

BAB 2

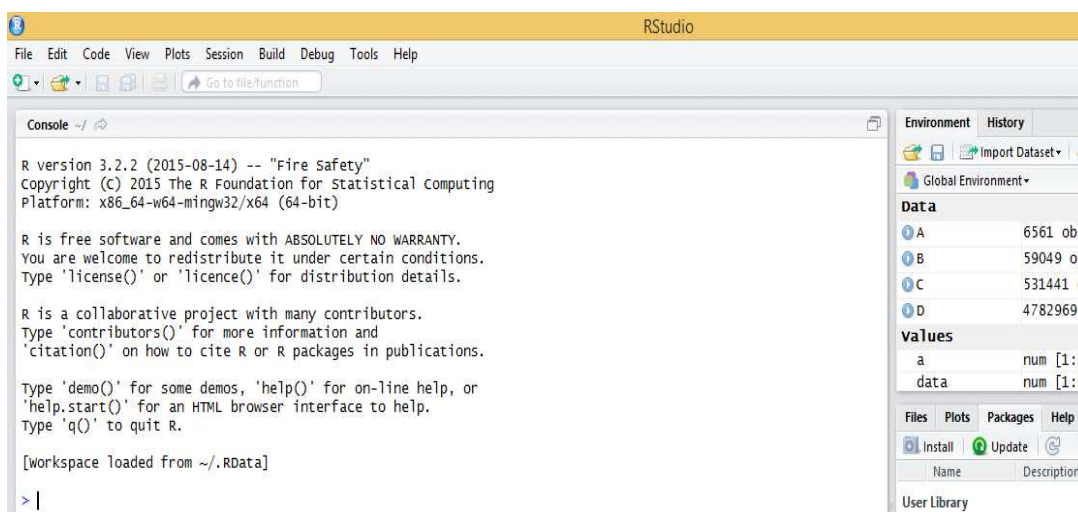
FUNGSI DASAR DALAM R

Memulai R

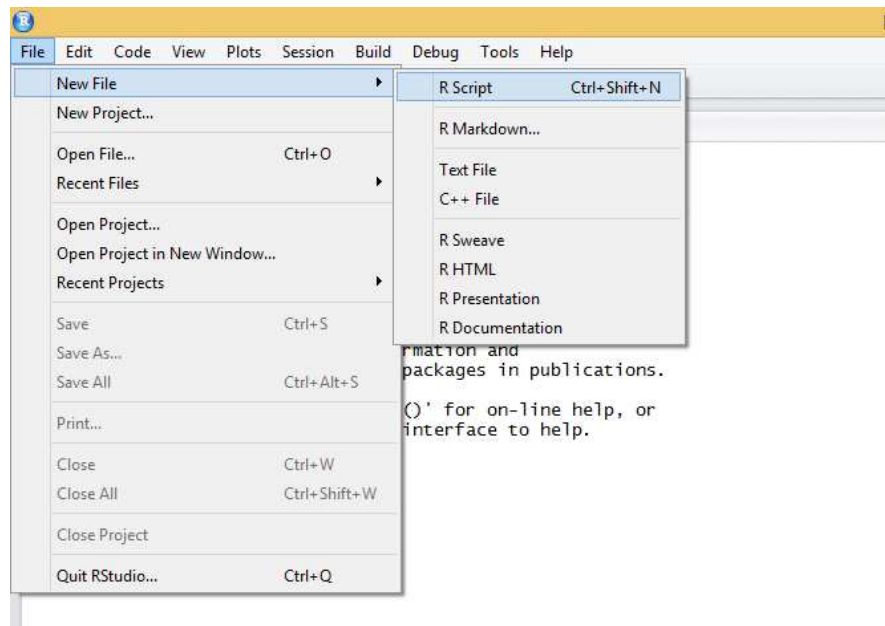
Berikut dipaparkan langkah-langkah untuk masuk ke dalam area kerja R. Aktifkan RStudio terlebih dahulu (Gambar 2.1), sehingga akan muncul tampilan seperti pada Gambar 2.2. Pada Gambar 2.2, pilih *File => New File => R Script* (lihat Gambar 2.3), sehingga muncul tampilan seperti pada Gambar 2.4. Gambar 2.4 merupakan area kerja R, di mana pada pembahasan selanjutnya, kode R akan diinput pada area tersebut. Setelah kode R diinput, selanjutnya kode R tersebut dieksekusi, sehingga muncul *output* berdasarkan eksekusi kode R tersebut.



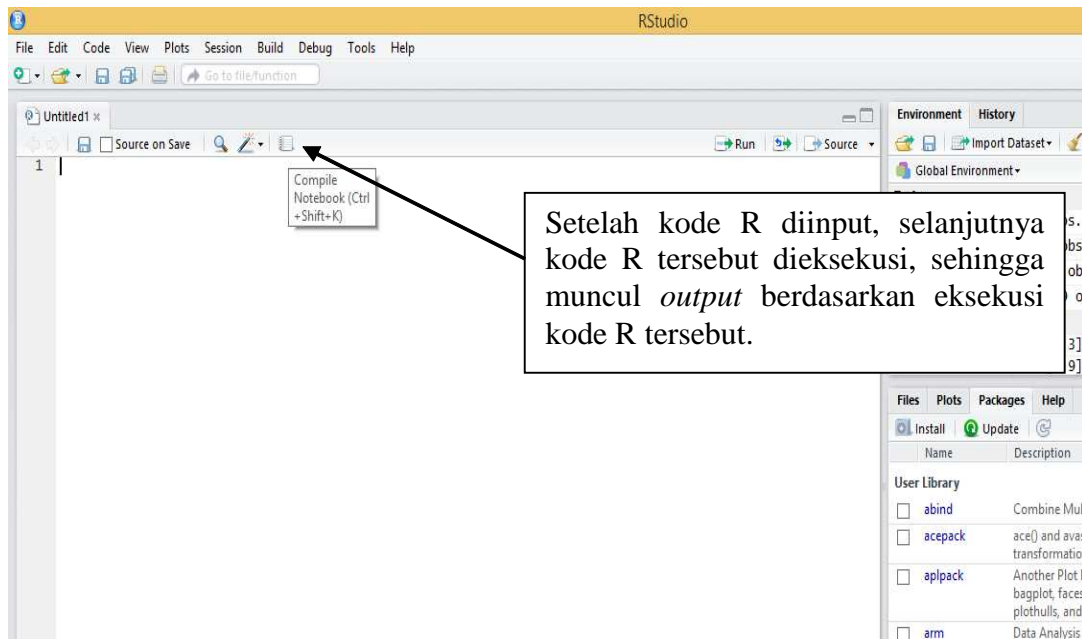
Gambar 2.1



Gambar 2.2



Gambar 2.3

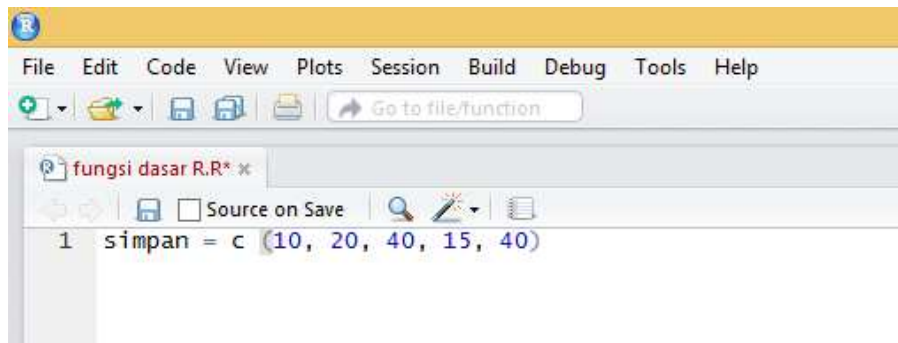


Gambar 2.4

Menyimpan Data dalam Variabel (Fungsi c)

Andaikan suatu data terdiri dari bilangan 10, 20, 40,15, 40. Misalkan data tersebut akan disimpan dalam variabel yang diberi nama **simpan**. Dalam R, fungsi **c** digunakan untuk menggabungkan satu nilai data, dengan nilai data lainnya. Perhatikan kode R berikut (lihat juga Gambar 2.5).

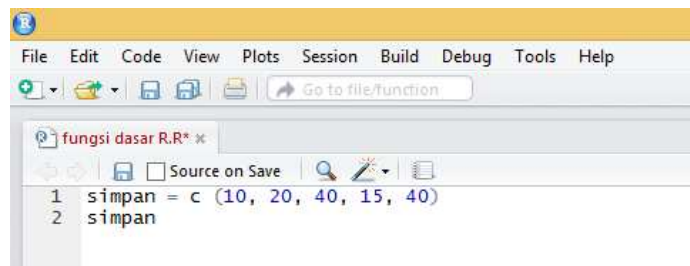
simpan = c (10, 20, 40, 15, 40)



Gambar 2.5

Kode R **simpan = c (10, 20, 40, 15, 40)** atau pada Gambar 2.5, dapat diartikan variabel **simpan** ditugaskan untuk menyimpan data dengan nilai 10, 20, 40, 15, 40. Data-data tersebut diapit oleh tanda buka-tutup kurung biasa, dan masing-masing nilai data dipisahkan oleh tanda koma. Perhatikan kode R berikut (lihat juga Gambar 2.6).

simpan

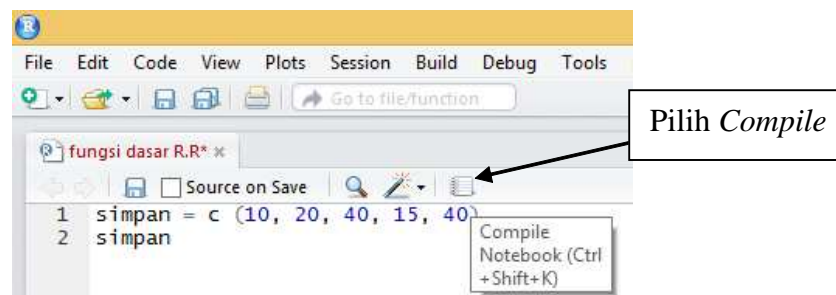


Gambar 2.6

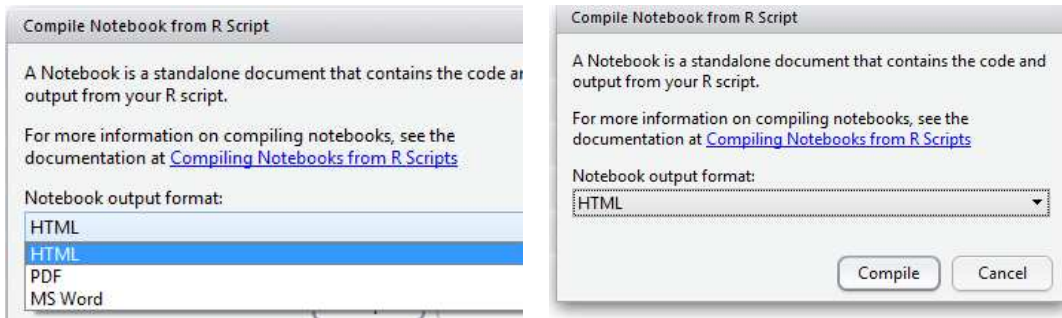
Kode R pada Gambar 2.6, baris ke-2, dapat diartikan menyajikan atau menampilkan nilai data yang disimpan dalam variabel **simpan**.

Mengeksekusi Kode R

Sekarang, kode R pada Gambar 2.6 akan dieksekusi. Pilih *Compile* (perhatikan Gambar 2.7), sehingga muncul tampilan seperti pada Gambar 2.8. Pada Gambar 2.8, *output* dari hasil eksekusi kode R pada Gambar 2.6, dapat berformat HTML, PDF, dan Ms Word. Dalam percobaan kali ini, pilih HTML dan *Compile*. Hasilnya diperlihatkan pada Gambar 2.9.



Gambar 2.7



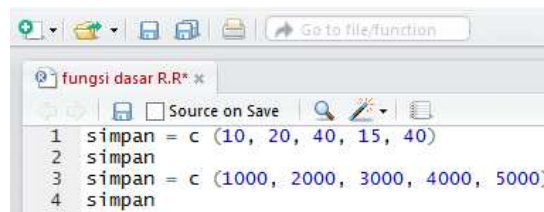
Gambar 2.8

```
simpan = c (10, 20, 40, 15, 40)
simpan

## [1] 10 20 40 15 40
```

Gambar 2.9

Pada Gambar 2.9, `## [1] 10 20 40 15 40` merupakan hasil eksekusi kode R pada baris ke-2. Perhatikan kode R pada Gambar 2.10, pada baris ke-3 dan ke-4.



Gambar 2.10

Gambar 2.11 merupakan hasil eksekusi kode R pada Gambar 2.10.

```
simpan = c (10, 20, 40, 15, 40)
simpan

## [1] 10 20 40 15 40

simpan = c (1000, 2000, 3000, 4000, 5000)
simpan

## [1] 1000 2000 3000 4000 5000
```

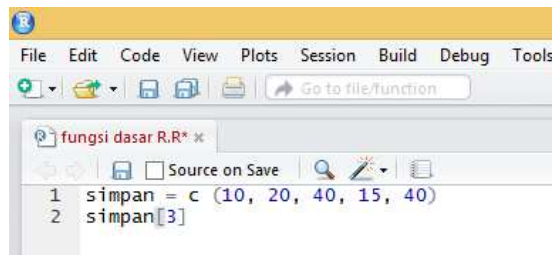
Gambar 2.11

Mengakses Nilai Data dalam Variabel secara Individual

Misalkan variabel **simpan** menyimpan nilai 10, 20, 40, 15, dan 40. Andaikan hanya ingin ditampilkan nilai dari variabel **simpan**, pada posisi ke-3, yakni nilai 40. Perhatikan kode R berikut.

```
simpan[3]
```

Kode R di atas berarti menampilkan nilai dalam variabel **simpan**, pada posisi ke-3, yakni 40. Ilustrasi dalam R diperlihatkan pada Gambar 2.12 dan Gambar 2.13.



Gambar 2.12

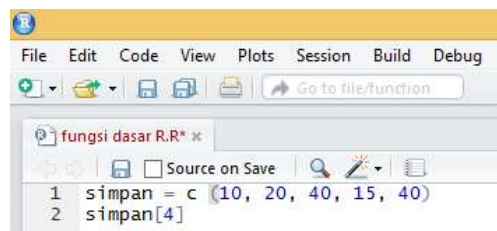


Gambar 2.13

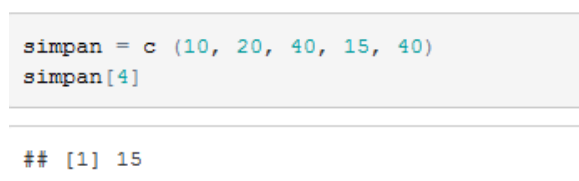
Perhatikan kode R berikut.

simpan[4]

Kode R di atas berarti menampilkan nilai dalam variabel **simpan** pada posisi ke-4, yakni 15. Ilustrasi dalam R diperlihatkan pada Gambar 2.14 dan Gambar 2.15.



Gambar 2.14



Gambar 2.15

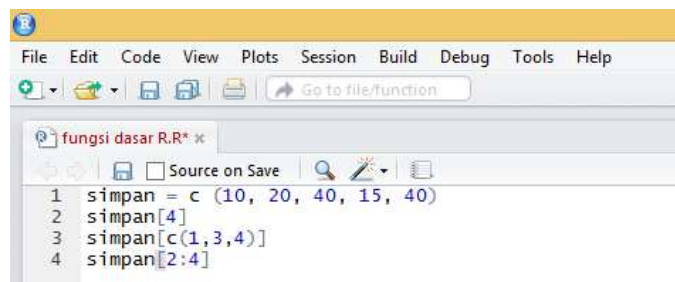
Berikut merupakan kode R untuk menampilkan 3 buah nilai dalam variabel **simpan** pada posisi ke 1,3, dan 4.

simpan[c(1,3,4)]

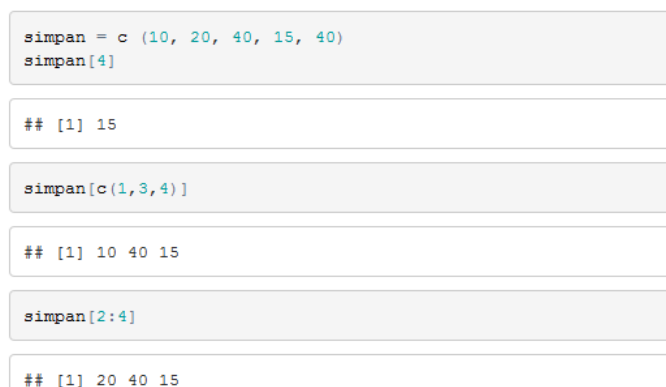
Kode **simpan[c(1,3,4)]** berarti menampilkan nilai dalam variabel **simpan** pada posisi ke 1, 3, dan 4, yakni 10, 40, dan 15. Berikut merupakan kode R untuk menampilkan 3 buah nilai dalam variabel **simpan** pada indeks ke 2,3,4.

simpan[2:4]

Kode **simpan[2:4]** berarti menampilkan nilai dalam variabel **simpan** pada posisi ke-2, sampai posisi ke-4, yakni 10, 20, dan 40. Ilustrasi dalam R diperlihatkan pada Gambar 2.16 dan Gambar 2.17.



Gambar 2.16



Gambar 2.17

Mengubah Nilai Data dalam Variabel

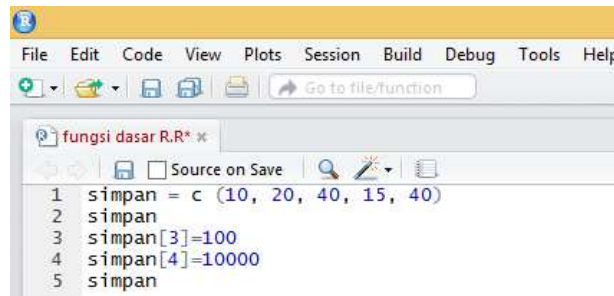
Pada pemaparan sebelumnya, diketahui variabel **simpan** menyimpan nilai 10, 20, 40, 15, dan 40. Andaikan nilai dari variabel **simpan**, pada posisi ke-3, yakni nilai 40, **akan diubah** menjadi 100. Perhatikan kode R berikut.

simpan[3]=100

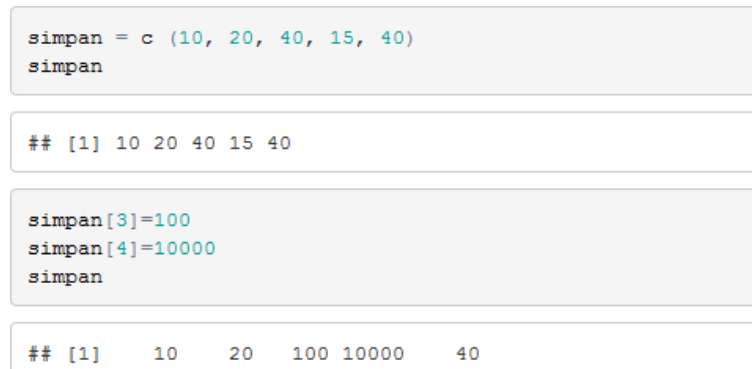
Kode R tersebut, yakni **simpan[3]=100**, dapat diartikan mengubah nilai data variabel **simpan** pada posisi ke-3 dengan nilai 100. Misalkan nilai pada posisi ke-4, yakni 15, ingin diubah menjadi 10000. Berikut merupakan kode dalam R untuk mengubah nilai pada posisi ke-4, yakni 15 menjadi 10000.

simpan[4]=10000

Ilustrasi dalam R diperlihatkan pada Gambar 2.18 dan Gambar 2.19.



Gambar 2.18



Gambar 2.19

Menghapus Nilai Data dalam Variabel

Misalkan suatu variabel bernama **NILAI** menyimpan 5 nilai, yakni 10, 40, 45, 30, dan 80. Berikut kode dalam R untuk menyimpan 5 nilai tersebut ke dalam variabel **NILAI**.

NILAI=c(10, 40, 45, 30, 80)

Misalkan nilai 45 pada variabel **NILAI** akan dihapus, sehingga nilai dalam variabel **NILAI** menjadi 10, 40, 30, dan 80. Diketahui nilai 45 berada pada posisi atau indeks ke-3. Berikut kode R untuk menghapus nilai 45 dalam variabel **NILAI**.

NILAI=NILAI[-3]

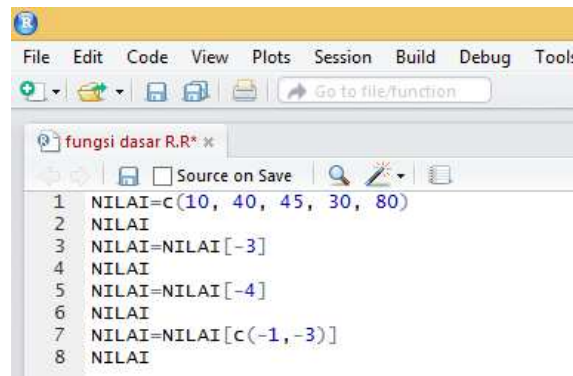
Setelah kode R **NILAI=NILAI[-3]** dieksekusi, maka nilai yang tersimpan pada variabel **NILAI** saat ini adalah 10, 20, 30, dan 80. Misalkan nilai 80 pada variabel **NILAI** akan dihapus, sehingga nilai dalam variabel **NILAI** menjadi 10, 20, dan 30. Perhatikan kode R berikut.

NILAI=NILAI[-4]

Diketahui nilai yang tersimpan pada variabel **NILAI** saat ini adalah 10, 20, dan 30. Misalkan nilai 10 dan 30 pada variabel **NILAI** akan dihapus, sehingga nilai dalam variabel **NILAI** adalah 20. Perhatikan kode R berikut.

NILAI=NILAI[c(-1,-3)]

Nilai dalam variabel **NILAI** saat ini adalah 20. Ilustrasi dalam R diperlihatkan pada Gambar 2.20 dan Gambar 2.21.



Gambar 2.20



Gambar 2.21

Penggunaan Operator > (Lebih Besar Dari)

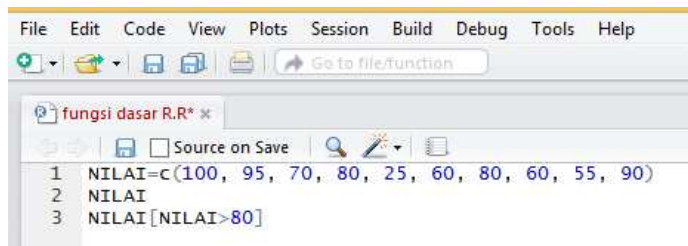
Misalkan suatu variabel bernama **NILAI** menyimpan 10 nilai, yakni 100, 95, 70, 80, 25, 60, 80, 60, 55, 90. Berikut kode R untuk menugaskan variabel **NILAI** menyimpan kesepuluh nilai tersebut.

```
NILAI=c(100, 95, 70, 80, 25, 60, 80, 60, 55, 90)
```

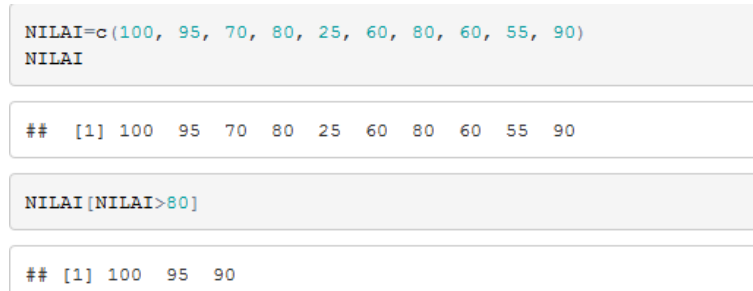
Misalkan akan ditampilkan nilai dari variabel **NILAI** dengan syarat lebih besar dari 80, yakni 100, 95, 90. Berikut kode dalam R.

```
NILAI[NILAI>80]
```

Ilustrasi dalam R diperlihatkan pada Gambar 2.22 dan Gambar 2.23.



Gambar 2.22



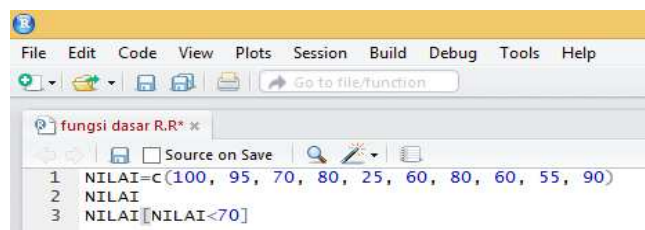
Gambar 2.23

Penggunaan Operator < (Lebih Kecil Dari)

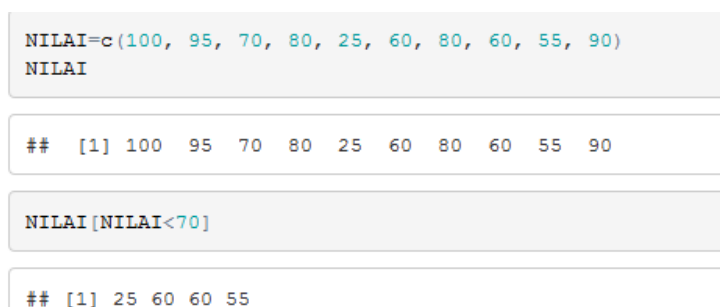
Diketahui sebelumnya bahwa variabel bernama **NILAI** menyimpan 10 nilai, yakni 100, 95, 70, 80, 25, 60, 80, 60, 55, 90. Misalkan akan ditampilkan nilai dari variabel **NILAI** dengan syarat lebih kecil dari 70, yakni 25, 60, 60, 55. Berikut kode dalam R.

NILAI[NILAI<70]

Ilustrasi dalam R diperlihatkan pada Gambar 2.24 dan Gambar 2.25.



Gambar 2.24



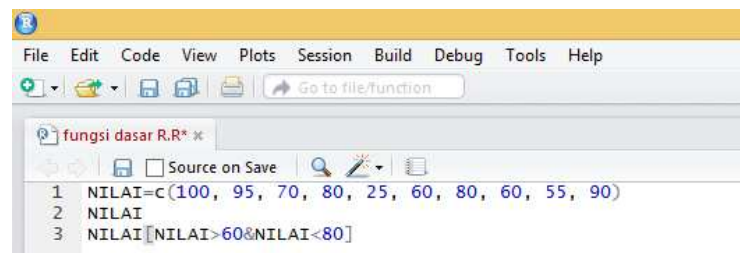
Gambar 2.25

Penggunaan Operator & (Dan)

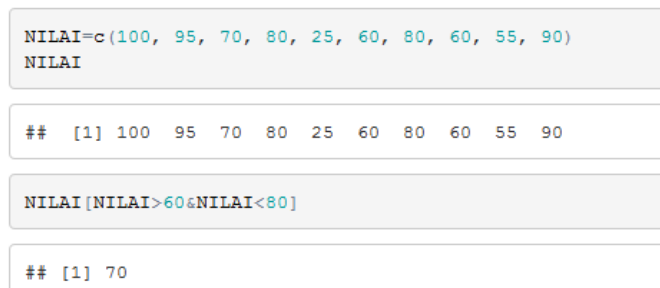
Diketahui sebelumnya bahwa variabel bernama **NILAI** menyimpan 10 buah bilangan, yakni 100, 95, 70, 80, 25, 60, 80, 60, 55, 90. Misalkan akan ditampilkan nilai dari variabel **NILAI** dengan syarat lebih besar 60 dan lebih kecil 80, yakni 70. Berikut disajikan kode R.

```
NILAI[NILAI>60&NILAI<80]
```

Ilustrasi dalam R diperlihatkan pada Gambar 2.26 dan Gambar 2.27.



Gambar 2.26



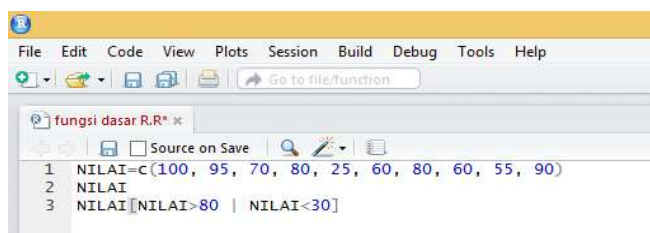
Gambar 2.27

Penggunaan Operator | (Atau)

Diketahui sebelumnya variabel bernama **NILAI** menyimpan 10 nilai, yakni 100, 95, 70, 80, 25, 60, 80, 60, 55, 90. Misalkan akan ditampilkan nilai dari variabel **NILAI** dengan syarat lebih besar 80 atau lebih kecil 30, yakni 100, 95, 25, 90. Berikut disajikan kode R.

```
NILAI[NILAI>80 | NILAI<30]
```

Ilustrasi dalam R diperlihatkan pada Gambar 2.28 dan Gambar 2.29.



Gambar 2.28

```

NILAI=c(100, 95, 70, 80, 25, 60, 80, 60, 55, 90)
NILAI

## [1] 100 95 70 80 25 60 80 60 55 90

NILAI[NILAI>80 | NILAI<30]

## [1] 100 95 25 90

```

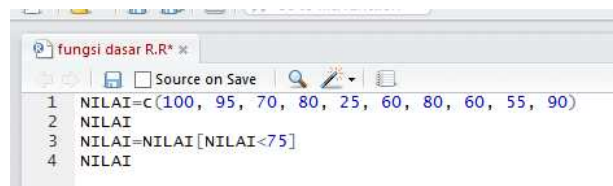
Gambar 2.29

Lebih Lanjut Penggunaan Operator < (Lebih Kecil Dari)

Diketahui variabel bernama **NILAI** menyimpan 10 nilai, yakni 100, 95, 70, 80, 25, 60, 80, 60, 55, 90. Misalkan nilai-nilai yang lebih besar atau sama dengan 75 akan dihapus dari variabel **NILAI**, sehingga nilai-nilai yang tersimpan dalam variabel **NILAI** adalah 70, 25, 60, 60, 55. Berikut disajikan kode R.

```
NILAI=NILAI[NILAI<75]
```

Ilustrasi dalam R diperlihatkan pada Gambar 2.30 dan Gambar 2.31.



```

fungsi dasar R.R* *
Source on Save
1 NILAI=c(100, 95, 70, 80, 25, 60, 80, 60, 55, 90)
2 NILAI
3 NILAI=NILAI[NILAI<75]
4 NILAI

```

Gambar 2.30

```

NILAI=c(100, 95, 70, 80, 25, 60, 80, 60, 55, 90)
NILAI

## [1] 100 95 70 80 25 60 80 60 55 90

NILAI=NILAI[NILAI<75]
NILAI

## [1] 70 25 60 60 55

```

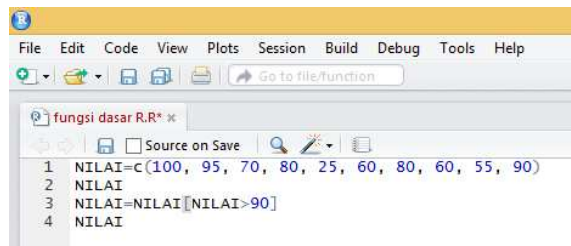
Gambar 2.31

Lebih Lanjut Penggunaan Operator > (Lebih Besar Dari)

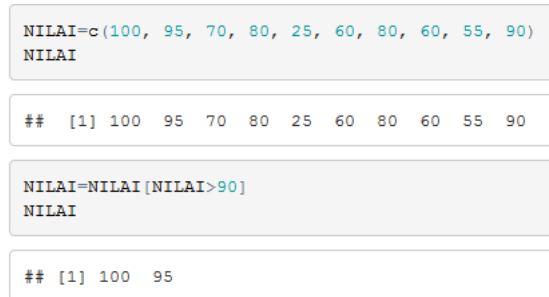
Diketahui variabel bernama **NILAI** menyimpan 10 nilai, yakni 100, 95, 70, 80, 25, 60, 80, 60, 55, 90. Misalkan nilai-nilai yang lebih kecil atau sama dengan 90 akan dihapus dari variabel **NILAI**, sehingga nilai-nilai yang tersimpan dalam variabel **NILAI** adalah 100 dan 95. Berikut kode dalam R.

```
NILAI=NILAI[NILAI>90]
```

Ilustrasi dalam R diperlihatkan pada Gambar 2.32 dan Gambar 2.33.



Gambar 2.32



Gambar 2.33

Contoh Sederhana Penggunaan dari Bahasa Pemrograman R

Misalkan suatu variabel bernama **NILAI** menyimpan 10 nilai, yakni 100, 95, 70, 80, 25, 60, 80, 60, 55, 90. Misalkan setiap nilai yang ada dalam variabel **NILAI**, yang **lebih kecil dari 65, ditambah dengan 10**. Nilai-nilai yang lebih kecil dari 65 adalah 25, 60, 60, 55. Nilai-nilai tersebut ditambah dengan 10.

$$\begin{aligned}
 25+10 &= 35 \\
 60+10 &= 70 \\
 60+10 &= 70 \\
 55+10 &= 65
 \end{aligned}$$

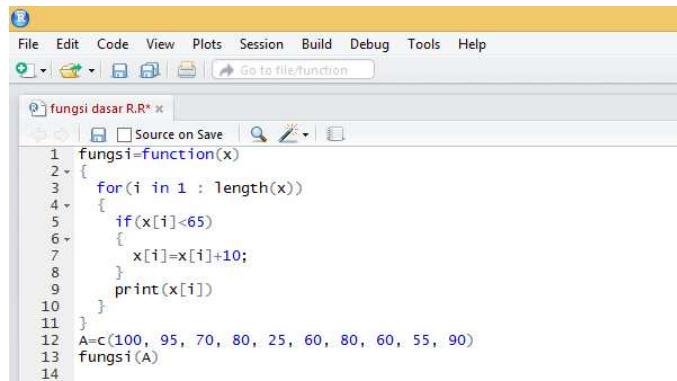
Maka nilai-nilai yang akan ditampilkan adalah 100, 95, 70, 80, 35, 70, 80, 70, 65, 90. Berikut merupakan contoh kode program dalam R.

```

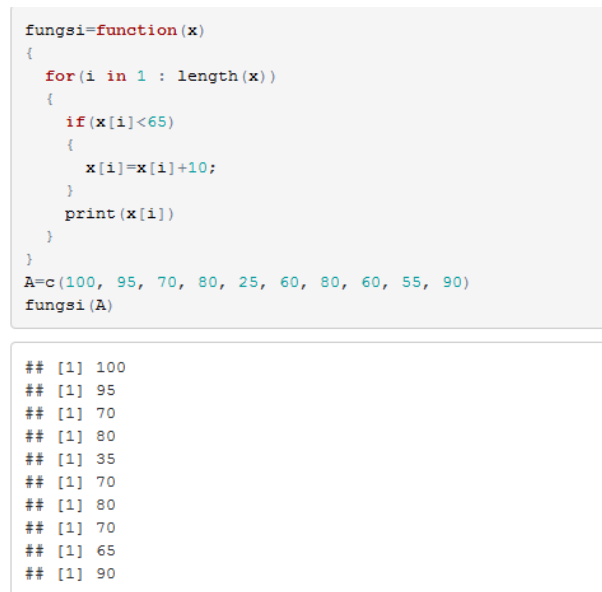
fungsi=function(x)
{
  for(i in 1 : length(x))
  {
    if(x[i]<65)
    {
      x[i]=x[i]+10;
    }
    print(x[i])
  }
}
A=c(100, 95, 70, 80, 25, 60, 80, 60, 55, 90)
fungsi(A)

```

Ilustrasi dalam R diperlihatkan pada Gambar 2.34 dan Gambar 2.35.



Gambar 2.34



Gambar 2.35

Jenis Data R

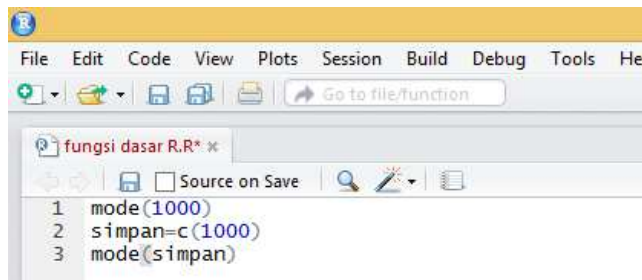
Berikut merupakan berbagai jenis dari jenis data dalam R.

- *Numeric* atau angka
- *Character* atau karakter
- *Logical* atau logika
- *Function* atau fungsi

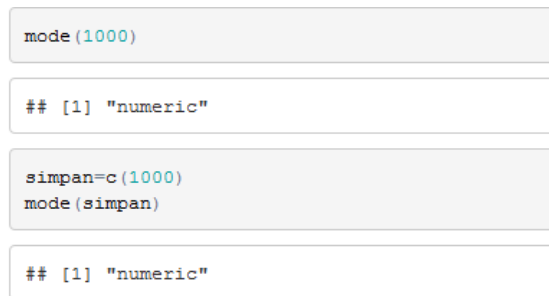
Data yang berupa angka merupakan data *numeric*. Sebagai contoh 1, 100, 1500, 3000, dan seterusnya, merupakan data berjenis numerik. Untuk mengetahui jenis dari suatu data dalam R, digunakan perintah **mode**. Berikut merupakan kode R untuk mengetahui bahwa data 1000 termasuk ke dalam data berjenis *numeric*.

mode(1000)

Ilustrasi dalam R diperlihatkan pada Gambar 2.36 dan Gambar 2.37.



Gambar 2.36

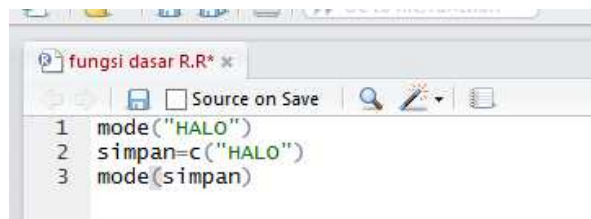


Gambar 2.37

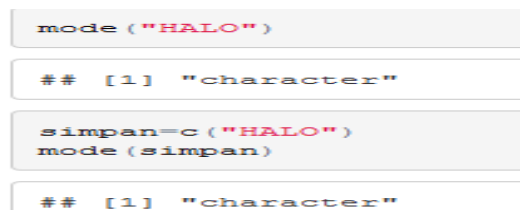
Data yang berupa karakter diapit dengan tanda petik ganda “ ”. Sebagai contoh “Halo”, “A”, “1”, “500”, “+”, dan seterusnya merupakan data berjenis karakter. Berikut merupakan perintah dalam R untuk mengetahui bahwa data “HALO” termasuk ke dalam tipe data karakter.

mode(“HALO”)

Ilustrasi dalam R diperlihatkan pada Gambar 2.38 dan Gambar 2.39.



Gambar 2.38



Gambar 2.39

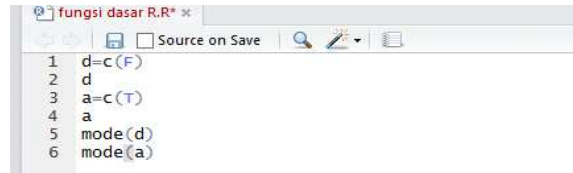
Nilai dari data yang berjenis logika terdiri dari 2 nilai, yakni T atau F. T merupakan singkatan dari *True*, yang berarti benar, sementara F merupakan singkatan dari *False*, yang berarti salah. Misalkan sebuah variabel bernama **d** menyimpan sebuah data berjenis logika, yakni F.

d=(F)

Berikut merupakan kode dalam R untuk mengetahui bahwa nilai yang tersimpan dalam variabel **d** berjenis logika.

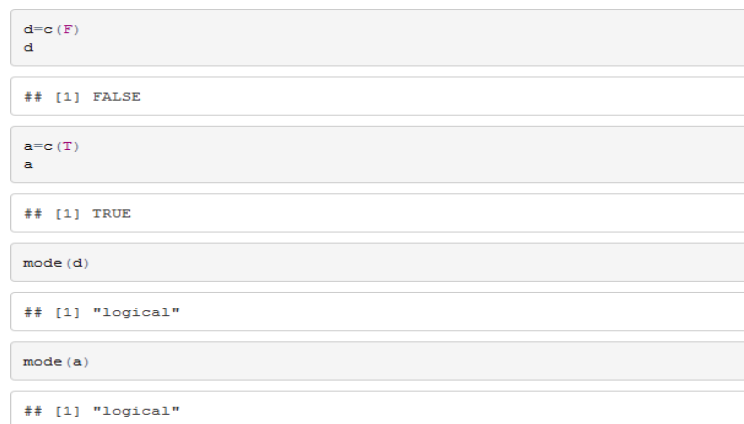
mode(d)

Ilustrasi dalam R diperlihatkan pada Gambar 2.40 dan Gambar 2.41.



```
fungsi dasar R.R* x
1 d=c(F)
2 d
3 a=c(T)
4 a
5 mode(d)
6 mode(a)
```

Gambar 2.40



```
d=c(F)
d
## [1] FALSE

a=c(T)
a
## [1] TRUE

mode(d)
## [1] "logical"

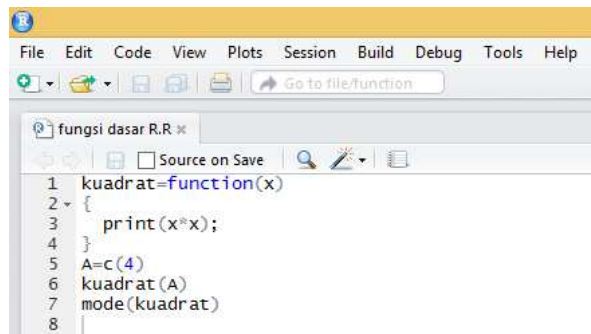
mode(a)
## [1] "logical"
```

Gambar 2.41

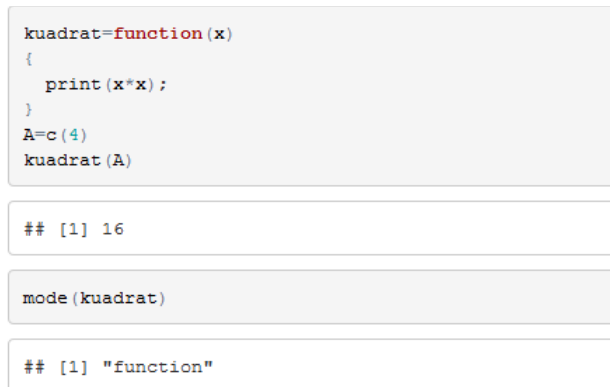
Data yang berjenis fungsi mempunyai ciri menggunakan kata **function**. Berikut merupakan contoh kode program R pembuatan fungsi kuadrat.

```
kuadrat=function(x)
{
  print(x*x);
}
A=c(4)
kuadrat(A)
mode(kuadrat)
```

Perhatikan bahwa **kuadrat** merupakan nama variabel yang menyimpan data berjenis fungsi. Ilustrasi dalam R diperlihatkan pada Gambar 2.42 dan Gambar 2.43.



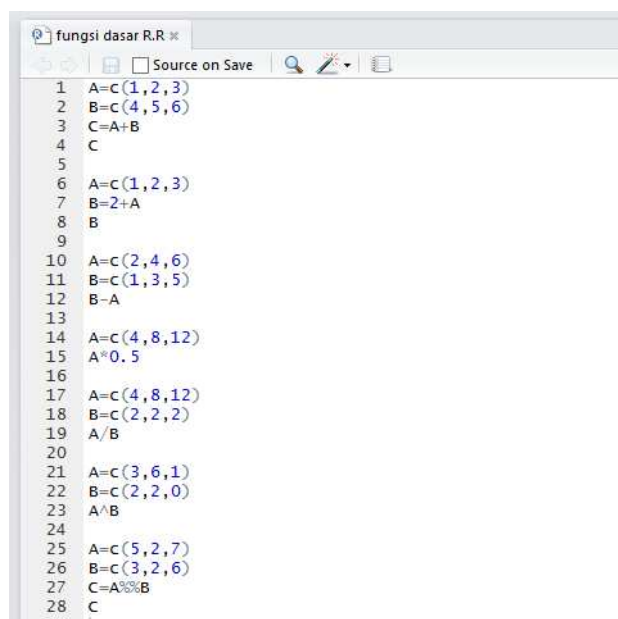
Gambar 2.42



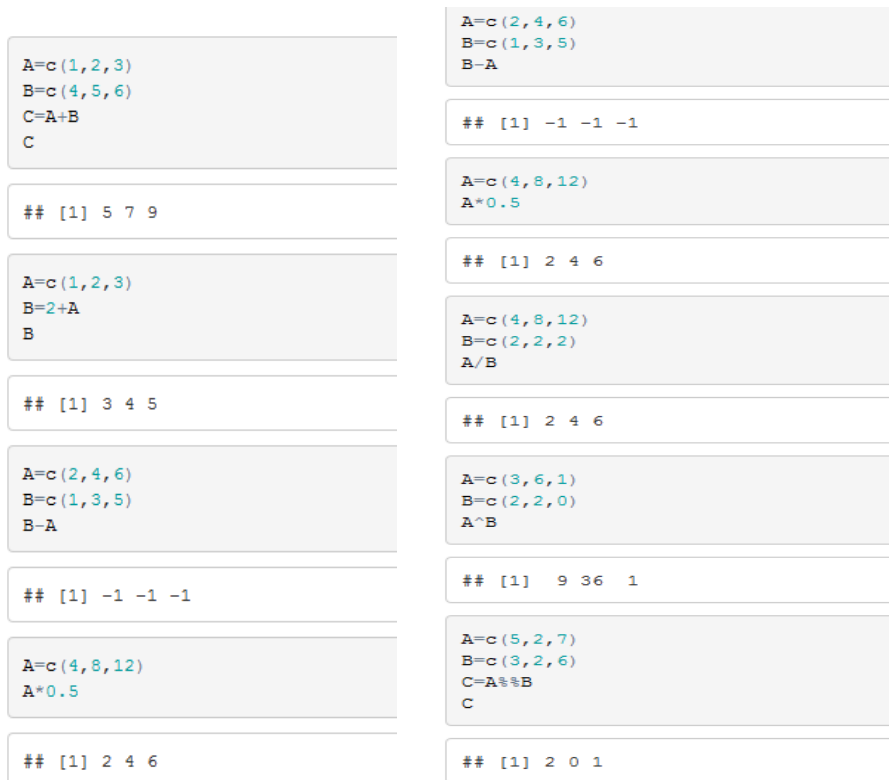
Gambar 2.43

*Operator Penjumlahan +, Pengurangan -, Perkalian *, Pembagian /, Pangkat ^, Sisa %%*

Gambar 2.44 dan Gambar 2.45 merupakan berbagai contoh kode R yang melibatkan penggunaan operator matematika.



Gambar 2.44



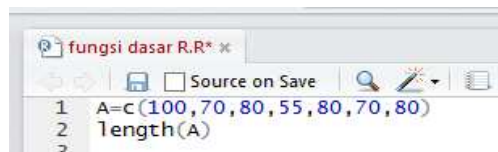
Gambar 2.45

Fungsi length

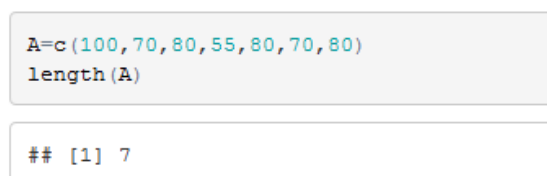
Fungsi **length** dalam R berfungsi untuk mengetahui jumlah elemen yang tersimpan atau terkandung dalam variabel. Misalkan suatu variabel bernama A menyimpan nilai 100, 70, 80, 55, 80, 70, 80. Maka banyaknya elemen dalam variabel A adalah 7. Berikut merupakan kode R untuk menentukan banyaknya elemen yang terkandung dalam variabel A.

```
A=c(100,70,80,55,80,70,80)
length(A)
```

Ilustrasi dalam R diperlihatkan pada Gambar 2.46 dan Gambar 2.47.



Gambar 2.46



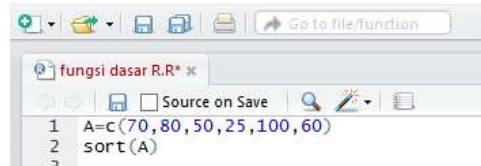
Gambar 2.47

Fungsi sort

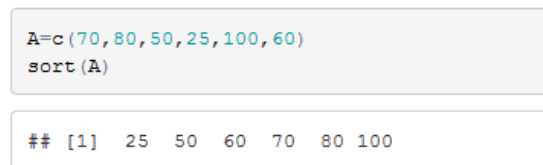
Fungsi **sort** dalam R berfungsi untuk mengurutkan data. Misalkan suatu variabel bernama A menyimpan nilai 70, 80, 50, 25, 100, 60. Berikut merupakan kode R untuk mengurutkan elemen-elemen atau nilai-nilai yang terkandung dalam variabel A.

```
A=c(70,80,50,25,100,60)
sort(A)
```

Ilustrasi dalam R diperlihatkan pada Gambar 2.48 dan Gambar 2.49.



Gambar 2.48



Gambar 2.49

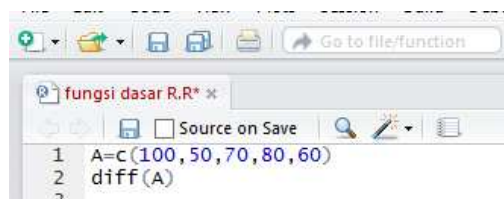
Fungsi diff

Misalkan diberikan data sebagai berikut. 100, 50, 70, 80, 60. Misalkan dilakukan perhitungan sebagai berikut.

- $50 - 100 = -50$
- $70 - 50 = 20$
- $80 - 70 = 10$
- $60 - 80 = -20$

Sehingga hasil akhirnya adalah $-50, 20, 10, -20$. Gambar 2.50 dan Gambar 2.51 merupakan penggunaan fungsi **diff** untuk mengilustrasikan contoh tersebut.

```
A=c(100,50,70,80,60)
diff(A)
```



Gambar 2.50

```
A=c(100,50,70,80,60)
diff(A)

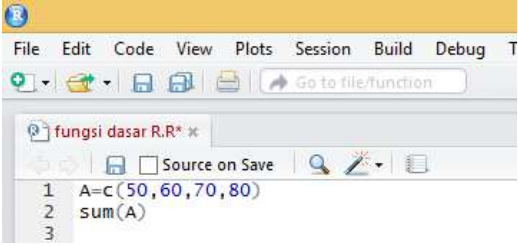
## [1] -50 20 10 -20
```

Gambar 2.51

Fungsi sum

Fungsi **sum** dalam R berfungsi untuk menjumlahkan seluruh nilai data. Misalkan variabel A menyimpan nilai 50, 60, 70, 80. Maka jumlah dari seluruh nilai dalam variabel A adalah 260.

```
A=c(50,60,70,80)
sum(A)
```



Gambar 2.52

```
A=c(50,60,70,80)
sum(A)

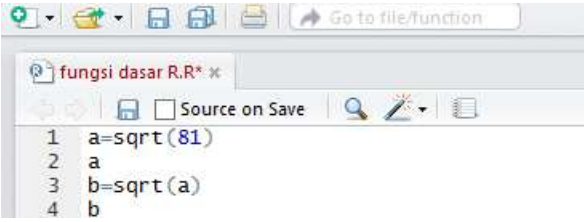
## [1] 260
```

Gambar 2.53

Fungsi sqrt

Fungsi **sqrt** dalam R berfungsi untuk menghitung nilai akar pangkat dua dari suatu bilangan. Sebagai contoh akar pangkat 2 dari 81 adalah 9, yakni $\sqrt[2]{81} = \sqrt{81} = 9$. Berikut merupakan kode R untuk menghitung nilai akar pangkat dua dari 81.

```
sqrt(81)
```



Gambar 2.54

```

a=sqrt(81)
a

## [1] 9

b=sqrt(a)
b

## [1] 3

```

Gambar 2.55

Fungsi max dan min

Fungsi **max** dalam R berfungsi untuk menentukan nilai maksimum dalam data. Misalkan diberikan data 10,25,90,75, 95, 57. Maka nilai maksimum dari data tersebut adalah 95. Berikut merupakan kode dalam R untuk menentukan nilai maksimum dari data tersebut.

```

A=c(10,25,90,75,95,57)
max(A)

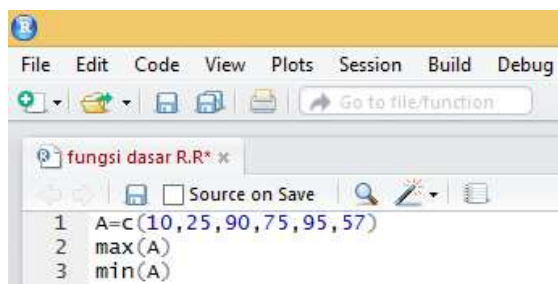
```

Fungsi **min** dalam R berfungsi untuk menentukan nilai minimum dalam data. Misalkan diberikan data 10,25,90,75, 95, 57. Maka nilai minimum dari data tersebut adalah 10. Berikut merupakan kode dalam R untuk menentukan nilai minimum dari data tersebut.

```

A=c(10,25,90,75,95,57)
min(A)

```



Gambar 2.56

```

A=c(10,25,90,75,95,57)
max(A)

## [1] 95

min(A)

## [1] 10

```

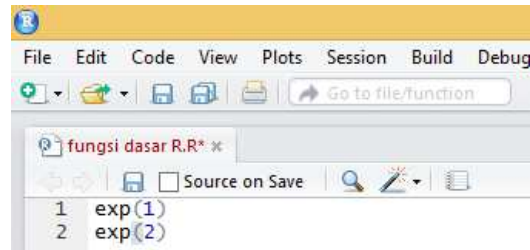
Gambar 2.57

Fungsi *exp*

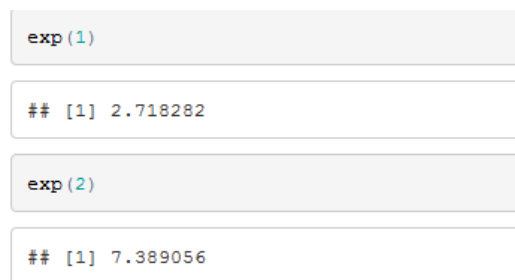
Exp merupakan singkatan dari *exponential* atau eksponensial. Nilai dari eksponensial adalah 2,71828182845...

$$\begin{aligned} \exp &= 2,71828182845 \\ \exp^1 &= 2,71828182845^1 = 2,71828182845 \\ \exp^2 &= 2,71828182845^2 = 7,389056096 \end{aligned}$$

Berikut diberikan contoh penggunaan fungsi **exp** dalam R (Gambar 2.58 dan Gambar 2.59).



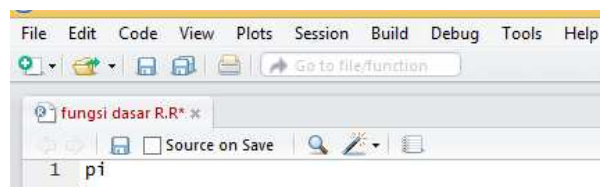
Gambar 2.58



Gambar 2.59

Fungsi *pi* atau π

Pi atau π bernilai 3,141593 ... Berikut diberikan contoh penggunaan fungsi **pi** dalam R (Gambar 2.60 dan Gambar 2.61).



Gambar 2.60

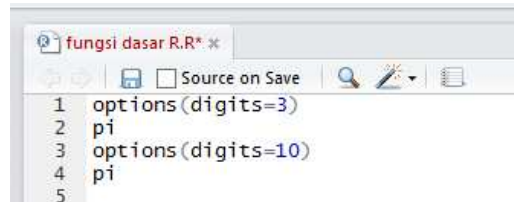


Gambar 2.61

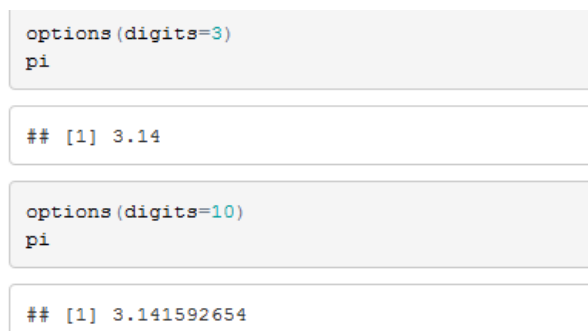
Fungsi options

Diketahui nilai pi adalah 3.141593... Misalkan hanya ingin ditampilkan 3 digit angka dari nilai pi, yakni 3.14. Berikut perintah dalam R untuk menampilkan hanya 3 digit angka dari bilangan pi.

```
options(digits=3)  
pi
```



Gambar 2.62



Gambar 2.63

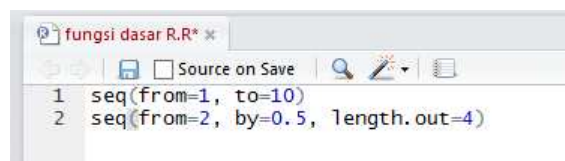
Fungsi seq

Misalkan ingin ditampilkan nilai data dari 1 sampai 10. Berikut perintah dalam R untuk menampilkan nilai data dari 1 sampai 10.

```
seq(from=1, to=10)
```

Misalkan ingin ditampilkan 4 buah nilai, dimulai dari 2 kemudian 2.5, 3, dan 3.5, dimana jaraknya adalah 0.5. Berikut merupakan perintah dalam R.

```
seq(from=2, by=0.5, length.out=4)
```



Gambar 2.64

```
seq(from=1, to=10)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

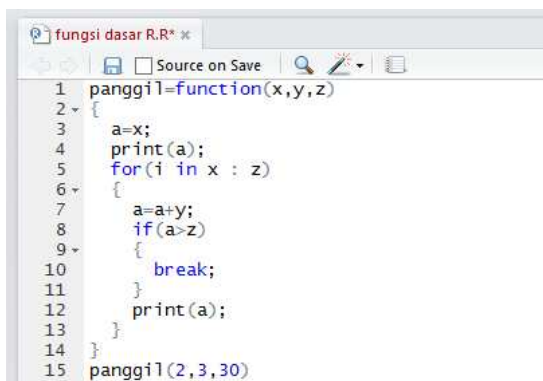
```
seq(from=2, by=0.5, length.out=4)
```

```
## [1] 2.0 2.5 3.0 3.5
```

Gambar 2.65

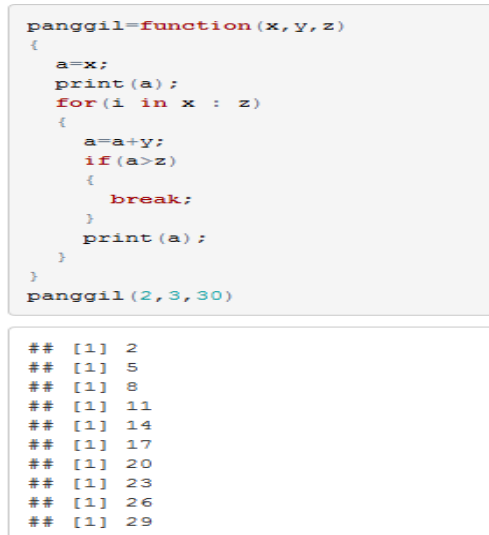
Misalkan ingin ditampilkan nilai-nilai kelipatan 3, dimulai dari angka 2 sampai 30. Adapun nilai-nilai tersebut adalah 2, 5, 8, 11, 14, 17, 20, 23, 26, 29. Berikut diberikan contoh kode program R untuk menyelesaikan permasalahan tersebut.

```
panggil=function(x,y,z)
{
  a=x;
  print(a);
  for(i in x : z)
  {
    a=a+y;
    if(a>z)
    {
      break;
    }
    print(a);
  }
}
panggil(2,3,30)
```



```
fungsi dasar R.R* x
1 panggil=function(x,y,z)
2 {
3   a=x;
4   print(a);
5   for(i in x : z)
6   {
7     a=a+y;
8     if(a>z)
9     {
10      break;
11    }
12    print(a);
13  }
14 }
15 panggil(2,3,30)
```

Gambar 2.66

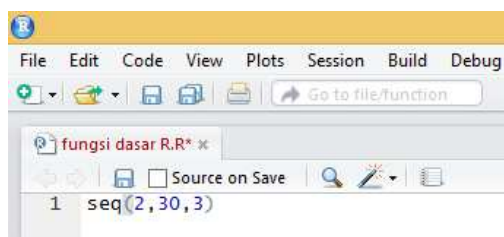


```
panggil=function(x, y, z)
{
  a=x;
  print(a);
  for(i in x : z)
  {
    a=a+y;
    if(a>z)
    {
      break;
    }
    print(a);
  }
}
panggil(2,3,30)

## [1] 2
## [1] 5
## [1] 8
## [1] 11
## [1] 14
## [1] 17
## [1] 20
## [1] 23
## [1] 26
## [1] 29
```

Gambar 2.67

Cara lain adalah sebagai berikut.



Gambar 2.68

```
seq(2, 30, 3)

## [1] 2 5 8 11 14 17 20 23 26 29
```

Gambar 2.69

Fungsi table

Fungsi **table** dalam R berfungsi untuk menyajikan data dalam bentuk tampilan tabel. Misalkan suatu variabel bernama A menyimpan data 10, 10, 30, 10, 30, 10, 10, 40, 40, 70, 90, 70, 80, 60, 60, 90. Berikut merupakan perintah atau kode R untuk menyajikan data pada variabel A dalam tabel.

```
A=c(10, 10, 30, 10, 30, 10, 10, 40, 40, 70, 90, 70, 80, 60, 60, 90)
table(A)
```

Penyajian secara tabel juga dapat disajikan dengan menampilkan informasi proporsi. Berikut merupakan perintah atau kode R untuk menyajikan tabel dengan informasi proporsi.

```
A=c(10, 10, 30, 10, 30, 10, 10, 40, 40, 70, 90, 70, 80, 60, 60, 90)
table(A)/length(A)
```

Contoh lain misalkan suatu survey yang dilakukan terhadap 10 orang sebagai sampel untuk melihat apakah seseorang tersebut terkena insomnia (ya) atau tidak (tidak). Hasil atau data disajikan sebagai berikut.

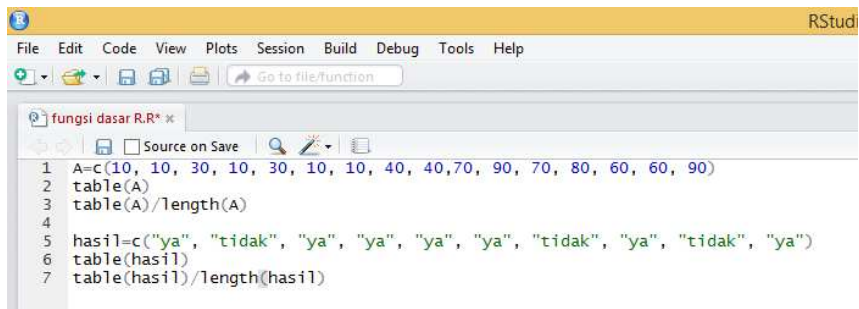
```
ya, tidak, ya, ya, ya, ya, tidak, ya, tidak, ya
```

Berikut merupakan perintah atau kode R untuk menyajikan data di atas dalam tabel.

```
hasil=c("ya", "tidak", "ya", "ya", "ya", "ya", "tidak", "ya", "tidak", "ya")
table(hasil)
```

Penyajian secara tabel juga dapat disajikan dengan menampilkan informasi proporsi. Berikut merupakan perintah atau kode R untuk menyajikan tabel dengan informasi proporsi.

```
hasil=c("ya", "tidak", "ya", "ya", "ya", "ya", "tidak", "ya", "tidak", "ya")
table(hasil)/length(hasil)
```

Gambar 2.70

```

A=c(10, 10, 30, 10, 30, 10, 10, 40, 40,70, 90, 70, 80, 60, 60, 90)
table(A)

## A
## 10 30 40 60 70 80 90
## 5 2 2 2 2 1 2

table(A)/length(A)

## A
## 10 30 40 60 70 80 90
## 0.3125 0.1250 0.1250 0.1250 0.1250 0.0625 0.1250

hasil=c("ya", "tidak", "ya", "ya", "ya", "ya", "tidak", "ya", "tidak", "ya")
table(hasil)

## hasil
## tidak ya
## 3 7

table(hasil)/length(hasil)

## hasil
## tidak ya
## 0.3 0.7

```

Gambar 2.71

Fungsi factor

Fungsi **factor** dalam R berfungsi untuk mengetahui keragaman level atau faktor dalam suatu data. Misalkan diberikan data sebagai berikut.

ikan, ikan, udang, ikan, udang, ikan, ikan, udang

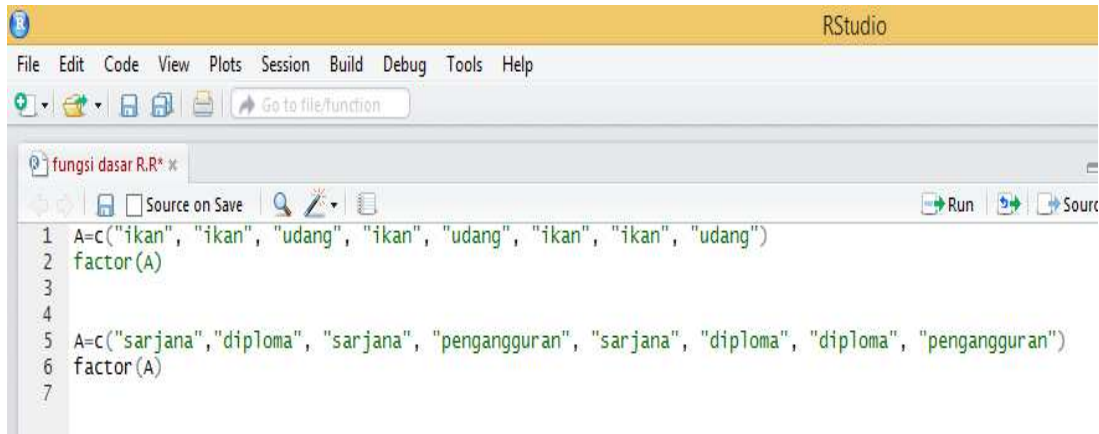
Berdasarkan data tersebut, terdapat dua faktor, yakni ikan dan udang. Misalkan diberikan data sebagai berikut.

**sarjana,diploma, sarjana, pengangguran, sarjana, diploma, diploma,
pengangguran**

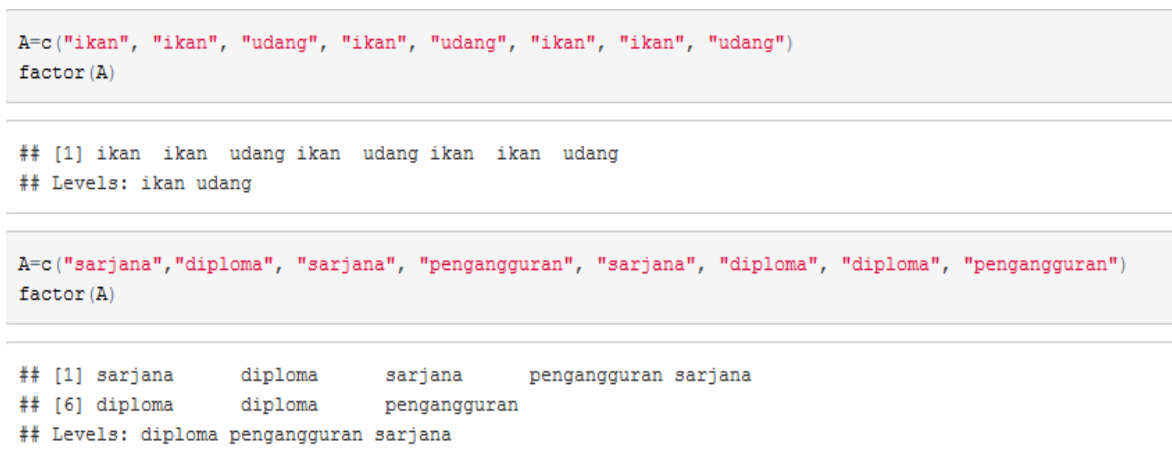
Berdasarkan data tersebut, terdapat tiga faktor, yakni sarjana, diploma, dan pengangguran. Berikut merupakan perintah atau kode R dalam penggunaan fungsi **factor()**.

```
A=c("ikan", "ikan", "udang", "ikan", "udang", "ikan", "ikan", "udang")
factor(A)
```

```
A=c("sarjana", "diploma", "sarjana", "pengangguran", "sarjana", "diploma",
"diploma", "pengangguran")
factor(A)
```



Gambar 2.72

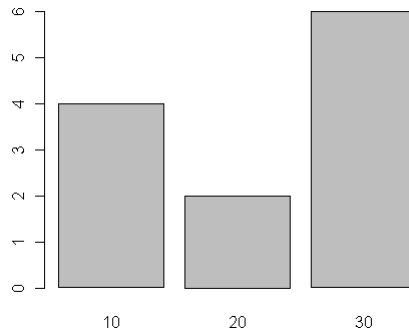


Gambar 2.73

Fungsi barplot

Fungsi **barplot** dalam R berfungsi untuk menyajikan data dalam bentuk diagram batang. Misalkan variabel A menyimpan data 10, 10, 10, 10, 20, 20, 30, 30, 30, 30, 30, 30. Berikut akan disajikan data pada variabel A dalam bentuk diagram batang.

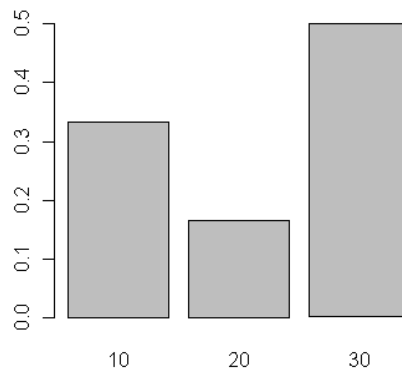
```
A=c(10, 10, 10, 10, 20, 20, 30, 30, 30, 30, 30, 30)
barplot(table(A))
```



Gambar 2.74

Perhatikan bahwa untuk data dengan nilai 10 mempunyai frekuensi sebanyak 4, data dengan nilai 20 mempunyai frekuensi sebanyak 2, dan data dengan nilai 30 mempunyai frekuensi sebanyak 6. Grafik batang di atas dapat diatur agar disajikan secara proporsi.

```
A=c(10, 10, 10, 10, 20, 20, 30, 30, 30, 30, 30, 30)  
barplot(table(A)/length(A))
```



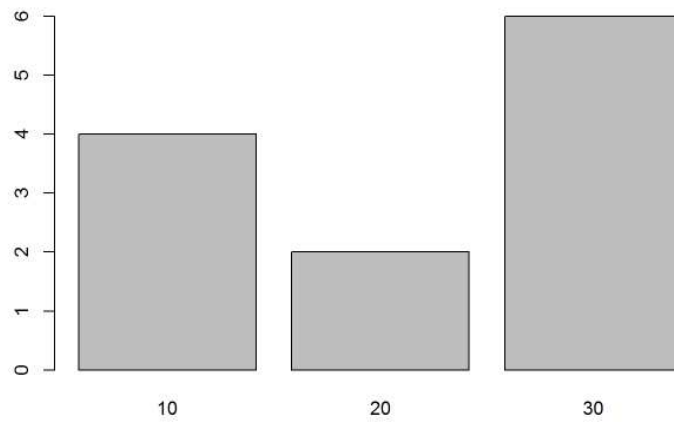
Gambar 2.75

Perhatikan bahwa nilai 0,3, 0,2, dan 0,5 masing-masing merupakan proporsi dari nilai 10, 20, dan 30.

```
fungsi dasar R.R* x
Source on Save
1 A=c(10, 10, 10, 10, 20, 20, 30, 30, 30, 30, 30, 30)
2
3 barplot(table(A))
4
5 barplot(table(A)/length(A))
6
```

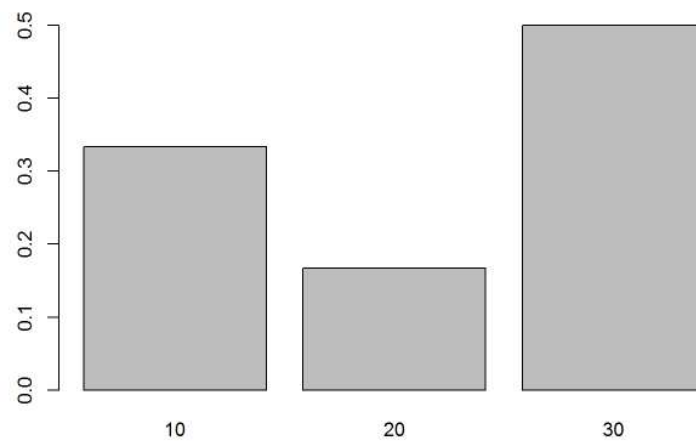
Gambar 2.76

```
A=c(10, 10, 10, 10, 10, 20, 20, 20, 30, 30, 30, 30, 30, 30)
barplot(table(A))
```



Gambar 2.77

```
barplot(table(A)/length(A))
```



Gambar 2.78

Fungsi plot

Misalkan variabel bernama A menyimpan data 10,10,10,10,10,20,20,20,30,30,40. Berikut akan digunakan fungsi **table** untuk mengetahui frekuensi dari masing-masing nilai data.

```
A=c(10,10,10,10,10,20,20,20,30,30,40)
table(A)
```

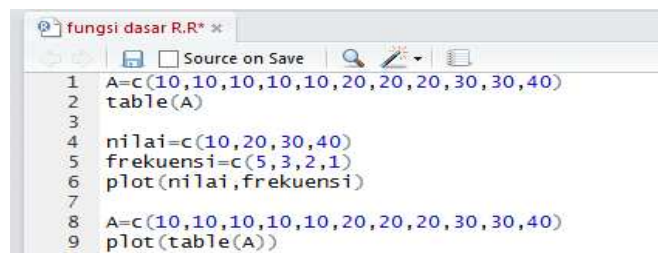
Diketahui nilai 10 muncul sebanyak 5, nilai 20 sebanyak 3, nilai 30 sebanyak 2, dan nilai 40 sebanyak 1. Berikut akan digunakan fungsi `plot()` untuk memplot data yang tersimpan dalam variabel A.

```
nilai=c(10,20,30,40)
frekuensi=c(5,3,2,1)
plot(nilai,frekuensi)
```

Alternatif lain untuk menyajikan data.

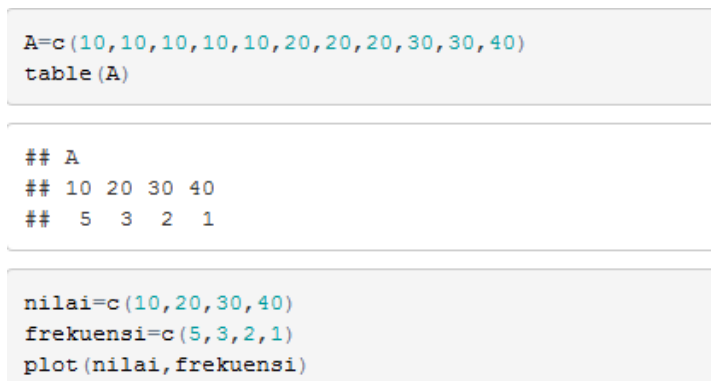
```
A=c(10,10,10,10,10,20,20,20,30,30,40)
plot(table(A))
```

Ilustrasi dalam R diperlihatkan pada Gambar 2.79 dan Gambar 2.82.



```
fungsi dasar R.R* *
Source on Save
1 A=c(10,10,10,10,10,20,20,20,30,30,40)
2 table(A)
3
4 nilai=c(10,20,30,40)
5 frekuensi=c(5,3,2,1)
6 plot(nilai,frekuensi)
7
8 A=c(10,10,10,10,10,20,20,20,30,30,40)
9 plot(table(A))
```

Gambar 2.79

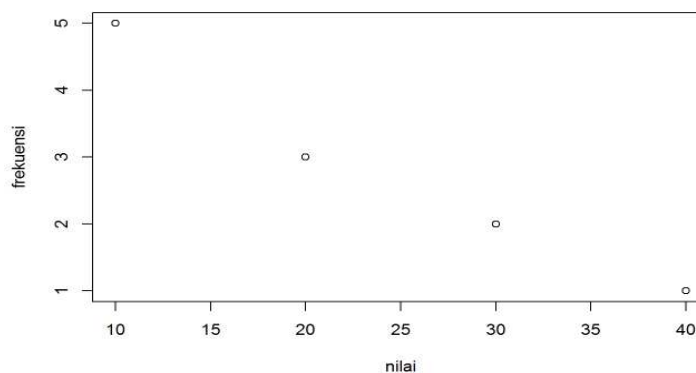


```
A=c(10,10,10,10,10,20,20,20,30,30,40)
table(A)

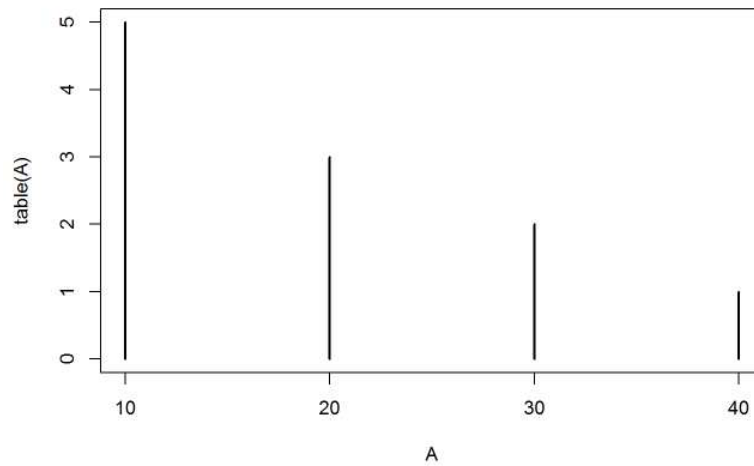
## A
## 10 20 30 40
## 5  3  2  1

nilai=c(10,20,30,40)
frekuensi=c(5,3,2,1)
plot(nilai,frekuensi)
```

Gambar 2.80



Gambar 2.81



Gambar 2.82

Referensi

1. Gio, P.U. dan E. Rosmaini, 2015. Belajar Olah Data dengan SPSS, Minitab, R, Microsoft Excel, EViews, LISREL, AMOS, dan SmartPLS. USUpres.
2. <http://www.statmethods.net/graphs/bar.html>
3. <http://www.r-tutor.com/elementary-statistics/qualitative-data/bar-graph>
4. <http://www.r-bloggers.com/using-r-barplot-with-ggplot2/>
5. <http://www.statmethods.net/graphs/line.html>
6. <http://www.statmethods.net/management/functions.html>
7. <http://www.r-bloggers.com/basic-mathematical-functions/>
8. <http://ww2.coastal.edu/kingw/statistics/R-tutorials/arithmetic.html>