

# KOMPUTER 1

Universitas Terbuka



BUKU MATERI POKOK  
SATS4111/3SKS/MODUL 1 - 9

# KOMPUTER I

- A.P. Hardhono
- Suhartono
- Imas Sukaesih Sitanggang



PENERBIT UNIVERSITAS TERBUKA

# Daftar Isi

Daftar Isi .....	2
1. Modul 1 : Pengenalan dan Instalasi Perangkat Lunak dan Lingkungan Pemrograman R.....	3
Pendahuluan .....	3
Perangkat lunak & Lingkungan Pemrograman R .....	3
Menjalankan R.....	5
Fasilitas Help .....	8
Bantuan Online Help .....	9
Demo pada R <i>Functions</i> .....	10
<i>Libraries</i> dalam bahasa R .....	10
Instalasi Packages .....	11
Command Line Versus Script .....	12
Data Editor .....	12
Perubahan Tampilan Layar Dari R.....	15
Fungsi Bantuan Lain .....	15
Test Formatif .....	16
2. Module 2 : Mekanisme Kerja, Pembuatan Objek, dan Informasi Bantuan dalam Sistem R ...	17
K. Belajar 1    Mekanisme Kerja Sistem R .....	17
K. Belajar 2    Pembuatan, Penayangan, serta Penghapusan Objek dalam R dan Informasi Bantuan Online. 21	
3. Modul 3 : Data dalam Sistem R.....	29
K. Belajar 1    Jenis-Jenis Objek dan Operasi Dasar Aritmatika dalam R.....	30
K. Belajar 2    Membaca dan Menulis <i>File</i> Data dalam R.....	35
Cara Mengakses Nilai dari Data Frame .....	36
Argumen atau Opsi <code>read.table()</code> .....	36
4. Lampiran.....	41

# 1. Modul 1 : Pengenalan dan Instalasi Perangkat Lunak dan Lingkungan Pemrograman R

## Pendahuluan

Modul 1 ini akan memperkenalkan anda sekalian pada perangkat lunak dan lingkungan pemrograman **R**. Lingkungan pemrograman **R** ini relatif baru dan belum banyak dikenal oleh kalangan di luar perguruan tinggi. Modul 1 ini akan menjelaskan langkah-langkah penginstalasian perangkat lunak dan lingkungan pemrograman **R**. Bagian akhir modul 1 ini akan dibahas dasar-dasar pengoprasian lunak dan pemrograman **R**.

Dalam modul 1, dipaparkan pembahasan mengenai pengenalan dan instalasi prangkat lunak dan lingkungan dalam Pemrograman **R**. Setelah mempelajari modul ini, diharapkan anda dapat;

1. memahami perangkat lunak bahasa R.
2. memahami lingkungan pemrograman bahasa R.

## Perangkat lunak & Lingkungan Pemrograman R

R adalah suatu perangkat lunak untuk membantu melakukan analisis stastik, perhitungan matriks, dan pembuatan grafik. R dikembangkan oleh Ross Ihaka & Robert Gentleman. R adalah sesuatu perangkat lunak yang termasuk dalam kategori *Open Source*.

R digunakan untuk melakukan perhitungan dan manipulasi data secara statistik serta untuk menampilkan data dalam bentuk grafis. R memiliki beberapa fasilitas, yakni;

1. Pengelilaan dan peyimpanan data.
2. Kumpulan operasi untuk perhitungan bentuk array khususnya matriks
3. Kumpulan fasilitas pengolahan data yang cukup besar dan komprehensif.
4. Sarana peembuatan grafis dan penampulan untuk penyajian pada layar monitor ataupun pada kertas, juga dapat disimpan sebagai file dalam berbagai bentuk format.

R Dikembangkan oleh sebuah tim dan menerima kontribusi dari berbagai pihak dalam pengembangannya sehingga dari waktu ke waktu faslitas yang ada dalam R selalu bertambah banyak dan meningkat kualitasnya. Daftar lengkap paket tersebut dapat dilihat pada

[https://cran.r-project.org/web/packages/available\\_packages\\_by\\_name.html](https://cran.r-project.org/web/packages/available_packages_by_name.html) . Berikut adalah beberapa paket pengelolaan, analisis, dan penampilan data yang terdapat pada R.

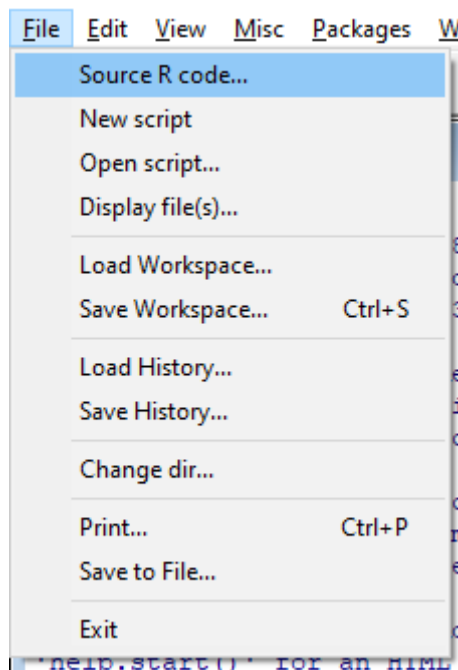
Tabel 1 Beberapa Paket dalam R

Nama Paket	Deskripsi
base	paket dasar R
boot	fungsi-fungsi Bootstrap R (S-Plus)
class	fungsi-fungsi untuk klasifikasi
cluster	fungsi-fungsi untuk clustering
datasets	paket R Datasets
foreign	membaca data yang disimpan oleh Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...
graphics	paket grafik R
grDevices	perangkat grafik R dan dukungan untuk warna dan font
grid	paket grafik grid
Kern Smooth	fungsi-fungsi untuk penghalusan ( <i>smoothing</i> ) kernel
lattice	grafik lattice
MASS	paket utama dari Venables dan Ripley's MASS
methods	metode-metode dan kelas formal
mgcv	GAMs dengan GCV <i>smoothness estimation</i> dan GAMMs oleh REML/PQL
nime	model-model efek gabungan linier dan non-linier
nnet	Feed-forward Neural Networks dan model-model Multinomial Log-Linear
rpart	pembuatan partisi rekursif
spatial	fungsi-fungsi untuk Kriging dan analisis pola titik
splines	fungsi-fungsi dan kelas untuk <i>Regression Spline</i>
stats	paket R Stats
stats4	fungsi statistik menggunakan kelas-kelas S4
survival	analisis <i>survival</i> , meliputi <i>penalised likelihood</i>
tcltk	antar muka Tcl/Tk
tools	alat untuk pembangunan paket
utils	paket R Utils

## Menjalankan R

Pada baris dibawah ini , tedapat delapan ikon alat (*icon tools*) dengan masing-masing memiliki fungsi yang telah ditentukan.

Gambar 1-1 Menu File



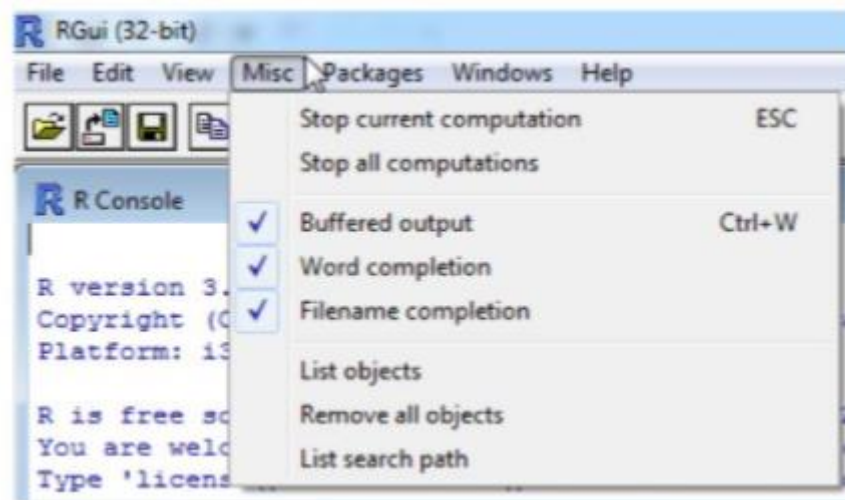
Fungsi-fungsi / sub menun dalam menu *file* sebagai berikut;

1. *Sour R code* berfungsi menjalankan program yang telah dibuat terlebih dahulu. Program disimpan dilam file dengan format R, Misalnya file `pembayra.R`
2. *New Script* berfungsi menulis program R baru atau *script* yang baru.
3. *Open Script* berfungsi membuka program R yang sebelumnya telah ditulis dan disimpan.
4. *Display File* berfungsi memperlihatkan daftar *file* yang ada didalam suatu *directory* atau *folder*.
5. *Load Workspace* berfungsi memasukan data dari tempat penyimpanan didalam *hard disk* kedalam memori untuk diolah.
6. *Save Workspace* berfungsi meyimpan data yang ada didalam memori kedalam *hard disk* atau media penyimpanan lainnya.
7. *Change Dir* berfungsi mengganti direktori kerja. Sebaiknya dalam mengerjakan sebuah proyek kita membuat sebuah folder yang terpisah dengan proyek yang lainnya.
8. *Print* berfungsi mencetak apa saja yang tersimpan dalam *workspace/R console* (ruang kerja yang tampak dalam layar monitor) kedalam *printer*.

9. *Save to file* berfungsi menyimpan segala sesuatu yang ada didalam *workspace* atau *R Console* kedalam suatu file teks.
10. *Exit* untuk mengakhiri atau menutup program R.

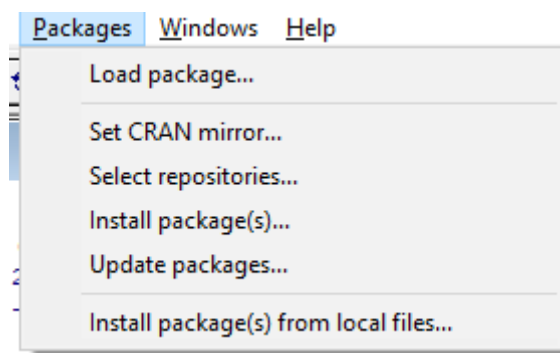
Untuk menu yang sejenis dengan aplikasi lain akan dilewatkan, dijelaskan dibawah ini fungsi dari menu ang sangat penting dan sering dipakai dalam menjalankan program R.

Gambar 1-2. Menu Misc



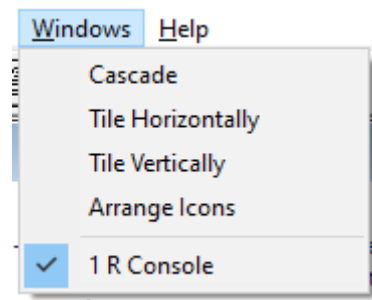
Pada menu *misc* (Gambar 1-2. Menu Misc) terdapat fungsi *stop current computation* yang berfungsi untuk menghentikan perhitungan yang sedang berlangsung, hal ini juga dapat digunakan dengan menekan tombol **ESC**. Fungsi lainnya adalah *List Objects* untuk menampilkan daftar objek dan *remove all objects* untuk menghapus semua objek.

Gambar 1-3 Menu Package



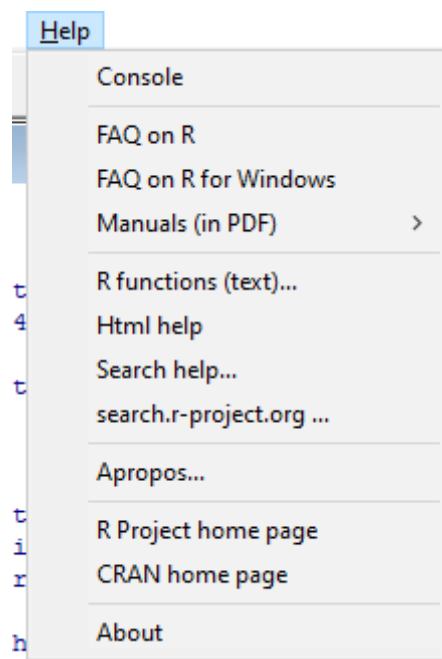
Menu *packages* berisi fungsi-fungsi untuk menambahkan paket statistik dan paket lainnya (*Load Package*), mengatur CRAN mirror (*Set CRAN Mirror*), memilih repositori tempat penyimpanan paket (*Select Repositories*), menginstal paket (*Intall Package*), dan memperbarui package (*Upadte Packages*), dan kita dapat menginstall paket dari *file zip* yang disimpan dalam komputer kita (*Install package(s) from local fils*).

Gambar 1-4 Menu Windows



Menu *window* menyediakan pilihan-pilihan untuk tampilan jendela *R Console*. Yaitu, *Cascade*, *Tile Horizontally*, *Tile Vertically*, dan *Arrange Icons*.

Gambar 1-5 Menu Help



Menu *help* menyediakan pertanyaan yang sering diajukan R (FAQ), panduan manual dalam bentuk PDF, bantuan penjelasan untuk fungsi-fungsi dalam R, tautan ke *website* R Project dan CRAN, serta mengenai versi R yang digunakan (*About*).

Gambar 1-6 Citaiton Function

```
> citation()

To cite R in publications use:

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical
Computing, Vienna, Austria. URL https://www.R-project.org/.

A BibTeX entry for LaTeX users is

@Manual{,
  title = {R: A Language and Environment for Statistical Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2021},
  url = {https://www.R-project.org/},
}

We have invested a lot of time and effort in creating R, please cite it when using it for data analysis. See also
'citation("pkgname")' for citing R packages.
```

Walaupun Bahasa R adalah program open source yang dapat digunakan secara bebas, namun kode etis bagi seorang yang terpelajar diwajibkan untuk memberikan kutipan didalam tugas atau kerjaan. Informasi yang dikutip. Fungsi `citation()` (Gambar 1-6 Citaiton Function) akan memberikan informasi yang dapat dimasukan pada kutipan didalam dokumen kerja.

## Fasilitas Help

Untuk mengetahui deskripsi, cara penggunaan dari sebuah fungsi yang sudah diketahui nama fungsinya dapat mengetik "?" setelah tanda prompt > diikuti dengan nama fungsi, > ?  
"read.tabel.

Gunakan fungsi `help.search("nama fungsi yang dicari")` untuk mengetahui fungsi yang diinginkan namun yang diketahui hanya subjek yang ingin dicari.

Fungsi lain yang berguna adalah *find* dan *apropos*. Fungsi *find* digunakan untuk mencari *packages* data dari suatu nama tertentu.

```
> find ("lowess")
[1] "package:stats"
```



Sementara itu, *apropos* berguna untuk mencari kata atau string yang berkaitan dengan fungsi yang ingin kita cari tahu maksudnya (secara menyeluruh atau parsial). Perhatikan contoh berikut

```
> apropos ("lm")
[1] ".colMeans"      ".lm.fit"         "colMeans"        "confint.lm"
"contr.helmert"   "dummy.coef.lm"   "glm"
[8] "glm.control"     "glm.fit"         "KalmanForecast"  "KalmanLike"
"KalmanRun"       "KalmanSmooth"    "kappa.lm"
[15] "lm"             "lm.fit"         "lm.influence"    "lm.wfit"
"model.matrix.lm" "nlm"            "nlminb"
[22] "predict.glm"     "predict.lm"      "residuals.glm"   "residuals.lm"
"summary.glm"     "summary.lm"
```

## Bantuan Online Help

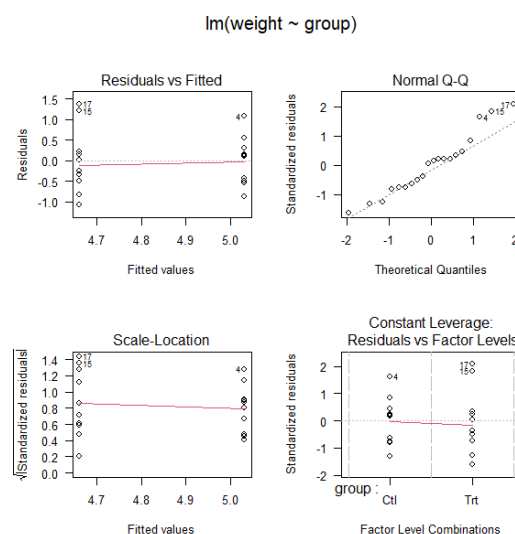
Informasi yang sangat banyak mengenai R anda dapat peroleh pada situs CARN <https://cran.r-project.org/> Dalam situs tersebut, Anda akan menjumpai beberapa manual penggunaan R seperti berikut.

- 1) *An Indroduction on R*
- 2) *R data import/export*
- 3) *R installation and Administration*

Untuk mencari contoh penulisan suatu fungsi dan contoh *output*-nya, kita gunakan perintah *example()* sebagai contoh fungsi *lm* (*linear model*),

```
> example (lm)
# Output
```

Gambar 1-7 Hasil perintah > example (lm)



Hasil perintah pada (Gambar 1-7) adalah proses visualisasi dari beberapa fungsi yang digunakan fungsi yang digunakan dan ditulis dapat dilihat pada (Lampiran 4-1 Kode hasil fungsi >example (lm)).

### Demo pada R Functions

Perintah ini untuk melihat jangkauan pekerjaan yang dapat dilakukan oleh bahasa R. Berikut ini adalah beberapa demo yang bisa anda coba.

```
> demo (persp)
> demo (graphics)
> demo (Hershey)
> demo (plotmath)
```

### Libraries dalam bahasa R

Secara sederhana, ketikkan salah satu fungsi *library* di dalam tanda kurung. Apa bila kita ingin menggunakan *library spatial* misalnya

Tabel 2 Beberapa Library dalam R

library	deskripsi
lattice	grafik <i>lattice</i> untuk plot panel atau <i>grafftrellis</i>
MASS	paket terkait buku Venables dan Ripley berjudul <i>Modern Applied Statistics using S-PLUS</i>
mgcv	<i>generalized additive models</i>
nlme	<i>mixed-effects models</i> (linear dan non linear)
nnet	<i>feed-forward neural networks</i> dan model-model multinomial <i>log-linear</i>
spatial	fungsi-fungsi untuk kriging dan analisis pola titik
survival	analisis <i>survival</i> , termasuk <i>penalised likelihood</i>

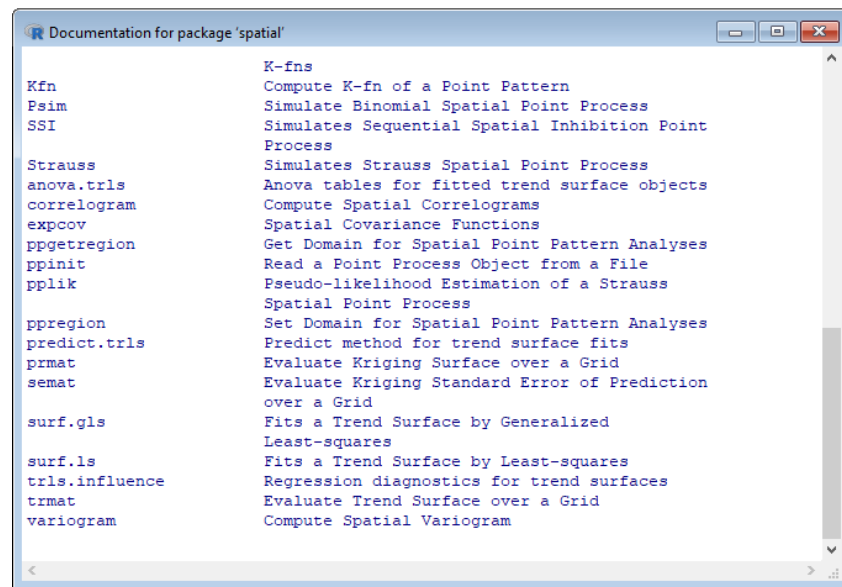
### Isi dari Library

Sesuatu yang mudah menggunakan fungsi *help* dalam mencari isi *library packages*. Sebagai contoh, perintah berikut untuk mencari informasi dari *spatial library*.

```
> library (help=spatial)
```

Maka itu tampil pada jendela pada (Gambar 1-8 Hasil dari library(help=spatial))

Gambar 1-8 Hasil dari library(help=spatial)



## Instalasi Packages

*Packages* dasar tidak berisi beberapa *library* yang dirujuk dalam buku ini, tetapi untuk mendownload *packages* ini sangatlah mudah. Gunakan fungsi `install.packages` untuk *download library* yang anda inginkan. Untuk mempercepat donwload, pilihlah *mirror* yang paling dekat dengan anda (Indonesia). Selanjutnya proses akan berjalan secara otomatis.

```
## Paramter 1 : nama packages yang ingin disinttal
## Parameter 2 : Mirror, atau sumber donwload package, pilih yang terdekat
agar download lebih cepat.

install.packages('RMySQL', repos='https://repo.bppt.go.id/cran/')
```

Downloaded packages are stored in

C:\Users\farras\AppData\Local\Temp\RtmpcMvGy\downloaded\_packages

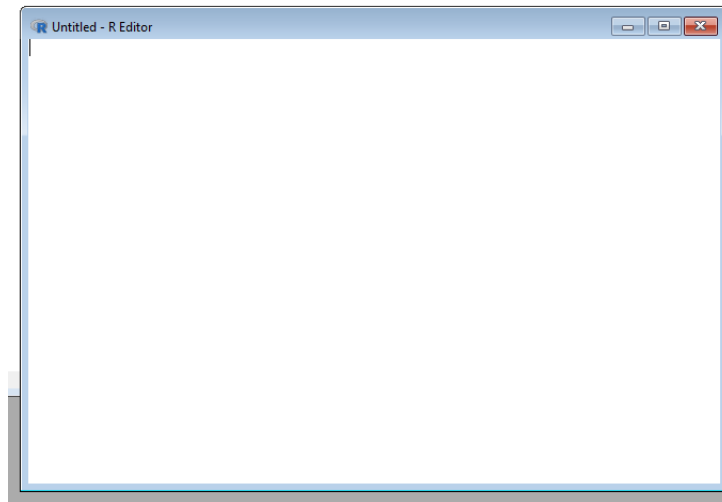
*Packages* yang digunakan dalam buku ini sebagai berikut

[illegible]

## Command Line Versus Script

Text editor didalam R adalah RGui, Untuk keperluan ini kita tinggal klik menu *file*. Pilihlah menu *new script*. Kemudian akan muncul jendela *untitled-R editor* (Gambar 1-9 UntitledR Editor).

Gambar 1-9 UntitledR Editor

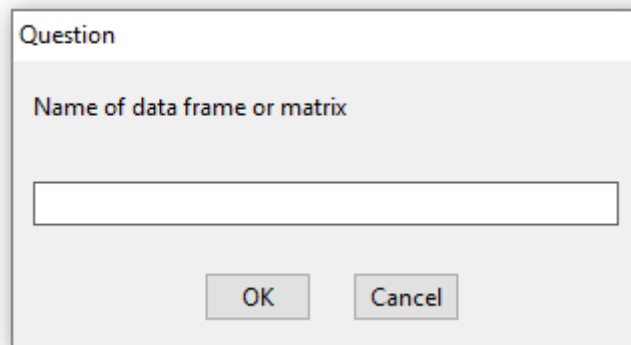


Dengan fasilitas ini, anda dapat menulis atau mengedit perintah. Untuk menjalankan perintah yang ada didalam R Editor tersebut dapat menekan **CTRL-R**. Baris perintah yang sudah diketik selanjutnya secara otomatis akan ditransfer ke *window R console* beserta hasil eksekusinya. Dengan **CTRL-S** anda dapat meyimpan peintah yang ditulis (*Script*) dalam jendela *untitled-R Editor*.

## Data Editor

*Data Editor* didalam R dapat diakses melalui *menu edit*, kemudian pilih *data editor*. Selanjutnya anda perlu memasukkan nama *data frame* atau matriks (*data frame* disini adalah *data frame* yang aktif dalam R). *Data Frame* digunakan untuk menyimpan tabel-tabel data. *Data Frame* berisi vektor-vektor dengan panjang yang sama.

Gambar 1-10 Jendela isian data frame atau matriks



Setelah anda memasukan nama *dataframe* dalam kota *question* (Gambar 1-10 Jendela isian data frame atau matriks), akan muncul jendela editor. Sebagai contoh, anda masukan salah satu nama *dataframe* yang terdapat dalam R session (*Bacteria* misalnya) sehingga akan muncul tampilan gambar (Gambar 1-11 Jendela Data Editor untuk Data Frame Bacteria), *Data frame bacteria* terdapat dalam *package* MASS. Agar anda dapat menggunakan *data frame* tersebut, terlebih dahulu ketikkan perintah berikut.

```
> library(MASS)
> attach(bacteria)
```

Gambar 1-11 Jendela Data Editor untuk Data Frame Bacteria

	y	ap	hilo	week	ID	trt	var7	var8
7	n	a	hi	6	X02	drug+		
8	y	a	hi	11	X02	drug+	S	
9	y	a	lo	0	X03	drug		
10	y	a	lo	2	X03	drug		
11	y	a	lo	4	X03	drug		
12	y	a	lo	6	X03	drug		
13	y	a	lo	11	X03	drug		
14	y	p	lo	0	X04	placebo		
15	y	p	lo	2	X04	placebo		
16	y	p	lo	4	X04	placebo		
17	y	p	lo	6	X04	placebo		
18	y	p	lo	11	X04	placebo		
19	y	p	lo	0	X05	placebo		
20	y	p	lo	2	X05	placebo		
21	y	p	lo	4	X05	placebo		
22	y	p	lo	6	X05	placebo		
23	y	p	lo	11	X05	placebo		
24	y	a	lo	0	X06	drug		
25	y	a	lo	2	X06	drug		

Pembahasan mengenai *data frame* akan dibahas dalam pembahasan selanjutnya dalam BMP ini.

Cara lain untuk menampilkan dan mengedit *data frame*, yaitu menggunakan fungsi *fix*. Misalnya anda ingin mengedit *data frame bacteria*. Setelah perintah diatas, dapa Anda ketikan perintah berikut ini.

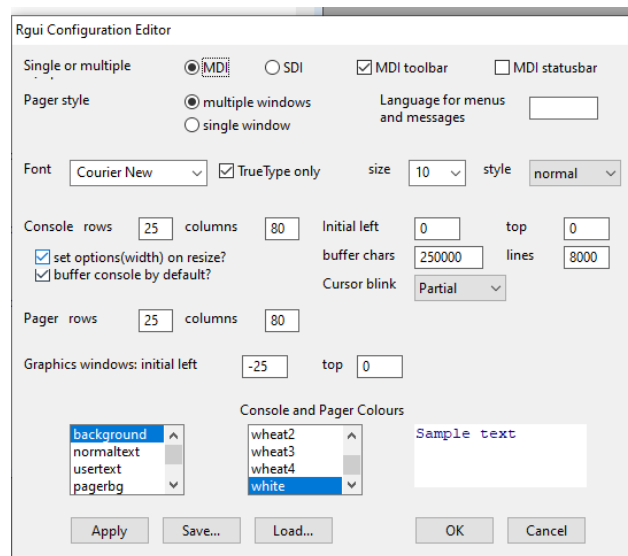
```
> fix(bacteria)
```

Bentuk tampilan dari RGui (*Data Editor*) tampak seperti tampilan *spreadsheet* Excel. Anda dapat mengubah judul dari kolom ataupun isi dari sel.

## Perubahan Tampilan Layar Dari R

Pengaturan *default* yang telah ada didama *window* R saat ini seperti telah standar dan memenuhi apa yang diinginkan. Namun, apabila ingin merubah tampilan *window* ini, Anda dapat masuk ke *R gui configuration editor* terdapat dalam menu *edit/GUI preferences* (Gambar 1-12 Jendela GUI Configuration Editor)

Gambar 1-12 Jendela GUI Configuration Editor



## Fungsi Bantuan Lain

Untuk melihat variabel apa yang telah Anda buat dalam *session* yang tengah berlangsung. Ketikkan perintah berikut.

```
> objects()
```

Untuk melihat *library* dan *dataframe* yang telah terlampir, ketikkan perintah berikut

```
> search()
```

Sementara itu, untuk menghapus variabel yang telah anda buat, digunakan perintah *rm*. Misalnya, dalam *session* sebelumnya, anda telah membuat variabel *x* dengan nilai 2,3, untuk menghapus variabel *x*, digunakan fungsi berikut.

```
# Mengisi variabel x
> x= x *123
# Menampilkan variabel x
> print (x)
[1] 2952
# Menghapus variabel x
> rm (x)
```

```
> print (x)
Error in print(x) : object 'x' not found
```

Perintah untuk menghapus *data frame* yang telah dilampirkan dapat menggunakan perintah fungsi *detach* seperti berikut

```
> detach (bacteria)
```

### Test Formatif

No	Jawaban	Hasil	No	Jawaban	Hasil
1	B	<i>ok</i>	6	A	<i>ok</i>
2	B	<i>ok</i>	7	C	<i>ok</i>
3	C	<i>ok</i>	8	C	<i>ok</i>
4	A	<i>ok</i>	9	<del>B</del> (D)	<i>not</i>
5	<del>B</del> (D)	<i>not</i>	10	A	<i>ok</i>



## 2. Module 2 : Mekanisme Kerja, Pembuatan Objek, dan Informasi Bantuan dalam Sistem R

### K. Belajar 1 Mekanisme Kerja Sistem R

Sebagaimana yang dijelaskan didepan bahwa R adalah perangkat lunak dan lingkungan pemrogram. R adalah bahasa yang langsung diterapkan oleh sistemnya, hal ini berbeda dengan bahasa pemrogram lainnya, seperti C, Fortran, dan Pascal yang harus dikompilasi terlebih dahulu untuk membangun suatu program yang lengkap. Dalam R, setiap perintah yang dituliskan langsung dijalankan oleh program R dan dapat langsung dilihat hasilnya.

Setiap fungsi didalam R, selalu ditulis dengan sepasang tanda kurung dibelakangnya walaupun fungsi tersebut tidak membutuhkan argumen yang harus ditulis diantara tanda kurung. Contoh :

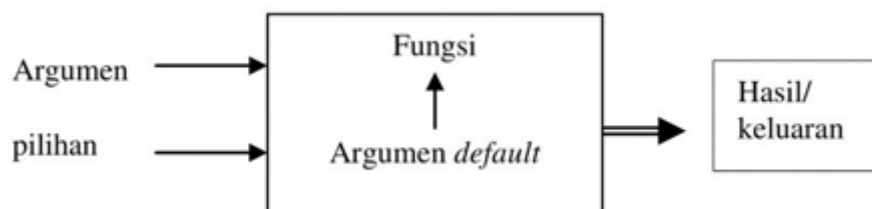
```
> ls()  
[1] "bacteria" "variane"
```

Penulisan fungsi dengan menggunakan tanda kurung dibelakang tujuannya untuk membedakan fungsi dari objek-objek lain di dalam R.

Ketika R sedang berfungsi, semua variabel, data, dan fungsi disimpan dalam bentuk objek di dalam *Random Access Memory (RAM)* komputer. Dalam beberapa literatur R, tempat pelaksanaan operasi disebut memori aktif. Anda dapat melakukan sesuatu pada objek-objek tersebut dengan operator, misalnya aritmatika, logika, perbandingan, serta fungsi.

Penggunaan fungsi sangat intuitif sebagaimana yang dijelaskan dalam bagan berikut (Gambar 2-1 Bagan Penggunaan Fungsi R)

Gambar 2-1 Bagan Penggunaan Fungsi R



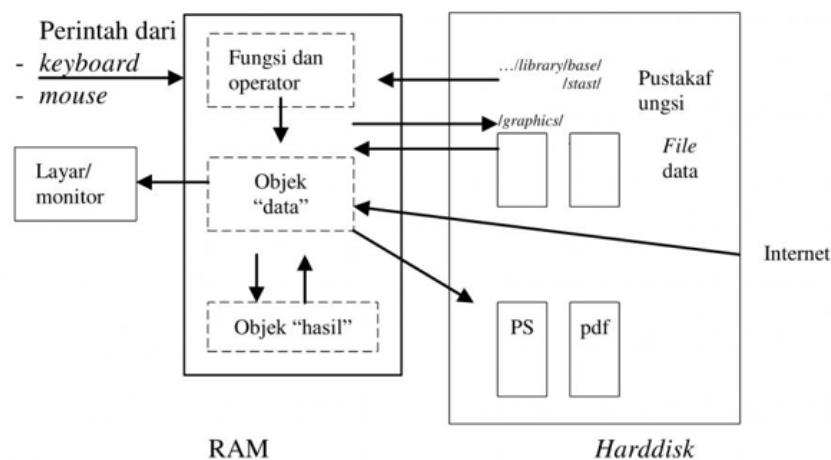
Nilai argumen *default* adalah nilai dari argumen yang telah ditentukan sebelumnya. Argumen dapat berupa objek (data, rumus atau ekspresi lainnya). Beberapa diantaranya mungkin telah ditentukan sebagai nilai *default*-nya. Anda dapat juga merubah nilai *default* tersebut dengan

menggunakan pilihan atau *option*. Fungsi-fungsi dalam R ada yang membutuhkan argumen, termasuk argumen *default*, tetapi ada juga yang tidak membutuhkan argumen sama sekali.

Sebagaimana yang telah dijelaskan sebelumnya, bahwa semua objek yang akan dioperasikan oleh R akan disimpan didalam RAM. RAM adalah jenis memori yang apabila mesin dimatikan isi dari RAM tersebut akan hilang.

Apakah setiap kali mengelola data dengan R harus memasukan kembali ke RAM dengena mengetikanya ? Jawabanya tidak selalu demikian karena perangkat lunak R telah melengkapi dirinya dengan sarana pembacaan (*Reading*) dan penulisan (*writing*) data yang diperlukan dala operasi dengan R. Sarana pembacaan dipakai untuk menyalin data yang disimpan terlebih dahulu dalam media lain, seperti *hard disk*, atau *flash disk*. Bahkan masukan data untuk R dapat dilakukan melalui internet. Sementara itu, faslitas penulisan digunakan untuk menyimpan hasil operasi R kedalam media lain. Gambar berikut menjelaskan mekanisme pemidahan data dari *hard disk* ke RAM dan sebaliknya.

Gambar 2-2 Mekanisme Pemindahan Data dari Hardisk ke RAm dan Sebaliknya



Dari Gambar 2-2 terlihat bahwa hasil analisis dari sistem R dapat disimpan dalam format *Post Script* (PS), *Portable Data Format* (PDF), dan format lainnya seperti jpg, png, bmp, emf, pictex, serta xfig.

Fungsi-fungsi yang tersedia didalam sistem R disimpan dalam direktori `R-4.1.0/library`. `R-4.1.0` adalah direktori ketika kita menginstal R, dalam hal ini, R yang terinstal adalah versi 4.1.0. Direktori ini merupakan tempat penyimpanan paket-paket fungsi secara terstruktur. Paket yang bernama *base* adalah sistem inti R dan berisi fungsi-fungsi utama sistem R, khususnya untuk pembacaan dan perhitungan serta analisis data. Setiap paket disimpan dalam direktori dan diberi

nama yang merupakan gabungan dari R dan nama paket itu sendiri, misalnya, `R\R-4.1.0\library\base\R\base.`

Beberapa contoh perintah yang relevan disini diantara lain adalah

```
# Mendapatkan informasi tempat kerja yang sedang berlangsung
> getwd()
[1] "C:/Users/farras/Documents"

# setwd() untuk mengganti tempat kerja yang sedang berlangsung.
# argumen pertama : nama direktori tujuan
> setwd ("D:/10. KULIAH/02. SEMESTER 2/01. KOMPUTER 1/Belajar R")

# dir() untuk menayangkan isi dari direktori kerja
> dir()
```

Contoh salah satu perintah yang paling sederhana adalah menuliskan nama objek untuk menampilkan isi dari suatu objek. Misalnya, suatu objek `myVariable` yang mempunyai nilai 15 seperti dibawah ini.

```
> myVariable <- 15
> myVariable
[1] 15
```

Angka 1 dalam siku kurung menunjukkan bahwa penayangan nilai `myVariable` dimulai dari nilai yang pertama. Perintah tersebut adalah pemakaian perintah `print(myVariable)` secara implisit. Itu artinya perintah `print(myVariable)` akan menghasilkan nilai yang sama dengan menuliskan `myVariable` saja.

Namun demikian, dalam beberapa situasi, misalnya dalam fungsi atau *loop*, perintah `print()` harus dituliskan secara eksplisit dan lengkap. Sebagai contoh, perintah berikut ini mencetak angka 1 sampai dengan 5 menggunakan pernyataan `for` untuk melakukan perulangann (*loop*).

```
> for (i in 1:5) { print (i) }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Penamaan suatu *object* bersifat *case sensitive*. Itu artinya huruf kecil dan kapital dianggap berbeda sehingga X dan x merupakan objek yang berbeda. Aturan penamaan objek dalam R adalah.

1. Diawali dengan huruf, baik huruf kapital maupun huruf kecil.
2. Diikuti oleh huruf, angka, titik, dan garis bawah.

## K. Belajar 2 Pembuatan, Penayangan, serta Penghapusan Objek dalam R dan Informasi Bantuan Online.

Suatu objek dapat dibuat dengan operator yang digambarkan dengan anak panah. Anak panah ini bisa mengarah ke kanan dan ke kiri. Variabel yang nilainya ditetapkan diletakan di depan anak panah.

```
> # invalid
> m ->5
Error in 5 <- m : invalid (do_set) left-hand side to assignment
> # valid
> m <-5
> 24 -> n
```

Nilai dari variabel `m` juga dapat merupakan hasil perhitungan dari beberapa angkat seperti berikut.

```
> m <- ((2+3)/5)+9
> m
[1] 10
```

Nilai suatu variabel juga dapat diperoleh dari nilai variabel lainnya. Sebagai contoh, perintah berikut mengisi variabel `n` dengan nilai dari variabel `m`, sehingga nilai `m` sama dengan nilai `n`.

```
> m -> n
> n
[1] 10
```

Disamping itu, nilai suatu variabel juga dapat merupakan hasil perhitungan dari satu atau lebih variabel menggunakan operator aritmatika dasar  $+$ ,  $-$ ,  $*$ ,  $/$ , dan  $^$ . Sebagai contoh, nilai variabel `y` berikut adalah hasil perhitungan persamaan  $x^2 + 5x + 12$ , yaitu nilai  $x$  terlebih dahulu diisi nilai 15.

```
> 15 -> x
> y <- x^2 + 5*x - 12
> y
[1] 288
```

Nilai suatu variabel juga dapat ditentukan menggunakan fungsi yang tersedia dalam R. Sebagai contoh, variabel `y` diberi nilai  $\sqrt{x} + \frac{\sqrt{x}}{2} + 10$ , nilai  $x = 144$

```
> x <- 144
> y = sqrt(x) + (sqrt(x)/2) + 10
> y
[1] 28
```

Dalam perintah diatas  $\sqrt{x}$  dinyatakan sebagai `sqrt(x)`, yaitu `sqrt()` merupakan fungsi aritmatika yang tersedia dalam R. Fungsi aritmatika lainnya yang tersedia didalam R, diantaranya, *log*, *exp*, *sin*, *cos*, dan *tan*.

R beroperasi tidak hanya pada variabel yang bernilai konstanta, tetapi juga pada struktur data yang bernama vektor numerik.

|| *Vektor numerik adalah entitas tunggal yang terdiri atas kumpulan terurut bilangan-bilangan.*

Sebagai contoh, untuk membuat vektor yang diberi nama `x` yang berisi lima bilangan 10, 6, 2, 41, dan 2, digunakan perintah R berikut.

```
> x <- c(10, 6, 2, 41, 2)
> x
[1] 10  6  2 41  2
```

Pernyataan penugasan `x <- (10, 6, 2, 41, 2)` menggunakan fungsi `c()` untuk merangkai nilai-nilai vektor.

Kita dapat membuat vektor-vektor numerik lainnya dari vektor numerik yang telah ada dengan menggunakan pernyataan aritmatika yang melibatkan vektor yang telah ada, konstanta, dan operator-operator aritmatika. Sebagai contoh, vektor `y` diperoleh dari vektor `x` dengan mengalikan setiap elemen dalam vektor `x` dengan 2.

```
> y <- x*2
> y
[1] 20 12  4 82  4
```

Contoh lain pernyataan aritmatika menggunakan vektor sebagai berikut.

```
> n <- x^2+2*x+1
> n
[1] 121 49 9 1764 9
```

Kita dapat menerapkan fungsi-fungsi yang tersedia didalam R terhadap vektor numerik.

```
> x <- c(1,4,6,2,8,7,9,1,10,3)
> length(x)
[1] 10
> min(x)
[1] 1
> max(x)
[1] 10
> sum(x)
[1] 51
> mean (x)
[1] 5.1
> sort (x)
[1] 1 1 2 3 4 6 7 8 9 10
```

Dengan menggunakan fungsi *sum()*, *mean()*, dan *length()*, kita juga dapat menghitung nilai *vairance* dari vektor *x* sebagai berikut. Namun R sendiri memiliki fungsi *var()* untuk menghitung nilai *variance*.

$$variance = \frac{\sum_{i=1}^n (x - \bar{x})^2}{(n - 1)}$$

```
> (sum((x-mean(x))^2))/(length(x)-1)
[1] 11.21111
> var (x)
[1] 11.21111
```

Kita juga dapat mengetahui nama-nama variabel dan vektor yang telah kita buat menggunakan fungsi `ls()`, `ls(pat = "...")`, atau `ls(pat="^...")`. `ls` merupakan kependekan dari *list simply*. Apabila anda mempunyai banyak objek, sebaiknya anda menamai objek seinformatif mungkin sehingga anda bisa menggunakan opsi *pattern* atau pola yang dapat disingkat dengan *pat* untuk menayangkan nama-nama objek yang mengandung karakter atau huruf tertentu saja.

```
> ls (pat = "va")
[1] "variane"

> ls (pat = "ia")
[1] "bacteria" "myVariabel" "myVariable" "variane"

# tanda ^ untuk mencari objek yang diawali dengan huruf m.
> ls (pat = "^m")
[1] "m" "myVariabel" "myVariable"
```

Perintah `ls.str()` dapat dipakai untuk menayangkan informasi objek secara lebih rinci (misalnya tipe dan nilai),

```
> ls.str()
bacteria : function ()
i : int 10
m : num 10
myVariabel : num 15
myVariable : num 15
n : num [1:5] 121 49 9 1764 9
variane : num 24
x : num [1:10] 1 4 6 2 8 7 9 1 10 3
y : num [1:5] 20 12 4 82 4

# dengan pola / pattern
> ls.str(pat = "^m")
m : num 10
myVariabel : num 15
myVariable : num 15
```



Untuk menghapus objek dalam memori, gunakanlah fungsi atau perintah `rm()`, `rm` merupakan kependekan dari *remove*.

Pernyataan	Makna
<code>rm(x)</code>	Menghapus variabel <i>x</i>
<code>rm(x,y)</code>	Menghapus variabel <i>x</i> & <i>y</i>
<code>rm (list=ls())</code>	Menghapus semua objek
<code>rm (list = ls (pat = "^m"))</code>	Menghapus semua objek yang namanya diawali dengan huruf "m"

```
> ls()
[1] "bacteria"  "i"         "m"         "myVariabel" "myVariable"
[6] "n"         "variane"   "x"         "y"
> rm (list = ls (pat = "^m"))

> ls()
[1] "bacteria" "i"         "n"         "variane"   "x"         "y"
# Objek myVariabel, myVariable, dan m dihapus.
```

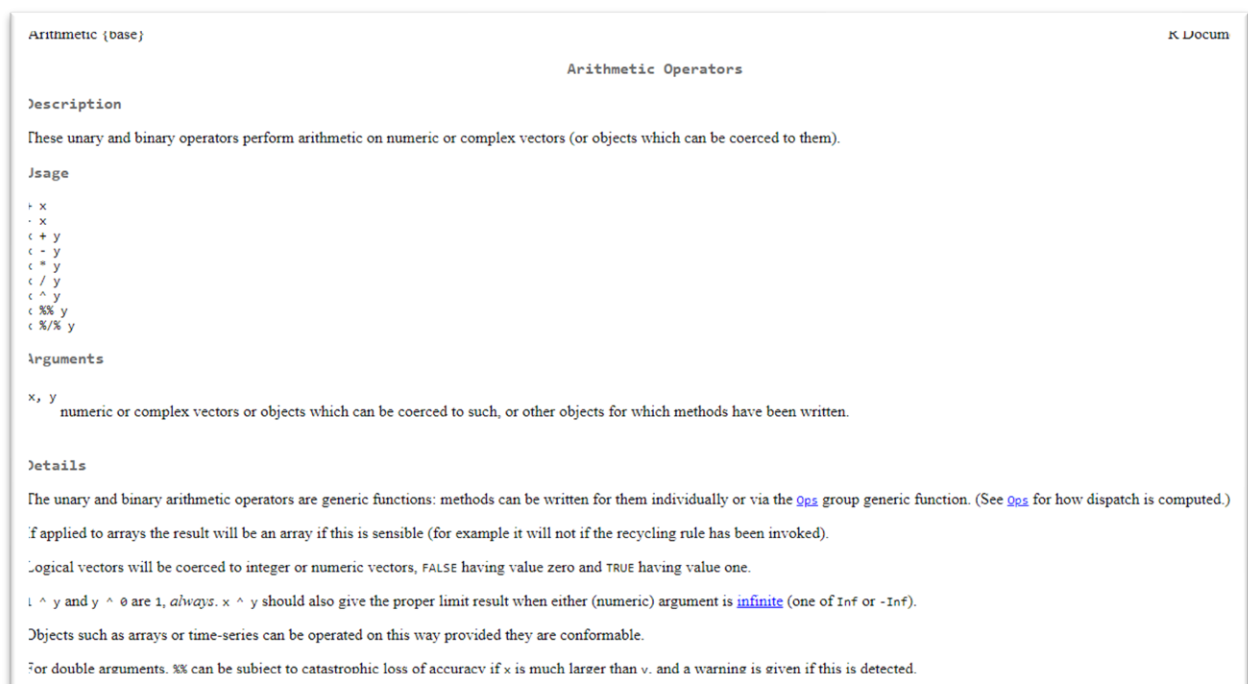
Bantuan *online* dalam R akan sangat membantu dalam menggunakan R karena bantuan tersebut memberikan informasi lengkap.

```
> ?ls
starting httpd help server ... done
> help (ls)
> help ("ls")
```

perintah diatas akan menayangkan informasi bantuan yang berkenaan dengan fungsi `ls()`.

```
> help ("*")
```

*Gambar 2-3 Tampilan yang Dihasilkan dari Perintah help("\*")*



Perintah *help* hanya akan menampilkan sistem R yang telah dimasukan dalam R atau memori aktif karena nilai yang telah ditetapkan sebelumnya untuk argumen fungsi *help*, yaitu `try.all.packages`, adalah *false*. Untuk melihat semua bantuan, argumen tersebut harus bernilai benar. Untuk melihat nilai argumen `try.all.packages` sebelum dan sesudah diubah perhatikan contoh berikut.

```
> help ("bs")
No documentation for 'bs' in specified packages and libraries:
you could try '??bs'
```

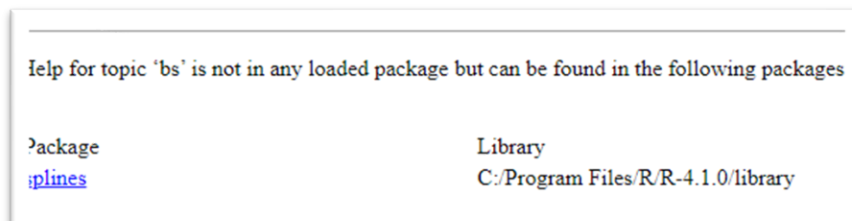
Pesan yang ditampilkan diatas menyatakan bahwa tidak ada dokumentasi dari "bs" dalam kumpulan paket dan pustaka yang telah dikumpulkan. Selain itu disarankan untuk menggunakan perintah `??bs`.

Cara menggunakan `try.all.packages` sebagai berikut.

```
> help ("bs", try.all.packages = TRUE)
```

Perintah diatas akan menghasilkan pesan berikut yang tampil pada halaman *web* pada *browser*.

*Gambar 2-4 Hasil dari help dengna argumen try.all.pacages=true*



Dalam kasus ini, informasi bantuan tidak ditayangkan secara langsung. Namun anda mendapatkan informasi bahwa informasi bantuan tersebut dapat ditemukan. Berdasarkan informasi diatas (Gambar 2-4), anda dapat menggunakan perintah berikut.

```
> help(package="splines")
```

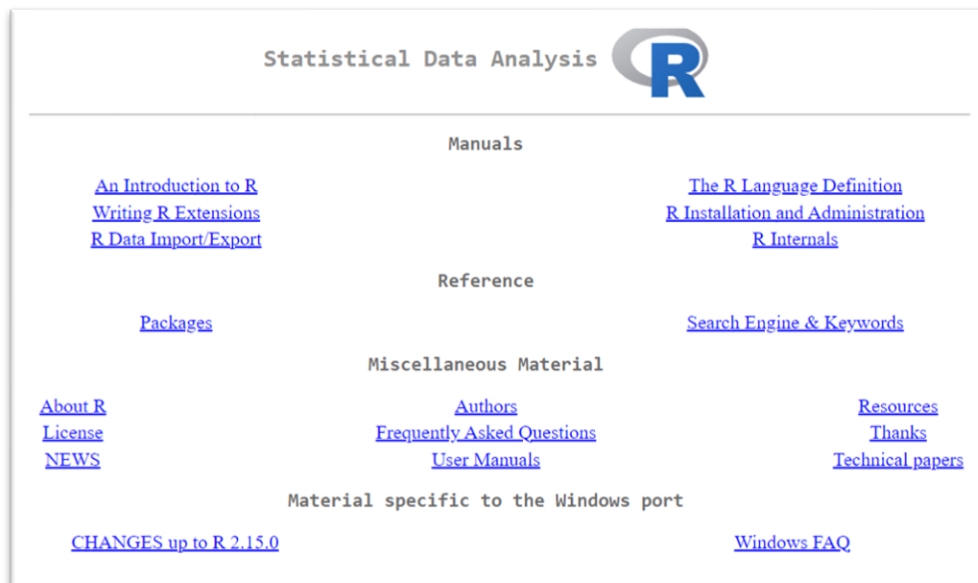
*Gambar 2-5 Tampilan yang Dihasilkan dari perintah help(package="splines")*



Anda juga dapat menampilkan informasi bantuan dalam format HTML yang dipakai untuk halaman web di Internet dengan perintah berikut. dan akan muncul seperti (Gambar 2-6)

```
> help.start ()  
If nothing happens, you should open  
'http://127.0.0.1:27911/doc/html/index.html' yourself
```

Gambar 2-6 Hasil dari `help.start()`



Pencarian informasi dengan memakai kata kunci dimungkinkan pada perintah `help` dengan memakai format html. perintah yang digunakan adalah `help.search()`.

```
> help.search("tree")
```

Pencarian informasi bantuan tersebut akan dilakukan pada semua paket yang terinstal. Oleh karena itu, Apabila anda baru saja melakukan instalasi paket baru, Anda perlu menggunakan opsi `rebuild` dan mengubah nilai dari opsi menjadi `true`.

```
> help.search("tree", rebuild=TRUE)
```

Perintah atau fungsi `apropos()` dipakai untuk menemukan informasi bantuan mengenai fungsi yang terdapat dalam argumen pemanggilan fungsi tersebut. Dalam hal ini, pencarian hanya dilakukan pada memori yang sedang aktif (RAM). Berikut adalah contoh untuk menampilkan informasi bantuan yang mengandung "help".

```
> apropos ("help")  
[1] "help"          "help.request"  "help.search"   "help.start"
```

### 3. Modul 3 : Data dalam Sistem R

Pembahasan pada modul ini meliputi jenis objek beserta karakteristiknya, pembacaan dan penyimpanan data dari *file* komputer. serta pembuatan data dan manipulasi data dengan berbagai operasi dan fungsi.

Capaian :

- ♦ Mengetahui Jenis Objek dan karakteristiknya dalam R
- ♦ Mampu membaca data dari *file* yang sudah tersedia di media penyimpanan.
- ♦ Menulis objek dalam bentuk *file* di media penyimpanan.
- ♦ Membangun data dengan berbagai fungsi dasar dalam R.
- ♦ Membuat dan memanipulasi data dengan berbagai operasi aritmatika dan fungsi sederhana.

## K. Belajar 1 Jenis-Jenis Objek dan Operasi Dasar Aritmatika dalam R

Setiap objek yang dipakai selalu memiliki nama. Nilai serta atribut yang menentukan jenis data dinyatakan oleh suatu objek. Dalam pengolahan data suatu variabel yang bernilai 1, 2 dan 3, belum tentu variabel memiliki satu jenis data saja. Variabel tersebut dapat berjenis bilangan riil atau bilangan bulat, seperti jumlah siswa, atau bisa juga variabel kategorikal, gabungan antara beberapa jenis data seperti kode unit pelaksanaan belajar jarak jauh (UPBJJ) yang dipakai di lingkungan Universitas Terbuka.

*Operasi pengolahan data tergantung dari tipe data yang tersedia. Jika salah dalam pengolahan data maka hasilnya akan sia-sia.*

Dengan memahami jenis datanya, tentu konsekuensi operasi dalam pengolahan data atau analisis statistiknya menjadi berbeda. Anda tidak bisa menjumlahkan angka-angka kode UPBJJ untuk mendapatkan nilai rata-rata kode UPBJJ diantara 100 mahasiswa yang ada dalam penelitian. Walaupun secara manual dapat saja mendapatkan nilai rata-rata tersebut dengan menjumlahkan kode UPBJJ dan dibagi dengan jumlah mahasiswa; tindakan tersebut tidak bermakna sama sekali. Hal ini disebabkan angka-angka tersebut hanyalah kode yang mewakili nama-nama UPBJJ.

Semua objek dalam R memiliki dua atribut instrinsik, diantaranya adalah;

### 1. Mode

Jenis atau tipe suatu objek. Mode dari suatu objek dapat berupa Tabel 3 Jenis Mode (Tipe-Tipe).

Untuk melihat jenis dari suatu objek dapat menggunakan fungsi `mode` atau `class`. Gunakan fungsi `in.{tipe objek}(value)` akan mengembalikan nilai `TRUE` jika benar dan sebaliknya.

*Tabel 3 Jenis Mode (Tipe-Tipe)*

Jenis	Dalam R mode (penulisan)	Contoh
Numerik	<b>num</b>	<pre>&gt; mode (27) [1] "numeric"  &gt; is.numeric (27.0) [1] TRUE</pre>

Jenis	Dalam R mode (penulisan)	Contoh
		<pre># 27 tetap dianggap sebagai numerik oleh R. Salah satu kekurangan dari R. &gt; is.integer (27) [1] FALSE</pre>
Karakter	<b>char</b>	<pre>&gt; mode ("Hai") [1] "character"  # "27" akan menjadi karakter, bukan numerik &gt; mode ("27") [1] "character"  &gt; is.character ("27") [1] TRUE</pre>
Complex	<b>com</b>	<pre>&gt; mode (4i) [1] "complex"  &gt; is.complex (4i) [1] TRUE</pre>
Logic	<b>log</b>	<pre>&gt; mode (true) Error in mode(true) : object 'true' not found  # Penulisan tipe objek logikal harus huruf kapital &gt; mode (TRUE) [1] "logical"  &gt; is.logical (TRUE) [1] TRUE</pre>

## 2. Panjang

Panjang dari suatu objek dapat diketahui dengan menggunakan fungsi `length()` dengan parameter sebuah nilai atau variabel

```
> length (24)
[1] 1
> length (24)
[1] 1
> x <- c (2,5,2,1,5)
> length (x)
[1] 5
```

Bilangan yang sangat besar dan sangat kecil dapat dinyatakan secara eksponensial. Misalnya, 17.000.000 dapat dinyatakan dengan `17e6`

```
> n = 17000000
> m = 17e6
> n
[1] 1.7e+07
> m
[1] 1.7e+07
> n == m
[1] TRUE
```

Sistem R mempunyai cara yang standar atau mirip dengan sistem yang lain dalam menangani data yang hilang (N/A, *No Availabel*), bilangan tak hingga ( $\text{Inf } +\infty / -\infty$ ), dan sesuatu yang seharusnya bilangan tapi bukan bilangan (NaN, *Not a Number*).

```
> x <- 5
> y = x/0
> y
[1] Inf
> y-y
[1] NaN
```



Nilai dari variabel dengan mode karakter dimasukan dalam tanda petik *double* atau *single*. Dimungkinkan pula untuk memasukan karakter dengan diikut dengan tanda \. Kombinasi tanda \ dengan " akan diperlukan secara khusus oleh beberapa fungsi, misalnya `cat` yang berkaitan dengan penayangan di layar atau `write.table` untuk menulis hasil kedalam *disk*.

```
> x <- "jum'at"
> x
[1] "jum'at"
> y <- 'jum'at'
Error: unexpected symbol in "y <- 'jum'at"
> y <- 'jum\'at'
> y
[1] "jum'at"
> cat ("hari ini hari ",y)
hari ini hari  jum'at
```

Tabel 4 Tipe-tipe Objek untuk Penyimpanan Data berikut ini memberi gambaran berbagai jenis objek untuk meyimpan data.

Tabel 4 Tipe-tipe Objek untuk Penyimpanan Data

Objek	Mode (Tipe)	Kemungkinan beberapa mode dalam satu objek
vector	numerik, karakter, kompleks, atau logika	tidak
faktor	numerik atau karakter	tidak
array	numerik, karakter, kompleks, atau logika	tidak
matriks	numerik, karakter, kompleks, atau logika	tidak
frame data	numerik, karakter, kompleks, atau logika	ya
ts (time series)	numerik, karakter, kompleks, atau logika	tidak
list	numerik, karakter, kompleks, atau logika fungsi, ekspresi	ya

- ❖ Vektor adalah variabel dalam makna yang telah disepakati secara umum.
- ❖ Faktor adalah variabel kategorikal.
- ❖ Array adalah tabel berdimensi  $k$ .  $k$  adalah jumlah atau panjang suatu,
- ❖ Matriks adalah *array* berdimensi dua.
- ❖ Frame data adalah suatu tabel yang terdiri atas satu atau beberapa vektor yang sama panjangnya, tetapi dimungkinkan mempunyai mode yang berbeda.
- ❖ ts (*time series*) adalah data runtun waktu yang mengandung informasi tambahan, misalnya waktu, tanggal, dan frekuensi.
- ❖ List adalah objek yang dapat mengandung sembarang objek, termasuk list itu sendiri.

Dalam suatu vektor, mode dan panjangnya sudah cukup untuk menjelaskan datanya. Namun untuk objek lain, informasi lain masih diperlukan dan informasi ini diberikan oleh atribut nonintrinsik, diantaranya atribut `dim` yang mengacu pada dimensi dari objek. Sebagai contoh matriks dengan dua baris dan dua kolom.

## K. Belajar 2 Membaca dan Menulis *File* Data dalam R

R menggunakan direktori tempat kerja atau *working directory*. Untuk mengetahui direktori tempat kerja ini digunakan perintah `getwd()`. Untuk mengubah atau pindah direktori digunakan perintah `setwd()`, sementara untuk mengetahui isi dari direktori, gunakan perintah `dir()`.

R mampu membaca data dalam berbagai format, misalnya ASCII, Excel, SAS, dan SPSS. R dapat juga membaca data yang disimpan dalam basis data yang diakses dengan SQL (*Structured Query Language*). Namun *package* untuk membaca data dalam format tersebut tidak terdapat dalam *base package*, kita harus menginstall *library* tambahan kedalam sistem R.

Tabel 5 Cara Membaca File Dan Penjelasannya<sup>1</sup>

Fungsi	Penggunaan	Penjelasan
<code>getwd()</code>	<pre>&gt; getwd() [1] "D:/direktori/lokasi/"</pre>	Menampilkan alamat dari <i>working directory</i>
<code>setwd(Path)</code>	<pre>&gt; setwd("direktori//direktori lain//tujuan") atau &gt; setwd("direktori\\direktori lain\\tujuan")</pre>	Berpindah <i>working directory</i>
<code>dir()</code>	<pre>&gt; dir() [1] "Data" "RScript"  # Nested, untuk melihat isi dari subdirektori &gt; dir(dir()[1])[1] [1] "data_saya.dat"</pre>	Untuk melihat isi dari direktori
<code>read.table()</code>	<pre>&gt;myData&lt;- read.table("data/data_saya.dat")</pre>	Untuk membaca isi <code>data_saya.dat</code> dan menyimpannya dalam <i>frame data</i> yang diberi nama <code>myData</code> .
	<pre>&gt; myData</pre>	Untuk menampilkan isi <i>framedata</i>

<sup>1</sup> Script : Belajar R\Modul 3\K.B 2\RScript\1. Cara Membaca File dan Penjelasannya.R

## Cara Mengakses Nilai dari Data Frame

Untuk mengakses nilai dari *data frame* tersebut dapat dilakukan dengan cara sebagai berikut: `myData$V1`, `myData$V2`, dan `myData$V3`. Bisa juga dengan cara `myData["V1"]`, `myData["V2"]`, dan `myData["V3"]`. Selain itu juga bisa dengan `myData[,1]`, `myData[,2]`, `myData[,3]`. Hasil lengkapnya dapat dilihat di Lampiran 4-2 Berbagai Cara untuk Mengakses Nilai dari Data Frame

## Argumen atau Opsi `read.table()`

### Deskripsi

Membaca *file* dengan format tabel dan menjadikannya sebuah *data frame*, with cases corresponding to lines and variabel to fields in the file.

### Penggunaan

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = FALSE,
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```

### Argumen - File<sup>2</sup>

*Bernilai String* : Alamat dan nama dari *file* yang ingin dibaca. Setiap *row* (baris) dari tabel ditampilkan sebagai satu garis di dalam *file*. Jika argumen ini tidak berisi alamat absolut, maka nama *file* bisa berupa alamat relatif dari *working director* saat ini, `getwd()`.

*file* juga dapat diambil dari URL.

```
# Alamat relatif
> dataMahasiswaRlf <- read.table ("Data/nilai_mahasiswa.dat")
# Alamat absolut
> dataMahasiswaAbs <- read.table ("D:{alamat}/Data/nilai_mahasiswa.dat")
```

<sup>2</sup> Script : [Belajar R\Modul 3\K.B 2\RScript\3. read.table - file.R](#)

### Argumen - header<sup>3</sup>

*Bernilai Logic* : TRUE atau FALSE. Nilai tersebut mengidentifikasi apakah *file* tersebut memiliki nama variabel pada baris pertamanya. Jika tidak ada (header=FALSE) maka, nilai *header* ditentukan oleh *file* format. Seperti V1, V2. atau V3. *Default value* dari argumen ini adalah FALSE, jika TRUE maka baris pertama dieksekusi sebagai nilai *header*.

```
> dataMahasiswaNoHeader <- read.table ("Data/nilai_mahasiswa.dat", header = FALSE)
```

Gambar 3-1 Data Fram dengan Argumen header = FALSE

	V1	V2	V3	
1	nim	nama	nilai	
2	042910503	Muhammad Farras Ma'ruf	98	Header ditentukan oleh file format
3	042412566	Richard Hendric	97,5	
4	059251522	Mark L Murphy	92,3	
5	052824875	Abu Bakar Ash Shidiq	84,2	
6	082885728	Jamesh Goshling	86,2	

```
> dataMahasiswaWHeader <- read.table ("Data/nilai_mahasiswa.dat", header = TRUE)
```

Gambar 3-2 Data Frame dengan argumen header = TRUE

	nim	nama	nilai	
1	42910503	Muhammad Farras Ma'ruf	98	Header adalah nilai dari baris pertama
2	42412566	Richard Hendric	97,5	
3	59251522	Mark L Murphy	92,3	
4	52824875	Abu Bakar Ash Shidiq	84,2	
5	82885728	Jamesh Goshling	86,2	

### Argumen - sep<sup>4</sup>

*Bernilai Character* : Pemisah kolom yang digunakan dalam *file* data. Nilai dari setiap baris pada sebuah *file* dipisahkan oleh *Character* ini. Jika jika *sep* = "" (Nilai bawaan) maka pemisahannya adalah 'white space'.

Jika pemisahannya adalah tab maka bisa gunaka nilai *sep*="\t". Jika koma bisa kita gunakan *sep*=", ".

```
> path <- "Data/nilai_mahasiswa_separator.dat"
> dataMahasiswaSep <- read.table (path, header = TRUE, sep="|")
> dataMahasiswaSep
```

	nim	nama	nilai
1	42910503	Muhammad Farras Ma'ruf	98
2	42412566	Richard Hendric	97,5
3	59251522	Mark L Murphy	92,3
4	52824875	Abu Bakar Ash Shidiq	84,2

<sup>3</sup> Script : [Belajar R\Modul 3\K.B 2\RScrip\4. read.table-header.R](#)

<sup>4</sup> Script : [Belajar R\Modul 3\K.B 2\RScrip\5. read.table-sep.R](#)

```
5 82885728      Jamesh Goshling  86,2
```

### Argumen - quote<sup>5</sup>

*Bernilai Character* : Kumpulan dari *quoting character* sebagai karakter *string* tunggal.

Penggunaan `quote` biasa digunakan seperti dibawah ini, dimana `sep` adalah argumen seperator.

```
quote = if(identical(sep, "\n")) "" else "'\""
```

Agar lebih mudah dipahami, argument `quote` digunakan untuk memberitahu R jika ada sebuah nilai yang diawali dan diakhiri dengan *quoting character* seperti ( ' , / , \* , & ) maka anggap string di dalamnya sebagai sebuah *character string* tunggal.

Contoh, kita memiliki data yang aneh seperti dibawah ini ;

```
nim nama &nama panggilan& nilai
042910503 'Muhammad Farras Maruf' 'faras maruf' 98
042412566 'Richard Hendric' 'richard' 97,5
059251522 *Mark L Murphy* 'Mark Murp' 92,3
052824875 'Abu Bakar Ash Shidiq' 'ABS' 84,2
082885728 "Jamesh Goshling" 'Jams' 86,2
```

Jika kita menjalankan perintah

```
> dataMahasiswaQuote <- read.table (path, header = TRUE)
Error in scan(file = file, what = what, sep = sep, quote = quote, dec = dec,
:
line 1 did not have 6 elements
```

Maka akan muncul pesan *error* yang mengatakan bahwa baris pertama tidak memiliki 6 elemen. Hal tersebut terjadi karena sebuah seluruh baris dari *file* tersebut tidak memiliki jumlah elemen yang konsisten. Dimana dijelaskan sebagai berikut;

- *Header* dari *file* tersebut (baris pertama) hanya memiliki 5 elemen (`nim`, `nama`, `&nama`, `panggilan&`, dan `nilai`)
- Baris ke-2 hanya memiliki 4 elemen (042910503, Muhammad Farras Maruf, dan 98)
- Baris ke-4 memiliki 6 elemen (049251522, \*Mark, L, Murphy\*, Mark Murp, dan 92,3)
- dan baris ke-6 hanya memiliki 4 elemen (082885728, Jamesh Goshling, Jams, dan 86,2)

Nilai 'Muhammad Farras Ma'ruf', 'richard', "James Ghoslin" dan sejenisnya dibaca menjadi satu kesatuan karakter string karena karakter " (Petik Ganda) dan ' (Petik) termasuk dalam quoting

<sup>5</sup> Script : [Belajar R\Modul 3\K.B 2\Script\6. read.table-quote.R](#)

*character. Sedangkan karakter & (Ampersand) dan \*(Asterisk) tidak, sehingga harus dimasukkan dalam argumen `quotes`*

Untuk mengetahui lebih dalam mari kita coba menambahkan argumen *fill* dengan nilai *true* (InsyaAllah argumen *fill* dibahas setelah ini). Intinya argumen *fill* akan mengisi nilai *blank* (kosong) jika ada kasus baris-baris tidak memiliki jumlah elemen yang sama.

```
> dataMahasiswaQuote <- read.table (path, header = TRUE, fill=TRUE)
> fix(dataMahasiswaQuote)
```

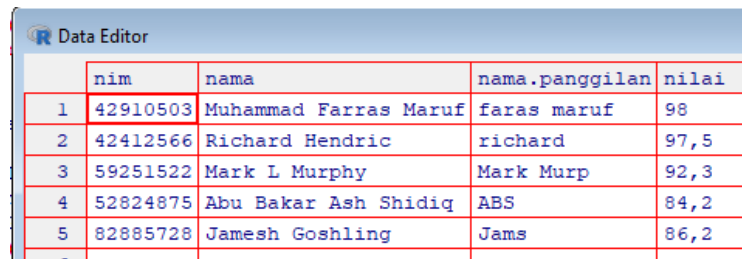
Gambar 3-3 Hasil dari `fix(dataMahasiswaQuote)`

row.names	nim	nama	X.nama	panggilan	nilai
1	042910503	Muhammad Farras Maruf	faras maruf	98	
2	042412566	Richard Hendric	richard	97,5	
3	059251522	*Mark	L	Murphy*	Mark Murp
4	052824875	Abu Bakar Ash Shidiq	ABS	84,2	
5	082885728	Jamesh Goshling	Jams	86,2	

Pada Gambar 3-3 Hasil dari `fix(dataMahasiswaQuote)` dapat dilihat pada border kuning bahwa nilai dari 2 elemen pada baris ke-1,2 dan ke-4,5 berisi *Blank* untuk menyesuaikan dengan jumlah elemen dari baris ke-3. Dan pada border hijau berisi nilai *row.names* yang mana nilai tersebut sama untuk menyesuaikan dengan baris ke-3. Untuk membentuk dan membuat *data frame* sesuai dengan yang kita inginkan dengan kondisi *file* seperti yang dijelaskan diawal. Dapat menggunakan argumen *quote* untuk memasukan daftar karakter yang membuat R paham bahwa karakter tersebut termasuk *quoting character*.

```
> dataMahasiswaQuote <- read.table (path, header = TRUE, quote="\",'*&")
> dataMahasiswaQuote
      nim      nama nama.panggilan nilai
1 42910503 Muhammad Farras Maruf  faras maruf    98
2 42412566   Richard Hendric    richard  97,5
3 59251522      Mark L Murphy    Mark Murp  92,3
4 52824875 Abu Bakar Ash Shidiq      ABS  84,2
5 82885728   Jamesh Goshling      Jams  86,2
> fix(dataMahasiswaQuote)
```

Gambar 3-4 Hasil dari `fix(dataMahasiswaQuote)` dengan argumen `quote`



	nim	nama	nama.panggilan	nilai
1	42910503	Muhammad Farras Maruf	faras maruf	98
2	42412566	Richard Hendric	richard	97,5
3	59251522	Mark L Murphy	Mark Murp	92,3
4	52824875	Abu Bakar Ash Shidiq	ABS	84,2
5	82885728	Jamesh Goshling	Jams	86,2

Argumen `quote="\"*&"` memberitahu R bahwa nilai yang berada antara karakter `"`, `'`, `*`, dan `&` adalah sebuah satu kesatuan *character string*.



## 4. Lampiran

### Lampiran 4-1 Kode hasil fungsi >example (lm)

```
> example (lm)

lm> require(graphics)

lm> ## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
lm> ## Page 9: Plant Weight Data.
lm> ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)

lm> trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)

lm> group <- gl(2, 10, 20, labels = c("Ctl","Trt"))

lm> weight <- c(ctl, trt)

lm> lm.D9 <- lm(weight ~ group)

lm> lm.D90 <- lm(weight ~ group - 1) # omitting intercept

lm> ## No test:
lm> anova(lm.D9)
Analysis of Variance Table

Response: weight
          Df Sum Sq Mean Sq F value Pr(>F)
group      1  0.6882  0.68820   1.4191  0.249
Residuals 18  8.7292  0.48496

lm> summary(lm.D90)

Call:
lm(formula = weight ~ group - 1)

Residuals:
    Min       1Q   Median       3Q      Max
-1.0710 -0.4938  0.0685  0.2462  1.3690

Coefficients:
```

```
      Estimate Std. Error t value Pr(>|t|)
groupCtl    5.0320      0.2202   22.85 9.55e-15 ***
groupTrt    4.6610      0.2202   21.16 3.62e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6964 on 18 degrees of freedom
Multiple R-squared:  0.9818,    Adjusted R-squared:  0.9798
F-statistic: 485.1 on 2 and 18 DF,  p-value: < 2.2e-16

lm> ## End(No test)
lm> opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))

lm> plot(lm.D9, las = 1)      # Residuals, Fitted, ...
Waiting to confirm page change...

lm> par(opar)

lm> ## Don't show:
lm> ## model frame :
lm> stopifnot(identical(lm(weight ~ group, method = "model.frame"),
lm+                      model.frame(lm.D9)))

lm> ## End(Don't show)
lm> ### less simple examples in "See Also" above
```

*Lampiran 4-2 Berbagai Cara untuk Mengakses Nilai dari Data Frame*

```
> # Berbagai cara untuk mengakses nilai dari data frame
> myData$V1
[1] "Nama" "Muhammad Farras Ma'ruf" "Tania Dwi Haryanti"
[4] "Muhammad Faris Ma'ruf" "Another Person"
> typeof (myData$V1);mode (myData$V1);class (myData$V1)
[1] "character"
[1] "character"
[1] "character"
> myData$V1[2]
[1] "Muhammad Farras Ma'ruf"
> # atau -----
> myData[,1]
[1] "Nama" "Muhammad Farras Ma'ruf" "Tania Dwi Haryanti"
[4] "Muhammad Faris Ma'ruf" "Another Person"
> typeof (myData[,1]);mode (myData[,1]);class (myData[,1])
[1] "character"
[1] "character"
[1] "character"
> myData[,3][2]
[1] "98"
> #-----
> # Mengakses kolom dari data frame
> myData[1]
      V1
1      Nama
2 Muhammad Farras Ma'ruf
3      Tania Dwi Haryanti
4 Muhammad Faris Ma'ruf
5      Another Person
> typeof (myData[1]);mode (myData[1]);class (myData[1])
[1] "list"
[1] "list"
[1] "data.frame"
> myData[5,1]
[1] "Another Person"
```

