



# Pengantar Komputasi Statistika

#9 Meeting

Basic Data Structure

Ferdian Bangkit Wijaya, S.Stat., M.Si

NIP. 199005202024061001



# R dan R Studio



Sebelum memulai, kita perlu membedakan dua hal penting:

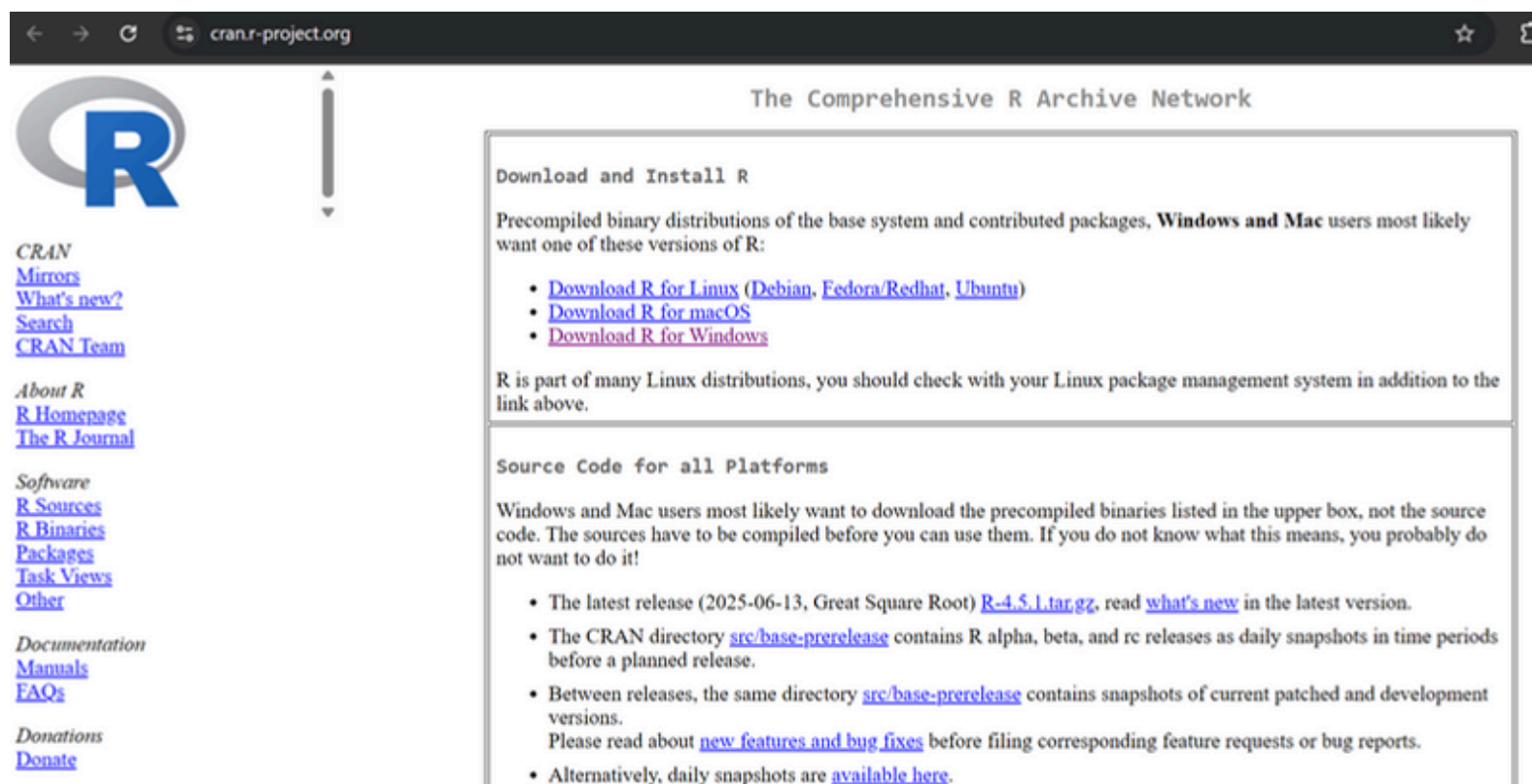
- R: Ini adalah bahasa pemrograman dan software untuk analisis statistika dan grafis. Anggap R sebagai mesin mobil. Dia yang melakukan semua perhitungan berat.
- RStudio: Ini adalah Integrated Development Environment (IDE). Anggap RStudio sebagai kokpit mobil (dashboard, setir, panel instrumen). RStudio menyediakan antarmuka (UI/UX) yang jauh lebih mudah dan nyaman untuk menggunakan R.

Perlu menginstal keduanya.

Aturan Penting: Selalu instal R terlebih dahulu, baru instal RStudio.

# Install R

## Install R Programming : Link CRAN R



The Comprehensive R Archive Network

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](#) (Debian, Fedora/Redhat, Ubuntu)
- [Download R for macOS](#)
- [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.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2025-06-13, Great Square Root) [R-4.5.1.tar.gz](#), read [what's new](#) in the latest version.
- The CRAN directory [src/base-prerelease](#) contains R alpha, beta, and rc releases as daily snapshots in time periods before a planned release.
- Between releases, the same directory [src/base-prerelease](#) contains snapshots of current patched and development versions. Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Alternatively, daily snapshots are [available here](#).

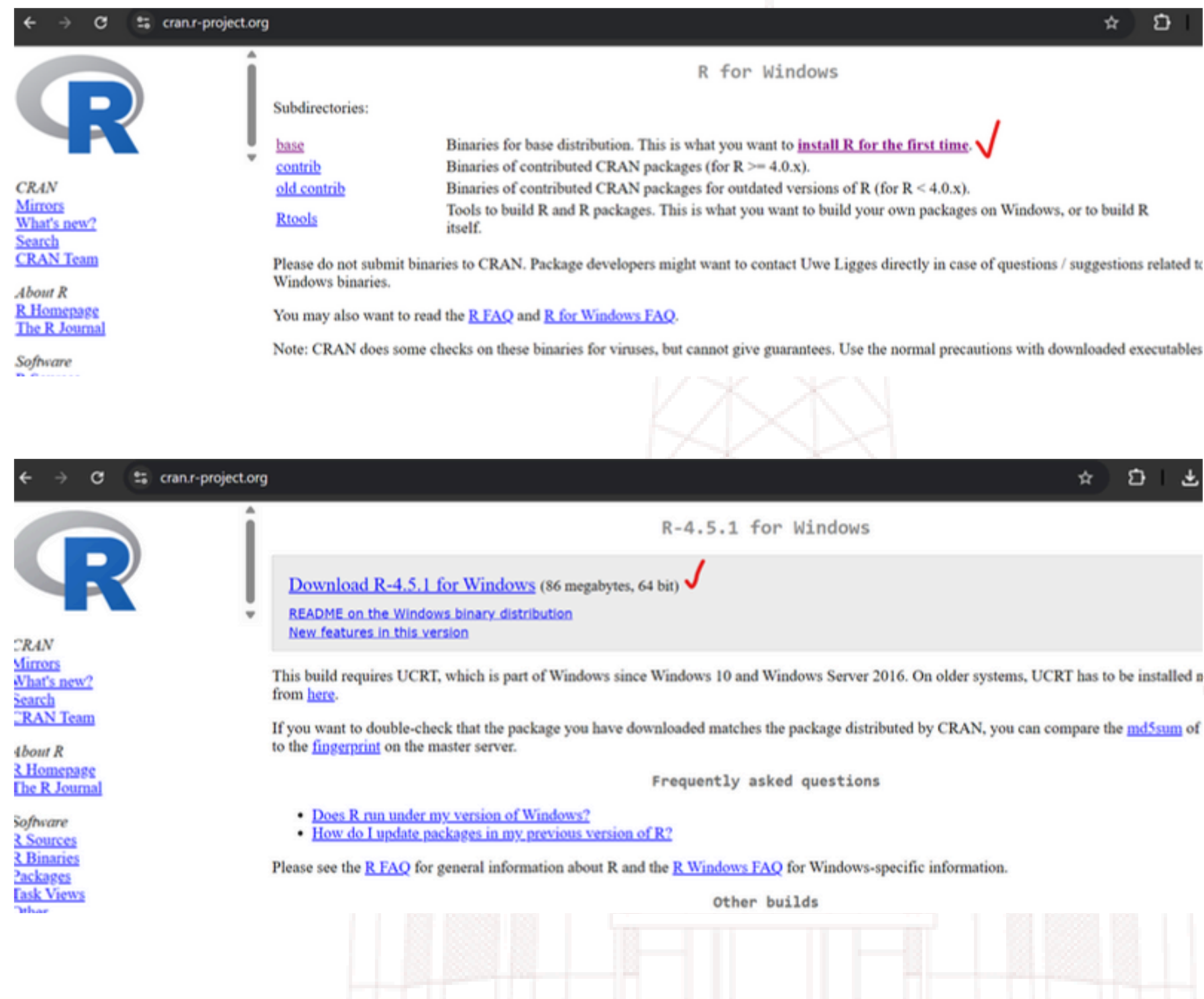
CRAN  
[Mirrors](#)  
[What's new?](#)  
[Search](#)  
[CRAN Team](#)

About R  
[R Homepage](#)  
[The R Journal](#)

Software  
[R Sources](#)  
[R Binaries](#)  
[Packages](#)  
[Task Views](#)  
[Other](#)

Documentation  
[Manuals](#)  
[FAQs](#)

Donations  
[Donate](#)



R for Windows

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 >= 4.0.x).
- [old.contrib](#): Binaries of contributed CRAN packages for outdated versions of R (for R < 4.0.x).
- [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

R-4.5.1 for Windows

[Download R-4.5.1 for Windows](#) (86 megabytes, 64 bit) ✓

[README on the Windows binary distribution](#)  
[New features in this version](#)

This build requires UCRT, which is part of Windows since Windows 10 and Windows Server 2016. On older systems, UCRT has to be installed from [here](#).

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of to the [fingerprint](#) on the master server.

Frequently asked questions

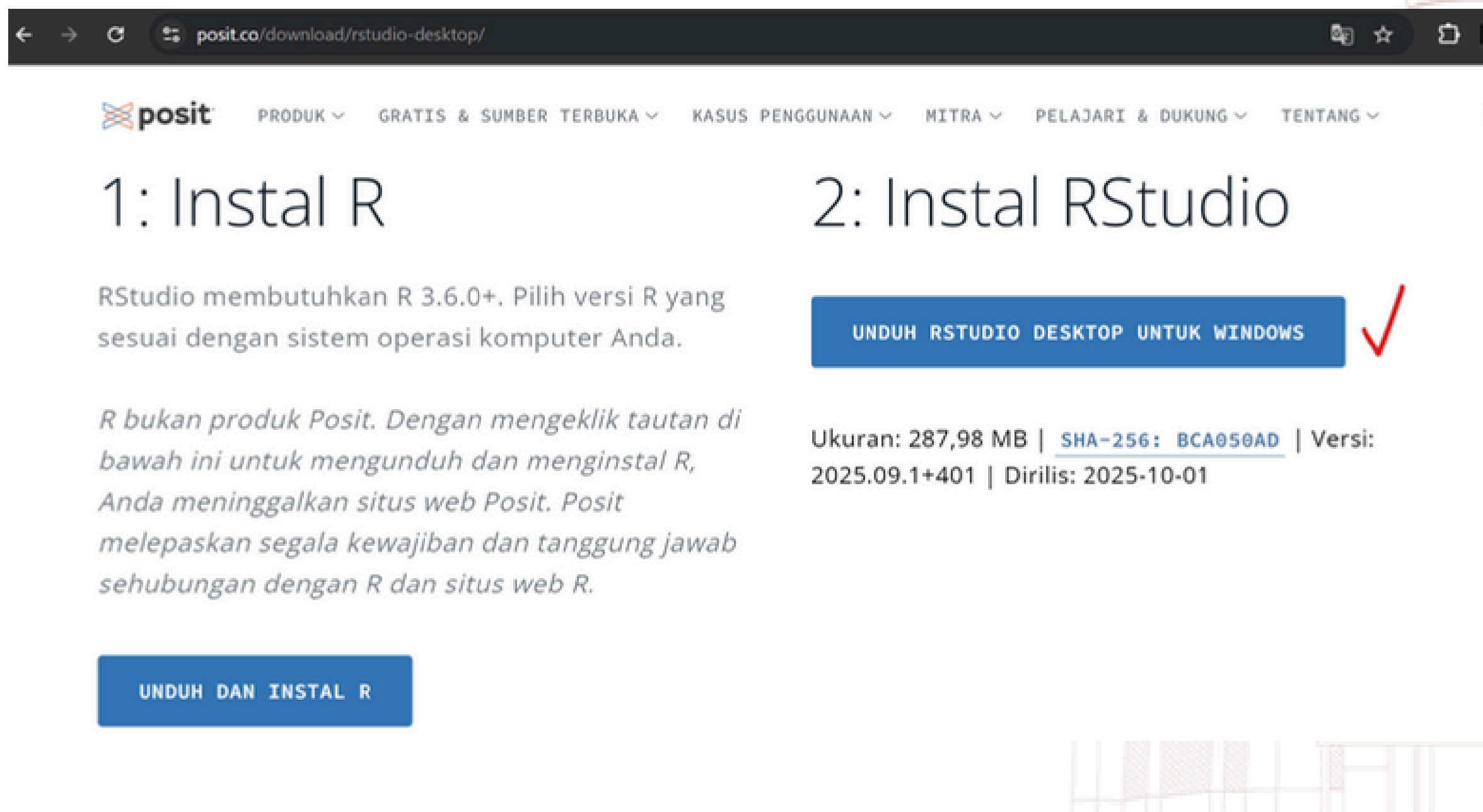
- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

# Install R Studio

Install R Studio : [Link R Studio](https://posit.co/download/rstudio-desktop/)



posit PRODUK GRATIS & SUMBER TERBUKA KASUS PENGGUNAAN MITRA PELAJARI & DUKUNG TENTANG

## 1: Instal R

RStudio membutuhkan R 3.6.0+. Pilih versi R yang sesuai dengan sistem operasi komputer Anda.

*R bukan produk Posit. Dengan mengeklik tautan di bawah ini untuk mengunduh dan menginstal R, Anda meninggalkan situs web Posit. Posit melepaskan segala kewajiban dan tanggung jawab sehubungan dengan R dan situs web R.*

UNDUH DAN INSTAL R

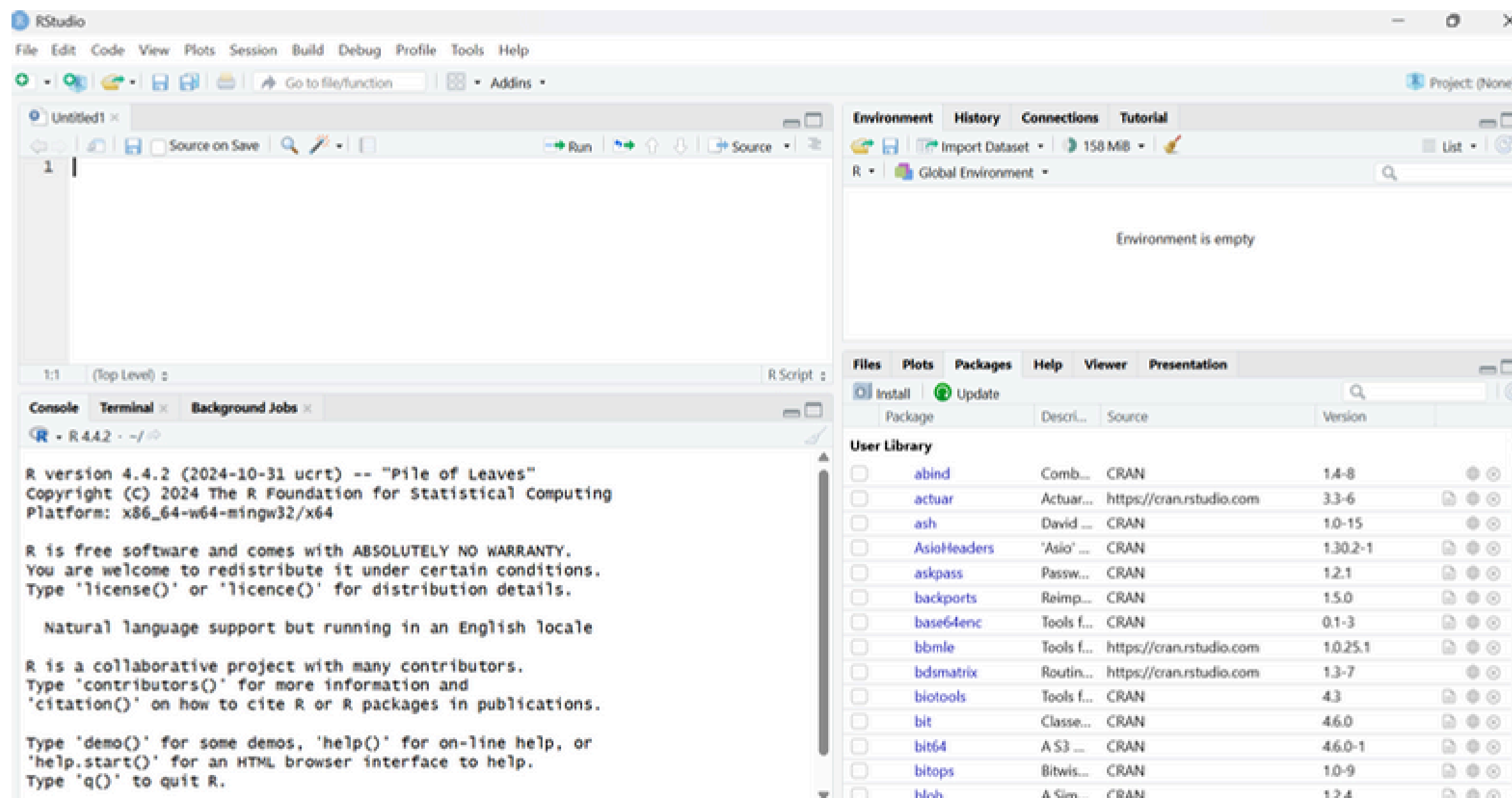
## 2: Instal RStudio

UNDUH RSTUDIO DESKTOP UNTUK WINDOWS ✓

Ukuran: 287,98 MB | [SHA-256: BCA050AD](#) | Versi: 2025.09.1+401 | Dirilis: 2025-10-01

# Install R dan R Studio

Buka R Studio





# Penyiapan Folder Project



Sangat penting untuk bekerja secara terorganisir!

1. Jangan menyimpan file pekerjaan di sembarang tempat (seperti Desktop atau Downloads).
2. Buat sebuah folder baru khusus untuk mata kuliah ini di lokasi yang mudah diakses (misal: di dalam Documents atau D:).
3. Contoh Lokasi Folder:
  - Windows: C:/Users>NamaAnda/Documents/Kuliah/KomputasiStatistika
  - macOS: /Users>NamaAnda/Documents/KomputasiStatistika

Tips: Hindari penggunaan spasi pada nama folder Anda. Gunakan underscore (\_) atau camelCase (misal: KomputasiStatistika bukan Komputasi Statistika).





# Working Directory



- Direktori Kerja (atau Working Directory) adalah folder default tempat R akan mencari file untuk dibaca dan menyimpan file (plot, data, dll) yang Anda buat.
- Perintah `cd` (Change Directory) di Terminal atau Command Prompt setara dengan perintah `setwd()` (Set Working Directory) di R.

Kenapa Ini Penting? Jika Anda tidak mengaturnya, R mungkin akan membaca dari folder `C:/Users>NamaAnda/Documents` sementara file data Anda ada di `D:/Kuliah/Data`. Ini akan menyebabkan eror "File not found".







# Set Working Directory



# Mengecek direktori saat ini (Anda tidak perlu mengetik ini)  
getwd()

```
[1] "C:/Users>NamaAnda/Documents"
```

# Mengganti direktori ke folder proyek kita

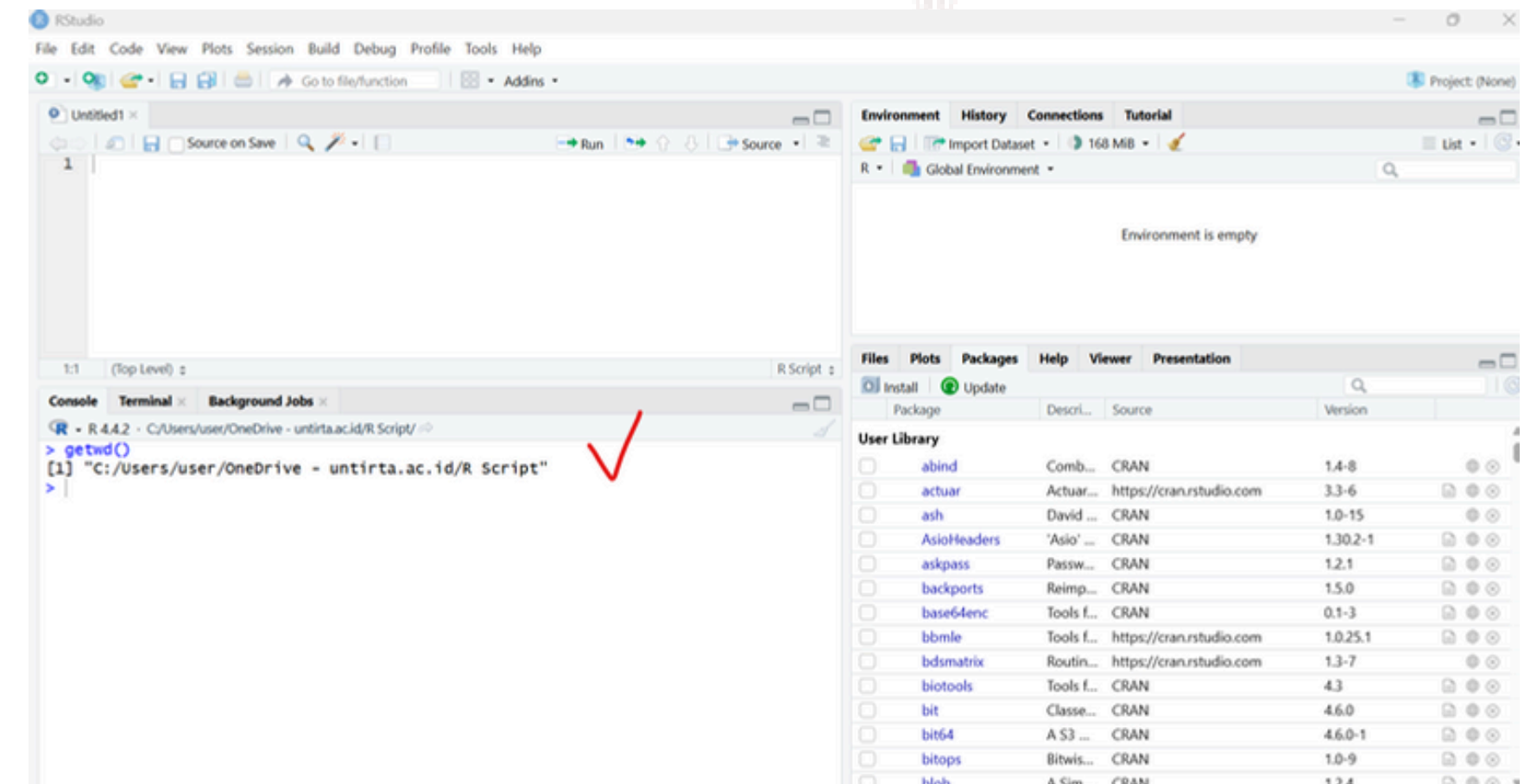
# PERHATIKAN: R menggunakan forward-slash (/) BUKAN  
back-slash (\)

```
setwd("C:/Users>NamaAnda/Documents/Kuliah/KomputasiStatistika")
```

# Cek lagi untuk memastikan sudah berubah  
getwd()

```
[1]
```

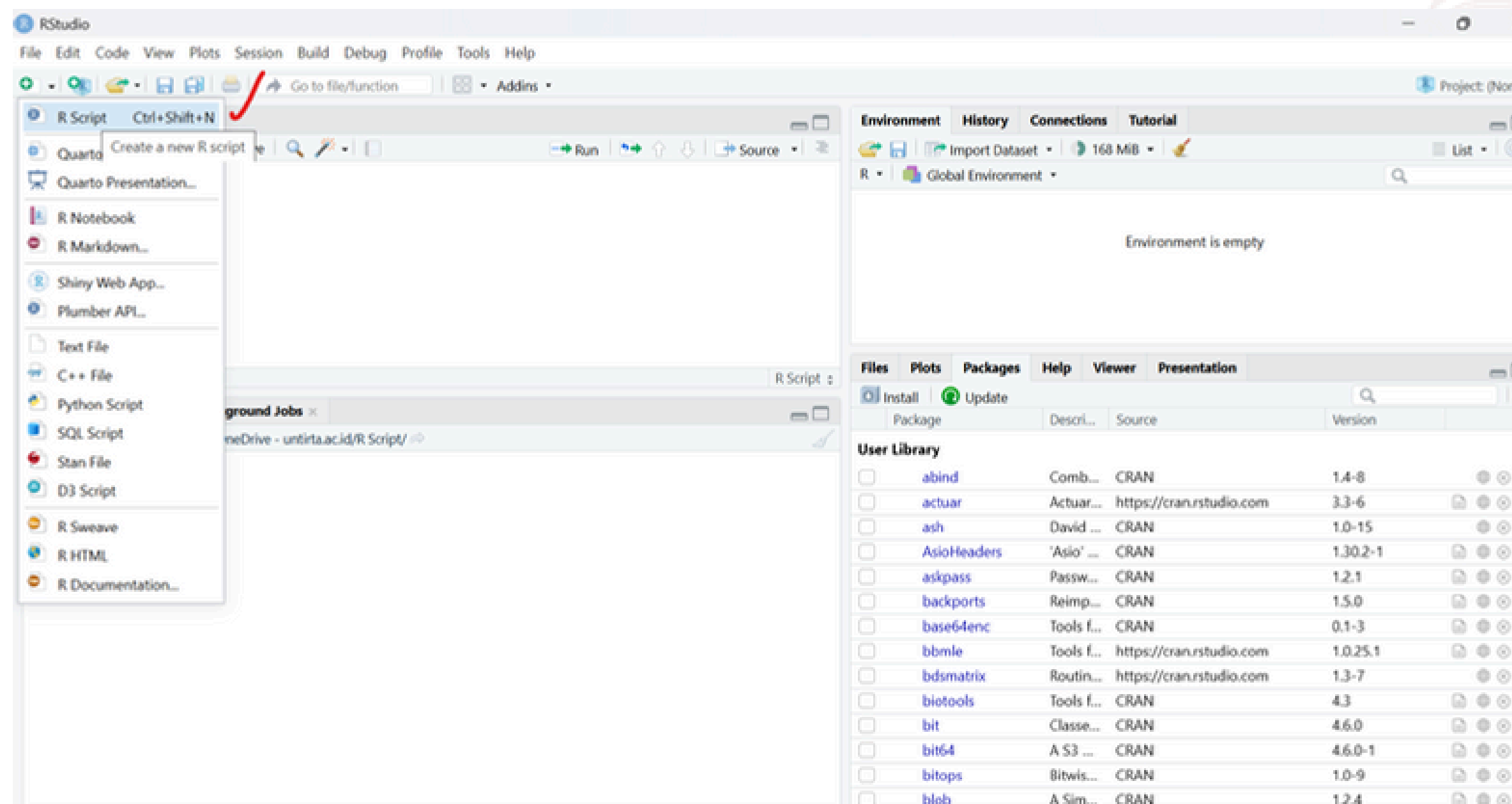
```
"C:/Users>NamaAnda/Documents/Kuliah/KomputasiStatistika"
```





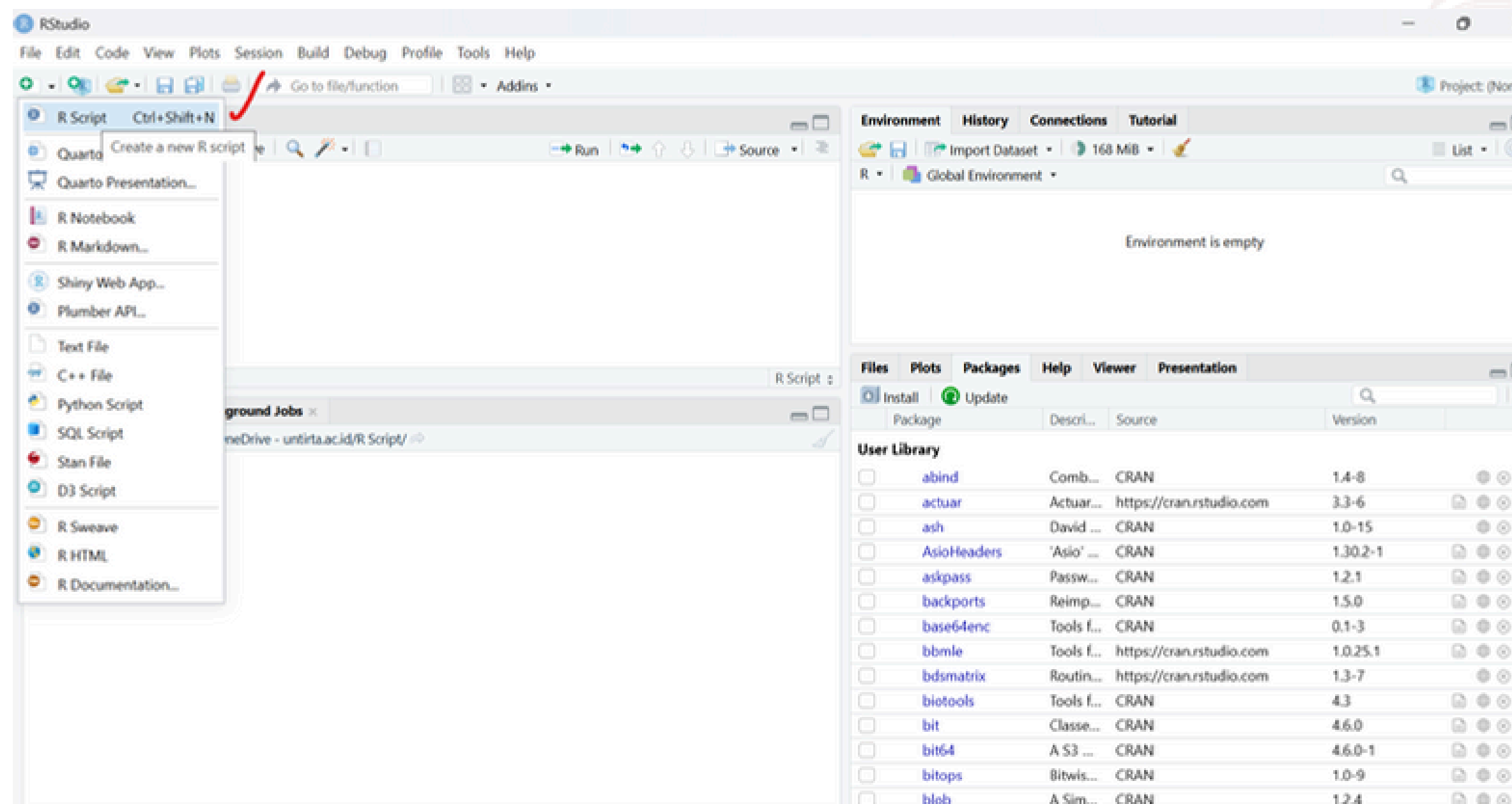
# New File

Buat New File pada R Studio untuk melakukan koding



# New File

Buat New File pada R Studio untuk melakukan koding





# Variabel dan Tipe Data



Apa itu Variabel? Tempat untuk menyimpan nilai. Anggap saja seperti kotak penyimpanan yang Anda beri label (nama variabel).

Apa itu Tipe Data? Jenis nilai yang disimpan di dalam "kotak" tersebut.

R memiliki 6 tipe data atomik:

- 1.numeric (atau double): Angka desimal atau angka bulat. Contoh: 10.5, 77.
- 2.integer: Angka bulat murni. Dibuat dengan menambahkan L. Contoh: 10L, 100L. (Ini adalah bagian khusus dari numeric).
- 3.logical: TRUE atau FALSE.
- 4.character: Teks. Contoh: "hello".
- 5.complex: Bilangan kompleks. Contoh:  $5 + 2i$  (jarang dipakai di statistika dasar).
- 6.raw: Untuk data mentah dalam bentuk bytes (sangat jarang dipakai).





# Contoh Variabel dan Operator



Assignment (Menyimpan Nilai) Kita menggunakan <- (panah) untuk memasukkan nilai ke dalam variabel.

# Tipe Numeric

nilai\_uts <- 85

tinggi\_badan <- 170.5

# Tipe Character (Teks)

nama\_mahasiswa <- "Budi"

mata\_kuliah <- "Statistika"

# Tipe Logical

lulus <- TRUE

mengulang <- FALSE

Operator Aritmatika Sama seperti matematika: +, -, \*, /, ^ (pangkat)

# Operasi pada variabel

nilai\_uas <- 90

nilai\_akhir <- (nilai\_uts + nilai\_uas) / 2

# Mencetak isi variabel ke Console

print(nilai\_akhir)

[1] 87.5





# Vektor



Apa itu Vektor? Struktur data paling dasar di R. Ini adalah kumpulan nilai dengan tipe data yang SAMA.

- Anggap saja seperti rangkaian kotak yang saling terhubung.
- Dibuat menggunakan fungsi `c()` (singkatan dari `combine` atau `concatenate`).

# Vektor numeric

```
vektor_nilai <- c(85, 90, 78, 92, 88)
```

# Vektor character

```
vektor_nama <- c("Budi", "Ani", "Citra", "Doni")
```

# Vektor logical

```
vektor_lulus <- c(TRUE, TRUE, FALSE, TRUE)
```

Vektor harus berisi tipe data yang sama. Jika Anda mencampurnya, R akan "memaksa" (coercion) semuanya menjadi tipe data yang paling fleksibel (biasanya character).

# Contoh Coercion (Pemaksaan Tipe)

```
campuran <- c("Budi", 80, TRUE)
```

```
print(campuran)
```

```
# [1] "Budi" "80"  "TRUE" <- Semuanya menjadi teks!
```

# Pengindeksan Vektor

```
vektor_nilai <- c(85, 90, 78, 92, 88)
```

```
length(vektor_nilai)
```

```
[1] 5 (Jumlah elemen)
```

```
sum(vektor_nilai)
```

```
[1] 433 (Total jumlah)
```

```
mean(vektor_nilai)
```

```
[1] 86.6 (Rata-rata)
```

```
min(vektor_nilai)
```

```
[1] 78 (Nilai terkecil)
```

```
max(vektor_nilai)
```

```
[1] 92 (Nilai terbesar)
```

```
summary(vektor_nilai)
```

Memberikan ringkasan statistik lengkap

Pengindeksan Vektor (Mengambil Elemen) R menggunakan indeks berbasis 1 (angka pertama adalah 1, bukan 0 seperti di Python).

```
vektor_nama <- c("Budi", "Ani", "Citra", "Doni")
```

```
# Mengambil elemen ke-2
```

```
vektor_nama[2]
```

```
# [1] "Ani"
```

```
# Mengambil elemen ke-1 dan ke-3
```

```
vektor_nama[c(1, 3)]
```

```
# [1] "Budi" "Citra"
```

```
# Mengambil semua KECUALI elemen ke-2
```

```
vektor_nama[-2]
```

```
# [1] "Budi" "Citra" "Doni"
```



# Array dan Matrix



Apa itu Array? Mirip seperti vektor, tetapi bisa memiliki lebih dari satu dimensi (multi-dimensi).

- Anggap saja seperti tabel (2 dimensi) atau kubus data (3 dimensi).
- Array juga harus berisi tipe data yang SAMA.
- Dibuat menggunakan fungsi `array()`.

## Matriks (Khusus)

- Matriks hanya merujuk pada array yang memiliki tepat 2-dimensi (baris dan kolom).
- Karena struktur 2D ini sangat umum dalam statistika dan matematika (aljabar linear), R memberikan fungsi khusus untuknya, yaitu `matrix()`.

```
# Cara 1: Membuat matriks dengan fungsi matrix()
matriks_saya <- matrix(data = 1:6, nrow = 2, ncol = 3)
```

```
# Cara 2: Membuat array 2-dimensi dengan fungsi array()
array_saya <- array(data = 1:6, dim = c(2, 3))
```

# Keduanya akan menghasilkan output yang SAMA PERSIS:

```
#      [,1] [,2] [,3]
```

```
# [1,]  1  3  5
```

```
# [2,]  2  4  6
```

# Ini akan GAGAL

```
matriks_3d <- matrix(data = 1:24, dim = c(2, 3, 4))
```

```
# Error: 'dim' must be a vector of length 2
```

Anda tidak bisa membuat objek 3D dengan fungsi `matrix()` (karena matriks harus 2D).

# Ini BERHASIL

```
array_3d <- array(data = 1:24, dim = c(2, 3, 4))
```

```
# Ini adalah "kubus data" (4 tumpukan matriks 2x3)
```





# Pengindeksan Array



```
dim(array_saya)
# [1] 2 3 (Mengecek dimensi: 2 baris, 3 kolom)
length(array_saya)
# [1] 6 (Total elemen)
```

```
# array_saya
# [,1] [,2] [,3]
# [1,] 1 3 5
# [2,] 2 4 6
```

```
# Mengambil elemen di Baris 1, Kolom 3
array_saya[1, 3]
# [1] 5
```

```
# Mengambil SELURUH Baris 2
array_saya[2, ]
# [1] 2 4 6
```

```
# Mengambil SELURUH Kolom 1
array_saya[, 1]
# [1] 1 2
```

```
# Mengambil elemen di Baris 1, Kolom 2 dan 3
array_saya[1, c(2, 3)]
# [1] 3
```





# Pengindeksan Matrix



```
# Membuat data angka 1 sampai 12
```

```
data_angka <- 1:12
```

```
# Membuat matriks 3 baris, 4 kolom
```

```
# byrow = TRUE artinya kita mengisi datanya per  
BARIS (kiri ke kanan)
```

```
mat <- matrix(data_angka, nrow = 3, ncol = 4, byrow  
= TRUE)
```

```
# Mencetak matriks 'mat'
```

```
print(mat)
```

```
      [,1] [,2] [,3] [,4]  
[1,]   1   2   3   4  
[2,]   5   6   7   8  
[3,]   9  10  11  12
```

```
# 1. Mengambil SATU elemen (Baris 2, Kolom 3)
```

```
mat[2, 3]
```

```
# [1] 7
```

```
# 2. Mengambil SELURUH Baris 1
```

```
# (Kosongkan indeks kolom)
```

```
mat[1, ]
```

```
# [1] 1 2 3 4
```

```
# 3. Mengambil SELURUH Kolom 4
```

```
# (Kosongkan indeks baris)
```

```
mat[, 4]
```

```
# [1] 4 8 12
```



# Pengindeksan Matrix



# 4. Mengambil SUB-MATRIKS (Blok)

# (Baris 1 dan 2, Kolom 3 dan 4)

mat[1:2, 3:4]

# [,1] [,2]

# [1,] 3 4

# [2,] 7 8

# 5. Mengambil Baris yang TIDAK BERURUTAN

# (Baris 1 dan 3, SEMUA kolom)

mat[c(1, 3), ]

# [,1] [,2] [,3] [,4]

# [1,] 1 2 3 4

# [2,] 9 10 11 12

# 6. Menggunakan LOGIKA

# (Ambil semua elemen yang nilainya LEBIH DARI 8)

# Hasilnya akan menjadi VEKTOR, bukan matriks lagi

mat[mat > 8]

# [1] 9 10 11 12



# List

Apa itu List? Struktur data yang paling fleksibel di R.

- List adalah kumpulan "kotak" 📦🎁🍲 yang bisa berisi TIPE DATA APAPUN.
- Satu elemen list bisa berisi vektor, elemen lain bisa berisi array, dan elemen lainnya bisa berisi teks.
- Anggap saja seperti tas belanjaan yang bisa diisi apa saja.
- Dibuat menggunakan fungsi list()

# Membuat list data mahasiswa

```
mahasiswa_1 <- list(  
  nama = "Budi Hartono",  
  npm = "12345678",  
  usia = 21,  
  lulus = FALSE,  
  nilai = c(85, 90, 78) # Elemen ini adalah sebuah vektor  
)
```

# Mencetak list

```
print(mahasiswa_1)
```





# Pengindeksan List



Mengambil elemen dari list bisa menggunakan dua cara:

1. \$ (Tanda Dolar): Cara paling umum, menggunakan nama elemen.
2. [...] (Kurung Siku Ganda): Menggunakan nama atau nomor indeks.

# Menggunakan \$ (paling umum)

mahasiswa\_1\$nama

# [1] "Budi Hartono"

# Mengambil vektor nilai

mahasiswa\_1\$nilai

# [1] 85 90 78

# Mengambil nilai ke-2 dari vektor 'nilai' di dalam list

mahasiswa\_1\$nilai[2]

# [1] 90

# -----

# Menggunakan [...]

# Mengambil elemen ke-1 (nama)

mahasiswa\_1[[1]]

# [1] "Budi Hartono"

# Mengambil elemen dengan nama "npm"

mahasiswa\_1[["npm"]]

# [1] "12345678"



# SEE YOU NEXT WEEK !

**Ferdian Bangkit Wijaya, S.Stat., M.Si**

**NIP. 199005202024061001**

**[ferdian.bangkit@untirta.ac.id](mailto:ferdian.bangkit@untirta.ac.id)**