

# Pengantar Komputasi Numerik dengan R

Saifuddin Arief

[Saifuddin.Arief@rocketmail.com](mailto:Saifuddin.Arief@rocketmail.com)

## Lisensi Dokumen:

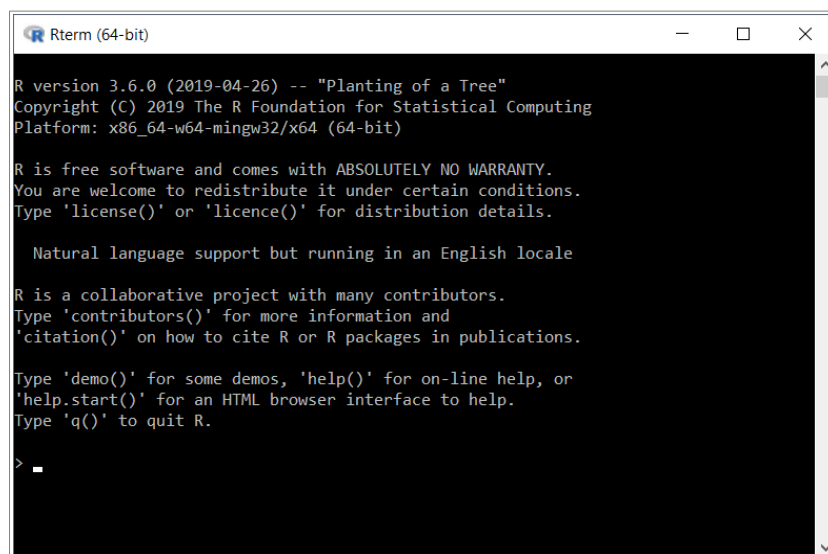
Copyright © 2003-2020 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

R adalah bahasa pemrograman tingkat tinggi serta lingkungan komputasi yang dikembangkan untuk komputasi numerik, statistik dan visualisasi grafik. R mempunyai sintak yang elegan dan intuitif. File instalasi dan dokumentasi untuk R dapat diperoleh melalui situs <https://cran.r-project.org/>.

## Interaksi dengan R

Pada sistem operasi Windows, R dapat dijalankan melalui terminal, antarmuka RGui atau RStudio. Pada sistem operasi Linux, R dapat dijalankan melalui terminal atau Rstudio. RStudio adalah program lain yang terpisah dan harus diinstal secara tersendiri.



```
Rterm (64-bit)
R version 3.6.0 (2019-04-26) -- "Planting of a Tree"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

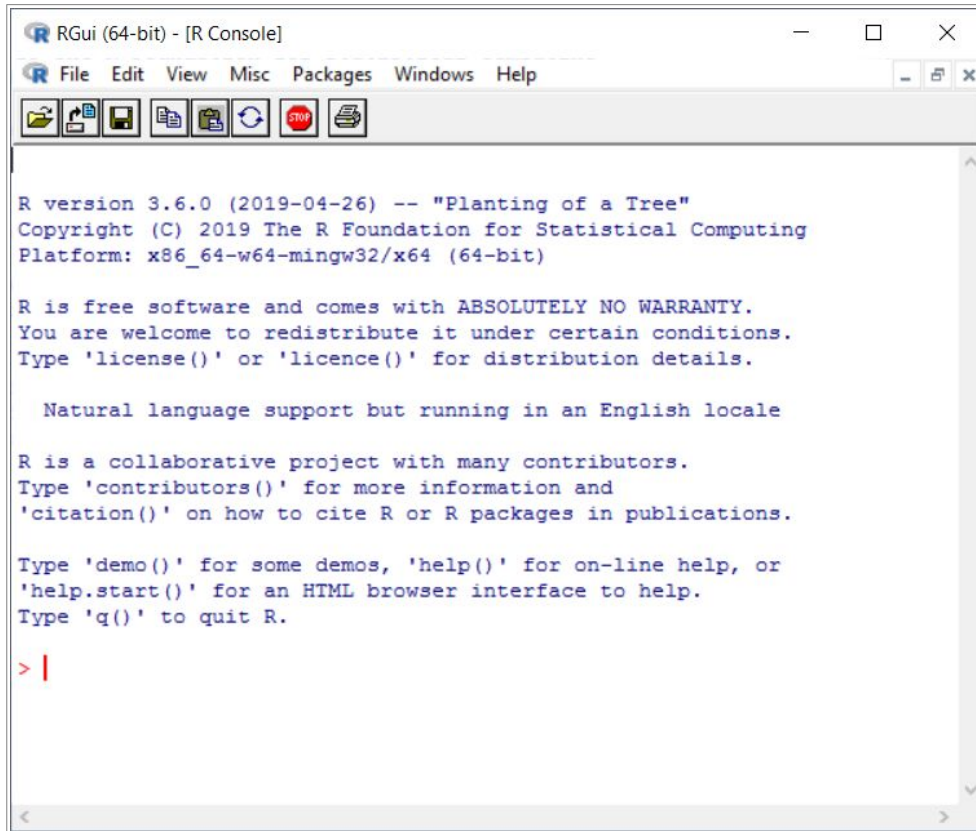
  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

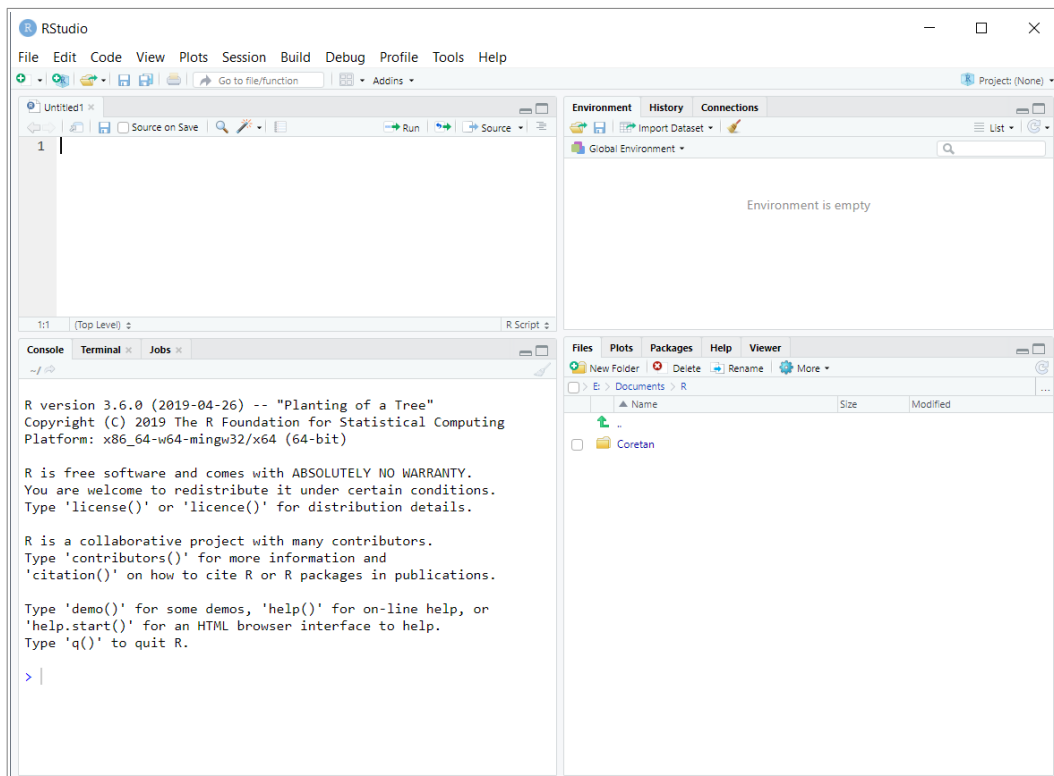
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> _
```

Gambar 1. R pada terminal (Rterm)



Gambar 2. RGui



Gambar 3. Rstudio

Simbol `>` yang terdapat pada jendela Rterm, RGui atau RStudio adalah tempat untuk memasukkan suatu perintah. Sebagai contoh, untuk melakukan perhitungan  $1.23 + 4.56$  maka ekspresi matematika tersebut diketikkan setelah simbol `>` kemudian menekan tombol `enter` untuk menjalankan perintahnya. R akan menampilkan hasil perhitungannya pada baris berikutnya.

```
> 1.23 + 4.56
[1] 5.79
>
```

Simbol `>` yang muncul kembali menunjukkan bahwa R siap untuk mengerjakan perintah berikutnya.

R dilengkapi dengan sistem bantuan dan dokumentasi yang sangat baik. Sistem bantuan dapat diakses pada menu `help` yang terdapat pada jendela RGui atau RStudio. Sejumlah dokumentasi tentang penggunaan R dapat diunduh melalui situs R.

Perintah `help` dapat digunakan untuk mendapatkan penjelasan mengenai suatu perintah, fungsi atau operator tertentu. Sebagai contoh, penjelasan tentang fungsi `abs` dapat diperoleh dengan menjalankan perintah `help(abs)`.

Untuk keluar dari program R dapat dilakukan dengan menggunakan perintah `q()`, melalui menu `File - Exit` yang terdapat pada antarmuka RGui, atau melalui menu `File - Quit Session` yang terdapat pada menu Rstudio. atas dari jendela R.

## Operator, Fungsi Matematika dan Variabel

Operator untuk perhitungan aritmatika yaitu `+`, `-`, `*`, `/` dan `^`. Simbol tersebut masing-masing melambangkan operasi penjumlahan, pengurangan, perkalian, pembagian dan pemangkatan. Di dalam R juga terpasang fungsi-fungsi matematika, seperti `sqrt`, `abs`, `exp`, `sin`, `cos`, `tan` dan lain sebagainya.

```
> 1010*(sqrt(9.81^2 + 7^2)*((0.07 + 0.0214)*cos(35.5/180*pi)))
[1] 905.7136
```

Nilai suatu obyek atau hasil perhitungan dapat disimpan ke dalam suatu variabel dengan statemen sebagai berikut:

```
x = ekspresi
x <- ekspresi
```

dimana `x` adalah nama variabel. Simbol `=` atau `<-` merupakan operator penugasan. Dalam penggunaan variabel, huruf kecil dan huruf besar adalah berbeda. Untuk menampilkan nilai yang tersimpan pada suatu variabel dilakukan dengan memanggil nama variabel tersebut.

Di dalam R terdapat variabel khusus `pi` yang menyatakan konstanta matematika  $\pi = 3.1415927$ .

Untuk memperjelas perintah-perintah yang dibuat, dapat ditambahkan suatu baris komentar. Baris komentar dapat ditulis sebagai suatu baris tersendiri atau ditulis di belakang suatu statemen. Kumpulan karakter yang terletak setelah tanda `#` akan dianggap sebagai baris komentar.

```
> luas = 12.5*8; luas
[1] 100
> r = 10;           # radius lingkaran
> A = pi*r^2       # luas lingkaran
> A
[1] 314.1593
```

## Bilangan Kompleks

Bilangan kompleks  $z = x + iy$  dapat dinyatakan dengan  $z = x + yi$  dimana  $i$  adalah variabel khusus yang merepresentasikan konstanta  $\sqrt{-1}$ .

Operasi aritmatika terhadap bilangan kompleks dapat dilakukan dengan menggunakan notasi yang sama dengan notasi pada bilangan real.

```
> z1 = 6 - 8i
> z2 = 3 + 1i
> p = z1/z2
> p
[1] 1-3i
> z1 = 6 - 8i
> z2 = 3 + 1i
> p = z1 + z2
> p
[1] 9-7i
> q = z1 - z2
> z1*z2
[1] 26-18i
> z2/z1
[1] 0.1+0.3i
```

Bilangan yang menyatakan komponen imajiner harus dinyatakan secara eksplisit. Sebagai contoh, statemen  $3 + 1i$  apabila dinyatakan dengan  $3 + i$  tanpa angka 1 maka akan muncul suatu pesan kesalahan, seperti di bawah ini.

```
> z3 = 3 + i
Error: object 'i' not found
```

Argumen untuk operasi yang menghasilkan bilangan kompleks juga harus dinyatakan dalam notasi bilangan kompleks, seperti contoh di bawah ini.

```
> sqrt(-100)
[1] NaN
Warning message:
In sqrt(-100) : NaNs produced
> sqrt(-100 + 0i)
[1] 0+10i
```

## Larik, Matrik dan Vektor

Salah satu kelebihan R yaitu kemampuannya dalam menangani berbagai macam operasi manipulasi terhadap data larik dengan mudah. Suatu larik dapat dibuat dengan fungsi `c`, seperti yang ditunjukkan pada contoh di bawah ini.

```
> m = c(2,1,1,6,2,1,-2,2,1)
> m
[1] 2 1 1 6 2 1 -2 2 1
```

Pembuatan data matrik dan vektor dilakukan dengan menggunakan fungsi `matrix`, seperti pada contoh berikut ini.

```
> A = matrix(c(2,1,1,6,2,1,-2,2,1), nrow=3, byrow=TRUE)
```

```
> A
      [,1] [,2] [,3]
[1,]    2    1    1
[2,]    6    2    1
[3,]   -2    2    1
> b = matrix(c(1,-1,7), nrow=3, byrow=FALSE)
> b
      [,1]
[1,]    1
[2,]   -1
[3,]    7
```

Suatu larik atau sekuen yang nilai elemen-elemennya bertambah atau berkurang sebanyak satu satuan dari suatu nilai awal  $i$  sampai nilai akhir  $j$  dapat dibuat dengan sintak  $i:j$ .

```
> i = 1:10
> i
[1] 1 2 3 4 5 6 7 8 9 10
```

Fungsi seq dapat digunakan untuk membuat suatu sekuen bilangan dengan pertambahan atau jumlah elemen tertentu, seperti yang ditunjukkan di bawah ini.

```
> n = seq(0,1,by=0.25)
> n
[1] 0.00 0.25 0.50 0.75 1.00
> m = seq(1,10,by=2)
> m
[1] 1 3 5 7 9
> x = seq(0,10,length=11)
> x
[1] 0 1 2 3 4 5 6 7 8 9 10
```

## Operasi Berbasis Vektor

Di dalam R, secara umum operasi-operasi matematika terhadap obyek matrik dan vektor dapat dilakukan dengan mudah tanpa harus menggunakan suatu perulangan.

```
> x = seq(0,pi,pi/4)
> x
[1] 0.0000000 0.7853982 1.5707963 2.3561945 3.1415927
> cos(x)
[1] 1.000000e+00 7.071068e-01 6.123032e-17 -7.071068e-01 -1.000000e+00
> u = 1:3
> u
[1] 1 2 3
> y = exp(u)
> y
[1] 2.718282 7.389056 20.085537
> log(y)
[1] 1 2 3
```

Operasi Aljabar Linier, seperti penjumlahan, pengurangan dan perkalian, juga dapat dilakukan dengan sangat mudah tanpa harus menggunakan suatu ekspresi perulangan secara eksplisit. Pada operasi aljabar argumen-argumennya harus mempunyai dimensi yang kompatibel, jika dimensinya tidak kompatibel maka operasinya tidak dapat dieksekusi dan akan muncul suatu pesan kesalahan.

Operasi penjumlahan dan pengurangan terhadap matrik dan vektor dilakukan dengan operator yang sama dengan operasi skalar yaitu + dan -. Operasi perkalian terhadap matrik dan vektor dilakukan dengan operator %\*%.

```
> X = matrix(c(9,8,5,1,3,0,2,4,6), nrow=3, byrow=TRUE)
> X
      [,1] [,2] [,3]
[1,]    9    8    5
[2,]    1    3    0
[3,]    2    4    6
> Y = matrix(c(3,2,1,4,5,6,9,8,7), nrow=3, byrow=T)
> Y
      [,1] [,2] [,3]
[1,]    3    2    1
[2,]    4    5    6
[3,]    9    8    7
> A = X + Y
> A
      [,1] [,2] [,3]
[1,]   12   10    6
[2,]    5    8    6
[3,]   11   12   13
> B = X - Y
> B
      [,1] [,2] [,3]
[1,]    6    6    4
[2,]   -3   -2   -6
[3,]   -7   -4   -1
> p = matrix(c(1,2,3,8), nrow=2, byrow=T)
> p
      [,1] [,2]
[1,]    1    2
[2,]    3    8
> Z = A + p
Error in A + p : non-conformable arrays
> j = matrix(c(6,4), nrow=2, byrow=T)
> j
      [,1]
[1,]    6
[2,]    4
> g = p %*% j
> g
      [,1]
[1,]   14
[2,]   50
```

Di dalam R dapat juga dilakukan operasi perkalian dan pembagian terhadap obyek matrik atau larik dengan melakukan operasi secara elemen dengan elemen. Notasi untuk perkalian dan pembagian secara elemen dengan elemen adalah \* dan /.

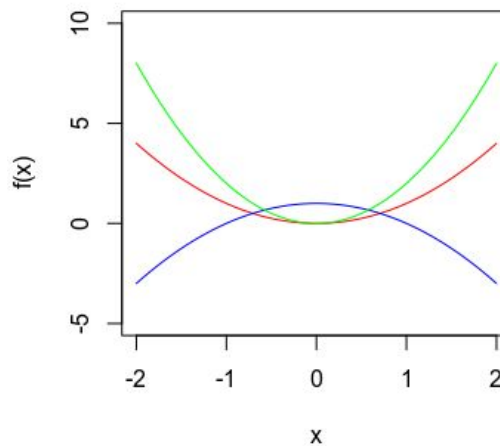
```
> x = matrix(c(1,2,3,4), nrow=2, byrow=T)
> x
      [,1] [,2]
[1,]    1    2
[2,]    3    4
```

```
> y = matrix(c(5,6,7,8), nrow=2, byrow=T)
> y
      [,1] [,2]
[1,]    5    6
[2,]    7    8
> u = x*y
> u
      [,1] [,2]
[1,]    5  12
[2,]   21  32
> z = y/x
> z
      [,1] [,2]
[1,] 5.000000    3
[2,] 2.333333    2
```

## Visualisasi Data

R dapat digunakan untuk melakukan visualisasi data, baik secara dua dimensi maupun tiga dimensi. Untuk membuat grafik dua dimensi dapat digunakan perintah `plot`, kemudian untuk grafik tiga dimensi dapat digunakan perintah `surf`. Grafik yang dihasilkan oleh perintah-perintah grafis akan ditampilkan pada jendela tersendiri, yaitu jendela grafik.

```
> % Contoh penggunaan fungsi plot (lihat Gambar 3)
> x = linspace(-1,1,61)';
> y1 = x.^2; y2 = 1 - y1; y3 = 2*y1;
> plot(x,[y1 y2 y3])
> title('Contoh penggunaan fungsi plot')
```



Gambar 4. Contoh penggunaan fungsi plot

## Perulangan dan Kondisional

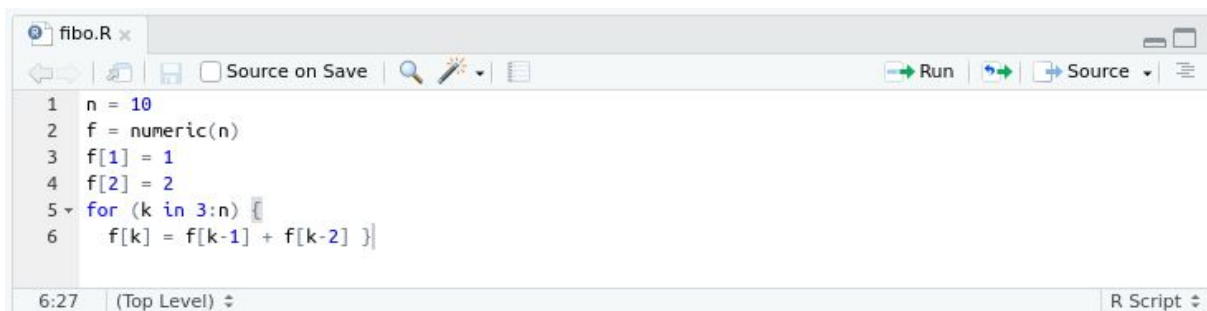
R menyediakan kontrol pemrograman yang dapat digunakan untuk mengatur jalannya eksekusi suatu program dengan menggunakan statemen perulangan `for`, `while` dan kondisional `if-elseif-else`.

```
> # Contoh Statemen perulangan
> H = matrix(0,4,4)
```

```
> for (i in 1:4) {  
+   for (j in 1:4) { H[i,j] = 1/(i+j-1) }  
+ }  
> H  
      [,1]      [,2]      [,3]      [,4]  
[1,] 1.0000000 0.5000000 0.3333333 0.2500000  
[2,] 0.5000000 0.3333333 0.2500000 0.2000000  
[3,] 0.3333333 0.2500000 0.2000000 0.1666667  
[4,] 0.2500000 0.2000000 0.1666667 0.1428571  
  
> # Contoh statemen perulangan while  
> a = 12.34; x = a/2; tol = 1e-6;  
> while (abs(a - x^2) > tol) {  
+   x = (x + a/x)/2  
+ }  
> x  
[1] 3.512834  
  
> # Contoh statemen kondisional  
> nilai = function(kode) {  
+   if (kode=='A') {n = 4}  
+   else if (kode=='B') {n = 3}  
+   else if (kode=='C') {n = 2}  
+   else {n = 1}  
+   return(n)}  
> nilai("A")  
[1] 4  
> nilai("C")  
[1] 2
```

## Skrip

Skrip adalah sebuah file teks yang di dalamnya terdapat perintah-perintah R. Apabila suatu skrip dijalankan maka perintah-perintah yang terdapat di dalamnya akan dieksekusi seolah-olah kita mengetikkannya pada jendela perintah. Sebuah file skrip dapat dibuat melalui program file editor , seperti geany, notepad2, atau melalui editor yang disediakan oleh Rgui dan RStudio. Gambar 5 adalah contoh sebuah skrip fibo.R yang menggambarkan perhitungan 10 suku pertama dari deret Fibonacci.



Gambar 5.

Statemen-staten yang terdapat di dalam sebuah skrip dapat dijalankan melalui perintah `source`. Apabila skrip fibo.R dijalankan maka akan memberikan hasil sebagai berikut:

```
> source('~\Documents\R\Coreetan\fibo.R')  
> f  
[1] 1 2 3 5 8 13 21 34 55 89
```



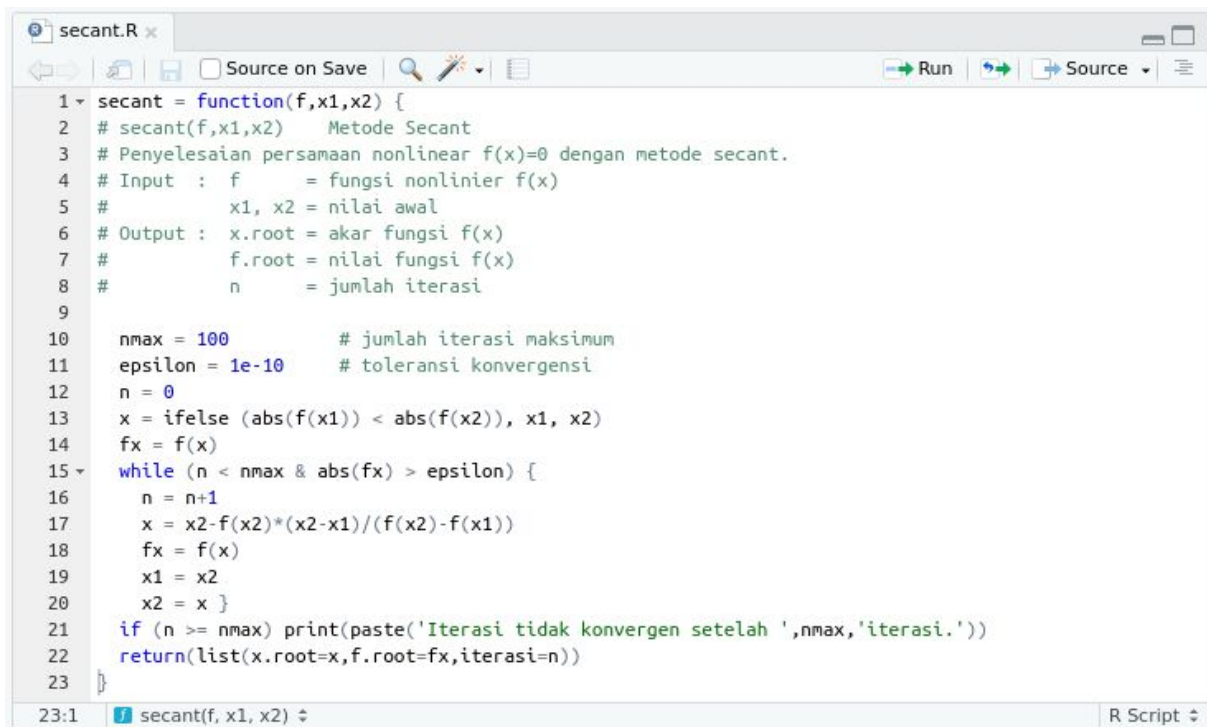
## Fungsi

Fungsi merupakan kumpulan dari statemen-statemen R yang dapat melakukan suatu komputasi atau perhitungan tertentu. Fungsi dapat dibuat dalam bentuk suatu file atau secara inline pada jendela perintah. Bentuk umum dari suatu fungsi adalah sebagai berikut:

```
fname = function(in1, in2, ...) {  
  statemen-statemen  
  return(out)  
}
```

dimana `fname` adalah nama fungsi, `in1`, `in2` adalah argumen input dan `out` adalah argumen output.

Berikut ini adalah ilustrasi dari suatu fungsi yang melalui editor pada program Rstudio.



```
1 secant = function(f,x1,x2) {  
2 # secant(f,x1,x2) Metode Secant  
3 # Penyelesaian persamaan nonlinear f(x)=0 dengan metode secant.  
4 # Input : f = fungsi nonlinier f(x)  
5 # x1, x2 = nilai awal  
6 # Output : x.root = akar fungsi f(x)  
7 # f.root = nilai fungsi f(x)  
8 # n = jumlah iterasi  
9  
10 nmax = 100 # jumlah iterasi maksimum  
11 epsilon = 1e-10 # toleransi konvergensi  
12 n = 0  
13 x = ifelse (abs(f(x1)) < abs(f(x2)), x1, x2)  
14 fx = f(x)  
15 while (n < nmax & abs(fx) > epsilon) {  
16 n = n+1  
17 x = x2-f(x2)*(x2-x1)/(f(x2)-f(x1))  
18 fx = f(x)  
19 x1 = x2  
20 x2 = x }  
21 if (n >= nmax) print(paste('Iterasi tidak konvergen setelah ',nmax,'iterasi.'))  
22 return(list(x.root=x,f.root=fx,iterasi=n))  
23 }
```

Gambar 6.

Selanjutnya, contoh pembuatan suatu fungsi secara inline adalah sebagai berikut:

```
> f = function(x) x - exp(-x) # fungsi f(x)
```

Fungsi yang telah dibuat dapat digunakan sebagaimana fungsi yang telah terpasang pada R. Fungsi yang tersimpan pada suatu file harus muat terlebih dahulu ke ruang kerja dengan perintah `source`.

Berikut ini adalah contoh penggunaan fungsi `secant` dan `f` yang telah dibuat.

```
> source('~\Documents\R\Coretan\secant.R')  
> sol = secant(f,0,1)  
> sol  
$x.root  
[1] 0.5671433  
  
$f.root  
[1] -1.24234e-13
```

\$iterasi  
[1] 5

## Komputasi Numerik

Di dalam R, berbagai macam persoalan komputasi numerik dapat diselesaikan dengan mudah, seperti yang ditunjukkan pada contoh-contoh di bawah ini.

Sistem persamaan linear  $Ax = b$  dapat diselesaikan dengan fungsi `solve(A,b)`.

```
> A = matrix(c(1,1,-1,6,0,2,6,-4,0), nrow=3, byrow=T)
> A
      [,1] [,2] [,3]
[1,]    1    1   -1
[2,]    6    0    2
[3,]    6   -4    0
> b = matrix(c(0,10,24), nrow=3)
> b
      [,1]
[1,]    0
[2,]   10
[3,]   24
> x = solve(A,b)
> x
      [,1]
[1,]    2
[2,]   -3
[3,]   -1
```

Fungsi `solve` dengan argumen tunggal akan menghasilkan matrik inverse. Nilai determinan dapat dihitung dengan fungsi `det`.

```
> solve(A)
      [,1]      [,2]      [,3]
[1,] 0.1818182 0.09090909 0.04545455
[2,] 0.2727273 0.13636364 -0.18181818
[3,] -0.5454545 0.22727273 -0.13636364
> det(A)
[1] 44
```

Penyelesaian suatu persamaan nonlinier  $f(x) = 0$  dapat dilakukan dengan fungsi `uniroot`, seperti yang ditunjukkan pada contoh di bawah ini.

```
> f = function(x) x - exp(-x)      # fungsi f(x)
> xsol = uniroot(f,c(0,1))
> xsol
$root
[1] 0.5671439

$f.root
[1] 9.448109e-07

$iter
[1] 3
```

```
$init.it  
[1] NA
```

```
$estim.prec  
[1] 7.424999e-05
```

Diperoleh bahwa akar dari  $f(x) = x - \exp(-x) = 0$ , adalah  $x = 0.5671$ .

Terdapat banyak persamaan integral tertentu yang sulit atau bahkan tidak dapat diselesaikan secara analitis, salah satunya adalah seperti yang terdapat pada contoh berikut ini.

$$Q = \int_0^5 \frac{\sin(3x)}{\sqrt{x^2 + x + 1}} dx$$

Nilai suatu integral tertentu dapat dihitung dengan menggunakan fungsi `quad`. Misalkan  $h(x)$  adalah fungsi yang diintegalkan pada persamaan di atas:

```
> h = function(x) sin(3*x)/sqrt(x^2 + x + 1)
```

Selanjutnya, nilai integral dapat dihitung dengan perintah sebagai berikut.

```
> Q = integrate(h, 0,5)  
> Q  
0.3648728 with absolute error < 3.2e-09
```

Diperoleh  $Q = 0.36487$ .