

Pengenalan bahasa pemrograman R: untuk non-programmer

Dasapta Erwin Irawan dan Prana Ugi

August 23, 2015

Contents

1	Data punya peran sentral	1
2	Tipe data yang dikenal R	2
2.1	<i>Numeric</i>	3
2.2	<i>Integer</i>	3
2.3	<i>Complex</i>	3
2.4	<i>Logical</i>	3
2.5	<i>Character</i>	4
3	Data disimpan sebagai <i>Objects</i> (obyek)	4
3.1	Penamaan obyek	5
3.2	Mencoba membuat obyek	5
3.3	Jenis obyek	6
4	Mengatur direktori kerja	9
5	Mengimpor data ke dalam R	9
6	References	10

... 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

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.

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.

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.
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 Tipe data yang dikenal R

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 Data disimpan sebagai *Objects* (obyek)

Data baik yang kita ketik secara langsung maupun yang diimpor dari luar akan disimpan ke dalam R sebagai obyek.

3.1 Penamaan obyek

Setiap obyek harus memiliki nama, seperti halnya yang sudah anda lakukan di beberapa baris di atas. 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.

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.

3.2 Mencoba membuat obyek

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. Namun simbol `=` akan sering digunakan dalam *argument*. Jadi agar tidak bingung biasakan menggunakan simbol `<-`.

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, atau ketik `ls()`.

3.3 Jenis obyek

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

- *vector*
- *list*
- *array*
- *matrix*
- *data frames*

3.3.1 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.

3.3.2 *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)
```

3.3.3 *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
```

3.3.4 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).

3.3.5 *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 Mengatur direktori kerja

Seperti halnya piranti lunak yang lain, R perlu mengetahui di direktori manakah kita akan bekerja. Untuk mengaturnya bisa menggunakan perintah baris atau menggunakan menu di menu bar.

Bila menggunakan perintah baris, anda dapat mengetik perintah:

```
setwd("pathname")
```

misal

```
setwd("d:/data/Latihan R")
```

Bila anda menggunakan OS Windows, jangan bingung. Simbol `slash (/)` digunakan dalam sistem R, walaupun Windows hanya mengenal `backslash (\)`.

Untuk mengetahui posisi direktori terakhir, gunakan perintah:

```
getwd()
```

5 Mengimpor data ke dalam R

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` atau `readr` anda akan bisa membuka format apa saja, termasuk `xls` (dan variannya), `sav`, dan `dta`.

Sekarang kita coba untuk mengimpor data dari beberapa jenis format yang umum ke dalam R. Untuk melakukan hal ini kita punya fungsi dasar `read.tabel`. Fungsi ini sangat handal untuk mengimpor data berformat text (`ASCII`) seperti `txt` dan `csv`. Kita coba selangkah demi selangkah.

Dalam latihan ini file data perlu kita unduh lebih dahulu. Data yang diberikan adalah data kualitas air yang diambil di beberapa mata air di kawasan Bandung Selatan. Tabel tersebut disimpan di situs [blog saya](#) yang merupakan pendamping buku ini.

Ada dua cara:

1. anda bisa mengunduh lebih dahulu empat jenis file yang disertakan di halaman situs tersebut atau,
2. anda bisa secara langsung membacanya dari R.

```
# Membaca file format *.csv secara langsung dari R
# dan disimpan dalam obyek bernama "data_csv"
data_csv <- read.csv("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_csv.csv")
```

```
# anda dapat melihat jendela "Environment"
# anda akan melihat ada 7 observasi (baris) dan 33 variabel (kolom)
# Coba baca obyek dengan beberapa perintah berikut
```

```

View(data_csv)
head(data_csv)
tail(data_csv)

# membuka file format *.txt
data_txt <- read.table("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_txt.txt")
# lihat di jendela "Environment" ada 9 obs dan 33 variable.

# coba perintah berikut
data_txt <- read.table("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_txt.txt", header=T)
# lihat di jendela "Environment" ada 8 obs dan 33 variable.
# apa yang tidak sama? Jumlah barisnya. Apa yang beda?
View(data_txt)
# anda bisa melihat nama "Situ Kince" yang mestinya berada dalam satu kolom dipenggal menjadi dua kolom

# membuka file zip
unzip("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_zip.zip", unzip="internal")
# ada pesan kesalahan dan unzip tidak berhasil.
# oleh karena itu kami sarankan anda menyimpan file dalam format *.csv atau *.xls

# Membuka file format *.xls menggunakan package "readr"
install.packages("readxl") # instalasi package
library(readxl) # loading library
# https://github.com/hadley/readxl

data_xls <- read_excel("http://dasaptaerwin.net/wp/wp-content/uploads/2016/01/data_xls.xls")

# didapatkan pesan kesalahan
# harus diunduh lebih dulu baru dibuka.

data_xls <- read_excel("/Users/dasaptaerwinirawan/Documents/2016/Github/Rforbeginners/tab-fig/data_xls.xls")
View(data_xls)
# Anda dapat melihat ada 7 obs dan 33 variables

```

Dari latihan di atas, kami sarankan:

- simpanlah file data dalam format *.csv atau *.xls, manfaatkan package seperti readxl, foreign, atau gdata untuk membaca berbagai format file data lainnya.
- unduhlah terlebih dahulu file data sebelum diimpor ke dalam R.

6 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
- Readxl R-package
- Foreign package tutorial
- Gdata package tutorial