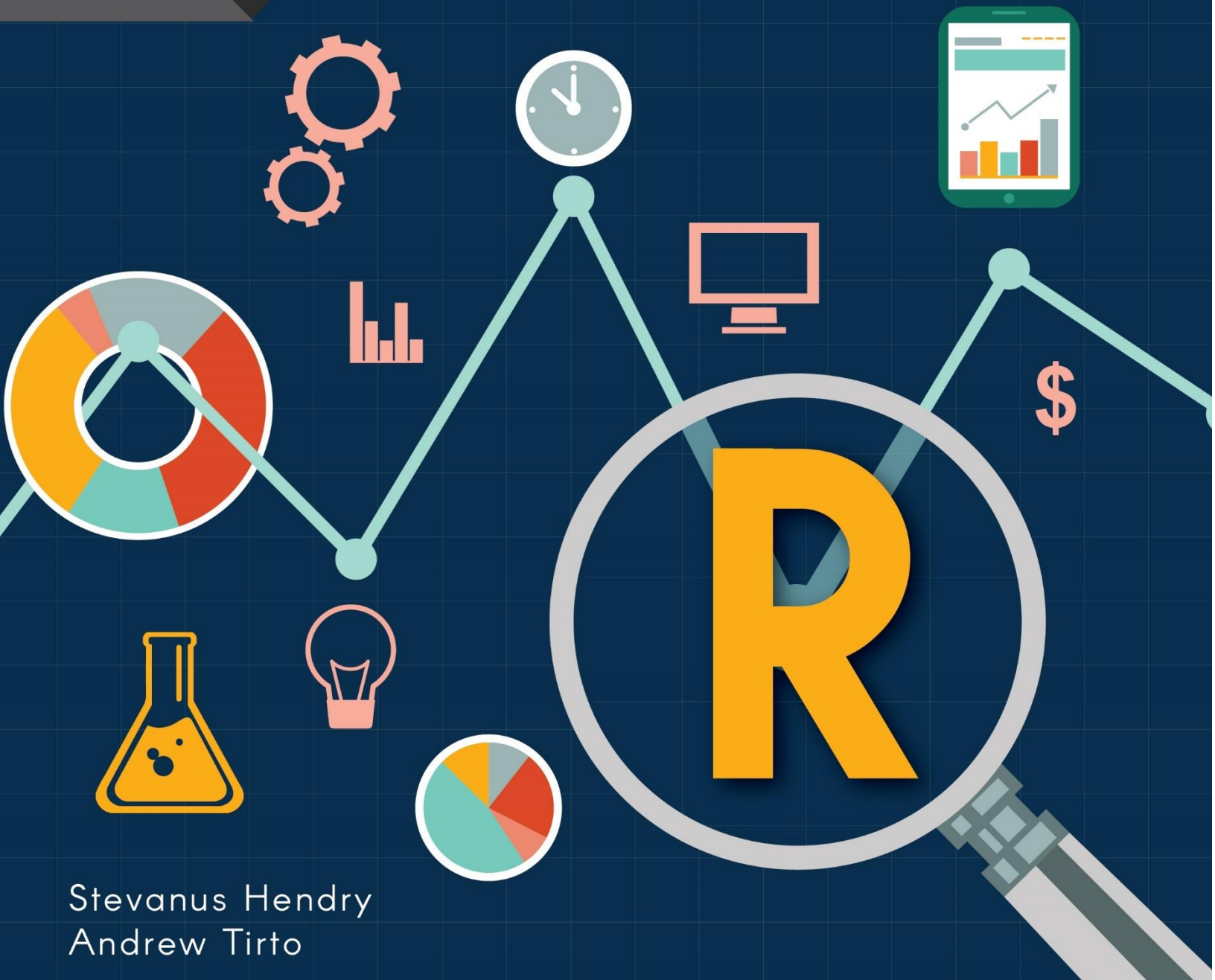


# DASAR BAHASA PEMROGRAMAN “R”

Machine Learning



Stevanus Hendry  
Andrew Tirto

## Profile Penulis



### Stevanus Henry Christian

Lahir di Jakarta pada 12 Juni 1992, Stevanus menyelesaikan Pendidikan S1 nya di Universitas Multimedia Nusantara dengan jurusan Sistem Informasi.

Ketertarikannya untuk mengeksplorasi sesuatu yang baru, terutama dalam hal Azure dan Machine Learning /R, membuat penulis mencoba menuliskan analisisnya dalam bentuk e-book yang dapat mengedukasi orang lain.



### Andrew Tirto Kusumo

Lahir di Semarang pada 10 April 1996, Andrew sedang menjalankan magang di PHI-Integration sebagai salah satu syarat untuk menyelesaikan pendidikan S1 nya di Universitas Multimedia Nusantara dengan jurusan Sistem Komputer. Setelah ditekuni sejak awal 2017, Penulis merasa tertantang untuk mendalami data analysis karena berkembangnya IoT dan data analysis adalah salah satu bidang yang sangat menarik. Selain itu, penulis sudah menekuni bahasa pemograman R sejak 2016.

Daftar Isi

Profile Penulis ..... 1

Daftar Isi ..... 2

Pendahuluan..... 3

Instalasi R ..... 4

Tampilan R ..... 7

Instalasi Rstudio ..... 8

Environment Introduction..... 9

Tipe Data Umum Dan Operasinya ..... 11

Operasi matematika..... 12

Pengenalan Data dan Plotting..... 21

Pengenalan Data Kualitatif ..... 22

Pengenalan Data Kuantitatif ..... 27

Operasi Statistik Dasar Menggunakan R dan Pemograman R ..... 32

Operasi Statistik..... 33

Subset Data dan Membuat Data ..... 41

Subset Data ..... 42

Pemrograman R

# Pendahuluan



## R Background

**R** adalah **bahasa pemrograman** untuk perhitungan statistika dan grafik. Ini adalah proyek dari *GNU* dan dikembangkan di Bell Laboratories oleh John Chambers dan temannya. R menyediakan banyak sekali jenis statistika dan teknik grafis, dan sangat mudah untuk menambahkan fungsi lain. R menyediakan jalur *Open Source* untuk orang dapat berpartisipasi dalam pengembangannya. Salah satu keunggulan R adalah untuk menghasilkan plot/grafik yang mempunyai kualitas sangat baik.

R adalah software terintegrasi untuk memanipulasi data, melakukan kalkulasi, dan menampilkan grafik. Ini meliputi :

Penanganan data yang efektif dan fasilitas storage

Operator untuk menangani perhitungan array / matrix

Kumpulan alat bantu yang banyak untuk analisa data

Bahasa pemrograman yang sangat teruji meliputi *conditionals*, *loops*, *recursive*, *function*, *input*, dan *output*.

## Instalasi R

Anda dapat melakukan instalasi program R dengan mengikuti proses dibawah ini.

Buka link <https://cran.r-project.org/mirrors.html> , unduh melalui *0-Cloud* atau di server Indonesia.

---

CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: [main page](#), [windows release](#), [windows old release](#).

If you want to host a new mirror at your institution, please have a look at the [CRAN Mirror HOWTO](#).

0-Cloud  
<https://cloud.r-project.org/>  
<http://cloud.r-project.org/>

Automatic redirection to servers worldwide, currently sponsored by Rstudio  
 Automatic redirection to servers worldwide, currently sponsored by Rstudio

Indonesia  
<https://repo.bppt.go.id/cran/>

Agency for The Application and Assessment of Technology

Unduh sesuai sistem operasi Anda.

**Download and Install R**

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Unduh melalui *subdirectories base*.

R for Windows

Subdirectories:

<a href="#">base</a>	Binaries for base distribution (managed by Duncan Murdoch). This is what you want to <a href="#">install R for the first time</a> .
<a href="#">contrib</a>	Binaries of contributed CRAN packages (for R >= 2.11.x; managed by Uwe Ligges). There is also information on <a href="#">third party software</a> available for CRAN Windows services and corresponding environment and make variables.
<a href="#">old.contrib</a>	Binaries of contributed CRAN packages for outdated versions of R (for R < 2.11.x; managed by Uwe Ligges).
<a href="#">Rtools</a>	Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Unduh program R versi terbaru


**R-3.4.1 for Windows (32/64 bit)**

[Download R 3.4.1 for Windows](#) (62 megabytes, 32/64 bit)

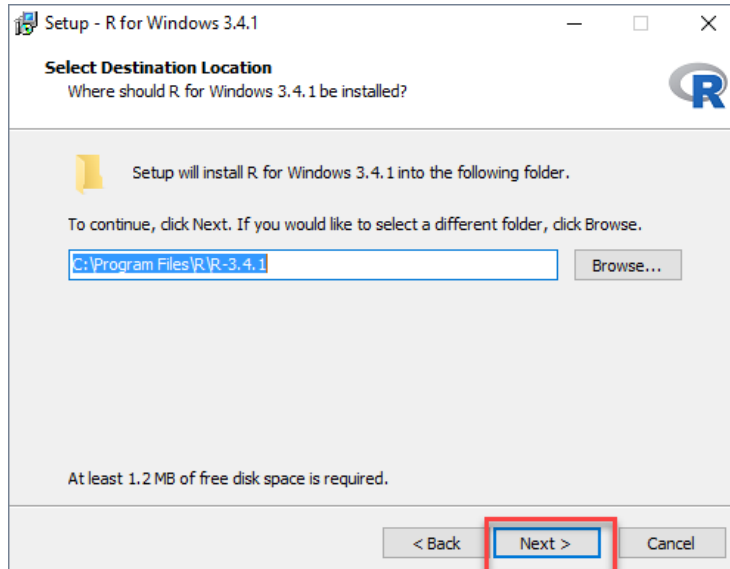
[Installation and other instructions](#)

[New features in this version](#)

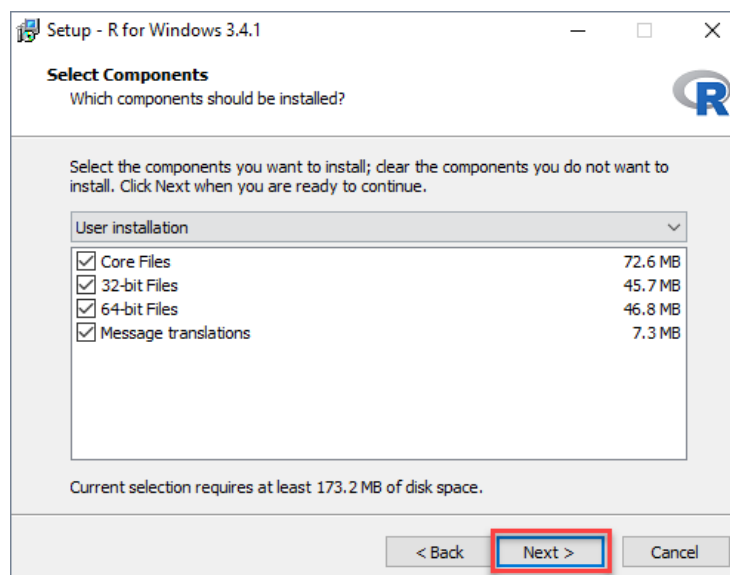
Buka folder tempat Anda melakukan pengunduhan

 R-3.4.1-win.exe	7/18/2017 10:16 AM	Application	76,257 KB
---	--------------------	-------------	-----------

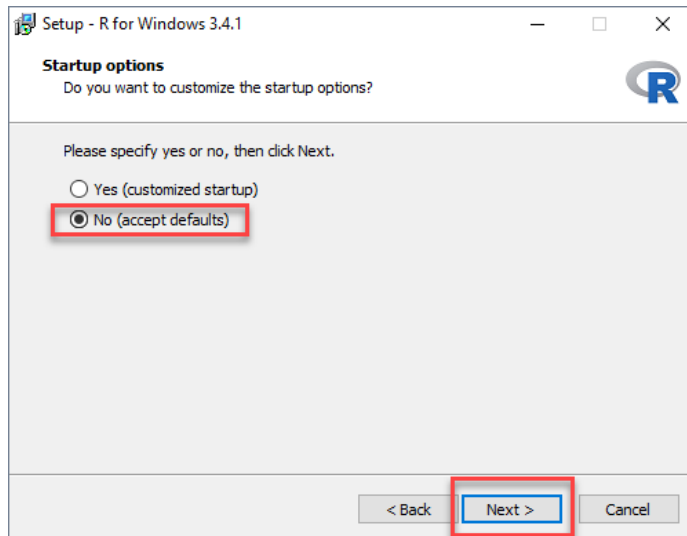
Tentukan folder penyimpanan R Anda, Klik **next**.



Install default sesuai rekomendasi, klik **next**.



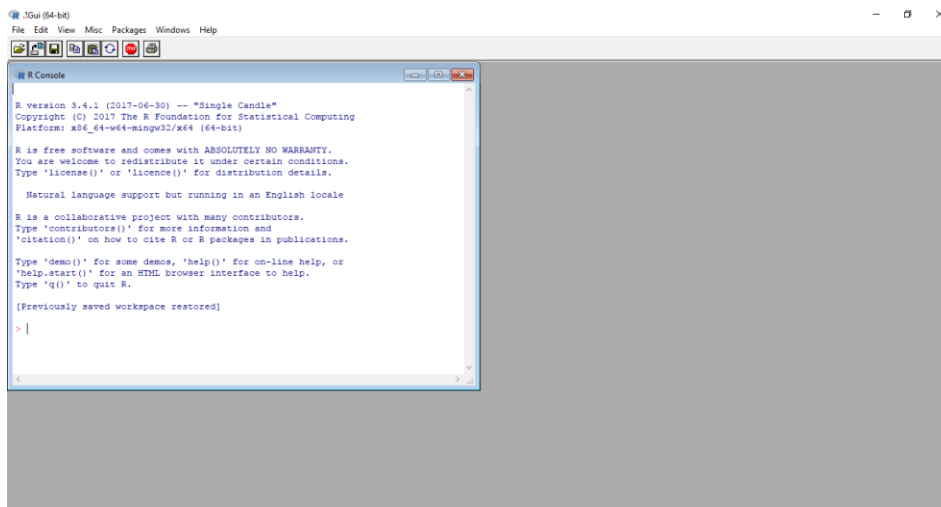
Pilih **No** untuk customized startup.



**Next** saja hingga installasi dimulai.

Sekarang anda dapat membuka R yang sudah diinstall dari icon di desktop.

## Tampilan R



Tapi kita tidak akan menggunakan R karena ada beberapa penyedia IDE untuk R dan menyediakan banyak sekali alat bantu. R yang kita install adalah untuk dasar *Environment* yang nantinya digunakan di Rstudio.



## Instalasi Rstudio

Buka link <https://www.rstudio.com/products/rstudio/> dan download versi Open Source yang gratis. RStudio adalah tools yang sangat powerful untuk membantu kita dalam environment R.


Unduh sesuai sistem operasi anda.

RStudio requires R 2.11.1+. If you don't already have R, download it [here](#).

### Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.0.143 - Windows Vista/7/8/10	81.9 MB	2017-04-19	76bb84296b9202759b3eb1de555a2231
RStudio 1.0.143 - Mac OS X 10.6+ (64-bit)	71.2 MB	2017-04-19	c7f1ed865428b225b202fd1b431954b4
RStudio 1.0.143 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	85.5 MB	2017-04-19	21ca14bffc1a2361ead2d763d0313d
RStudio 1.0.143 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	92.1 MB	2017-04-19	75761eae209158d8415d562b3771fbec
RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	84.7 MB	2017-04-19	2c356d4ee50667ad4042ee196afb3c53
RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	85.7 MB	2017-04-19	7ab5fc240351debe491c6c5a7acb6068

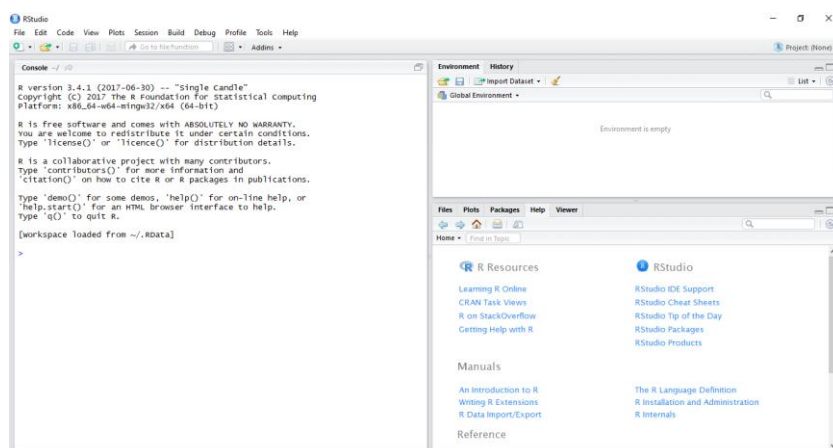
Buka folder tempat anda menyimpan dan jalankan installer RStudio.

 RStudio-1.0.143.exe	7/18/2017 10:30 AM	Application	83,892 KB
---	--------------------	-------------	-----------

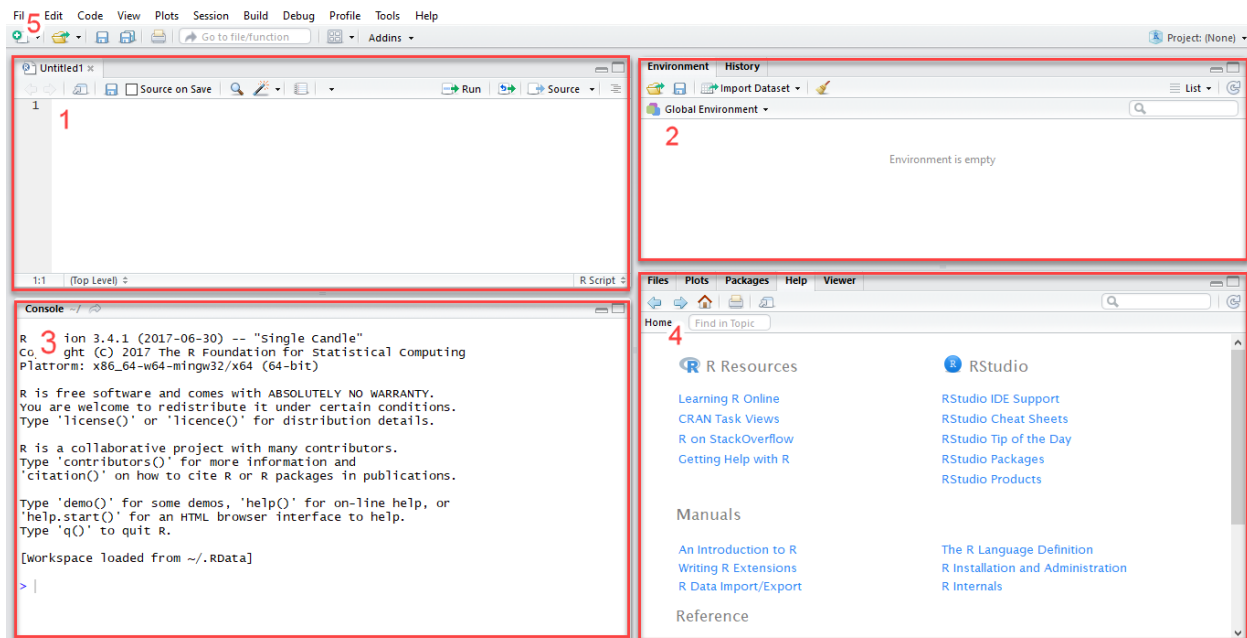
Tidak ada settingan yang perlu diubah, next terus sampai instalasi selesai.

Setelah selesai, buka RStudio yang sudah diinstall.

Berikut adalah tampilan RStudio.



## Environment Introduction



Berikut adalah penjelasan environment di RStudio :

Script Editor, ini adalah tempat untuk menulis script dan menjalankan script.

Environment Editor, ini adalah tempat melihat variable apa saja yang ada di environment kita beserta history dari command yang sudah dijalankan.

Console, ini adalah main window kita untuk melihat console yang dijalankan.

Bagian ini digunakan untuk melihat File, Plot, Packages, dan Help.

Navigation Bar di atas adalah bagian untuk melakukan navigasi di Environment R.

Join our 3 days Complete Training  
**“Data Warehouse with Open Source Tools”**  
 with **Feris Thia** as a Founder of **PHI-Integration** that focus on data  
 management solution for the past 15 years.



**Complete Training**  
**Data Warehouse with Open Source Tools**

**SET THE  
FOUNDATION OF  
INTELLIGENCE**

**TOP 3 PARTICIPANTS WILL RECEIVE  
JOB OFFERING (GUARANTEE)**

Trainer: **Feris Thia**



Feris Thia, Is a founder of PHI-Integration that focus on data management solution for the past 15 years, and recently play in big data and machine learning area. He is very passionate on anything related to data, math and artificial intelligence.

Feris also involved extensively in delivering technical solutions to clients such as Dirjen Pajak, Kementerian BUMN, Telkomsel, The Body Shop, Dima (Guinnes Distributor), Bank BTPN, etc

**Training Date: 30 Oct - 1 Nov 2017**

**Day 1:**

1. Data Warehouse and Data Mart Concept
2. Introducing Kettle, Open Source ETL
3. Software and sample database setup (PHI-Minimart)
4. ETL walkthrough

**Day 2:**

1. Multidimensional Modelling with Fact and Dimension Tables
2. Introduction to 34 Subsystems of ETL
3. Slowly Changing Dimension
4. Optimization with Staging Database

**Day 3:**

1. Advanced Kettle Features
2. Environment Variables
3. Error Handling
4. Automation

**Venue: Hotel Akmani,**  
 Jl. KH. Wahid Hasyim, Menteng, Jakarta

**Target participants:**

- Prerequisite: Basic SQL
- Role: IT or BI or DWH/ Data Management related area
- Professional, academican, researcher, Manager, consultant, fresh graduate

**Fee:**  
 Early Bird IDR 4,500,000 (before 12 Oct)  
 Normal IDR 5,500,000 (per 12 Oct)

installment available, with 0% interest  
 Limited seats for 40 Participants

**REGISTRATION:**  
 Send email with subject DWH\_YourName  
 to [nabil@iykra.com](mailto:nabil@iykra.com)

Payment instruction will be sent through email  
 Contact point NABIL +62857-1170-5552

**iykra. Education & Hiring solution provider for business data technology field**

Pemrograman R

# Tipe Data Umum Dan Operasinya



## Operasi matematika

Di dalam R, operasi matematika dalam kehidupan sehari-hari dapat kita selesaikan dengan mudah. R menyediakan banyak fitur untuk menyelesaikan permasalahan ini.

Angka yang akan kita gunakan dapat disimpan di dalam sebuah variable. Variable ini akan terus berada di environment kita selama tidak kita hapus / ganti sesinya.

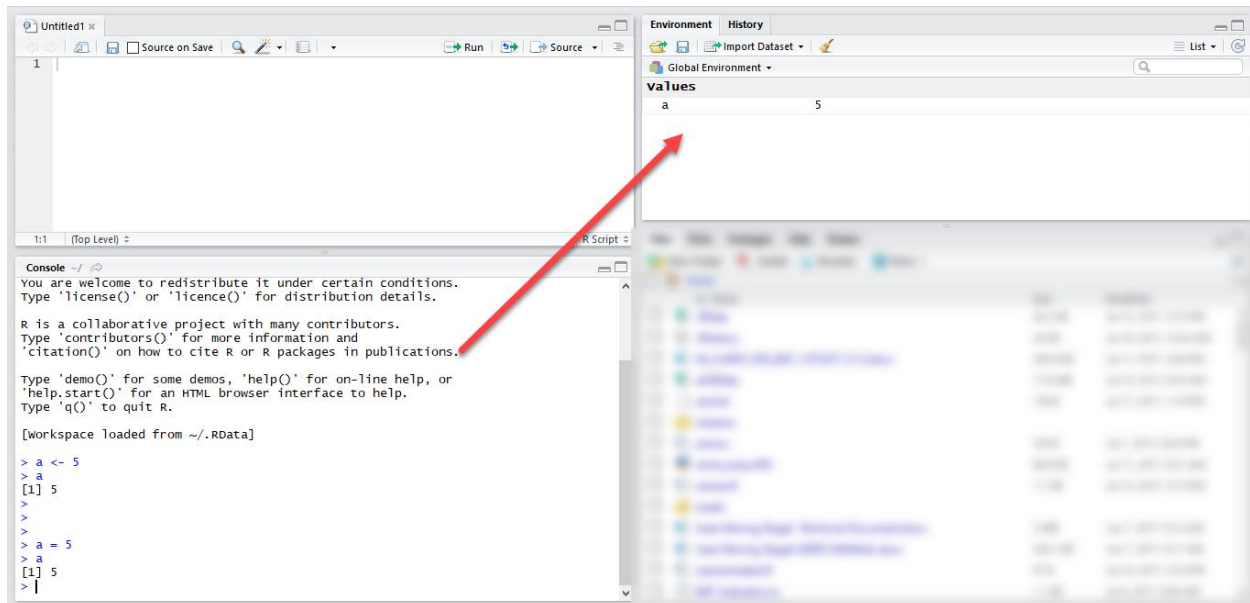
Sekarang kita coba masukkan nilai **5** ke variable **A** melalui Console.

Dalam R, penamaan variable tidak serumit bahasa pemrograman lainnya. Anda dapat menggunakan '.', '-', '\_', dan yang lain untuk penamaan antar kata.

Kita dapat menggunakan "<-" atau "=" untuk menyimpan nilainya.

```
> a <- 5      > a = 5
> a           > a
[1] 5         [1] 5
```

Keduanya menghasilkan hasil yang sama. Untuk mengecek nilai sebuah variable, kita dapat mengetikkan nama variable tersebut langsung di Console, atau melihat ke Environment Variable di atas kanan.



Masukkan nilai **5** ke variable **A** dan nilai **6** ke variable **B**.

```
> a <- 5
> b <- 6
```

Sekarang kita coba operasi matematika **Tambah (+)** , **Kurang (-)**, **Kali (\*)**, dan **Bagi (/)**.

```
> a+b
[1] 11
> a-b
[1] -1
> a*b
[1] 30
> a/b
[1] 0.8333333
> |
```

Hasil operasi matematika juga dapat langsung disimpan ke dalam suatu variable, coba masukkan **A** dikali **B** ke variable **C**.

```
> c<-a*b
> c
[1] 30
> |
```

Sekarang nilai **C** adalah 30.

Di dalam R juga ada beberapa *command built in* yang dapat langsung dipanggil. Coba kita cari akar dari 16. Gunakan command **sqrt(16)**.

```
> sqrt(16)
[1] 4
```

Ini adalah beberapa contoh *basic commands* yang ada di R :

**exp()** untuk eksponensial , **log()** untuk mencari log, **pi** merupakan nilai  $\pi$  (22/7) , **abs()** untuk mencari nilai absolute.

Menginstall Paket

R adalah sebuah IDE yang *Open Source*, sehingga memungkinkan banyak sekali penambahan yang dapat dilakukan.

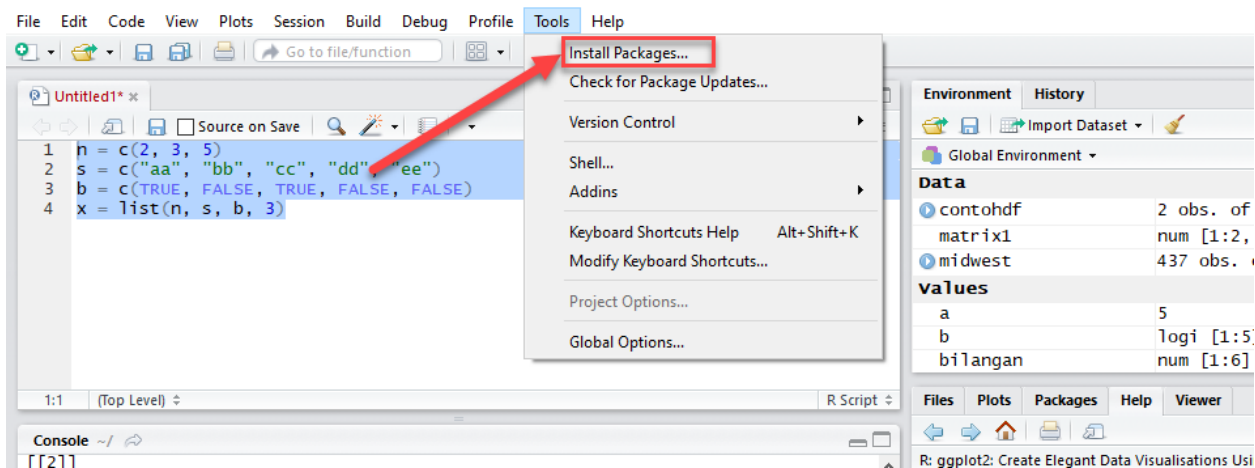
Sebuah library bernama **ggplot2** digunakan untuk membuat visualisasi plot di R semakin mudah dan bagus. Kita coba menginstall library ini dengan command **install.packages("ggplot2")**.

```
> install.packages("ggplot2")
warning in install.packages :
  cannot open URL 'http://www.stats.ox.ac.uk/pub/Rwin/src/contrib/PACKAGES.rds':
HTTP status was '404 Not Found'
Installing package into 'C:/Users/andre/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
warning in install.packages :
  cannot open URL 'http://www.stats.ox.ac.uk/pub/Rwin/bin/windows/contrib/3.4/PACKAGES.rds': HTTP status was '404 Not Found'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/ggplot2_2.2.1.zip'
Content type 'application/zip' length 2784759 bytes (2.7 MB)
downloaded 2.7 MB

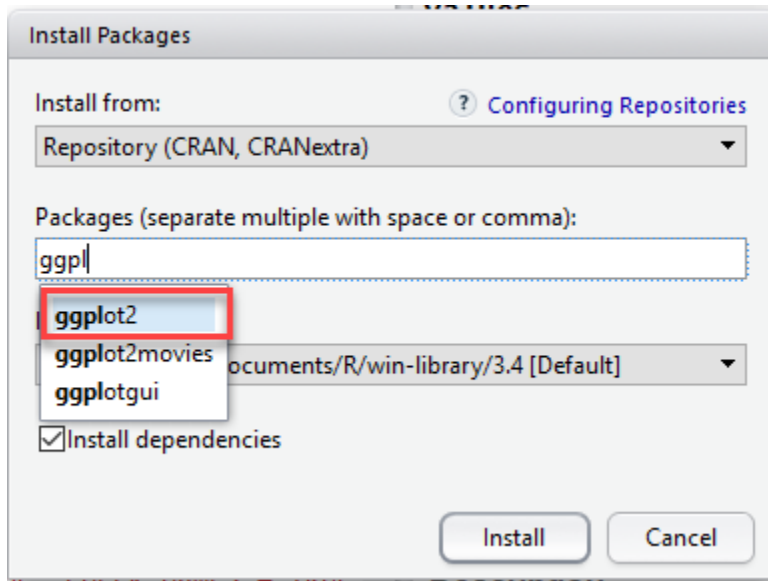
package 'ggplot2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\andre\AppData\Local\Temp\RtmpaAq2eE\downloaded_packages
```

**Cara lain** menginstall paket adalah dengan menggunakan navbar di bagian atas.



Lalu klik **ggplot2**.



Klik **Install**.

Setelah paket berhasil diinstall, kita dapat memanggil paket dengan command **library(ggplot2)**.

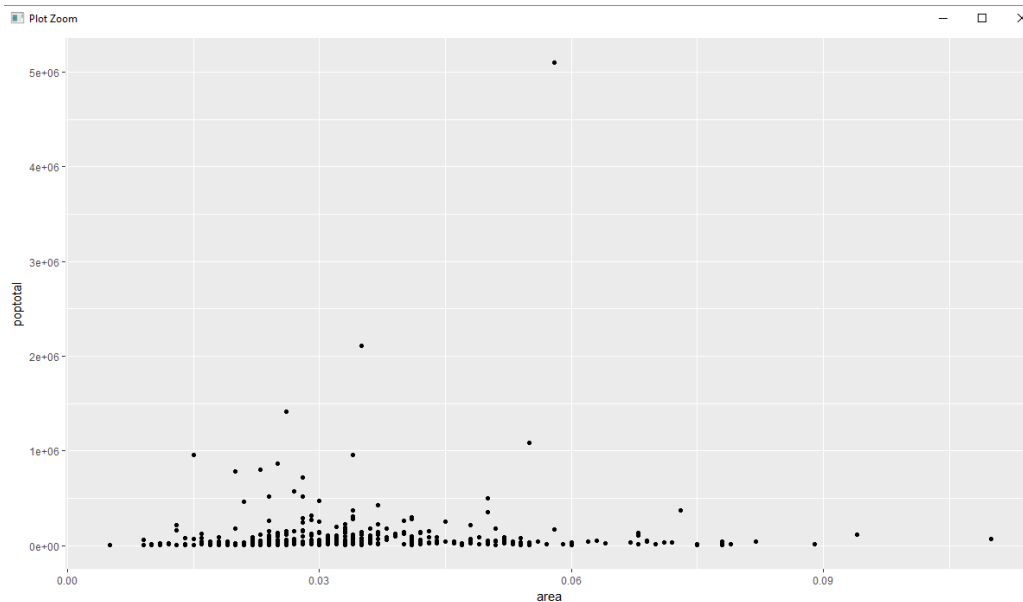
Sekarang kita coba fungsi dari **ggplot2**.

Commandnya adalah `ggplot(midwest, aes(x=area, y=poptotal)) + geom_point()` .

```
> library(ggplot2)
> ggplot(midwest, aes(x=area, y=poptotal)) + geom_point()
> |
```



Sekarang kita bisa melihat hasil ggplot2 di sebelah kanan.



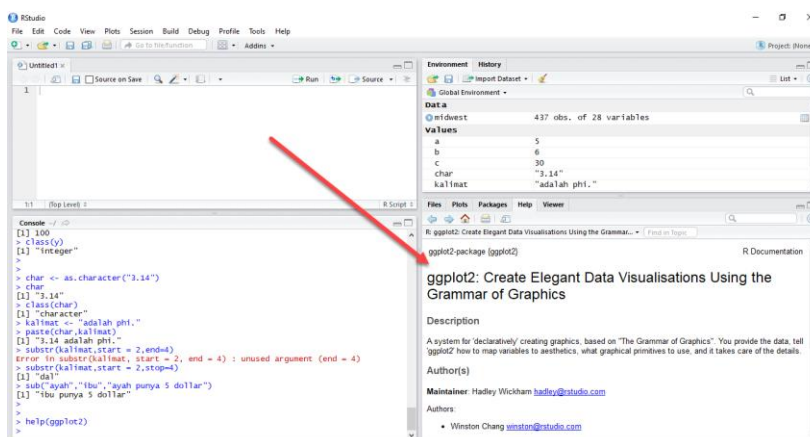
Ini adalah salah satu contoh paket yang dapat diinstall di dalam R.

Jika kita bingung dengan sebuah command dari paket, kita dapat memanggil bantuan dengan command : **help(x)**. X adalah nama paket/fungsi dalam paket.

Misalkan **help(ggplot2)**.

```
> help(ggplot2)
```

Akan muncul di sebelah kanan kolom Help.



## Tipe Data Numerik

Tipe data yang kita masukkan ke dalam suatu variable umumnya bersifat numerik.

```
> x=10.5
> x
[1] 10.5
> class(x)
[1] "numeric"
```

Untuk mengecek tipe dari sebuah data, dapat menggunakan command **class(x)**.

## Tipe Data Integer

Untuk merubah sebuah numerik menjadi integer, kita perlu menggunakan command **as.integer(x)**.

```
> y <- as.integer(100)
> y
[1] 100
> class(y)
[1] "integer"
> |
```

Integer tidak bisa mempunyai angka pecahan dan harus bulat, sehingga ketika angkanya tidak bulat, integer akan otomatis merubah nilainya.

```
> y <- as.integer(100.55)
> y
[1] 100
> class(y)
[1] "integer"
> |
```

## Tipe Data Character

Tipe data karakter di R mirip dengan string di bahasa pemrograman lainnya.

Cara merubah sebuah angka menjadi character adalah dengan command **as.character(x)**.

```
> char <- as.character("3.14")
> char
[1] "3.14"
> class(char)
[1] "character"
> |
```

Berikut adalah beberapa command yang dapat dilakukan kepada character.

```
> char <- as.character("3.14")
> char
[1] "3.14"
> class(char)
[1] "character"
> kalimat <- "adalah phi."
> paste(char,kalimat)
[1] "3.14 adalah phi."
> |
```

```
> substr(kalimat,start = 2,stop=4)
[1] "dal"
> |
```

```
> sub("ayah","ibu","ayah punya 5 dollar")
[1] "ibu punya 5 dollar"
> |
```

## Tipe Data Complex

Tipe data complex dapat menyimpan data berupa bilangan kompleks.

```
> z = 1+2i
> z
[1] 1+2i
> class(z)
[1] "complex"
>
```

## Tipe Data Logical

Tipe data logical merupakan tipe data yang hanya dapat menyimpan **TRUE** atau **FALSE**.

```
> x=1; y=2;
> z <- x>y
> z
[1] FALSE
> class(z)
[1] "logical"
> |
```

Karena **X<Y** maka nilai yang dihasilkan adalah **FALSE**.

## Type Data Vector

Vektor adalah kumpulan tipe data yang sama dalam 1 kelompok.

```
> bilangan <- c(2,3,5,6,7,10)
> bilangan
[1] 2 3 5 6 7 10
> class(bilangan)
[1] "numeric"
> |
```

Bilangan adalah sebuah vector dengan tipe data numerik.

```
> huruf <- c('a','b','c')
> huruf
[1] "a" "b" "c"
> class(huruf)
[1] "character"
> |
```

Huruf adalah sebuah vector dengan tipe data character.

## Type Data Matrix

Matriks adalah kumpulan vector dalam bentuk 2 dimensi.

```
> matrix1 = matrix(
+ c(2, 4, 3, 1, 5, 7),
+ nrow=2,
+ ncol=3,
+ byrow = TRUE)
> matrix1
      [,1] [,2] [,3]
[1,]    2    4    3
[2,]    1    5    7
> |
```

**Nrow** menentukan banyaknya baris.

**Ncol** menentukan banyaknya kolom.

**Byrow** menentukan urutannya berdasar baris.

## Tipe Data List

List adalah sebuah objek yang berisi dari 2 atau lebih vektor.

```
> n = c(2, 3, 5)
> s = c("aa", "bb", "cc", "dd", "ee")
> b = c(TRUE, FALSE, TRUE, FALSE, FALSE)
> x = list(n, s, b, 3)
> x
[[1]]
[1] 2 3 5

[[2]]
[1] "aa" "bb" "cc" "dd" "ee"

[[3]]
[1] TRUE FALSE TRUE FALSE FALSE

[[4]]
[1] 3

> |
```

List bisa terdiri dari vektor yang berbeda tipe data.

## Tipe Data Dataframe

Dataframe adalah tipe data yang paling umum di dalam pengolahan data menggunakan R.

Dataframe memudahkan user untuk membaca data.

Contoh merubah data menjadi dataframe adalah dengan command **as.data.frame(x)**.

```
> contohdf <- as.data.frame(matrix1)
> contohdf
  v1 v2 v3
1  2  4  3
2  1  5  7
> |
```

Untuk mengakses elemen dataframe, kita dapat memanggil nama kolomnya dengan menyertakan **\$** lalu nama kolomnya.

```
> contohdf$v1
[1] 2 1
> |
```

Pemrograman R

# Pengenalan Data dan Plotting



## Pengenalan Data Kualitatif

Mulai dari bagian ini, kita akan menggunakan dataframe yang sudah disediakan di dalam modul pelatihan ini untuk memudahkan pembelajaran.

Modul dapat diunduh di : <http://tinyurl.com/ebook-r-phi>

File yang akan kita gunakan adalah **phi\_dummy.csv**, Bukalah file csv yang sudah anda download untuk melihat struktur datanya. Ada 4 kolom yang mempunyai judul, lalu di bawahnya langsung data.

	A	B	C	D
1	jeniskelar	usia	goldarah	asaldaerah
2	P	36	O	Bogor
3	L	24	O	Jakarta
4	L	31	O	Jakarta
5	L	21	O	Tangerang
6	P	31	B	Tangerang
7	P	21	A	Jakarta
8	L	19	A	Tangerang
9	P	21	AB	Tangerang
10	P	21	AB	Bogor
11	L	21	A	Jakarta
12	P	31	B	Jakarta
13	L	27	A	Bogor
14	L	25	A	Tangerang
15	P	20	B	Tangerang
16	L	20	O	Jakarta
17	P	22	A	Jakarta
18	L	20	A	Bekasi
19	L	24	O	Tangerang
20	L	29	A	Bogor

Letakkan file csv ini satu directory dengan working directory R Anda.

Cara mengecek working directory Anda adalah dengan command **getwd()**.

```
> getwd()
[1] "C:/Users/andre/Documents"
```

Setelah Anda meletakkan file **phi\_dummy.csv** di *working directory* Anda, Anda dapat membaca file csv ini dengan command **read.csv()** di R. Hasil dari read.csv harus dimasukkan ke sebuah variable untuk dapat dipanggil di environment R kita.

```
| > datalatihan <- read.csv(file="phi_dummy.csv",header=TRUE)
```

**Header=TRUE** digunakan untuk membaca judul sehingga pada saat kita panggil **datalatihan** di console, judulnya akan keluar.

```
> datalatihan
  jeniskelamin usia goldarah asaldaerah
1           P   36         O      Bogor
2           L   24         O      Jakarta
3           L   31         O      Jakarta
4           L   21         O      Tangerang
5           P   31         B      Tangerang
6           P   21         A      Jakarta
7           L   19         A      Tangerang
8           P   21        AB      Tangerang
9           P   21        AB      Bogor
10          L   21         A      Jakarta
11          P   31         B      Jakarta
12          L   27         A      Bogor
13          L   25         A      Tangerang
14          P   20         B      Tangerang
15          L   20         O      Jakarta
16          P   22         A      Jakarta
17          L   20         A      Bekasi
18          L   24         O      Tangerang
19          L   29         A      Bogor
20          L   19         A      Tangerang
```

Setelah berhasil membaca data csv ini di R, Anda melihat tipe data dari data yang sudah di baca pada R pada step sebelumnya dengan menggunakan fungsi **class()**.

```
> class(datalatihan)
[1] "data.frame"
```

Karena tipe datanya adalah **data.frame**, Anda dapat memanggil masing-masing kolom dengan \$.

Contoh **datalatihan\$goldarah**.

```
> datalatihan$goldarah
 [1] O  O  O  O  B  A  A  AB AB A  B  A  A  B  O  A  A  O  A  A  A  O  A  B
[26] B  O  A  AB A  B  B  A  O  AB O  O  O  A  O  AB O  O  O  A  A  O  A  B
[51] AB B  A  A  O  A  AB O  A  O
Levels: A AB B O
>
```



Sekarang kita akan coba fungsi `table` untuk mengelompokkan data kualitatif.

```
> goldarahtable <- table(datalatihan$goldarah)
> goldarahtable
```

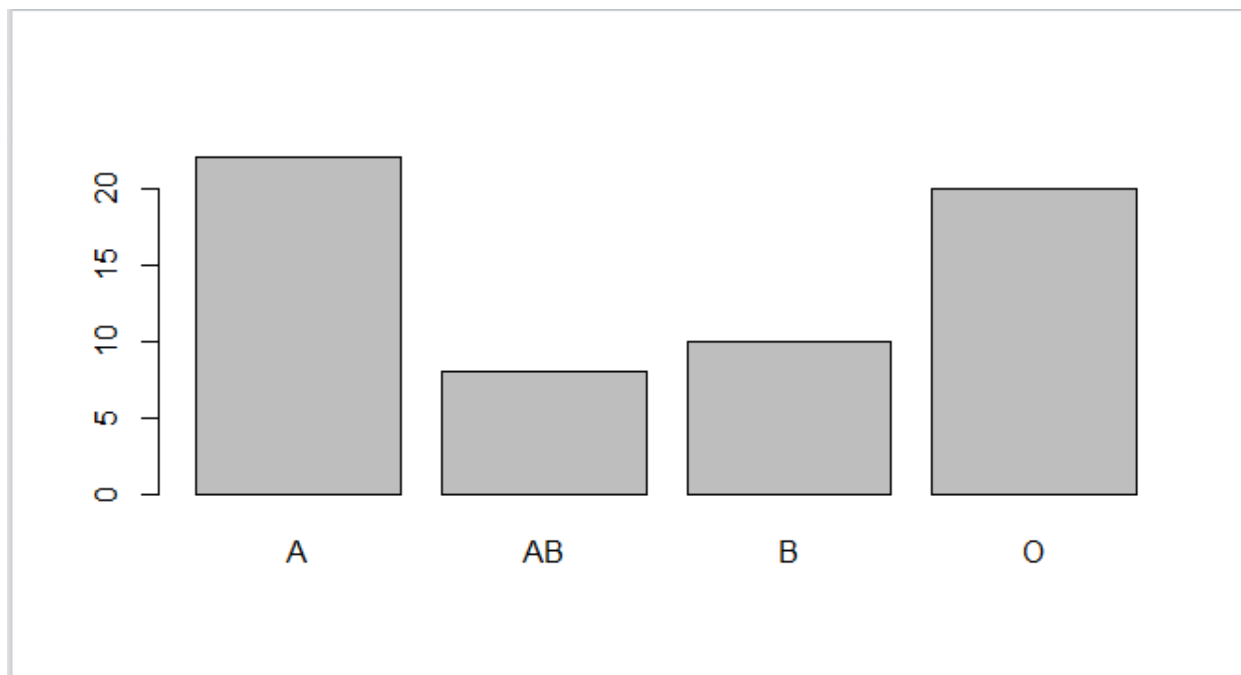
A	AB	B	O
22	8	10	20

Dari yang sebelumnya **goldarah** berisi persebaran golongan darah, setelah diberi fungsi `table` hasilnya menjadi dikelompokkan.

Selanjutnya kita akan membuat barplot dari hasil `table` tersebut dengan fungsi **`barplot()`**.

```
> barplot(goldarahtable)
```

Hasilnya dapat dilihat di kanan (plot).



Hasil dari barplot ini masih sangat sederhana. Kita dapat menambahkan warna dan label ke plot ini.

```
> warna <- c("red", "blue", "violet", "green")
```

Pertama kita harus mendeklarasi warna plot yang ingin kita gunakan. Warna dapat dilihat di <http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf>.

```
> barplotlatihan <- barplot(goldarhtable, col=warna, xlab="Golongan Darah",  
ylab="Jumlah", ylim = c(0,26))
```

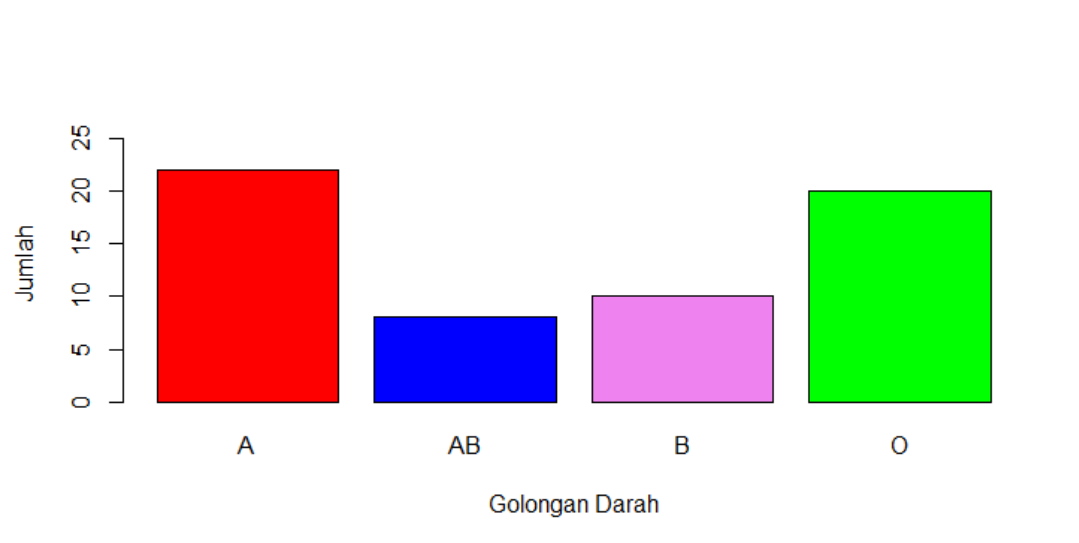
Karena kita nanti ingin menggunakan fungsi *text*, maka barplot ini kita masukkan ke sebuah variable.

Berikut penjelasan dari penggunaan fungsi **BARPLOT()** diatas:

Fungsi **col** dalam **barplot()** berfungsi untuk menentukan warna.

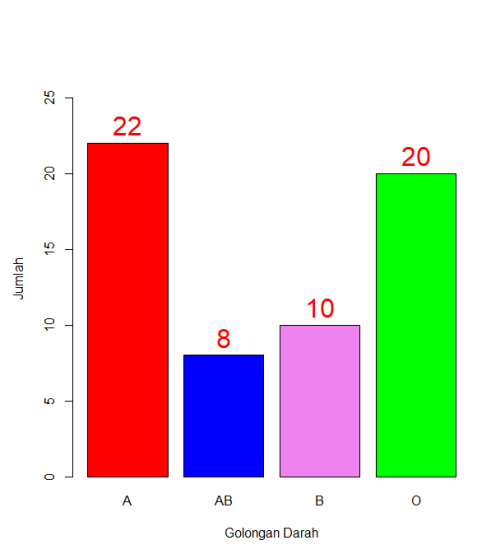
Fungsi **xlab** dan **ylab** berfungsi untuk memberikan label di plot.

Fungsi **ylim** berfungsi untuk memberikan batas sumbu Y. C(0,26) berarti dari 0-26.



Jika kita ingin menambahkan *text*, kita dapat menggunakan fungsi **text()**.

```
> text(x=barplotlatihan,y=goldarahtable,label=goldarahtable,pos=3,cex=2,col="red")
```

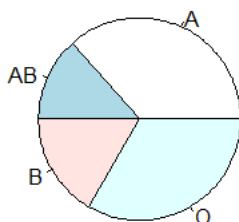


**X** diisi dengan variable barplot kita, **Y dan label** diisi dengan nilai dari table, **pos** diisi angka 1-4 (Bawah,Kiri,Atas,Kanan), **cex** adalah besarnya text, **col** adalah warna text.

Dalam kasus ini **pos** kita isi dengan angka 3 supaya berada di atas dan ukuran text adalah 2.

Untuk membuat piechart, kita dapat menggunakan fungsi **pie()**.

```
> pie(goldarahtable)
```



## Pengenalan Data Kuantitatif

Di dalam data kuantitatif, tidak ada pengelompokkan dalam nilai data. Berbeda dengan kualitatif.

Data yang digunakan dalam bagian ini dapat diunduh di : <http://tinyurl.com/ebook-r-phi>

Nama file : **phi\_gempa.csv**

File terdiri dari 2 kolom : ukuran gempa (dalam skala richter) dan lama gempa (dalam second).

ukurangempa	lamagempa
6.2	10
3.1	5
4.5	6.6
1.6	2.8
6.4	10.2
4.2	7.2
5.8	9.2
3.3	5.3
4	5.1
8.2	12.8
7.7	10.9
3.2	5.3
5.3	8.3
3.2	4.8
3.3	4.9
4.3	5.8
6.3	10.5
7	11
2.8	4.3
2.7	3.7

Anda harus meletakkan file csv tersebut di working directory, sama seperti cara mengambil data dari bagian sebelumnya.

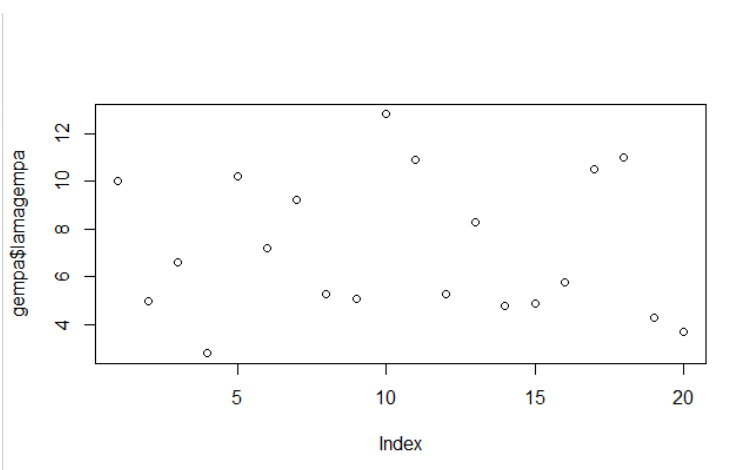
Gunakan **read.csv()** untuk mengambil data ke R.

Masukkan data tersebut ke sebuah variable bernama **gempa**.

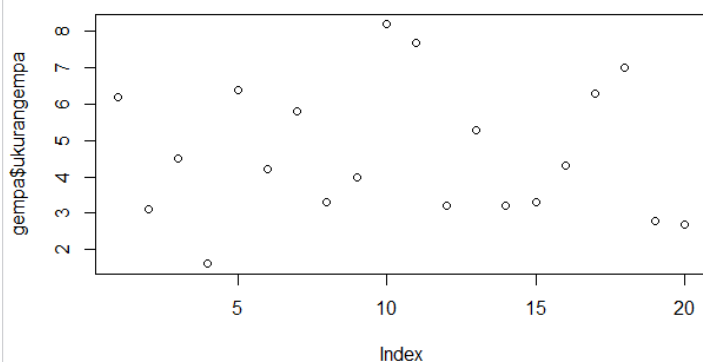
```
> gempa <- read.csv("phi_gempa.csv",header=TRUE)
> gempa
  ukur angempa lamagempa
1      6.2      10.0
2      3.1       4.5
3      4.5       6.1
4      1.6       2.6
5      6.4      10.2
6      4.2       4.9
7      5.8       8.8
8      3.3       4.8
9      4.0       5.1
10     8.2      12.0
11     7.7      10.9
12     3.2       5.3
13     5.3       8.0
14     3.2       4.8
15     3.3       4.9
16     4.3       5.8
17     6.3      10.5
```

Sekarang yang akan kita pelajari adalah bagaimana cara membuat plot dari data kuantitatif. Fungsi **plot()** akan digunakan.

```
> plot(gempa$lamagempa)
```

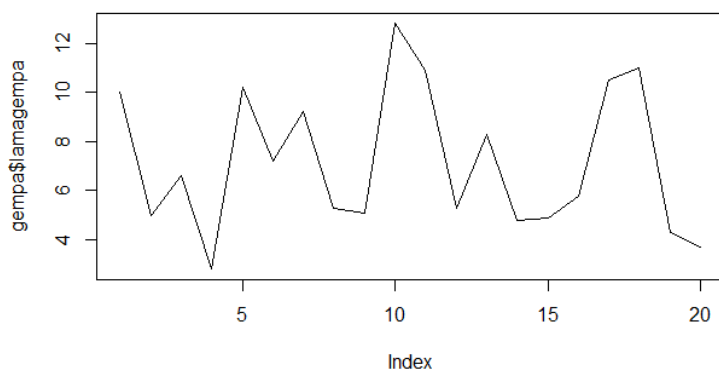


```
> plot(gempa$ukur angempa)
```

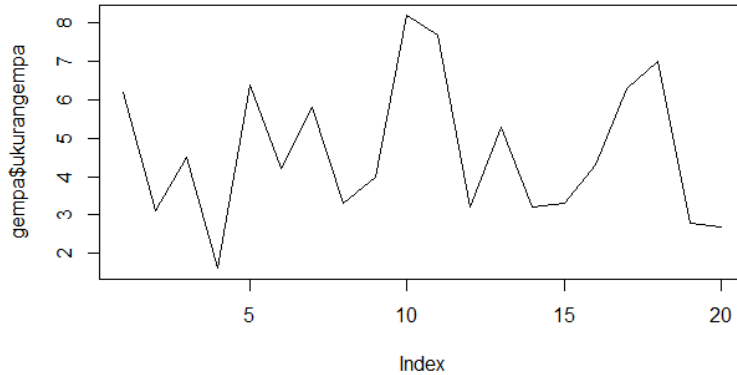


Secara kasat mata sangat susah melihat apa yang digambarkan. Tetapi jika kita menambahkan opsi **type = "l"** di dalam **plot()**. Anda dapat melihat gerak plotnya.

```
> plot(gempa$lamagempa, type = "l")
```



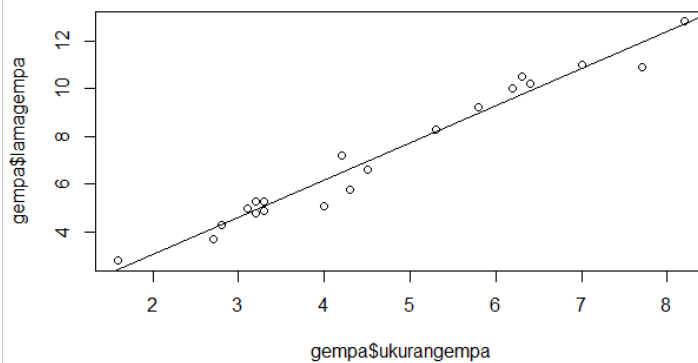
```
> plot(gempa$ukurangempa, type = "l")
```



Secara sekilas grafiknya hampir mirip, tetapi berbeda nilai di sumbu Y. Sumbu X dibaca sebagai index.

Sekarang kita ingin melihat scatter plot dari kedua nilai di atas dan melihat korelasinya.

```
> plot(gempa$ukurangempa, gempa$lamagempa)
> abline(lm(gempa$lamagempa ~ gempa$ukurangempa))
```



Dapat dilihat bahwa plot ini menunjukkan korelasi ukuran gempa dan lama gempa sangat kuat. Hampir tidak ada nilai yang menjauhi garis normal.

```
> cor(gempa$ukurangempa, gempa$lamagempa)
[1] 0.9842339
```

Untuk memeriksa nilai korelasi kedua variable, kita dapat menggunakan fungsi **cor()**.

Nilai yang dihasilkan sangat tinggi sehingga dapat disimpulkan bahwa hubungan dua variable sangat kuat.

Semakin tinggi ukuran gempa, semakin lama gempa itu bertahan.



Pemrograman R

# Operasi Statistik Dasar Menggunakan R dan Pemrograman R



## Operasi Statistik

Operasi statistik dasar seperti *mean* dan *standar deviasi* kini sudah dapat dihitung di calculator. Tetapi jika untuk data yang skala besar, R dapat memudahkan kita untuk melakukan operasi ini.

Untuk melakukan percobaan dengan operasi statistik kita menggunakan data **phi\_gempa.csv** yang telah digunakan pada Bab sebelumnya.

Setelah kita memasukkan file csv tersebut ke satu variable bernama *gempa*, kita akan menghitung *Mean, Median, Quartile, Percentile, Range, Variance, Standar Deviation, Covariance, Correlation, Skewness* dengan fungsi yang sudah ada pada *package* pada R.

Fungsi **mean()** untuk *gempa\$lamagempa*.

```
> gempa <- read.csv("phi_gempa.csv",header=TRUE)
> mean(gempa$lamagempa)
[1] 7.185
> |
```

Fungsi **median()**

```
> median(gempa$lamagempa)
[1] 6.2
> |
```

Fungsi **quantile()**

```
> quantile(gempa$lamagempa)
 0%    25%    50%    75%   100%
2.800  4.975  6.200 10.050 12.800
>
```

Untuk **percentile**, kita menggunakan fungsi yang sama seperti **quantile**, tapi dengan menambahkan opsi vector berapa saja persentase yang ingin dikeluarkan.

```
> quantile(gempa$lamagempa, c(0.31, 0.48, 0.89))
 31%    48%    89%
5.089  5.896 10.864
```

Dalam kasus ini vector ada 3 di 0.31, 0.48, dan 0.89.

Untuk mencari range dalam statistik dengan mencari nilai terbesar dikurangi dengan nilai terkecil. Sehingga kita dapat menggunakan fungsi **max()** dan **min()**.

```
> max(gempa$lamagempa) - min(gempa$lamagempa)
[1] 10
```

Untuk mencari Variance dalam R, kita menggunakan fungsi **var()**.

```
> var(gempa$lamagempa)
[1] 8.541342
```

Untuk mencari standar deviasi dalam R, kita menggunakan fungsi **sd()**.

```
> sd(gempa$lamagempa)
[1] 2.922557
> |
```

Untuk mencari persebaran dari sebuah data, kita bisa menggunakan **density()**.

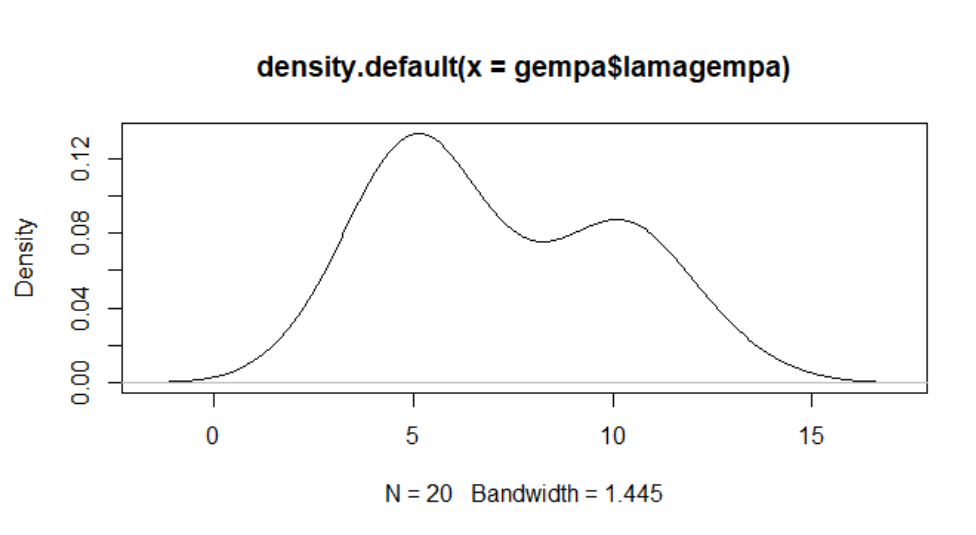
```
> density(gempa$lamagempa)

Call:
density.default(x = gempa$lamagempa)

Data: gempa$lamagempa (20 obs.);      Bandwidth 'bw' = 1.445

      x              y
Min.  :-1.534   Min.  :0.0001586
1st Qu.: 3.133   1st Qu.:0.0075082
Median : 7.800   Median :0.0548665
Mean   : 7.800   Mean   :0.0535059
3rd Qu.:12.467   3rd Qu.:0.0856808
Max.   :17.134   Max.   :0.1331891
> plot(density(gempa$lamagempa))
```

Dan ketika di plot hasilnya seperti ini.

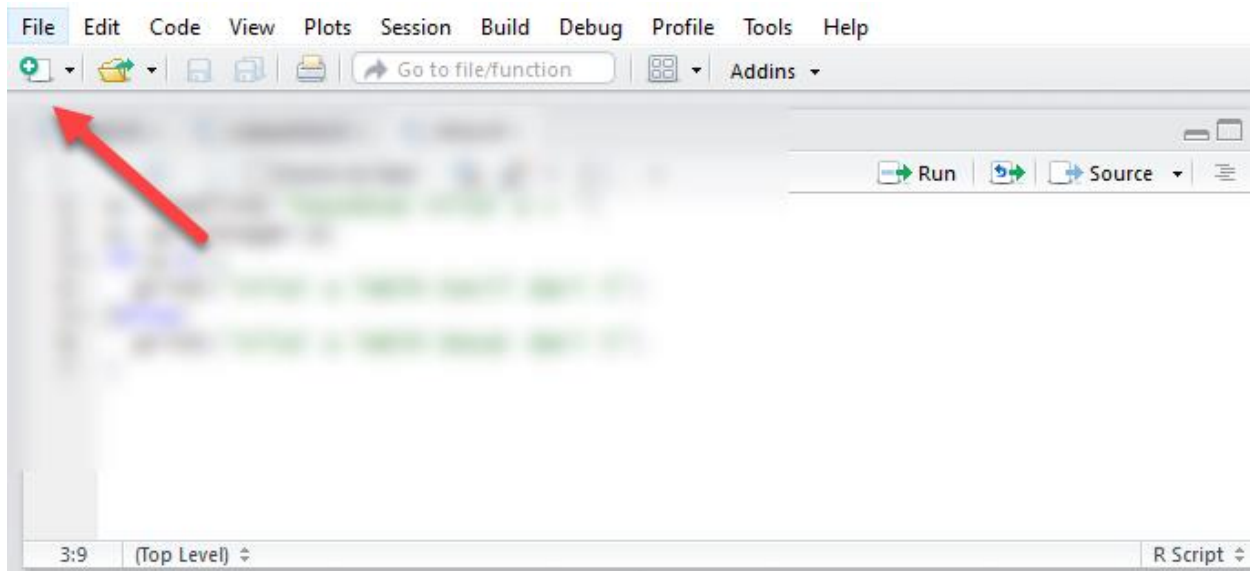


## Dasar Pemrograman R

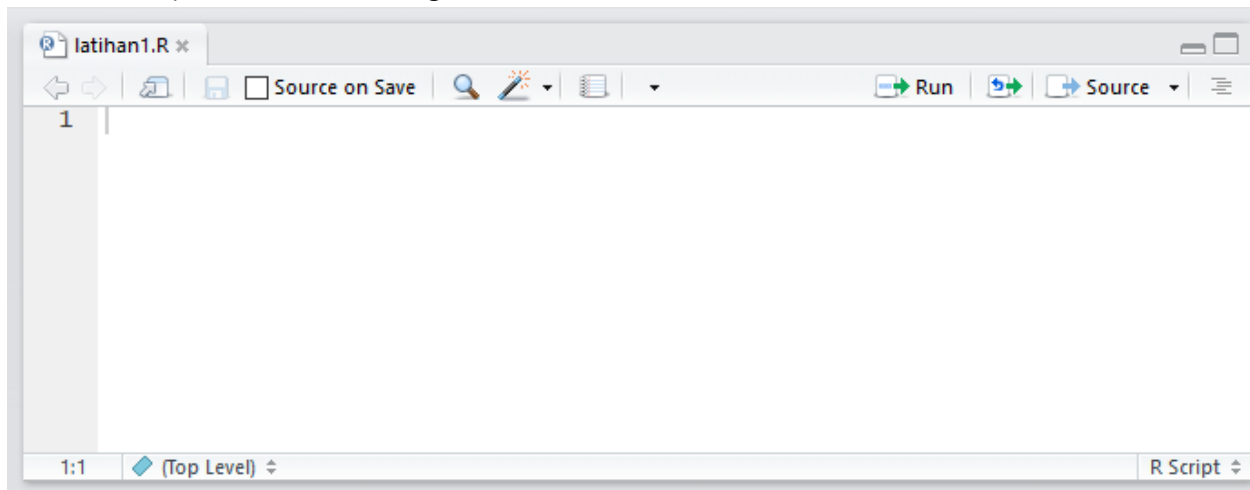
Pemrograman dalam R meliputi Input Output, Kondisi If, Loop, dan Fungsi. Tapi dalam kesempatan kali ini kita hanya akan membahas sampai Loop saja.

Untuk operasi input kita menggunakan command **readline()**.

Untuk operasi output kita menggunakan command **print()**.

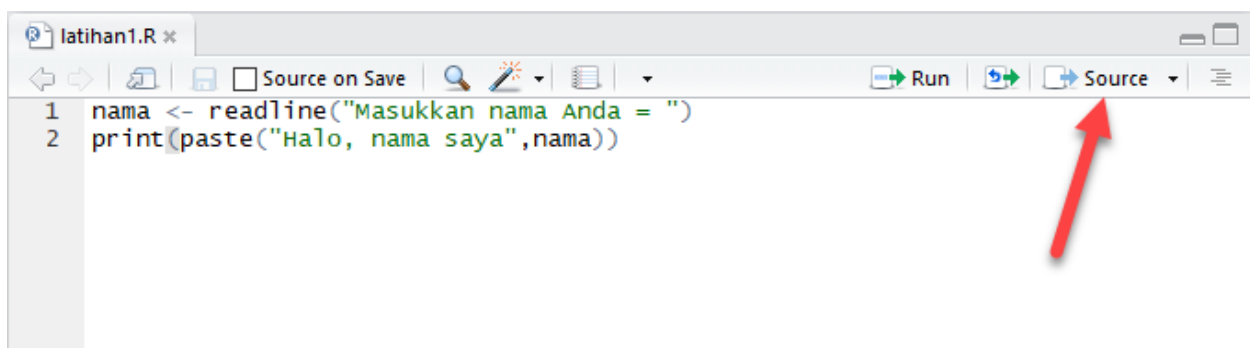


Buatlah Script baru, save dengan nama **latihan1**.



Sekarang Anda coba untuk membuat input dan output dalam R.

Lakukanlah seperti pada gambar di bawah.



Jika sudah, klik tombol **Source. Script** yang anda buat sekarang berjalan di **console**.

```
> source('~/.latihan1.R')
Masukkan nama Anda = |
```

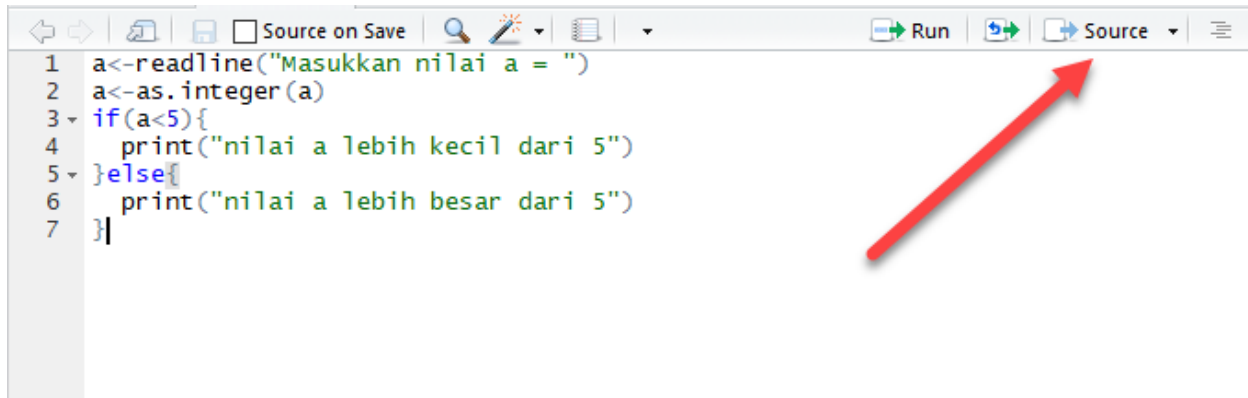
Masukkan nama Anda, lalu klik **Enter**.

```
> source('~/.latihan1.R')
Masukkan nama Anda = Komi
[1] "Halo, nama saya Komi"
> |
```

Sekarang program akan menuliskan sesuai yang sudah kita buat di script.

Untuk If, Then, Else, syntax pemrograman di R mirip dengan bahasa pemrograman lain. Tetapi dalam pemrograman R tidak memerlukan tanda semicolon ";" untuk mengakhiri statement. Buatlah Script baru dengan nama **ifelse**.

Coba salin seperti tampilan di bawah di script tersebut.



```

1 a<-readline("Masukkan nilai a = ")
2 a<-as.integer(a)
3 if(a<5){
4   print("nilai a lebih kecil dari 5")
5 }else{
6   print("nilai a lebih besar dari 5")
7 }
  
```

Jika sudah, klik tombol **Source** untuk menjalankan program.

Program ini akan meminta input dari user dan dimasukkan ke variable **A**. Semua tipe data yang dibaca oleh readline adalah character, sehingga harus diganti menjadi integer dengan function **as.integer()**.

Syntax if then else di R adalah `if("condition"){ } else{ }`.

Program diatas akan mengecek apakah angka lebih besar dari 5 atau lebih kecil dari 5. Jika lebih kecil dari 5 program akan mengeluarkan output **"nilai a lebih kecil dari 5"**, jika lebih besar outputnya adalah **"nilai a lebih besar dari 5"**.

Setelah program dijalankan tampilannya seperti ini.

```

> source('~./ifelse.R')
Masukkan nilai a = |
  
```

Jika dimasukkan 4, hasilnya seperti ini.

```

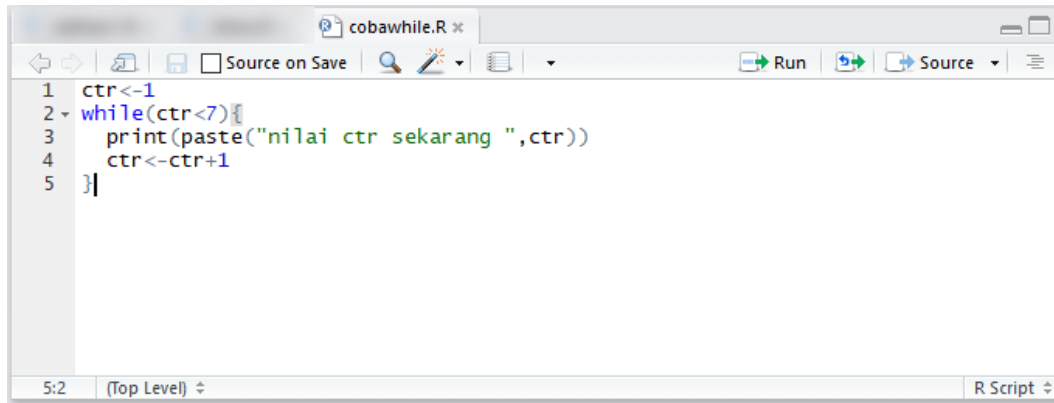
> source('~./ifelse.R')
Masukkan nilai a = 4
[1] "nilai a lebih kecil dari 5"
> |
  
```

Jika dimasukkan 6, hasilnya seperti ini.

```

> source('~./ifelse.R')
Masukkan nilai a = 6
[1] "nilai a lebih besar dari 5"
> |
  
```

Untuk perulangan, di dalam R ada **for()** dan **while()**. Berikut contoh script untuk *for* dan *while*.



```

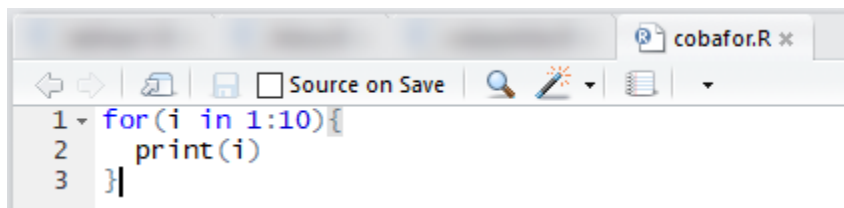
1 ctr<-1
2 while(ctr<7){
3   print(paste("nilai ctr sekarang ",ctr))
4   ctr<-ctr+1
5 }
  
```

Hasilnya ketika tombol **Source** diklik adalah :

```

> source('~/.cobawhile.R')
[1] "nilai ctr sekarang 1"
[1] "nilai ctr sekarang 2"
[1] "nilai ctr sekarang 3"
[1] "nilai ctr sekarang 4"
[1] "nilai ctr sekarang 5"
[1] "nilai ctr sekarang 6"
> |
  
```

Dalam penggunaan *while* di R, suatu perulangan akan terus dilakukan sampai memenuhi suatu kondisi. Dalam kasus ini perulangan akan terus dilakukan selama nilai **ctr** dibawah 7. Yang dilakukan dalam perulangan adalah mencetak "**nilai ctr sekarang 'ctr'**".



```

1 for(i in 1:10){
2   print(i)
3 }
  
```

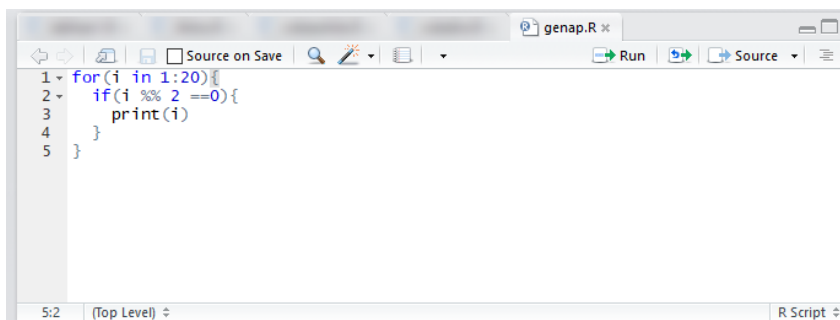
Hasilnya ketika tombol **Source** dipencet adalah :

```
> source('~/.cobafor.R')
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

Penggunaan for dalam R sedikit unik, dimana kita harus memberikan nilai awal dan akhir langsung dalam kondisi perulangannya. **i in 1:10** artinya selama nilai **i** dimulai dari 1 dan sampai nilai **i** bernilai 10. Yang dilakukan dalam perulangan ini adalah mencetak nilai **i**.

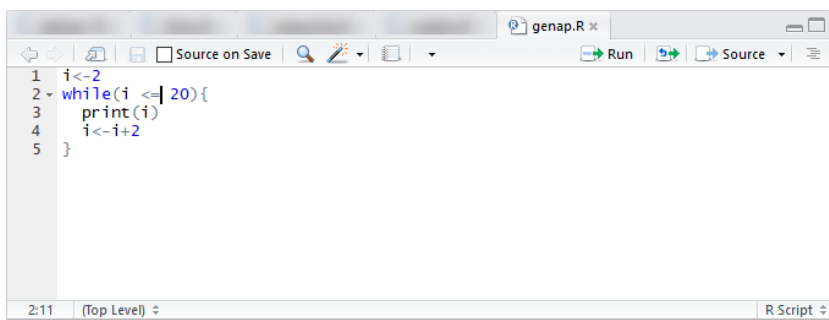
Sekarang cobalah membuat perulangan untuk mengeluarkan nilai genap dari 1 sampai 20.

Ini adalah cara pertama menggunakan for.




```
1 for(i in 1:20){
2   if(i %% 2 == 0){
3     print(i)
4   }
5 }
```

Ini adalah cara kedua menggunakan while.



```
1 i<-2
2 while(i <= 20){
3   print(i)
4   i<-i+2
5 }
```






SKYSTAR  
VENTURES

EARLY BIRD: 50K | NORMAL: 100K

EXECUTIVE LOUNGE, UMN


## The Quick Path to Start Data Science 101 Career



SATURDAY

**28 / 10**

9 AM - 1 PM



**Feris Thia**  
Founder of PHI-Integration

### Learning Outcomes:

- A clear roadmap of data technologies that will help you understand and decide your career path
- A basic applicable knowledge in theory and in direct application for real businesses case.
- Understand basic steps of Data Science from data integration to prescriptive analytics.
- Introduction to popular tools in Data Mining, Machine Learning, and AI.
- Opportunity to career referral or internship to work on data project in big company (enterprise) in Indonesia.

CONTACT:  
**BAGUS 0812-2590-3056**

[skystarventures.com/category/events](https://skystarventures.com/category/events)

Find More About PHI-Integration at our Youtube Account:  
<https://www.youtube.com/user/phiintegration>

Pemrograman R

# Subset Data dan Membuat Data



## Subset Data

Subset data seringkali digunakan untuk melakukan *sampling* terhadap sebuah kumpulan data. Subset ini sangat berguna jika kita mempunyai data yang sangat banyak.

Sekarang kita menggunakan **datatest** yang diisi data dari 1 sampai 10

```
> datatest <- c(1:10)
>
> datatest
[1] 1 2 3 4 5 6 7 8 9 10
```

Sekarang misalkan kita ingin memilih angka 1-5 untuk dimasukkan ke variable baru. **Select** berisi angka 1-5 dan nantinya akan menjadi acuan untuk subset data. Untuk mengambil data yang hanya sesuai dengan **select**, command nya adalah seperti berikut.

```
> select <- c(1:5)
>
> newdata <- datatest[select]
> newdata
[1] 1 2 3 4 5
> |
```

Setelah kita melakukan command **newdata <- datatest[select]** di console. Yang terjadi adalah newdata akan diisi data dari datatest yang hanya sesuai dengan select. Dimana data yang ada di select adalah 1 sampai 5 saja.

Cara lain adalah dengan langsung menggunakan vector ke pemilihan data.

```
> newdata <- datatest[c(1:5)]
> newdata
[1] 1 2 3 4 5
```

Kedua cara di atas adalah tipe *include*. Jika kita ingin *exclude*, gunakan - (minus) di depan vector.

```
> newdata <- datatest[-c(6:10)]
> newdata
[1] 1 2 3 4 5
> |
```

Dengan cara ini, kita meng *exclude* 6-10 dari **datatest**.

Jika kita ingin memilih data tertentu, kita dapat menggunakan *vector*.

```
> newdata <- datatest[c(1,4,7)]
>
> newdata
[1] 1 4 7
>
```

Untuk exclude data tertentu.

```
> newdata <- datatest[-c(1,4,7)]
> newdata
[1] 2 3 5 6 8 9 10
>
```

Sekarang kita coba menggunakan fungsi **subset()**. Import terlebih dahulu data **phi\_gempa.csv** untuk kita gunakan.

```
> gempa <- read.csv("phi_gempa.csv",header=TRUE)
```

Lakukan seperti ini, di dalam fungsi **subset**, kita dapat memberikan kondisi data apa saja yang dapat dipilih. Dari data gempa yang jumlahnya 20 disubset menjadi yang lama gempanya diatas 6 detik.

```
> subsetgempa <- subset(gempa,gempa$lamagempa > 6)
> subsetgempa
  ukur angempa lamagempa
1         6.2      10.0
3         4.5       6.6
5         6.4      10.2
6         4.2       7.2
7         5.8       9.2
10        8.2      12.8
11        7.7      10.9
13        5.3       8.3
17        6.3      10.5
18        7.0      11.0
>
```

Untuk kondisi yang lebih dari satu kita dapat menggunakan “ | “ untuk **OR** dan “ & “ untuk **AND**.

Sekarang untuk mengambil sample acak, kita dapat menggunakan fungsi **sample()**.

```
> subsetgempa <- gempa[sample(1:nrow(gempa),10),]
> subsetgempa
   ukur angempa  lamagempa
5          6.4      10.2
13         5.3       8.3
3          4.5       6.6
20         2.7       3.7
7          5.8       9.2
1          6.2      10.0
6          4.2       7.2
12         3.2       5.3
17         6.3      10.5
14         3.2       4.8
> |
```

**Nrow()** adalah jumlah baris yang dimasukkan.

Perintah di atas mengambil sample dari seluruh data gempa dan yang diambil jumlahnya adalah

Membuat Data

Membuat data adalah salah satu keunggulan R dimana kita tidak perlu repot menulis satu per satu.

Cara pertama adalah dengan menggunakan fungsi **sample()** seperti pada bagian sebelumnya. Kita bisa memberikan kategori apa saja yang harus dibuat dan seberapa besar kemungkinannya. Misalnya kita ingin membuat data untuk golongan darah, dengan kemungkinan masing-masing A=38%, B=12%, o=38%, AB=12%.

```
> goldarah <- sample(c("A","B","O","AB"), size =60,replace=TRUE,prob=c(0.38,0.12,0.38,0.12))
> goldarah
 [1] "O" "O" "A" "AB" "O" "AB" "A" "O" "O" "AB" "O" "AB" "O" "O" "AB"
[16] "A" "O" "O" "O" "O" "AB" "O" "A" "A" "A" "A" "O" "O" "O" "B"
[31] "A" "O" "A" "O" "A" "O" "AB" "A" "AB" "A" "AB" "A" "A" "O" "B"
[46] "AB" "O" "AB" "B" "O" "O" "O" "A" "O" "A" "O" "A" "A" "A" "A"
> |
```

Yang pertama adalah kita mendeklarasi apa saja kategori yang ingin dimunculkan dengan vector **c("A","B","O","AB")**. Selanjutnya berapa jumlah data kita **size=60**. Lalu **replace=TRUE** untuk overwrite data jika sudah ada. Yang terakhir adalah probabilitas masing-masing kategori.

Setelah kita buat table, akan terlihat persebarannya.

```
> goldarah.table<-table(goldarah)
> goldarah.table
goldarah
  A AB  B  O
20 11  3 26
> |
```

Hasilnya akan selalu berubah-ubah karena untuk mengenerate data, kita menggunakan RNG dan ada seed yang selalu berbeda. Kita akan coba dengan **set.seed(123)**. Fungsi **set.seed()** adalah untuk menghasilkan sesuatu yang random dan berbeda setiap saat.

```
> set.seed(123)
> goldarah <- sample(c("A","B","O","AB"), size =60,replace=TRUE,prob=c(0.38,0.12,0.38,0.12))
> goldarah.table<-table(goldarah)
> goldarah.table
goldarah
  A AB  B  O
23  4 11 22
```

Hasilnya terlihat beda dari yang sebelumnya.

Selanjutnya untuk membuat data spesifik dengan ketentuan **mean**, **standard deviation** tertentu kita dapat menggunakan fungsi **rnorm()**.

```
> set.seed(124)
> newdata <- rnorm(10,mean = 50,sd = 3)
> mean(newdata)
[1] 50.6443
> sd(newdata)
[1] 2.590623
> |
```

Kita dapat mendeklarasikan berapa jumlah data yang diinginkan, **mean**, dan **standard deviation**-nya.

Setelah itu, kita dapat cek dengan fungsi yang sudah pernah kita pelajari, **mean()** dan **sd()**.

Walaupun hasilnya tidak 100% tepat, tetapi bisa membuat data dengan ketepatan yang cukup bagus.

# *Thankyou*

