

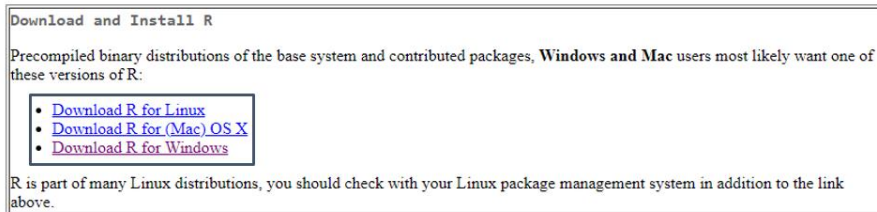
# Praktikum Komputasi Statistika I

## Pertemuan 1

### Pengenalan R

#### Instalasi R-Gui

Download di website <https://cran.r-project.org/>



Subdirectories:

[base](#) Binaries for base distribution. This is what you want to [install R for the first time](#).  
[contrib](#) Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.  
[old-contrib](#) Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).  
[Rtools](#) Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

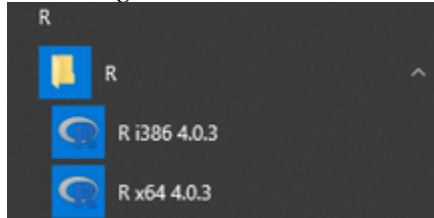
Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

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

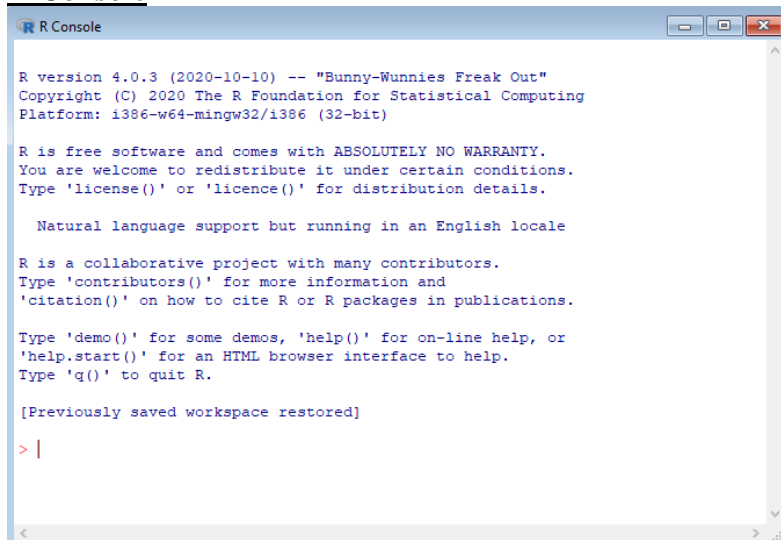
Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

#### Membuka R-Gui

Membuka R-Gui dapat dilakukan melalui *shortcut* yang sudah ada pada layar desktop, atau dengan *searching* R di *Start Windows*.



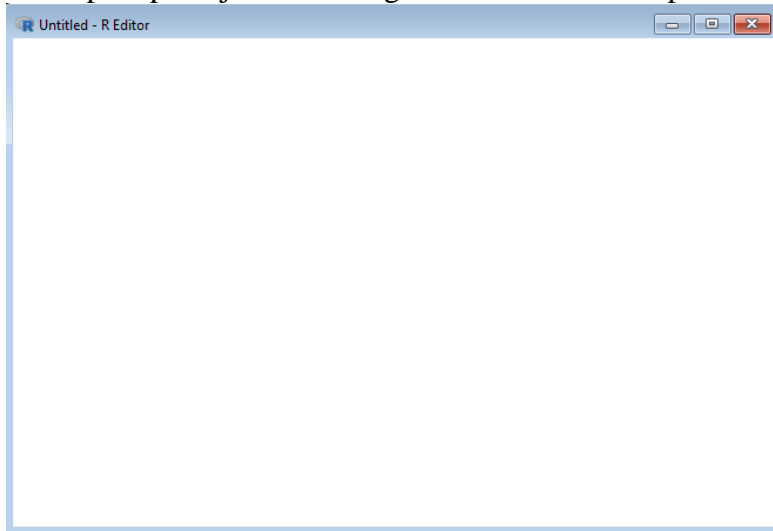
#### R Console



R Console dapat digunakan untuk menulis dan menjalankan *syntax* langsung per baris dengan menekan tombol “enter” sesudah selesai menulis *syntax* (*output* akan langsung keluar jika ada).

## **R Script**

R Script dapat dijalankan dengan : File → New script



Pada R Script, kita dapat menuliskan terlebih dahulu seluruh *syntax* sampai selesai.



Untuk menjalankan *syntax* dapat dengan ctrl + R atau klik pada icon diatas. *Syntax* dapat dijalankan per baris atau bisa keseluruhan dengan mengeblok seluruh *syntax*. Outputnya akan muncul di R-Console.

## **Data dan Tipe Data**

Dalam R, terdapat 6 struktur data yang akan dijumpai, yakni vektor, matriks, *data frame*, *list*, dan *factors*. Struktur data ini akan terdiri dari data-data, dalam R terdapat 5 tipe data seperti terlihat di bawah, untuk memeriksa tipe data dapat menggunakan fungsi `class()`.

Tipe Data	Contoh
Numeric	Double >4.1 >6.2
	Integer >1 >2
Complex	>2+5i
Logical	>TRUE

	>FALSE >c(T,F,T,F) >year>2000
Character	>c("SD","SMP","SMA") >c("T","F","3")

### **Nama Variabel**

Nama dapat terdiri dari kombinasi dari huruf besar, huruf kecil, angka, *underscore*, dan titik, akan tetapi kita tidak dapat menggunakan *Extended ASCII character* atau karakter spesial seperti \$%#\* dan lainnya. Nama hanya dapat diawali dengan huruf dan titik, jika diawali dengan titik tidak dapat dilanjutkan dengan angka. Berikut contoh dari nama objek yang valid :

```
mydata
my.data
my.data.2
data.test.no.2
```

Berikut contoh nama objek yang tidak valid :

```
4data (dimulai dari angka)
data-1 (operator '-' tidak dapat digunakan)
```

Selain itu kita juga harus menghindari penggunaan nama fungsi/command built-in R. Apabila kita menggunakan nama fungsi/data built-in maka kita tidak dapat mengakses fungsi dan data built-in ini sampai kita menghapus nama objek yang dibuat. Perlu diperhatikan pula bahwa nama di R *case sensitive* yang berarti Nama berbeda dengan nama.

### **Variabel di R**

Variabel di R dapat berisikan hanya 1 nilai, dapat juga berisikan lebih dari 1 nilai berupa vektor.

#### **Variabel dengan 1 nilai**

Variabel dengan 1 nilai dapat dituliskan dengan " = " atau " <- "

```
> mydata = 3
> mydata
[1] 3
```

```
> my.data <- 5
> my.data
[1] 5
```

```
> dataku = ('hai dunia')
> dataku
[1] "hai dunia"
```

### **Variabel berisikan vektor (data *array* satu dimensi)**

Vektor merupakan struktur data paling dasar yang dikenal dalam R, yang merupakan suatu *array*/himpunan dari bilangan, character/string, logical value, dll. Yang perlu diingat kita harus menggunakan satu tipe data yang sama untuk data kita, yakni kita tidak dapat menggabungkan dua data atau lebih yang berbeda tipe ke dalam satu objek vektor. Jika ini dilakukan, maka R akan mengubah data ke mode yang terumum.

```
> c(T,2,3) #vektor terdiri dari data dengan mode logical dan numeric
[1] 1 2 3
```

```
> c("M",T) #vektor terdiri dari data dengan mode character dan logical
[1] "M"      "TRUE"
```

```
> c("M",2,T) #vektor terdiri dari data dengan mode character, numeric
dan logical
[1] "M"      "2"      "TRUE"
```

```
> w = c(1,3) #w berisikan 1 dan 3
> print(w)
[1] 1 3
```

```
> x = c(2:5) #x berisikan 2, 3, 4, dan 5 (2 sampai 5)
> print(x)
[1] 2 3 4 5
```

```
> y = c('aku', 'dan', 'kamu')
> y
[1] "aku"      "dan"      "kamu"
```

```
> z = rep(1,4) #z berisikan 1 sebanyak 4
> z
[1] 1 1 1 1
```

*\*tanda pagar “#” merupakan komentar yang berarti kata - kata setelah tanda # tidak akan di eksekusi R*

Untuk vektor, dapat diketahui jumlah data dalam vektor atau panjangnya vektor dengan fungsi `length()`.

### **Ekstraksi sebagian Data Vector**

Misalkan kita memiliki sebuah vektor yang terdiri dari 5 elemen, yaitu 5, 14, 8, 9,5 :

```
> y=c(5,14,8,9,5)
> y
[1] 5 14 8 9 5
```

Kita dapat membentuk data baru dengan melakukan ekstraksi data. Sebagai contoh :

- Menampilkan elemen pertama  

```
> y[1]  
[1] 5
```
- Menampilkan elemen kedua dan kelima  

```
> y[c(2,5)]  
[1] 14 5
```
- Menampilkan semua elemen kecuali elemen keempat  

```
> y[-4]  
[1] 5 14 8 5
```
- Menampilkan semua elemen kecuali elemen pertama dan ketiga  

```
> y[-c(1,3)]  
[1] 14 9 5
```
- Menampilkan semua elemen yang lebih besar dari 8  

```
> y[y>8]  
[1] 14 9
```
- Memberi nama/menyimpan nilai vektor  

```
> vektor=y[y>8]  
> vektor  
[1] 14 9
```

### **Operator Operasi di R**

Terdapat 3 macam operator operasi di R, yakni aritmatika, perbandingan, dan logika, *syntax* dan penjelasan lebih lanjutnya adalah sebagai berikut:

#### **Operator Aritmatika**

<b>Simbol</b>	<b>Keterangan</b>
+	<i>addition</i> - operasi penjumlahan
-	<i>subtraction</i> - operasi pengurangan
*	<i>multiplication</i> - operasi perkalian
/	<i>division</i> - operasi pembagian
^	<i>exponentiation</i> - operasi pemangkatan
%%	<i>modulus</i> - mencari sisa pembagi
/%	<i>integer</i> - mencari bilangan bulat hasil pembagian saja

## Operator Perbandingan

Simbol	Keterangan
==	sama dengan, bernilai TRUE jika kedua objek bernilai sama
!=	tidak sama dengan
>	lebih besar dari
<	lebih kecil dari
>=	lebih besar sama dengan
<=	lebih kecil sama dengan

Pada operator perbandingan, akan bernilai TRUE jika hasilnya benar dan bernilai FALSE jika hasilnya salah.

## Operator Logika

Simbol	Keterangan
&&	operator logika AND
	operator logika OR
!	operator logika NOT
&	operator logika AND element wise
	operator logika OR element wise

## Mengimpor Data, Fungsi Statistika Dasar dan R Studio

---

### Mengimpor Data

Langkah sederhana untuk mengimpor data yang berasal dari Excel dapat digunakan perintah **read.delim()** dimana cara kerjanya adalah pertama kita membuka file yang berisi data yang kita miliki, lalu *copy* bagian data yang kita ingin *import* dan selanjutnya tulis perintah yang sesuai. Sebagai contoh :

	A	B	C
1	Tahun	Populasi	
2	2000	120	
3	2001	145	
4	2002	160	
5	2003	189	
6	2004	201	
7	2005	222	
8	2006	245	
9	2007	289	
10	2008	300	
11	2009	340	
12	2010	467	
13			

```
> populasi=read.delim('clipboard')  
> populasi
```

```
      Tahun Populasi  
1    2000      120  
2    2001      145  
3    2002      160  
4    2003      189  
5    2004      201  
6    2005      222  
7    2006      245  
8    2007      289  
9    2008      300  
10   2009      340  
11   2010      467
```

Kita dapat menggunakan fungsi *head* untuk memperoleh output data hanya pada sebagian data pertama. *Default* banyak data yang muncul adalah 6, tetapi dapat dimodifikasi sesuai keinginan dengan menambahkan angka seperti berikut :

```
> head(populasi)
  Tahun Populasi
1  2000      120
2  2001      145
3  2002      160
4  2003      189
5  2004      201
6  2005      222

> head(populasi,3)
  Tahun Populasi
1  2000      120
2  2001      145
3  2002      160
```

Kita dapat menggunakan fungsi *tail* untuk memperoleh output data hanya pada sebagian data terakhir. *Default* banyak data yang muncul adalah 6, tetapi dapat dimodifikasi sesuai keinginan dengan menambahkan angka seperti berikut :

```
> tail(populasi)
  Tahun Populasi
6  2005      222
7  2006      245
8  2007      289
9  2008      300
10 2009      340
11 2010      467

> tail(populasi,3)
  Tahun Populasi
9  2008      300
10 2009      340
11 2010      467
```



### **Fungsi Operasi di R (Descriptive Statistics)**

Melalui R kita dapat melakukan analisis deskriptif dari data yang diberikan, seperti menghitung nilai mean, median, minimum, maksimum, variansi, standar deviasi, jumlahan dan sebagainya. Berikut contohnya :

```
> data=c(20,10,30,100,50)
> mean(data)#menghitung nilai mean
[1] 42
> median(data) #menghitung nilai median
[1] 30
> min(data) #menghitung nilai minimum
[1] 10
> max(data) #menghitung nilai maksimum
[1] 100
> var(data) #menghitung nilai variansi
[1] 1270
> sd(data) #menghitung nilai standar deviasi
[1] 35.63706
> sum(data) #menjumlahkan data
[1] 210
```

Kita juga dapat menggunakan fungsi summary() untuk memunculkan nilai statistik seperti di atas secara bersamaan :

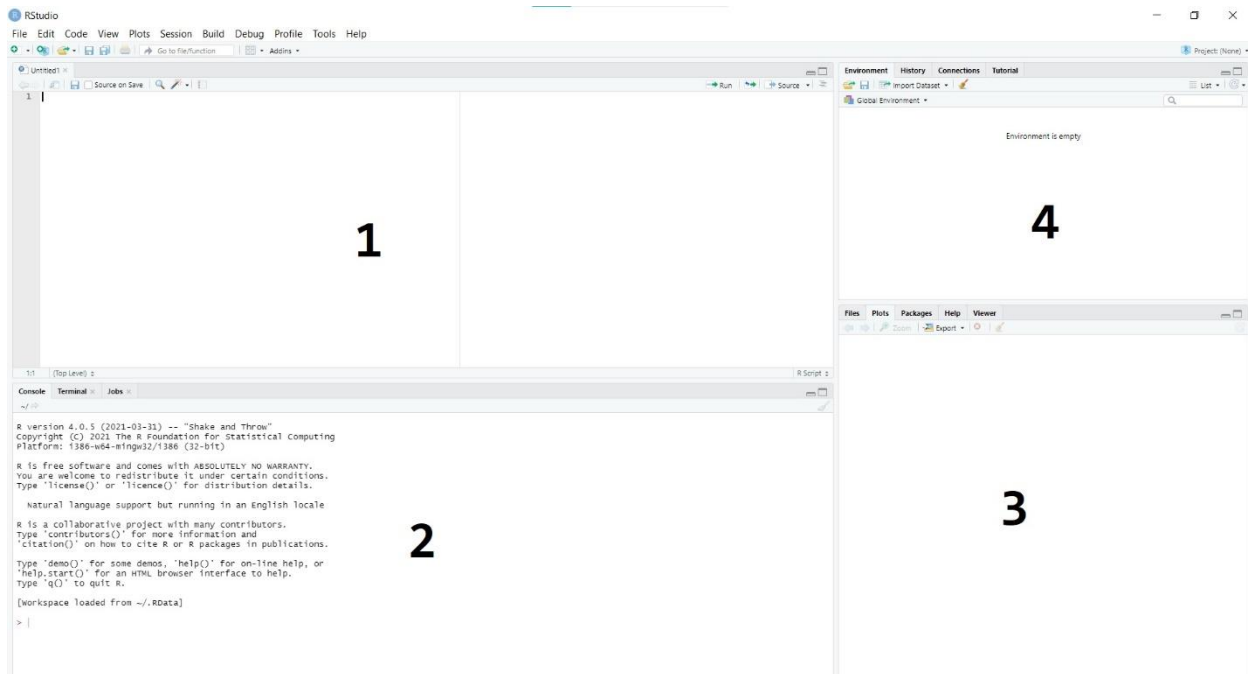
```
> summary(data)
   Min.   1st Qu.   Median     Mean 3rd Qu.    Max.
    10     20      30      42    50     100
```

Apabila kita perhatikan ternyata data pada vektor tidak urut, namun kita dapat menggunakan perintah sort() untuk mengurutkannya.

```
> sort(data)
[1] 10 20 30 50 100
```

### **R Studio**

R Studio merupakan pengembangan IDE (*Integrated Development Environment*) untuk R, yang memberikan tampilan *interface* R yang lebih baik dan cocok digunakan untuk pekerjaan yang lebih profesional dan kompleks. Perlu diketahui dalam menjalankan R Studio kita perlu menginstall R terlebih dahulu.



Keterangan:

- Nomor 1 : R Script
- Nomor 2 : R Console
- Nomor 3 : *Global environment* dari *working space* kita
- Nomor 4 : Menunjukkan hasil plot, file dalam direktori kerja, fitur help, dan *viewer*.

Secara keseluruhan tidak ada perbedaan dalam menuliskan *syntax* yang ada dalam R pada R Studio. Adapun, dalam menjalankan *syntax* secara langsung dalam script pada R Studio dapat menggunakan `ctrl + Enter`.

## Matriks dan Data Frame

Matriks	Data Frame
Setiap kolom atau variabel harus terdiri dari tipe data yang sama, yakni numerik.	Setiap kolom atau variabel tidak harus terdiri dari tipe data yang sama, dapat memiliki tipe data yang berbeda seperti numerik, karakter, dll.
Dapat digunakan untuk melakukan segala jenis operasi aljabar linear.	Tidak dapat digunakan untuk melakukan segala jenis operasi aljabar linear.
Kurang nyaman digunakan jika sering merujuk ke sebuah kolom dengan nama.	Lebih nyaman digunakan jika sering merujuk ke sebuah kolom dengan nama.
Lebih hemat memori.	Kurang hemat memori.

### Matriks

```
> matriks1=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,ncol=3)
> matriks1
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

> matriks2=matrix(1:9,3)
> matriks2
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

Secara default, data akan diisikan kolom perkolom, seperti yang dapat dilihat sebagai berikut :

```
> data=c(5,7,6,4,6,9)
> matx=matrix(data,nrow=3,ncol=2)
> matx
      [,1] [,2]
[1,]    5    4
[2,]    7    6
[3,]    6    9
```

Untuk melakukan pengisian menurut baris perbaris, maka dapat digunakan optional argumen **byrow=T** pada command matriks. Lihat contoh berikut :

```
> matx2=matrix(data,nrow=3,ncol=2,byrow=T)
> matx2
      [,1] [,2]
[1,]    5    7
[2,]    6    4
[3,]    6    9
```

### Operator untuk operasi matriks

Operator Matriks	Keterangan
*	Perkalian elemen demi elemen matriks
%*%	Perkalian matriks
%o%	Outer
solve	Invers matriks
t	Transpose matriks
crossprod	Cross product yakni $X^t X$ , untuk X suatu matriks

### Menggabungkan satu kolom baru atau satu baris baru ke matriks lain

Dalam R bila kita ingin menggabungkan satu kolom atau satu baris baru ke dalam satu objek matriks lain dapat menggunakan perintah **rbind** (untuk menambahkan baris) dan **cbind** (untuk menambahkan kolom). Sebagai contoh :

```
> mt=matrix(c(2,3,4,5),2)
> mt
      [,1] [,2]
[1,]    2    4
[2,]    3    5

> #Menambah kolom
> mt1=cbind(mt,c(1,2))
> mt1
      [,1] [,2] [,3]
[1,]    2    4    1
[2,]    3    5    2
```

```
> mt2=cbind(c(1,2),mt)
> mt2
```

```
      [,1] [,2] [,3]
[1,]    1    2    4
[2,]    2    3    5
```

```
> #Menambah baris
> mt3=rbind(mt,c(1,2))
> mt3
```

```
      [,1] [,2]
[1,]    2    4
[2,]    3    5
[3,]    1    2
```

```
> mt4=rbind(c(1,2),mt)
> mt4
```

```
      [,1] [,2]
[1,]    1    2
[2,]    2    4
[3,]    3    5
```

Dapat dilihat bahwa posisi vektor yang ditambahkan dalam argumen mempengaruhi tata letak vektor pada matriks

### **Data Frame**

Objek data frame dapat dibuat menggunakan perintah **data.frame**. Contoh pembuatan data frame dapat dilihat berikut :

```
> nama=c("Elok","Simaster","Elisa")
> nilai=c(95,90,80)
> data.frame(nama,nilai)
```

```
      nama nilai
1     Elok    95
2 Simaster    90
3     Elisa    80
```

Sama seperti matriks data frame juga dapat disimpan pada suatu variabel untuk memudahkan akses jika diperlukan, sebagai contoh data frame di atas akan disimpan dalam variabel bernama website. Cara lain untuk memberi nama variabel pada setiap kolom pada data frame dengan menggunakan fungsi **names**

```
#memberi nama variabel kolom 1 dengan "nama"
names(website)[1]="nama"
#memberi nama variabel kolom 2 dengan "nilai"
names(website)[2]="nilai"
> #memberi nama variabel kolom 1 dengan "nama"
> names(website)[1]="nama"
> #memberi nama variabel kolom 2 dengan "nilai"
> names(website)[2]="nilai"
>
> website
      nama nilai
1     Elok    95
2 Simaster    90
3     Elisa    80
```

### Ekstraksi data matriks / data frame

Beberapa perintah untuk menampilkan data dalam matriks dan data frame ini dapat diberikan sebagai berikut ini,

```
matriks1 = matrix(1:12,4)
dataframe1 = data.frame(nilai = 1:7 ,
huruf = c("A", "B", "C", "D", "E", "F", "G"),
nilai1 = 8:14)
> matriks1 = matrix(1:12,4)
> dataframe1 = data.frame(nilai = 1:7 ,
+ huruf = c("A", "B", "C", "D", "E", "F", "G"),
+ nilai1 = 8:14)
>
> matriks1
      [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
> dataframe1
      nilai huruf nilai1
1         1     A      8
2         2     B      9
3         3     C     10
4         4     D     11
5         5     E     12
6         6     F     13
7         7     G     14
```

Untuk menampilkan baris dan kolom tertentu :

```
#elemen baris ke 2 kolom ke 3 dari matriks1
matriks1[2,3]
#elemen baris ke 5 kolom ke 2 dari dataframe1
dataframe1[5,3]
```

```

> #elemen baris ke 2 kolom ke 3 dari matriks1
> matriks1[2,3]
[1] 10
> #elemen baris ke 5 kolom ke 2 dari dataframe1
> dataframe1[5,3]
[1] 12

```

Menampilkan elemen kolom “huruf” :

```
dataframe1["huruf"]
```

```

> dataframe1["huruf"]
huruf
1      A
2      B
3      C
4      D
5      E
6      F
7      G

```

#dengan memanggil elemen (kolom) kedua

```
dataframe1[2]
```

```

> dataframe1[2]
huruf
1      A
2      B
3      C
4      D
5      E
6      F
7      G

```

#output berupa vektor

```
dataframe1[,2]
```

```

> dataframe1[,2]
[1] "A" "B" "C" "D" "E" "F" "G"

```

#output sama tetapi nama kolom berbeda

```
as.data.frame(dataframe1[,2])
```

```

> as.data.frame(dataframe1[,2])
dataframe1[, 2]
1              A
2              B
3              C
4              D
5              E
6              F
7              G

```

#menyimpan nama kolom yang baru ke dalam objek bernama test

```
test = as.data.frame(dataframe1[,2])
```

```
test
```

```

> test = as.data.frame(dataframe1[,2])
> test
  dataframe1[, 2]
1              A
2              B
3              C
4              D
5              E
6              F
7              G

```

#mengubah nama kolom

```
names(test)="huruf"
```

```
test
```

```

> names(test)="huruf"
> test
  huruf
1     A
2     B
3     C
4     D
5     E
6     F
7     G

```

#menyimpan elemen baris 2 sampai 4 dan kolom 1 sampai 2

```
df_baru = dataframe1[2:4,1:2]
```

```
df_baru
```

```

> df_baru = dataframe1[2:4,1:2]
> df_baru
  nilai huruf
2     2     B
3     3     C
4     4     D

```

#menampilkan elemen baris 1,3,5 dan kolom 1,3 dari dataframe1

```
dataframe1[c(1,3,5),c(1,3)]
```

```

> dataframe1[c(1,3,5),c(1,3)]
  nilai nilai1
1     1      8
3     3     10
5     5     12

```

#menampilkan selain kolom ke 3 dari dataframe1

```
dataframe1[, -3]
```

```

> dataframe1[, -3]
  nilai huruf
1     1     A
2     2     B
3     3     C
4     4     D
5     5     E
6     6     F
7     7     G

```



```
#menampilkan selain baris ke 4 dari dataframe1  
dataframe1[-4,]
```

```
> dataframe1[-4,]  
  nilai huruf nilail  
1      1      A      8  
2      2      B      9  
3      3      C     10  
5      5      E     12  
6      6      F     13  
7      7      G     14
```

```
#menampilkan selain baris 4,5 dan kolom 1 dari dataframe1  
dataframe1[c(-4,-5),-1]
```

```
> dataframe1[c(-4,-5),-1]  
  huruf nilail  
1      A      8  
2      B      9  
3      C     10  
6      F     13  
7      G     14
```

### Latihan

1. Billy membeli 2 kg apel dan 3 kg jeruk. Harga satu kg apel adalah Rp 40.000 dan harga satu kg jeruk adalah Rp 25.000. Jika Billy membawa uang sebesar Rp 160.000, berapa sisa uangnya setelah dibelanjakan?
2. Diberikan data sebagai berikut.

Y	X1	X2
251	42	29
248	44	30
267	45	30
273	48	29
270	48	31
285	53	32
302	57	33

Carilah estimator  $\beta$  dari persamaan regresi  $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$  dengan menggunakan metode *least square*!

(**Hint** : gunakan operasi matriks untuk menyelesaikan persamaan  $\hat{\beta} = (X'X)^{-1}X'Y$ )

Output yang diharapkan :

```
> beta
      [,1]
cons 150.399181
x1    4.057946
x2   -2.450102
```