

Modul Praktik Statistika Untuk Perencanaan

Penggunaan R dalam Pengolahan Data Kuantitatif

Contents

Pengantar

Selamat datang di modul daring Praktikum Statistika untuk Perencanaan. Modul ini digunakan untuk mendampingi kalian mengikuti Praktikum dalam mata kuliah Statistika untuk Perencanaan, mata kuliah di Program Studi Perencanaan Wilayah dan Kota, Institut Teknologi Sumatera.

Modul praktikum ini akan mendampingi Anda mempelajari pengolahan statistik menggunakan bahasa pemrograman R. Untuk itu, modul ini terbagi menjadi **9 modul** yang disusun secara bertahap, dimulai dari pengenalan dasar hingga analisis multivariat:

Modul	Topik	Deskripsi Singkat
1	Pengolahan Data Terstruktur	Dasar <i>data wrangling</i> : Impor, seleksi, standardisasi, penggabungan, dan ekspor data/dataset
2	Statistika Deskriptif	Perhitungan proporsi, mean, median, standar deviasi, dan IQV
3	Visualisasi Data	Pembuatan grafik dengan ggplot2 : bar chart, histogram, boxplot, scatter plot
4	Distribusi Sampling & Interval Kepercayaan	Simulasi sampling, standard error, dan estimasi parameter
5	Uji Hipotesis	Uji hipotesis 1 dan 2 populasi untuk rata-rata dan proporsi
6	Korelasi Nominal & Ordinal	Koefisien V Cramer, Lambda, dan Gamma
7	Korelasi Metrik	Koefisien Spearman's dan Pearson's r
8	Regresi Linear	Regresi linear sederhana dan berganda
9	Analisis Komponen Prinsip	PCA dan Analisis Faktor

Setiap modul dilengkapi dengan **capaian pembelajaran**, **contoh kode**, dan **soal latihan** yang harus dikerjakan.

Pengenalan R dan RStudio

Selamat datang di dunia pengolahan data yang lebih canggih! Dalam praktikum ini, kalian akan berkenalan dengan teknik pengolahan dan analisis data dengan menggunakan bahasa pemrograman bernama R. R adalah bahasa pemrograman yang dibuat khusus untuk keperluan analisis statistik.

0.1 Apa itu R?

R adalah lingkungan perangkat lunak (*software environment*) dan bahasa pemrograman yang digunakan untuk komputasi statistik dan grafik. R dikembangkan oleh Ross Ihaka dan Robert Gentleman di Universitas Auckland, Selandia Baru, pada awal tahun 1990-an sebagai evolusi dari bahasa S yang sebelumnya dikembangkan di Bell Laboratories (?). R bersifat *open-source* (berlisensi GNU GPL), yang berarti dapat digunakan secara gratis dan dikembangkan secara kolaboratif oleh komunitas global (?).

Keunggulan utama R dalam konteks perencanaan dan pemerintahan meliputi:

1. **Ekosistem Paket yang Luas:** R memiliki puluhan ribu paket (*packages*) di CRAN untuk berbagai keperluan, mulai dari manipulasi data, pemodelan statistik, hingga analisis geospasial menggunakan *sf* dan *terra* (??).
2. **Visualisasi Data:** Dengan paket seperti *ggplot2* dan *Shiny*, R memungkinkan pembuatan grafik berkualitas tinggi dan dashboard interaktif yang informatif (?).
3. **Dukungan Kebijakan Berbasis Data:** R mendukung implementasi kebijakan “Satu Data Indonesia” dengan kemampuan standarisasi, interoperabilitas, dan transparansi metode analisis yang dapat diaudit (?). R memfasilitasi proses ETL (*Extract-Transform-Load*) otomatis untuk membersihkan data dari berbagai instansi (?).

Dalam bidang perencanaan wilayah dan kota, R digunakan untuk:

- **Analisis Geospasial:** Mengelola data vektor dan raster untuk pemetaan dan pemodelan lingkungan (?).
- **Perencanaan Transportasi:** Menggunakan paket seperti *stplanr* untuk analisis jaringan, rute, dan pemodelan transportasi berkelanjutan (??).
- **Simulasi Kebijakan:** Melakukan simulasi dampak kebijakan publik, seperti perubahan penggunaan lahan atau intervensi sosial, sebelum diterapkan secara nyata (?).

0.2 Mengapa Menggunakan R?

R memiliki kelebihan yang signifikan dibandingkan dengan alat analisis berbasis antarmuka grafis (*Graphical User Interface* - GUI) seperti SPSS atau Microsoft Excel. Berikut adalah beberapa alasan utama mengapa R menjadi standar *de facto* dalam sains data modern:

1. **Reproduksibilitas (*Reproducibility*):** Berbeda dengan perangkat lunak berbasis *point-and-click* di mana langkah-langkah analisis seringkali sulit dilacak kembali, R bekerja berbasis skript (*script-based*). Setiap langkah analisis, mulai dari pembersihan data hingga pembuatan model, terekam secara eksplisit dalam kode. Hal ini memungkinkan analisis untuk diverifikasi, diaudit, dan dijalankan ulang oleh orang lain (atau diri kita sendiri di masa depan) dengan hasil yang identik, sebuah prinsip fundamental dalam penelitian ilmiah yang kredibel (??).
2. **Fleksibilitas dan Kemampuan Kostumisasi:** R tidak membatasi pengguna pada menu yang sudah tersedia. Sebagai bahasa pemrograman, R memungkinkan kita untuk membangun alat analisis sendiri atau memodifikasi metode yang sudah ada sesuai dengan kebutuhan spesifik penelitian, memberikan fleksibilitas yang hampir tak terbatas dibandingkan perangkat lunak tertutup (*proprietary*) (?).
3. **Gratis dan *Open Source*:** Sebagai perangkat lunak *open source*, R dapat digunakan sepenuhnya secara gratis. Ini menghilangkan hambatan biaya lisensi yang mahal yang sering ditemui pada perangkat lunak komersial, membuatnya sangat aksesibel bagi mahasiswa, peneliti, dan pemerintah daerah dengan anggaran terbatas.
4. **Komunitas yang Masif:** Dukungan komunitas R sangat luar biasa melalui platform seperti Stack-Overflow dan R-Bloggers. Jika Anda menemui masalah, kemungkinan besar orang lain sudah pernah menghadapinya dan solusinya sudah tersedia secara daring.

0.3 Apa itu RStudio?

RStudio (yang kini bertransformasi menjadi **Posit**) adalah *Integrated Development Environment* (IDE) untuk R. Diluncurkan pertama kali pada tahun 2009, RStudio menyediakan antarmuka visual yang lebih ramah pengguna dibandingkan antarmuka baris perintah (*command line*) asli R (?). Fitur-fitur seperti penyorotan sintaksis (*syntax highlighting*), *auto-completion*, dan manajemen *workspace* membuat pengguna baru lebih mudah mempelajari dan menggunakan R. Transformasi menjadi Posit mencerminkan komitmen terhadap ekosistem *data science* modern yang lebih luas, termasuk dukungan untuk Python (?).

Gambar ?? memberikan penjelasan singkat tentang tampilan jendela RStudio yang terdiri atas 4 panel.

1. Panel *Environment* yang menampilkan daftar data atau variabel yang telah kita impor atau buat.
2. Panel *Files, Plots, dan Help*, tempat kita melihat daftar *file* yang tersedia, grafik yang dihasilkan, dan menemukan dokumen bantuan untuk berbagai bagian dari R.
3. Panel *Console*, digunakan untuk menjalankan kode.
4. Panel *Script*, digunakan untuk menulis kode. Di sinilah kita akan menghabiskan sebagian besar waktu kerja kita.

0.4 Instalasi R dan RStudio

Sebelum memulai praktikum, kalian perlu menginstal R dan RStudio terlebih dahulu. R harus diinstal **sebelum** RStudio, karena RStudio memerlukan instalasi R yang berfungsi untuk dapat berjalan.

Langkah 1: Instalasi R

1. Kunjungi situs resmi CRAN di **cran.r-project.org**
2. Pilih sistem operasi Anda (Windows/macOS/Linux)
3. Untuk **Windows**: Klik “base” lalu unduh installer terbaru
4. Untuk **macOS**: Pilih versi sesuai prosesor (Apple Silicon M1/M2/M3/M4/M5 atau Intel)
5. Jalankan file installer dan ikuti petunjuk hingga selesai

Langkah 2: Instalasi RStudio

1. Kunjungi situs resmi Posit di **posit.co/downloads**
2. Unduh **RStudio Desktop** versi gratis sesuai sistem operasi Anda
3. Jalankan installer dan ikuti petunjuk hingga selesai

Verifikasi: Buka RStudio setelah instalasi selesai. RStudio akan otomatis mendeteksi instalasi R yang sudah ada.

0.5 Cara Menjalankan R

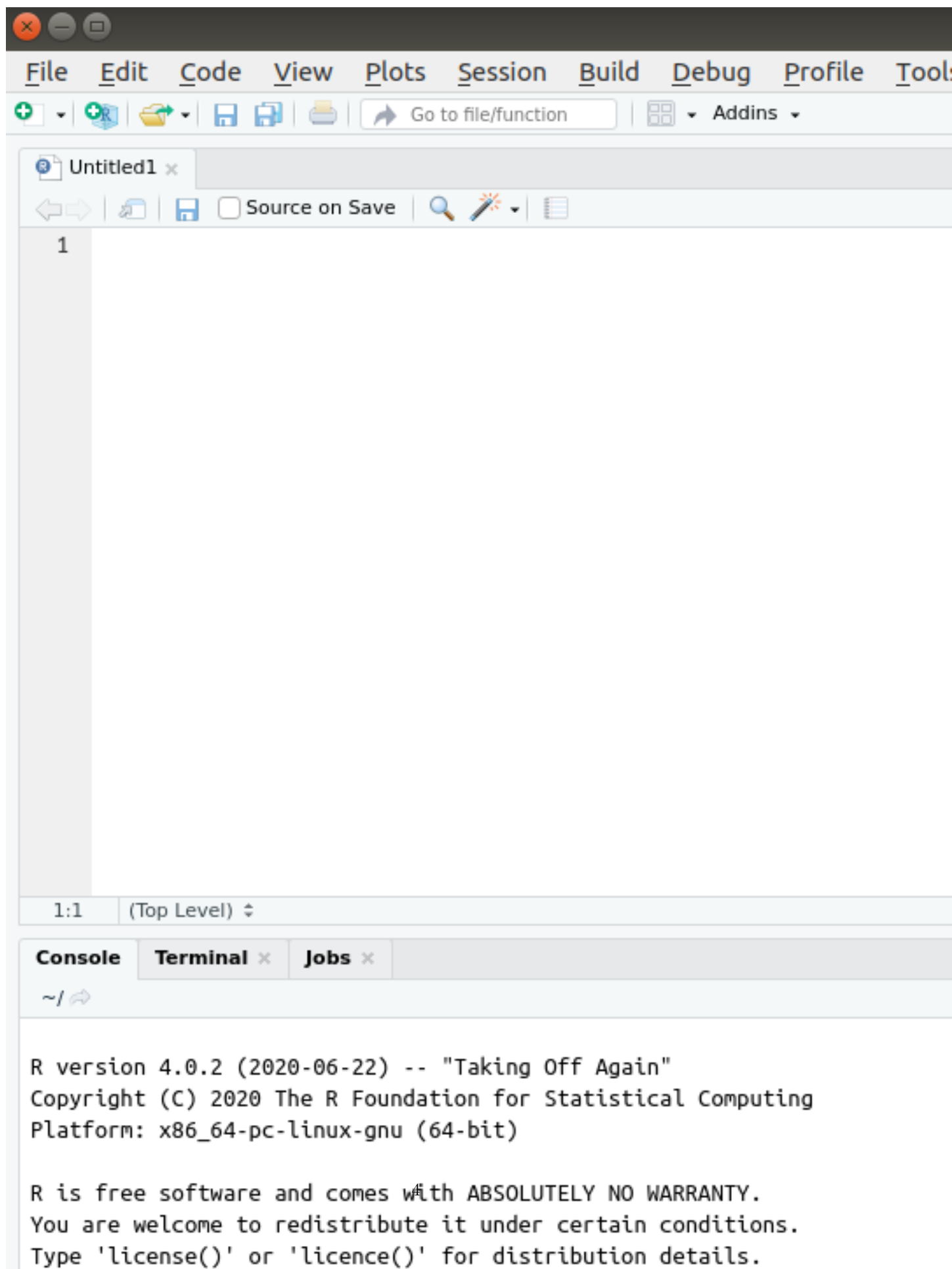
0.5.1 Langsung di *Console*

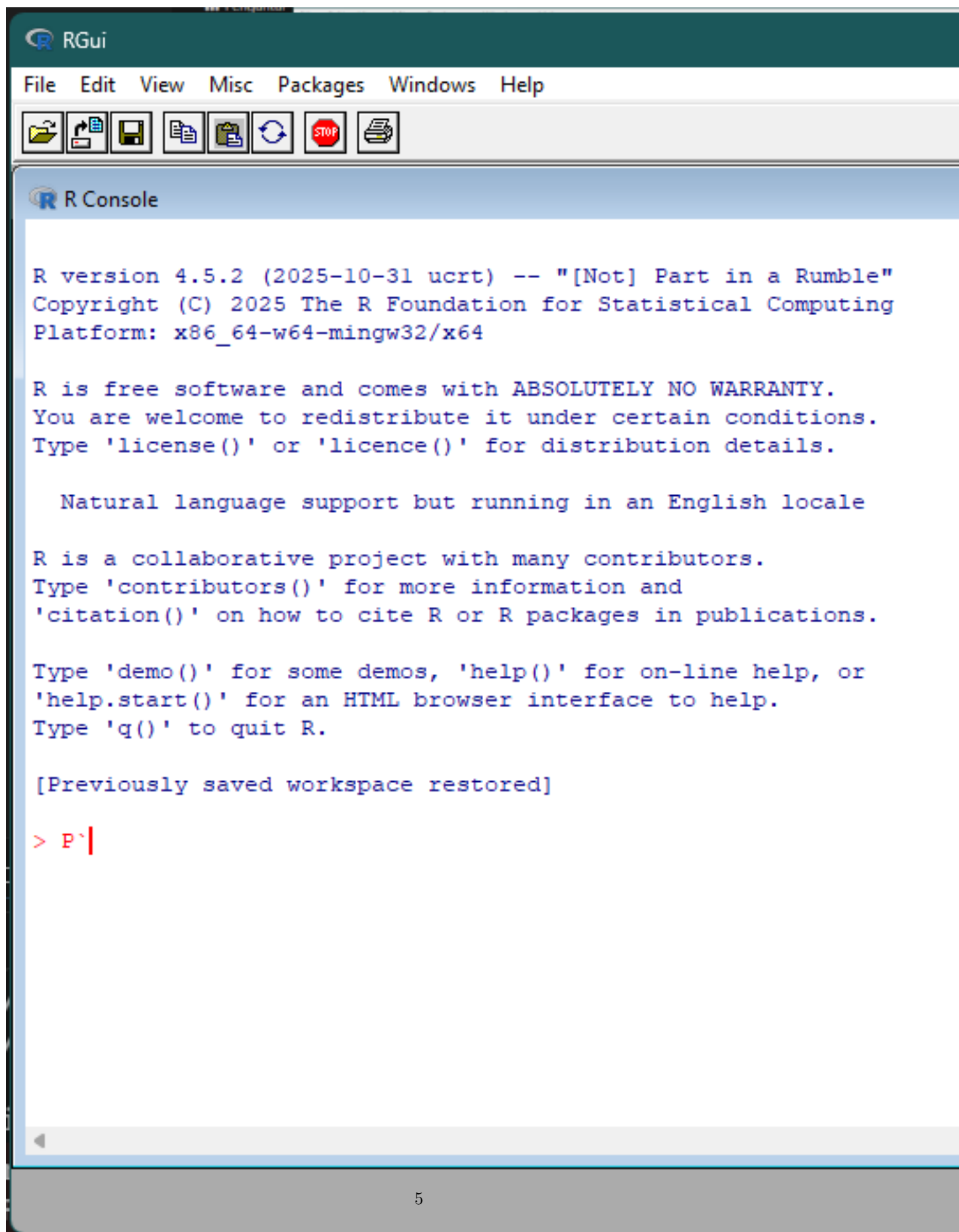
Pertama, silakan buka program **RGui**. Caranya, buka Start Menu, ketikkan **R x.x.x**. Ganti **x.x.x** dengan nomor versi yang Anda unduh sebelumnya. Kemudian tekan *Enter*. Tampilannya adalah seperti yang terlihat pada Gambar ?? berikut.

Aktivitas Mandiri: Kode R Pertama

Cobalah ketik perintah berikut di jendela yang terdapat kursor di dalamnya, lalu tekan Enter setelah mengetikkan kode berikut:

```
print(2 + 2)
```





Anda akan melihat angka 4 sebagai hasilnya.

Selamat! Anda baru saja menjalankan kode R pertama Anda.

Anda akan menyadari bahwa antarmuka RGui tidak bersahabat dengan pengguna (*user friendly*). Selain itu tidak ada juga alat-alat yang memudahkan kita manajemen pekerjaan seperti menampilkan lokasi file atau *script* kode sumber.

Oleh karena itu, kita membutuhkan cara kedua, yakni menggunakan **RStudio** melalui panel **Console** (jendela nomor 3 pada Gambar ??).

Aktivitas Mandiri: Kode R pertama di RStudio

Cobalah ketik perintah berikut di Console, lalu tekan Enter:

```
print(2 + 2)
```

Anda akan melihat hasil yang sama seperti yang Anda dapat di RGui. Inilah tempat menjalankan kode R di RStudio.

Selamat! Anda baru saja menjalankan kode R pertama Anda di RStudio.

0.5.2 Menggunakan *File Kode Sumber*

Namun, mengetik satu per satu di Console akan melelahkan jika perintahnya banyak. Oleh karena itu, kita biasanya menulis kode dalam sebuah file *source code* yang berekstensi (berakhiran) **.R**.

0.5.2.1 Membuat *File Kode Sumber R* (.R) File kode sumber R (berekstensi **.R**) adalah dokumen teks yang berisi rangkaian perintah R yang dapat disimpan, diedit, dan dijalankan berulang kali. Ini adalah cara profesional dalam bekerja dengan R, karena kode kita menjadi **terdokumentasi**, **dapat direproduksi**, dan **mudah dibagikan**.

Aktivitas Mandiri: Membuat *file Kode Sumber R Pertama*

1. Di RStudio, pilih menu **File** → **New File** → **R Script** (atau tekan **Ctrl + Shift + N** di Windows/Linux, **Cmd + Shift + N** di Mac)
2. Sebuah tab baru akan muncul di panel *Script* (jendela kiri atas)
3. Ketikkan kode R berikut di dalam tab tersebut

```
# File kode R pertama saya
# Dibuat pada: [isi dengan tanggal hari ini]

print("Halo dunia!")
print("Hari ini saya mulai belajar R!")

# Operasi matematika sederhana
1 + 1
2 * 3
```

4. Simpan file dengan memilih **File** → **Save** (atau **Ctrl + S** / **Cmd + S**) di direktori/folder pertama muncul di jendela Anda tersebut
5. Beri nama file dengan ekstensi **.R**, misalnya **kode_R_pertama_saya.R**

0.5.2.2 Menjalankan kode dari file .R Untuk menjalankan kode dari file .R ke Console tanpa mengetik ulang:

1. Anda bisa melakukan salah satu dari sekian cara ini:
 1. Sorot (blok) baris-baris kode yang ingin dijalankan di panel *Script*. Ini akan menjalankan kode yang Anda sorot saja.
 2. Klik angka baris di sebelah kiri untuk memilih seluruh kode yang ada dalam satu baris.
 3. **REKOMENDASI**: menempatkan kursor pada baris yang kodenya akan dijalankan. Ini akan secara otomatis meminta R menjalankan baris kode yang dianggap satu paket.
2. Tekan kombinasi tombol **Ctrl + Enter** (Windows/Linux) atau **Cmd + Enter** (Mac)
3. Kode tersebut akan otomatis dikirim ke *Console* dan dijalankan

Aktivitas Mandiri: Menjalankan Kode R dari File Kode Sumber

1. Masih dari *script* kode_R_pertama_saya.R tadi, tempatkan kursor pada baris berikut

```
...  
print("Halo dunia!")  
...
```

2. Jalankan baris tersebut dengan menekan tombol kombinasi sesuai penjelasan.
3. Kemudian, tempatkan kursor pada baris berikut

```
...  
2 * 3  
...
```

4. Jalankan baris tersebut kembali.

Ini adalah cara kerja standar yang efisien dalam menggunakan R. Dengan cara ini, kita bisa:

- Menyimpan pekerjaan untuk digunakan di lain waktu
- Menjalankan ulang analisis dengan mudah
- Berbagi kode dengan rekan kerja atau asisten praktikum
- Mendokumentasikan setiap langkah analisis kita

Pengenalan Teknis R

Sebelum memulai praktikum dengan set data sesungguhnya, ada baiknya kita memahami beberapa konsep teknis dasar dalam bahasa pemrograman R. Bagian ini merangkum keterampilan-keterampilan dasar yang akan kalian temui tersebar di modul-modul praktikum.

0.6 Pengaturan Direktori

Salah satu keterampilan penting dalam bekerja dengan R adalah mengorganisasi *file* dan data dengan baik. Sepanjang semester ini, kalian akan bekerja dengan berbagai dataset, kode R, dan hasil analisis. Pengaturan direktori yang terstruktur akan memudahkan kalian dalam mengelola semua file tersebut dan memastikan kode dapat berjalan dengan lancar.

0.6.1 Struktur Direktori yang Ideal

Sebelum memulai praktikum, kalian perlu menyiapkan direktori kerja (*working directory*) yang terorganisir dengan baik. Berikut adalah struktur direktori yang **wajib** kalian gunakan sepanjang semester:

```
Praktikum STP 2026/  
  Praktikum STP 2026.Rproj  
  datasets/  
    data.csv  
    ...  
  output/  
    output_1.xlsx  
    ...  
  modul_1.R  
  modul_2.R
```

Komponen-komponen penting:

- **Praktikum STP 2026.Rproj**: File RStudio Project yang mengatur *working directory* secara otomatis dan menyimpan *environment* kerja kalian.
- **datasets/**: Folder khusus untuk menyimpan **semua dataset mentah** (file Excel, CSV, atau format lainnya yang kalian terima dari asisten praktikum).
- **output/**: Folder khusus untuk menyimpan **hasil-hasil analisis** seperti file Excel yang sudah diolah, file CSV hasil export, atau grafik yang dihasilkan.
- **modul_1.R, modul_2.R, ...**: File-file kode R (script) yang akan kalian buat untuk setiap modul praktikum.

Dengan struktur seperti ini, kode R kalian akan lebih **ringkas** dan **portabel**. Sebagai contoh, untuk membaca data, kalian cukup menulis:

```
data <- read.xlsx("datasets/data_penduduk.xlsx")
```

Tanpa perlu menulis *path* lengkap seperti C:\Users>NamaAnda\Documents\praktikum\data_penduduk.xlsx yang panjang, rumit, dan berbeda-beda di setiap komputer.

0.6.2 Membuat dan Mengatur Direktori Praktikum

Ikuti langkah-langkah berikut untuk menyiapkan direktori praktikum kalian:

0.6.2.1 Langkah 1: Membuat Folder Utama Buatlah sebuah folder di komputer kalian dengan nama **Praktikum STP 2026** (sesuai tahun ajaran yang sedang berjalan). Penamaan yang seragam ini **sangat penting** agar asisten praktikum dapat dengan mudah membantu kalian jika terjadi masalah atau kesalahan.

Folder ini akan menjadi **direktori kerja** (*working directory*) kalian selama satu semester penuh.

0.6.2.2 Langkah 2: Membuat Subfolder datasets dan output Di dalam folder Praktikum STP 2026, buatlah **dua subfolder**:

1. Subfolder **datasets** - untuk menyimpan semua data mentah
2. Subfolder **output** - untuk menyimpan hasil analisis

Cara membuat folder:

- **Di Windows:** Klik kanan di dalam folder **Praktikum STP 2026** → pilih *New* → *Folder* → beri nama **datasets**, lalu ulangi untuk membuat folder **output**
- **Di Mac/Linux:** Klik kanan di dalam folder **Praktikum STP 2026** → pilih *New Folder* → beri nama **datasets**, lalu ulangi untuk membuat folder **output**

Setiap kali kalian mendapatkan dataset baru untuk praktikum (misalnya file Excel atau CSV dari asisten), **simpanlah di dalam folder datasets**. Setiap kali kalian menghasilkan output analisis, **simpanlah di dalam folder output**.

0.6.2.3 Langkah 3: Mengubahnya Menjadi RStudio Project Setelah struktur folder dasar sudah siap, langkah selanjutnya adalah mengubahnya menjadi **RStudio Project**. RStudio Project adalah fitur yang sangat berguna untuk mengelola sesi kerja R kalian.

Keuntungan RStudio Project:

1. **Otomatis mengatur *working directory*:** Ketika kalian membuka sebuah Project, RStudio otomatis mengatur *working directory* ke folder tersebut, sehingga kalian tidak perlu menggunakan perintah `setwd()` lagi.

Perintah `setwd()` digunakan untuk mengatur *working directory* secara manual. Misalnya, jika kita ingin mengatur *working directory* ke folder **Praktikum STP 2026** yang terletak dalam folder **Documents** pada direktori **C:/Users>NamaAnda/**, kita bisa menggunakan perintah

```
setwd("C:/Users>NamaAnda/Documents/Praktikum STP 2026")
```

Namun, jika kita sudah membuat direktori Anda sebagai Project, dalam praktikum ini, kita tidak perlu menggunakan perintah ini lagi.

2. **Mempertahankan status kerja:** RStudio menyimpan *environment* (variabel, data yang sudah dimuat) sehingga kalian bisa melanjutkan pekerjaan dari tempat terakhir kalian berhenti.
3. **Memudahkan manajemen path:** Kalian cukup menulis *relative path* (misalnya **datasets/data.xlsx**) tanpa perlu mengetik *path* lengkap yang panjang dan berisiko error.
4. **Portabilitas:** Project folder bisa dipindahkan ke komputer lain atau dibagikan ke rekan tanpa perlu mengubah kode, karena semua path bersifat relatif.

Cara membuat RStudio Project:

1. Buka RStudio
2. Pilih menu **File** → **New Project...**
3. Pilih **Existing Directory** (karena folder sudah dibuat)
4. Klik **Browse...** dan cari folder **Praktikum STP 2026** yang sudah kalian buat
5. Klik **Create Project**

Setelah itu, RStudio akan membuat file **Praktikum STP 2026.Rproj** di dalam folder kalian. File **.Rproj** ini adalah “pintu masuk” ke Project kalian.

Setiap kali ingin bekerja: Cukup *double-click* file **Praktikum STP 2026.Rproj** tersebut, dan RStudio akan terbuka dengan *working directory* yang sudah otomatis diatur ke folder praktikum kalian. Kalian siap untuk mulai bekerja!

Catatan Penting: Selalu buka RStudio melalui file **.Rproj**, bukan dengan membuka RStudio terlebih dahulu lalu membuka *script*. Ini memastikan *working directory* selalu benar dan kode kalian berjalan sebagaimana mestinya.

0.6.3 Mengunduh Dataset Praktikum dari GitHub

Setelah struktur direktori dan Project sudah siap, langkah selanjutnya adalah **mengunduh dataset-dataset praktikum** yang akan digunakan sepanjang semester. Semua dataset praktikum disimpan di *repository* GitHub dan dapat diunduh secara gratis.

Mengapa Dataset Disimpan di GitHub? {.header}

Dataset praktikum disimpan di GitHub agar: - Semua mahasiswa mendapatkan dataset yang **sama** dan **terbaru** - Dataset dapat **diakses kapan saja** tanpa perlu menunggu distribusi dari asisten - Jika ada pembaruan dataset, kalian bisa **mengunduh ulang** dengan mudah - Dataset **tidak hilang** meskipun laptop rusak atau file terhapus

0.6.3.1 Langkah-Langkah Mengunduh Dataset Ikuti langkah-langkah berikut dengan seksama untuk mengunduh folder `datasets/` dari GitHub:

1. Buka halaman GitHub repository praktikum

Klik link berikut atau copy-paste ke browser kalian:

<https://github.com/abdulmubdibindar/praktikum-r-stp>

Kalian akan melihat halaman repository yang berisi semua file praktikum.

2. Download seluruh repository sebagai file ZIP

- Di halaman GitHub, cari dan klik tombol hijau **“Code”** di bagian kanan atas
- Pada menu yang muncul, pilih **“Download ZIP”**
- Simpan file ZIP di lokasi yang mudah kalian temukan (misalnya folder **Downloads** atau **Desktop**)
- Tunggu hingga proses download selesai

3. Ekstrak file ZIP yang sudah didownload

- Buka folder tempat kalian menyimpan file ZIP (misalnya **Downloads**)
- Klik kanan pada file **praktikum-r-stp-main.zip**
- Pilih **“Extract All...”** atau **“Extract Here”** (di Windows)
- Atau untuk Mac/Linux: double-click file ZIP, akan otomatis ter-ekstrak
- Setelah ekstrak selesai, akan muncul folder bernama **praktikum-r-stp-main**

4. Copy folder `datasets` ke dalam Project RStudio kalian

Ini adalah langkah paling penting! Ikuti dengan teliti:

- Buka folder **praktikum-r-stp-main** yang baru diekstrak
- Di dalam folder tersebut, cari folder bernama **datasets**
- **Klik kanan** pada folder **datasets**, lalu pilih **“Copy”** (atau tekan **Ctrl + C**)
- Buka folder **Praktikum STP 2026** (folder Project RStudio kalian)
- **Klik kanan** di dalam folder tersebut, lalu pilih **“Paste”** (atau tekan **Ctrl + V**)
- Folder **datasets** beserta seluruh isinya sekarang sudah tersalin ke Project kalian

5. Verifikasi dataset sudah tersimpan dengan benar

Untuk memastikan dataset sudah tersimpan di lokasi yang benar, buka RStudio (klik file **.Rproj** kalian), lalu jalankan kode berikut di **Console**:

Jika hasilnya **TRUE**, berarti folder **datasets** sudah berada di lokasi yang benar!

Selanjutnya, cek isi folder `datasets` dengan kode:

Kalian akan melihat daftar file-file dataset seperti `Data Praktikum 01.xlsx`, `Data Praktikum 02.xlsx`, `DataUtama_mhsUBL.csv`, dan lain-lain.

Struktur Folder yang Benar {.header}

Pastikan struktur folder Project kalian sekarang terlihat seperti ini:

```
Praktikum STP 2026/          ← Folder Project kalian
  Praktikum STP 2026.Rproj ← File Project RStudio
    datasets/                ← Folder datasets (HARUS ada di sini!)
      Data Praktikum 01.xlsx
      Data Praktikum 02.xlsx
      Data Praktikum 03.xlsx
      DataUtama_mhsUBL.csv
      DataUtama_mhsUINRIL.csv
      DataUtama_mhsUNILA.csv
      DataUtama_mhsITERA.csv
      data_mahasiswa.csv
      ... (file-file dataset lainnya)
    output/                  ← Folder untuk hasil analisis
    ... (file .R kalian)
```

PENTING: Jika folder `datasets` berada di lokasi lain (misalnya di **Desktop** atau **Downloads**), kode praktikum **tidak akan bisa** membaca dataset. Pastikan folder `datasets` berada **langsung di dalam** folder `Praktikum STP 2026`.

0.7 Sintaks Dasar Bahasa R

Dalam subbab ini, kita akan membahas sintaks dasar bahasa R yang akan sering digunakan dalam praktikum ini.

0.7.1 Komentar

Pada dasarnya, semua yang kita tulis di *script* akan dijalankan oleh R. Akan tetapi, kita bisa memberi tahu R untuk tidak menjalankan kode yang kita tulis dengan mengubah formatnya menjadi **komentar** (*comment*). Kita bisa menuliskan komentar dengan menggunakan tanda pagar (`#`) sebelum kode yang ingin kita komentari.

```
# Kode ini akan menghitung 1 + 1
1 + 1
```

Komentar sangat berperan penting dalam kode kalian. Malahan, ini adalah kekuatan terbesar dari penggunaan R dalam Kita akan sering menggunakan komentar dalam kode kalian. Jadi, jangan ragu untuk menggunakan komentar untuk menjelaskan kode kalian.

0.7.1.1 Komentar Blok (*Block Comment*) Untuk menulis komentar yang panjang (lebih dari satu baris), kita bisa menggunakan **komentar blok**. Ada dua cara:

Cara 1: Tanda Pagar di Setiap Baris

```
# Ini adalah komentar blok
# yang terdiri dari beberapa baris
# untuk menjelaskan konsep yang kompleks
hasil <- data.ubl |>
  filter(Jenis.Kelamin == "Laki-Laki") |>
  summarize(rata_rata = mean(jarak))
```

Cara 2: Menggunakan if(FALSE)

```
if(FALSE) {
  Ini adalah komentar blok alternatif.
  Semua teks di dalam kurung kurawal tidak akan dijalankan.
  Berguna untuk men-disable blok kode sementara saat debugging.
}
```

Kapan Menggunakan Komentar Blok?

- Menulis jawaban analisis yang panjang (misalnya interpretasi hasil statistik)
- Mendokumentasikan proses pemikiran Anda
- Men-disable sementara kode yang tidak ingin dihapus
- Menjawab soal Aktivitas Mandiri dalam bentuk teks

0.7.2 Variabel

Variabel adalah tempat menyimpan data dan informasi di R. Variabel terdiri atas nama variabel dan nilainya, Cara mendeklarasikan variabel adalah dengan operator `<-`. Operator deklarasi bisa dimasukkan dengan *hotkey* `Alt + I`.

Contoh:

Meskipun kita bisa menggunakan `=` untuk meng-*assign* nilai ke variabel, praktik tersebut tidak lazim dipakai dalam komunitas R. Oleh karena itu, tetap gunakan *assignment operator* `<-` untuk memberi nilai pada suatu variabel.

Variabel juga bisa dideklarasikan menggunakan operasi matematika dan variabel sebelumnya.

```
## [1] "Hasil: 3"
```

Untuk menampilkan nilai variabel tuliskan kembali nama variabelnya lalu jalankan.

```
## [1] 1
```

```
## [1] 2
```

```
## [1] 3
```

0.7.3 Konvensi Penamaan dalam R

R mendukung beberapa gaya penamaan variabel. Penting untuk diingat bahwa **R membedakan huruf besar dan kecil** (*case-sensitive*), sehingga `data` berbeda dengan `Data`, `JenisKelamin` berbeda dengan `jeniskelamin`, dan `"Laki-Laki"` berbeda dengan `"Laki-laki"`. Kesalahan dalam kapitalisasi adalah sumber *error* yang sangat umum bagi pemula.

Konvensi Penamaan Variabel:

R mendukung beberapa gaya penamaan variabel:

1. **snake_case** (disarankan untuk variabel R):
 - `data_mahasiswa`, `jarak_tempuh`, `biaya_sepekan`
 - Menggunakan garis bawah (`_`) untuk memisahkan kata
2. **dot.notation** (umum untuk nama kolom):
 - `data.ubl`, `Jenis.Kelamin`, `biaya.dalam.sepekan`
 - Menggunakan titik (`.`) untuk memisahkan kata
 - Sering muncul saat mengimpor data dari Excel
3. **camelCase** (jarang dipakai di R):
 - `dataMahasiswa`, `jarakTempuh`
 - Huruf besar menandai awal kata baru

Rekomendasi:

- Gunakan **snake_case** untuk nama variabel/objek R yang Anda buat sendiri
- Gunakan **dot.notation** untuk nama kolom dataset (konsisten dengan output dari Excel)
- Hindari spasi dan karakter khusus dalam nama variabel
- Gunakan nama yang deskriptif: `data_ubl` lebih baik dari `d1`
- Konsisten: Pilih satu gaya dan gunakan terus sepanjang kode Anda

Aktivitas Mandiri: Membuat Variabel Disertai Komentar

1. Buatlah variabel dengan nama `nama` yang nilainya adalah nama kalian.
2. Buatlah variabel dengan nama `nim` yang nilainya adalah NIM kalian.
3. Buatlah variabel dengan nama `prodi` yang nilainya adalah program studi kalian.
4. Buatlah variabel dengan nama `angkatan` yang nilainya adalah angkatan kalian.
5. Tambahkan komentar pada baris paling atas bahwa variabel-variabel di bawahnya adalah “Identitas Saya”.
6. Tampilkan nilai variabel `nama`, `nim`, `prodi`, dan `angkatan` dengan perintah berikut.

```
cat("Nama saya adalah",nama,". NIM saya adalah",nim,". Saya adalah mahasiswa",prodi," angkatan",angkatan)
```

0.8 Paket (*Packages*)

R memiliki ribuan paket tambahan yang menyediakan fungsi-fungsi khusus.

- **`install.packages("nama_paket")`**: Digunakan untuk mengunduh dan menginstal paket dari internet ke komputer kita. Ini cukup dilakukan satu kali saja. **Perhatikan bahwa nama paket ditulis dengan tanda " di awal dan akhir nama paket yang akan diinstal.**
- **`library(nama_paket)`**: Digunakan untuk memuat paket yang sudah terinstal ke dalam sesi kerja kita agar isinya bisa digunakan. Ini harus dilakukan setiap kali kita membuka RStudio baru. **Perhatikan bahwa tanda " tidak perlu lagi ditulis ketika kita ingin memuat paket yang sudah kita instal.**

Aktivitas Mandiri: Instalasi Paket

Berikut adalah paket-paket yang akan kita gunakan selama praktikum ini. Instalasilah paket-paket berikut di komputer kalian:

- `tidyverse`

- openxlsx
- gtsummary
- flextable
- treemapify
- MKinfer
- rcompanion
- RCPA3
- vcdExtra
- psych

Tunggu hingga masing-masing proses instalasi selesai. Ingat, tahap ini hanya perlu dilakukan satu kali saja.

0.9 Fungsi (*Functions*)

Fungsi adalah serangkaian instruksi yang dibungkus menjadi satu perintah untuk melakukan tugas tertentu. R memiliki banyak fungsi bawaan yang berguna untuk banyak hal, seperti `print()` untuk menampilkan output, `mean()` untuk menghitung rata-rata, atau `sum()` untuk menghitung jumlah.

0.9.1 Menggunakan Fungsi

Untuk menggunakan fungsi, kita memanggil namanya diikuti dengan tanda kurung (). Di dalam tanda kurung, kita bisa memasukkan **argumen** (input).

```
# Contoh fungsi bawaan: mean (rata-rata)
nilai <- c(10, 20, 30)
rata_rata <- mean(nilai) # Hasil: 20
```

0.9.2 Membuat Fungsi Sendiri

Kita dapat membuat fungsi buatan kita sendiri menggunakan perintah `function()`. Struktur dasarnya adalah sebagai berikut:

```
nama_fungsi <- function(input) {
  # Proses yang dilakukan
  hasil <- input * 2
  return(hasil)
}
```

Penjelasan:

- `nama_fungsi`: Nama yang akan kita gunakan untuk memanggil fungsi ini.
- `function(input)`: Mendefinisikan bahwa kita membuat sebuah fungsi yang menerima satu argumen bernama `input`.
- `{ ... }`: Blok kode yang berisi instruksi yang akan dijalankan.
- `return(hasil)`: Mengembalikan nilai `hasil` sebagai *output* dari fungsi.

Aktivitas Mandiri: Membuat Fungsi Sendiri

1. Buatlah dua variabel dengan nama `c` dan `d` dengan nilai 10 dan 20
2. Buatlah fungsi dengan nama `kali_dua` yang menerima satu argumen `x` sebagai berikut:

```
kali_dua <- function(x) {
  # Proses yang dilakukan
  hasil <- x * 2
  return(hasil)
}
```

3. Jalankan fungsi tersebut dengan argumen nilai `c` dan `d` dan simpan nilainya berturut-turut pada variabel `c_kali2` dan `d_kali2`.
4. Tampilkan nilai variabel `c_kali2` dan `d_kali2`.

Ini akan sangat berguna ketika kita ingin melakukan perhitungan yang sama berulang kali, seperti pada perhitungan IQV di Modul 2.

0.10 Struktur Data Dasar

Kita perlu mengenal jenis-jenis struktur data yang ada di R. Mereka akan sangat sering kita gunakan selama praktikum ini.

0.10.1 Vektor

Vektor adalah struktur data paling dasar di R yang dapat menampung lebih dari satu nilai. Kita membuatnya dengan fungsi `c()` (*combine*).

Ciri utama vektor adalah semua elemen di dalamnya harus memiliki tipe data yang sama. Jika tidak, R akan mengubah tipe data elemen tersebut menjadi tipe data yang paling umum.

```
angka <- c(1, 2, 3, 4, 5)
huruf <- c("a", "b", "c")
campuran <- c(1, "a", TRUE) # R akan mengubah tipe data elemen tersebut menjadi tipe data yang paling umum
```

Aktivitas Mandiri: Membuat Vektor

1. Buatlah vektor dengan nama `nilai_ujian` yang berisi nilai ujian 5 mahasiswa berikut: 85, 90, 78, 92, 88
2. Buatlah vektor dengan nama `fakultas` yang berisi nama-nama fakultas di ITERA
3. Buatlah vektor dengan nama `apa_saja` yang berisi nilai 1 dan "a"
4. Tampilkan tipe data dari variabel `nilai_ujian`, `fakultas`, dan `apa_saja`. Perhatikan hasil pada bagian `apa_saja`. Apa jenis nilai-nilai dalam vektor tersebut?

0.10.2 List

List adalah struktur data yang dapat menampung banyak nilai yang berbeda tipe datanya.

```
list_saya <- list("kucing", 1, TRUE)
list_saya # Untuk menampilkan nilai list_saya
```

List juga bahkan dapat menampung vektor.

```
list_saya <- list(c(1, 2, 3), c("a", "b", "c"), c(TRUE, FALSE, TRUE))
list_saya
```

Kita dapat memberi nama elemen-elemen dalam list dengan mengetikkan nama elemen dan diikuti tanda = sebelum nilai elemen tersebut.

```
list_saya <- list(nama = "Budi", nilai_ujian = c(85, 90, 78), fakultas = "FTI")
```

Aktivitas Mandiri: Membuat List

1. Buatlah list dengan nama `list_saya` yang berisi nama, nilai ujian, dan fakultas
2. Tampilkan nilai variabel `list_saya`

0.10.3 Tipe Data

Saat mengolah data, kalian akan sering menemukan singkatan tipe data berikut:

- **dbl (Double)**: Angka numerik (bisa desimal).
- **chr (Character)**: Teks atau *string*. Ditandai dengan tanda kutip.
- **fact (Factor)**: Data kategorik (bisa nominal atau ordinal). Factor menyimpan nilai sebagai angka di belakang layar namun menampilkan label teks, yang berguna untuk analisis statistik dan plot.

0.10.4 Data Frame dan Tibble

Data dalam R biasanya disimpan dalam bentuk tabel yang disebut *Data Frame*. Di era modern (tidyverse), kita sering menggunakan versi yang lebih canggih disebut **Tibble**. Strukturnya mirip spreadsheet: baris mewakili observasi, kolom mewakili variabel.

0.11 Operator Pipa (|>)

Dalam **tidyverse**, kita sering menggunakan operator pipa (`|>`) untuk merangkai beberapa perintah sekaligus. Bayangkan ini sebagai penyambung proses:

```
# Tanpa pipa
hasil <- fungsi_2(fungsi_1(data))

# Dengan pipa (lebih mudah dibaca)
data |>
  fungsi_1() |>
  fungsi_2()
```

Artinya: “Ambil `data`, lalu `fungsi_1`, kemudian hasilnya dimasukkan ke `fungsi_2`.”

Kita akan mempelajari penggunaan operator pipa (`|>`) secara lebih praktis dalam ??.

0.12 Manajemen Data (Input/Output)

Ini adalah kemampuan yang penting dalam pengolahan data di R karena seringkali kita perlu mengolah data dari file eksternal (seperti Excel) atau menyimpan data hasil pengolahan ke file eksternal.

Kita akan mempraktikkan kemampuan ini lebih banyak mulai di ??.