

# STK372 Komputasi Statistik 2

## Pengantar Pemrograman Fungsi R

Dr. Agus M Soleh



**IPB University**  
— Bogor Indonesia —

*agusms@apps.ipb.ac.id*

# Outline

- 1 Fungsi di R
- 2 Menciptakan Fungsi Sendiri
  - Argumen Fungsi
  - Output/Luaran Fungsi
  - Menggunakan Fungsi yang Telah dibuat
- 3 Penanganan Kesalahan
- 4 Teladan Membuat Fungsi
- 5 Package R
  - Instalasi Package R
  - Menggunakan Package R
  - Membuat Package R

# Fungsi di R

R telah menyiapkan banyak fungsi yang dapat dimanfaatkan untuk:

- Manajemen data dan Utilitas: `sort`, `rev`, dll
- Aritmetika: `log`, `sqrt`, dll
- Statistik: `sum`, `mean`, `var`, `sd`, dll
- Grafik: `hist`, `boxplot`, dll

Berkembangnya metode statistika mungkin membutuhkan pengembangan fungsi baru

# Menciptakan Fungsi Sendiri

- Fungsi yang tidak ada dalam R dapat diciptakan sendiri.
- Syntax:  
`namafungsi <- function (argumen) isifungsi`
- `isifungsi` merupakan satu objek data
- Jika membutuhkan beberapa baris dapat dikelompokkan dengan `{ }` dan diakhiri dengan satu objek data.
- Komentar dituliskan setelah symbol `#` dan tidak akan dieksekusi
- Teladan:

```
#Membuat fungsi simpangan baku  
std.dev <- function (x)  sqrt(var(x))
```

# Argumen Fungsi

- Fungsi umumnya memiliki argumen
- Argumen fungsi dalam R dapat:
  - diberikan sebuah nilai default  
contoh: `function(x=10)`
  - berulang/tidak didefinisikan pasti (menggunakan "...")  
contoh: `function(x=10, ...)`
- Argumen fungsi R dapat menerima nama fungsi R lain untuk dijalankan  
contoh: `apply(x,1,mean)` atau `apply(x,1,sum)`

# Output/Luaran Fungsi

- Output/luaran dari fungsi adalah objek
  - Dapat secara langsung dituliskan objeknya atau menggunakan return
  - Jika luarannya memiliki mode berbeda, umumnya digunakan objek list
- Contoh:

```
fungsi1 <- function(x) {  
  ...  
  list(hasil1,hasil2)  
}  
fungsi2 <- function(x) {  
  ...  
  return (list(hasil1,hasil2))  
}
```

# Menggunakan Fungsi yang Telah dibuat

- Untuk menggunakan fungsi yang telah dibuat gunakan syntax berikut:  
Syntax: `namafungsi (arg1, arg2, ...)`
- Nama argumen tidak perlu dituliskan, kecuali urutan diabaikan  
Contoh:

```
fungsi1(10,20)
```

```
fungsi1(arg2=20,arg1=10)
```

# Penanganan Kesalahan

Untuk menangani kesalahan dalam fungsi disediakan fungsi-fungsi:

- `try`: pembungkus untuk menjalankan ekspresi yang mungkin gagal dan memungkinkan kode pengguna untuk menangani pemulihan kesalahan.  
contoh: `try(log("a"))`
- `tryCatch`: menyediakan mekanisme untuk menangani kondisi yang tidak biasa, termasuk kesalahan dan peringatan.  
contoh: `tryCatch(log(10), finally = print("Hello"))`
- `warning`: menghasilkan pesan peringatan yang sesuai dengan argumennya
- `stop`: menghentikan eksekusi ekspresi saat ini



# Teladan

Membuat Fungsi untuk Melakukan Pengujian Hipotesis Nilai Tengah untuk Dua Populasi dengan ragam sama.

Algoritma :

- 1 hitung masing-masing  $n$
- 2 hitung galat baku
- 3 hitung statistik uji
- 4 hitung nilai- $p$
- 5 tetapkan keputusan penerimaan atau penolakan  $H_0$

# Teladan

```
ttest <- function(y1, y2, test = "dua-arah", alpha = 0.05)
{
  n1 <- length(y1); n2 <- length(y2)
  ndf <- n1+n2-2
  s2 <- ((n1-1)*var(y1) + (n2-1)*var(y2))/ndf
  tstat <- (mean(y1)-mean(y2))/sqrt(s2*(1/n1+1/n2))
  tail.area <-
    switch(test,
      "dua-arah" = 2 * (1-pt(abs(tstat), ndf)),
      kecil=pt(tstat,ndf),
      besar= 1-pt(tstat,ndf),
      {
        warning ("uji harus 'dua-arah', 'kecil' atau 'besar'")
        NULL
      }
    )
  list(tstat=tstat,df=ndf,reject=if(!is.null(tail.area))
      tail.area < alpha, tail.area=tail.area)
}
```

# Pengantar Package R

- Sebuah ekstensi dari sistem dasar R dengan kode, data dan dokumentasi dalam format standar
- Paket-paket R merupakan cara yang untuk memelihara koleksi fungsi dan data set R dan untuk menyebarkannya
- Sistem paket memungkinkan lebih banyak orang untuk berkontribusi R

# Instalasi Online Package R

- Paket umumnya disajikan dalam sebuah file source `namapaket.tar.gz`
- Instalasi paket menggunakan fungsi `install.packages` (harus terkoneksi ke internet)
- Jika koneksi menggunakan proxy, jalankan fungsi `setInternet2()` terlebih dahulu
- Jika terdapat beberapa yang disarankan dalam paket gunakan opsi `dep=c("Suggests")`  
Contoh: `install.packages("foo",dep=c("Suggests"))`

# Instalasi Manual Package R

- Paket dikumpulkan dalam repositori cran:
  - Repositori Utama: <https://cran.r-project.org/>
  - Indonesia: <https://repo.bbpt.go.id/cran/>
- Download `namapaket.tar.gz` (untuk windows alternatif `namapaket.zip`)
- Melalui command prompt sistem:
  - R CMD INSTALL `foo_version.tar.gz` (Unix)
  - Rcmd INSTALL `foo_version.tar.gz` (Windows)
- Untuk OS Windows mungkin dibutuhkan Rtools (<http://cran.r-project.org/bin/windows/Rtools/>)

# Menggunakan Package R

- Jika package R sudah terinstall, maka menggunakannya jalankan:  
`library (namapaket)` atau `require (namapaket)`
- Menggunakan data yang disediakan package R:  
`data (nama objek)`
- Melepas (Unload) package R:  
`detach ("package:namapaket")`

# Membuat Package R

- Siapkan semua fungsi, data dll
- Membuat paket skeleton  
`package.skeleton(name="namapackage",list=c("daftar objek"), path="folder")`
- Edit beberapa file dalam direktori yang dihasilkan. Folder dan file yang dihasilkan:  
`DESCRIPTION R Read-and-delete-me data man`
- Membangun file paket  
Dari sistem:
  - R CMD build dotchart (Unix)
  - Rcmd build dotchart atau  
`Rcmd build --binary --use-zip dotchart`  
(Windows)

# Akhir Pengantar Pemrograman R ...