Character</i> . . . . .	7
<b>3</b>	<b>Obyek (<i>Objects</i>)</b>	<b>7</b>
3.1	Jenis obyek . . . . .	7
3.2	Penamaan obyek . . . . .	8
3.3	Vektor . . . . .	8
3.4	<i>lists</i> . . . . .	9
3.5	<i>arrays</i> . . . . .	9
3.6	<i>matriks/matrices</i> . . . . .	10
3.7	<i>data frames</i> . . . . .	10
<b>4</b>	<b>References</b>	<b>11</b>

... By the way, the word "data" is plural, so ordinarily you would not say "data is" or "data was." You  
... [King WB](<http://ww2.coastal.edu/kingw/statistics/R-tutorials/objects.html>)

## 1 Data punya peran sentral

Data adalah salah satu bagian sentral dari R, bagian yang lainnya adalah kode dan output. Oleh karenanya kami menempatkan bagaimana menyiapkan dan memanipulasi data di bab kedua. Dalam diagram berikut ini dapat anda lihat bahwa mekanisme kerja dalam R adalah:

- input: berupa data
- proses: berupa kumpulan kode perintah
- output: berupa tabel dan plot

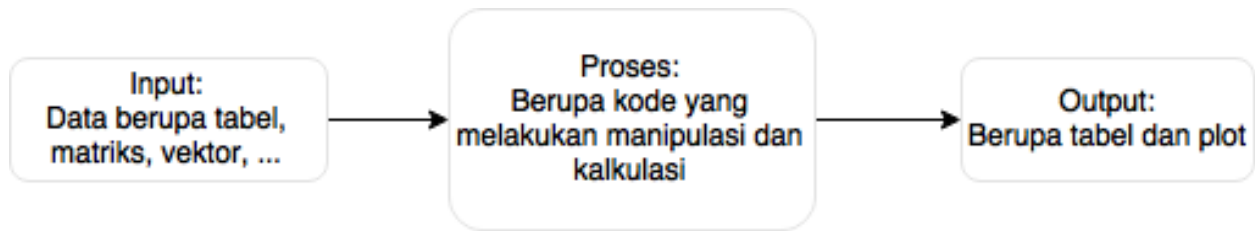


Figure 1: Komponen input dan output dalam R

Data adalah bahan baku R. Ia berperan sebagai input. Kalau anda mahasiswa pasti sering berhadapan dengan *spreadsheet* maka itulah salah satu contoh data, dalam hal ini adalah data eksternal. Anda harus mengimpor data tersebut ke dalam format yang dapat dibaca oleh R. Pada bagian berikutnya akan kami perlihatkan cara untuk melakukan langkah tersebut.

Selain mengimpor data dari luar, anda juga dapat memasukkan data secara langsung dari prompt R (R>). Ada tiga cara untuk meng-assign data kepada suatu obyek dalam R. Coba ketik beberapa baris berikut (selalu akhiri dengan **enter**). Simbol # digunakan untuk memberikan komentar. Semua yang diketik di belakang simbol # tidak akan diproses oleh R. Kita coba langsung ya agar anda terbiasa dengan *look and feel* syntax dalam R.

Ketikkan setiap baris di bawah ini setelah R> dan akhir dengan **enter**. Anda akan membuat obyek pertama anda bernama x berisi angka 1704.

```
# cara pertama (dengan beberapa variasi)
```

```
x<-1704
```

```
x
```

```
x <- 1704
```

```
x
```

```
x    <-    1704
```

```
x
```

```
# atau cara kedua
```

```
y=1704
```

```
y
```

```
y = 1704
```

```
y
```

```
z    =    1704
```

```
z
```

```
# atau cara ketiga
```

```
1704->z
```

```
z
```

```
1704 -> z
```

```
z
```

```
1704   <-   z
z
```

Bagaimana hasilnya?

Sama bukan.

Jadi dalam hal ini spasi tidak mempengaruhi R untuk mengerti perintah anda. Namun hal ini tidak berlaku umum. Nilai 1704 dimuat oleh R ke dalam memorinya dalam bentuk obyek bernama `x` untuk diproses lebih lanjut.

Simbol `<-` (tanpa spasi diantaranya) dan `=` mempunyai efek sama, tetapi tidak selalu. Simbol `=` akan sering digunakan dalam *argument*, tapi simbol `<-` tidak digunakan dalam *argument*. Oleh karenanya, supaya tidak bingung kita pakai saja simbol `<-`.

Apalagi *argument* ini?

Satu-satu ya akan kami jelaskan.

Sekarang kita coba lihat ada berapa obyek anda punya di memori R. Dengan perintah berikut, anda akan lihat bahwa di memori R sudah ada tiga obyek bernama: “x”, “y”, “z”. Coba lihat jendela **Environment** di kanan atas.

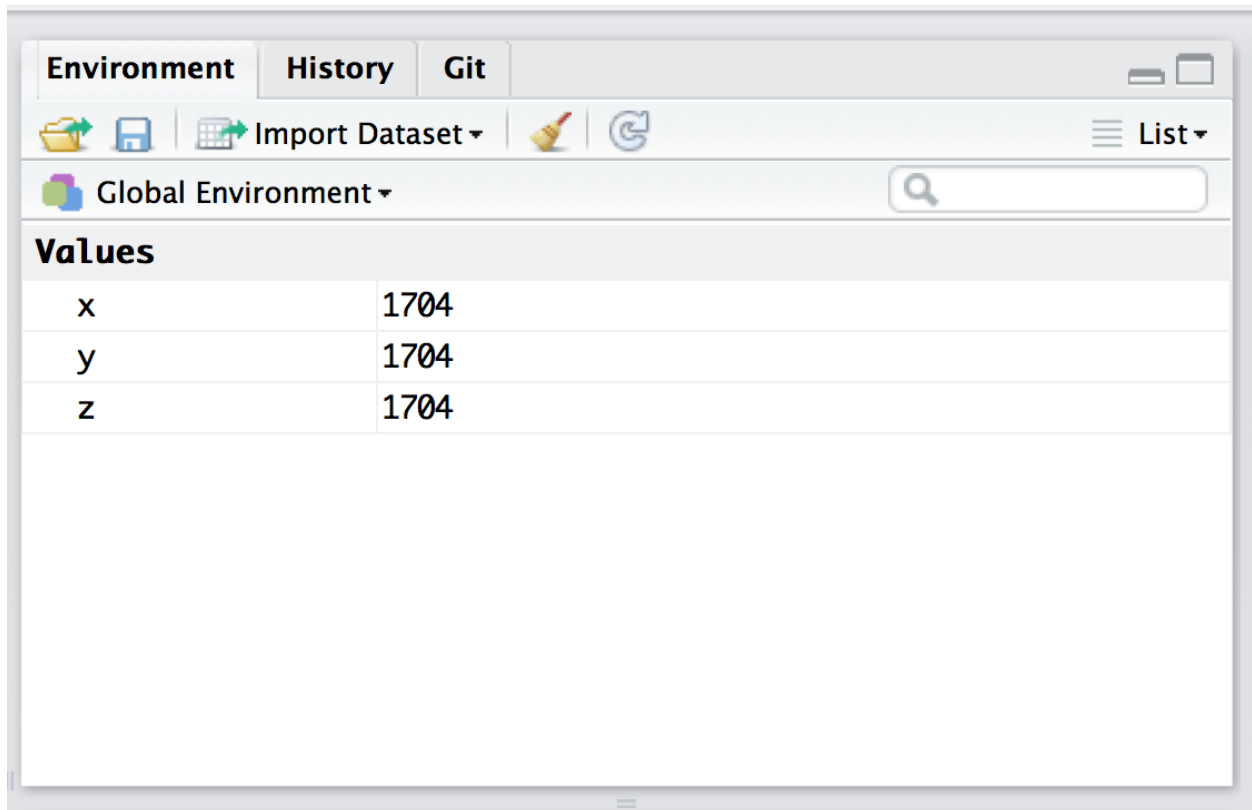


Figure 2: Tiga obyek di jendela **Global Environment**

Obyek sangat sensitif, begitu anda mengganti nilai yang tersimpan dalam suatu obyek, maka R akan langsung menindih obyek tersebut. Atau anda menggunakan nama obyek yang sama dengan nilai yang berbeda, maka obyek yang terakhirlah yang akan disimpan oleh R.

R memiliki beberapa data yang terinstalasi secara *default* di dalamnya. Ini dibuat untuk keperluan latihan biasanya, beberapa data yang sering dipakai sebagai contoh misalnya:

- **iris**: data iris hasil pengukuran (dalam cm) geometri 50 spesies bunga Iris (*Iris setosa*, *Iris versicolor*, dan *Iris virginica*).
- **mtcars**: data pengukuran 10 spesifikasi 32 merek mobil yang diambil dari majalah Motor Trend US terbitan tahun 1974.
- **rivers**: data panjang 141 sungai utama di Amerika Utara (dalam mil) yang dikompilasi oleh USGS.

Coba anda lihat jendela **Global Environments** kanan atas. Klik menu *dropdown* (seperti pada gambar berikut), maka anda akan melihat banyak dataset yang terinstalasi dalam sistem R. Masing-masing dataset tersebut bersifat bebas pakai dan dapat digunakan dalam latihan R anda.

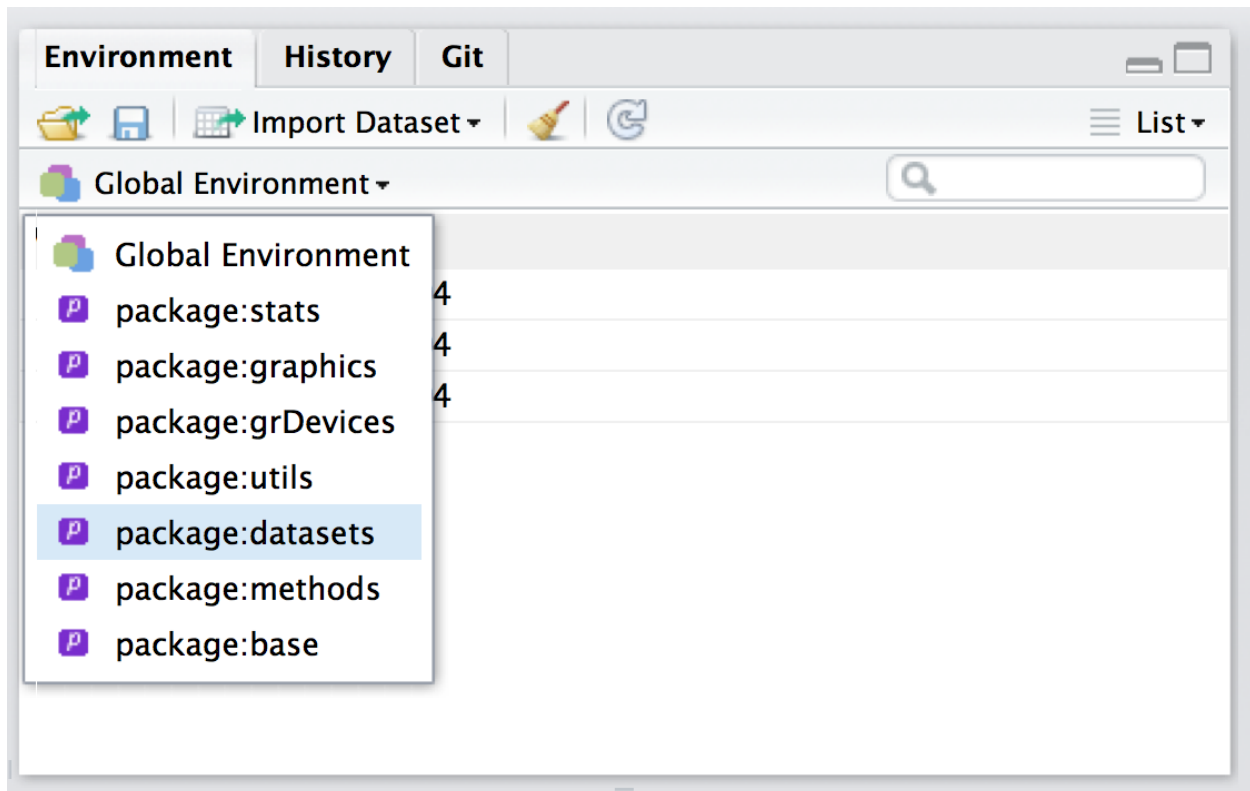


Figure 3: Menu dropdown **Global Environment** yang berisi dataset

Berikut ini adalah bagaimana cara melihat isi dataset. *Copy paste* baris per baris perintah berikut ini setelah R> di jendela **console** (kiri bawah) atau *copy paste* seluruhnya di jendela **Script** (kiri atas)

```
# for "iris" data
iris          # to see the complete data. It's messy.
?iris         # to see the name and the source of the data.
head(iris)    # to see the first six rows of the data.
tail(iris)    # to see the last six rows of the data.
str(iris)     # to see the structure and type of the data, eg (num = numeric, chr = character, etc).
class(iris)   # to see what kind of object the data is.

# for "mtcars" data
mtcars        # to see the complete data. It's messy.
?mtcars       # to see the name and the source of the data.
head(mtcars)  # to see the first six rows of the data.
tail(mtcars)  # to see the last six rows of the data.
```

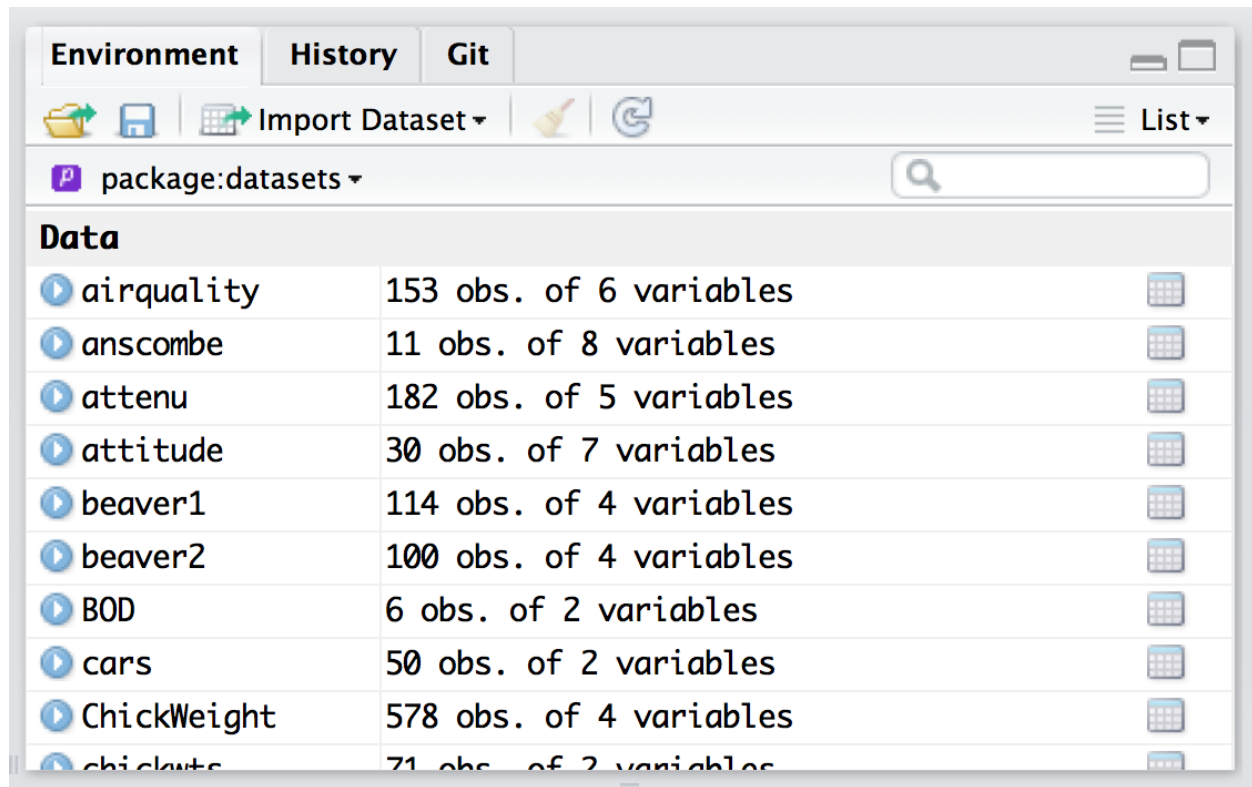


Figure 4: Daftar dataset yang terinstalasi dalam sistem R

```
str(mtcars)      # to see the structure and type of the data, eg (num = numeric, chr = character, etc).
class(mtcars)   # to see what kind of object the data is.

# for "rivers" data
rivers          # to see the complete data. It's messy.
?rivers         # to see the name and the source of the data.
head(rivers)    # to see the first six rows of the data.
tail(rivers)    # to see the last six rows of the data.
str(rivers)     # to see the structure and type of the data, eg (num = numeric, chr = character, etc).
class(rivers)   # to see what kind of object the data is.
```

## 2 Jenis data

Tiap piranti lunak (P/L) statistik biasanya akan membuat format data khusus, misal: format **sav** untuk *SPSS*, format **dta** untuk *Stata*, atau format **sta** untuk *Statistica*. Semuanya adalah format biner (*binary*). Jadi kalau tidak punya P/L nya maka anda tidak akan bisa membuka filenya.

R agak aneh sedikit. Ia bisa membuka file data dengan format yang umum, yakni text file atau ASCII, misalnya: **csv** (*comma separated value*) atau **txt**. Ini yang membuat R bisa dipakai di OS apa saja. Dalam buku ini kita akan banyak menggunakan format **csv**. Tapi dengan sedikit *tweaking* dan menginstalasi package bernama **foreign** anda akan bisa membuka format apa saja, termasuk **xls** (dan variannya), **sav**, dan **dta**.

Memang terasa sedikit mundur ke belakang, kalau kita belajar tipe data. Tapi percayalah sering kali masalah saat coding timbul karena kita (termasuk saya) lupa atau tidak tahu karakter dari data.

Ada beberapa tipe data yang bisa diproses oleh R:

- *Numeric*
- *Integer*
- *Complex*
- *Logical*
- *Character*

## 2.1 *Numeric*

Tipe data numerik adalah semua angka dari 0 sampai 9. Bisa positif, bisa negatif. Bisa mengandung pecahan (*fraction*) atau tidak. Yang penting adalah tidak bercampur dengan karakter.

```
x <- 1704      # assign a value to object named x
x              # print the value of x
class(x)       # print the class name of x

y <- 17.04     # assign a value to object named y
y              # print the value of y
class(y)       # print the class name of y
```

## 2.2 *Integer*

Jenis data integer sama dengan numerik, tapi tidak mengandung pecahan, atau angka yang bulat. Nilai integer boleh positif dan negatif.

Untuk memastikan bahwa nilai yang kita *assign* untuk suatu obyek adalah integer, kita bisa menggunakan perintah `as.integer()`. Jadi kita bisa juga memerintahkan R untuk menyimpan angka non pecahan sebagai integer.

```
x <- 1704
class(x)

y <- 17.04
class(y)

z1 <- as.integer(17.04)
z1
class(z1)

z2 <- as.integer(1704)
z2
class(z2)
```

Bisa dilihat bedanya? Apa yang anda lihat?

## 2.3 *Complex*

Bilangan kompleks adalah bilang yang memiliki struktur:  $a + b_i$ . Jadi mengandung bilangan  $a$  yang riil (*real*) dan bilangan  $b_i$  yang imajiner.

```
c <- 1 + 2i    # create a complex number
c              # print the value of z
class(c)       # print the class name of z
```

## 2.4 *Logical*

Nilai logikal dalam R biasanya bernilai TRUE atau FALSE.

```
x <- 1704
y <- 1511
z <- x > y    # is x larger than y? and store in z
z            # print the logical value in z
class(z)     # print the class name of z
```

Berikut adalah beberapa simbol untuk operasi logika baku dalam R:

- “&” (dan/*and*),
- “|” (atau/*or*),
- “!” (negasi/*negation*).

```
d <- TRUE
e <- FALSE
d & e      # u AND v
d | e      # u OR v
!d         # negation of u
```

## 2.5 *Character*

Data berjenis karakter adalah yang terdiri dari huruf (A-Z). Tipe data ini tidak akan diproses oleh R sebagai angka.

```
e <- "Ali"
e
class(e)

c <- "Adi"
c
class(c)
```

Tapi kita juga bisa mengetik angka sebagai teks atau memerintahkan R untuk membaca angka sebagai teks.

```
g <- "3.14"
g
class(g)
```

```
h <- as.character(3.14)
h
class(h)
```

# 3 Obyek (*Objects*)

## 3.1 Jenis obyek

Semua data yang bisa dibaca oleh R, dinamakan obyek/*Objects*. Berikut ini adalah beberapa jenis obyek dalam R:

- vektor/*vectors*
- *lists*
- *arrays*
- *matrix/matrices*
- *data frames*

## 3.2 Penamaan obyek

Setiap obyek harus memiliki nama, seperti halnya yang sudah anda lakukan di beberapa baris di atas. Coba sekarang lihat ada berapa obyek yang sudah anda buat.

Bagaimana caranya?

Lihat jendela *Environment* atau ketik `ls()` di jendela konsol.

```
ls()
```

Anda bisa memberi nama obyek sesuka anda. Aturan dasarnya adalah tidak boleh mengandung **spasi** atau simbol **dash**. Jadi anda bisa membuat nama obyek yang:

- panjang, tapi jangan terlalu panjang.
- campuran antara huruf kecil dan besar.
- mengandung simbol *underscore* (`_`) dan titik (`.`).

Berikut ini adalah beberapa contohnya:

- satu huruf seperti: `x`, `X`. Untuk kode yang pendek boleh saja, tapi kalau kode anda sudah panjang, akan susah mengingatnya.
- gabungan dua karakter seperti: `x2`, `X2`. Masih susah mengingatnya.
- gabungan tiga karakter seperti: `x.2`, `x_2`.
- sebuah frasa seperti: `myData`, `MyData`, `my_data`, `my.data`.
- tapi jangan terlalu panjang seperti: `my.data.from.the.geology.experiment`
- atau mengandung nama seseorang, seperti: `flinstone`, `Flinstone`, `FLINSTONE`, atau `Fred.Flinstone`
- atau jangan pula menggunakan nama perintah baku dalam R, misal: `mean`, `sd`, `plot`, dll.

## 3.3 Vektor

Karena bidang saya sumber daya air, maka coba kita lihat data “rivers”. Menurut perintah `class(rivers)` data ini berjenis `vector`. Beda dengan “iris” dan “mtcars” yang berjenis `data frame`. Vektor adalah jenis obyek yang paling sederhana. Sekarang kita coba membuat vektor sederhana.

```
c("Abi", "Aci", "Adi", "Afi", "Agi") # vector 1
c(76, 78, 79, 80, 83, 87) # vector 2
c(38, 34, 37, 36, 31, 27) # vector 3
```

Di atas anda sudah membuat tiga buah vektor, tapi keduanya belum memiliki nama. Ingat aturannya, setiap obyek, termasuk vektor, harus bernama. Sekarang kita coba beri nama.

```
nama <- c("Abi", "Aci", "Adi", "Afi", "Agi", "Ali") # vector 1
tahun.lahir <- c(76, 78, 79, 80, 83, 87) # vektor 2 named tahun.lahir
usia <- c(38, 36, 35, 34, 31, 27) # vektor 3 named usia
```



```

nama
str(nama)
class(nama)

tahun.lahir
str(tahun.lahir)
class(tahun.lahir)

usia
str(usia)
class(usia)

```

Setiap vektor akan terdiri dari nilai vektor dan no index atau no urut. Kalau anda ingin melihat data ketiga dalam vektor “nama”, “tahun.lahir”, dan “usia”, maka gunakan perintah berikut.

```

nama[3]
tahun.lahir[3]
usia[3]

```

Hasilnya adalah “Adi”, 79, dan 35.

Contoh lain, kalau anda ingin melihat data tertentu dalam vektor “rivers”, maka gunakan perintah berikut.

```

rivers[115]          # to see data no 115
rivers[10:20]        # to see data no 10 hingga 20
rivers[c(10, 15, 18, 20)] # to see data no 10, 15, 18 dan 20

```

Kalau anda melihat tulisan **NA**, maka artinya ada kolom yang kosong dalam data anda.

### 3.4 *lists*

List adalah gabungan dari campuran berbagai obyek yang tidak sebangun. Misal kita ingin membuat list bernama “daftar” dengan menggabungkan vektor “nama”, “tahun.lahir”, dan “usia” dengan angka 5.

```

daftar <- list("nama", "tanggal.lahir", "usia", 5)
daftar
str(daftar)
class(daftar)

```

### 3.5 *arrays*

Sebuah array bisa tersusun oleh matriks dan tabel. Kita bisa membuat sebuah array dengan tiga baris, empat kolom, dan dua tabel.

```

# make an array of number from 1 to 24
# of 2 tables, each table has 3 rows, 4 columns.
array1 <- array(1:24, dim=c(3,4,2))
array1

```

### 3.6 matriks/*matrices*

Sebuah matriks adalah kumpulan elemen data yang disusun secara dua dimensi, terdiri dari  $m$  baris dan  $n$  kolom.

Kita coba buat ya dengan menggunakan perintah:

```
namamatriks <- matrix(c(dataelement), nrow = m, ncol = n)

matriksA <- matrix(c(2, 4, 3, 1, 5, 7), nrow=2, ncol=3)
# or you can edit the code this way
matriksA <- matrix(c(2, 4, 3, 1, 5, 7),
                    nrow=2,
                    ncol=3)

matriksA
matriksA2 <- matrix(c(2, 4, 3, 1, 5, 7),
                    nrow=2,
                    ncol=3,
                    byrow=TRUE)

matriksA2

# here's the second matrix
matriksB <- matrix(c(2, 4, 3, 1, 5, 7),
                    nrow=3,
                    ncol=2)

matriksB
matriksB2 <- matrix(c(2, 4, 3, 1, 5, 7),
                    nrow=3,
                    ncol=2,
                    byrow=T)

matriksB2
```

perintah `byrow = TRUE` akan membuat matriks diisi berdasarkan baris (row).

### 3.7 data frames

Data frame adalah kumpulan dari vektor dengan dimensi yang sama. Bedakan dengan *list* ya. Jadi kita bisa membuat sebuah data frame dari awal, membaca sebuah tabel dan menyimpannya dalam sebuah data frame.

Ingat data frame “iris” atau “mtcars”.

```
class(iris)
str(iris)
class(mtcars)
str(mtcars)
```

Masih ingat tiga vektor kita: “nama”, “tahun.lahir”, dan “usia”? Karena ketiganya terdiri dari jumlah baris yang sama, yakni enam baris, maka kita bisa membuat data frame darinya. Ingat aturannya adalah vektor harus sama ukurannya, terutama jumlah barisnya agar bisa digabungkan dalam satu data frame.

```
str(nama)
str(tahun.lahir)
str(usia)
df <- data.frame(nama, tahun.lahir, usia)
df
```

Anda akan melihat bahwa data frame “iris” terdiri dari lima variabel (dalam kolom) dan 150 observasi (dalam baris). Data frame “mtcars” terdiri dari 11 variabel dan 32 observasi. Tipe obyek data frame akan sering kita pakai dalam R.

## 4 References

- Chongsuvivatwong, V, 2012, Analysis of Epidemiological Data using R and Epicalc Package, Epidemiology Unit Prince of Songkla University THAILAND
- Chi Yau, R Tutorial Blog, accessed 17 April 2014
- King, WB, R Tutorial, Coastal Carolina University
- Wikipedia online encyclopedia