

Bayu Widodo

31 January 2026

Daftar Isi

2	# Statistika Dasar dan Komputasi dengan R	11
2.1	Capaian Pembelajaran Bab 2	11
2.2	Pengertian dan Ruang Lingkup Statistika	11
2.2.1	Pengertian Statistika	11
2.2.2	Ruang Lingkup Statistika dalam Sains	11
2.2.3	Ruang Lingkup Statistika dalam Teknologi	12
2.2.4	Ruang Lingkup Statistika dalam Teknologi	12
2.2.5	Ruang Lingkup Statistika dalam Komputasi	12
2.3	Statistika Deskriptif dan Statistika Inferensial	13
2.3.1	Pengertian Statistika Deskriptif	13
2.3.2	Pengertian Statistika Inferensial	13
2.4	Berpikir Statistik untuk Pemecahan Masalah Berbasis Data	14
2.5	Skala Pengukuran Variabel	14
2.5.1	Pengertian Variabel dan Skala Pengukuran	14
2.6	Komputasi Statistik untuk Analisis Data	17
2.7	R dan RStudio	17
2.7.1	Instalasi R dan RStudio di Windows	17
2.7.2	Mengunduh dan Memasang RStudio	18
2.7.3	Verifikasi Instalasi	18
2.7.4	Mengenai Antarmuka RStudio	18
2.8	Pembuatan dan Eksekusi Skrip R	20
2.9	Objek di R	21
2.9.1	Karakteristik dan Keunggulan Bahasa R	23
2.10	Paket R	24
2.11	Memulai Bahasa R	25
2.11.1	R Session	25
2.11.2	Demo di R	26
2.11.3	R Sebagai Kalkulator	26
2.12	Instruksi Terpandu	27
2.13	Latihan Mandiri	29
2.14	Menjalankan Perintah Dasar R	30
2.15	Fungsi Dasar di R	32
2.16	Kesimpulan	34

Daftar Gambar

1	Tampilan Antarmuka RStudio	19
2	Session di R	25
3	Diagram Hirarki Konsep Statistika	36

2 # Statistika Dasar dan Komputasi dengan R

2.1 Capaian Pembelajaran Bab 2

Setelah mempelajari Bab 2, mahasiswa diharapkan mampu:

1. Menjelaskan konsep dasar statistika
 - Memahami pengertian statistika, ruang lingkup dalam sains, teknologi, dan komputasi.
 - Membedakan statistika deskriptif dan inferensial beserta kegunaannya.
2. Menerapkan berpikir statistik untuk pemecahan masalah berbasis data
 - Mengidentifikasi jenis data dan skala pengukuran variabel.
 - Menentukan skala data yang sesuai untuk analisis statistik sederhana.
3. Menguasai penggunaan R dan RStudio untuk analisis data dasar
 - Menginstal dan memverifikasi R dan RStudio.
 - Menjelajahi antarmuka RStudio serta memahami lingkungan kerja.
 - Menjalankan perintah dasar R dan fungsi-fungsi sederhana untuk manipulasi data.
4. Melakukan eksplorasi data sederhana menggunakan R
 - Membuat dan menampilkan vektor, matriks, dan data frame.
 - Melakukan ringkasan data sederhana (summary, head, tail).
 - Menyelesaikan latihan praktikum mandiri dengan bimbingan instruksi terpandu.
5. Menarik kesimpulan dari analisis data dasar menggunakan R
 - Menginterpretasikan output sederhana dari perintah R.
 - Mengaitkan hasil eksplorasi data dengan konsep statistika deskriptif.

Kuliah / Teori :

2.2 Pengertian dan Ruang Lingkup Statistika

2.2.1 Pengertian Statistika

Statistika adalah cabang ilmu yang mempelajari cara pengumpulan, pengolahan, analisis, interpretasi, dan penyajian data. Secara sederhana, statistika membantu kita mengubah data mentah menjadi informasi yang berguna untuk pengambilan keputusan. Dalam konteks sains dan teknologi, statistika bukan hanya soal menghitung rata-rata atau persentase, tetapi juga mencakup model matematis dan metode inferensi yang dapat memperkirakan tren, mengidentifikasi pola, dan memprediksi kejadian berdasarkan data yang tersedia. Gambaran keterkaitan antara konsep-konsep utama dalam statistika, mulai dari statistika deskriptif hingga statistika inferensial beserta penerapannya dalam pengambilan keputusan, disajikan pada Lampiran Bab ini.

Beberapa istilah kunci dalam statistika:

1. Populasi: Keseluruhan objek atau individu yang menjadi subjek penelitian.
2. Sampel: Bagian dari populasi yang diamati untuk mengambil kesimpulan.
3. Parameter: Nilai karakteristik dari populasi.
4. Statistik: Nilai karakteristik yang dihitung dari sampel.

2.2.2 Ruang Lingkup Statistika dalam Sains

Dalam dunia sains, statistika digunakan untuk:

1. Deskripsi Data (Descriptive Statistics)
 - Meringkas dan menyajikan data melalui mean, median, modus, standar deviasi, dan visualisasi grafik.
 - Contoh: Menentukan rata-rata pertumbuhan tanaman pada penelitian agronomi.
2. Inferensi Statistik (Inferential Statistics)
 - Membuat kesimpulan atau prediksi tentang populasi berdasarkan sampel.
 - Contoh: Mengestimasi efek obat baru terhadap populasi pasien.
3. Eksperimen dan Perancangan Percobaan (Experimental Design)
 - Mengatur percobaan sehingga data yang diperoleh valid dan dapat diulang.
 - Contoh: Mengatur uji coba laboratorium untuk memastikan perbedaan hasil bukan karena kebetulan.

2.2.3 Ruang Lingkup Statistika dalam Teknologi

Dalam teknologi dan industri, statistika menjadi fondasi data-driven decision making:

1. Kontrol kualitas: Menggunakan statistik untuk memantau dan meningkatkan kualitas produk (misal Six Sigma).
2. Analisis big data: Mengolah data besar untuk menemukan pola dan tren penting.
3. Machine learning & AI: Statistika menjadi dasar algoritma prediksi, klasifikasi, dan clustering.

2.2.4 Ruang Lingkup Statistika dalam Teknologi

Dalam teknologi dan industri, statistika menjadi fondasi data-driven decision making:

1. Kontrol kualitas: Menggunakan statistik untuk memantau dan meningkatkan kualitas produk (misal Six Sigma).
2. Analisis big data: Mengolah data besar untuk menemukan pola dan tren penting.
3. Machine learning & AI: Statistika menjadi dasar algoritma prediksi, klasifikasi, dan clustering.

2.2.5 Ruang Lingkup Statistika dalam Komputasi

Komputasi modern memperluas aplikasi statistika melalui:

1. Simulasi komputer: Menjalankan eksperimen virtual dengan berbagai skenario untuk memprediksi hasil.
2. Pemodelan probabilistik: Misal, Bayesian inference dalam prediksi dan pengambilan keputusan.
3. Analisis data berbasis R, Python, atau MATLAB: Mempermudah perhitungan statistik kompleks dan visualisasi data.

Statistika bukan sekadar “ilmu menghitung”, tetapi alat penting untuk memahami dunia nyata melalui data. Dalam sains, teknologi, dan komputasi, statistika memungkinkan:

1. Menemukan pola dan hubungan dalam data.
2. Membuat prediksi yang berbasis bukti.
3. Mendukung pengambilan keputusan secara sistematis dan akurat.

2.3 Statistika Deskriptif dan Statistika Inferensial

2.3.1 Pengertian Statistika Deskriptif

Statistika deskriptif adalah cabang statistika yang berfokus pada pengolahan, penyajian, dan ringkasan data agar mudah dipahami. Tujuan utamanya adalah menggambarkan kondisi atau karakteristik data secara jelas tanpa membuat kesimpulan tentang populasi lebih luas. Beberapa contoh metode statistika deskriptif:

1. Ukuran pemusatan data: mean, median, modus
2. Ukuran penyebaran data: range, variansi, standar deviasi
3. Visualisasi data: histogram, diagram batang, boxplot

Contoh aplikasi: Menentukan rata-rata nilai ujian mahasiswa di satu kelas, atau melihat distribusi umur penduduk di satu desa.

2.3.2 Pengertian Statistika Inferensial

Statistika inferensial adalah cabang statistika yang mengambil kesimpulan atau membuat prediksi tentang populasi berdasarkan data sampel. Statistika inferensial memungkinkan kita untuk:

1. Mengestimasi parameter populasi
2. Menguji hipotesis
3. Membuat prediksi atau model probabilistik

Contoh metode statistika inferensial:

- Estimasi parameter: interval kepercayaan untuk rata-rata populasi
- Uji hipotesis: t-test, chi-square, ANOVA
- Regresi dan korelasi: memprediksi hubungan antar variabel

Contoh aplikasi: Menentukan apakah rata-rata nilai ujian semua mahasiswa di universitas berbeda dengan standar tertentu, berdasarkan sampel.

Tabel 1: Perbedaan dan Keterkaitan Statistika Deskriptif dan Inferensial

Aspek	Statistika Deskriptif	Statistika Inferensial
Tujuan	Mendeskripsikan data	Menarik kesimpulan atau prediksi dari sampel ke populasi
Data yang digunakan	Seluruh data atau sampel	Biasanya sampel untuk estimasi populasi
Hasil	Ringkasan data, tabel, grafik	Estimasi parameter, uji hipotesis, prediksi
Contoh	Rata-rata, median, histogram	Interval kepercayaan, t-test, regresi

2.4 Berpikir Statistik untuk Pemecahan Masalah Berbasis Data

Berpikir statistik (statistical thinking) adalah pendekatan sistematis dalam mengumpulkan, menganalisis, dan menafsirkan data untuk memecahkan masalah secara logis dan berbasis bukti. Konsep ini menekankan bahwa data bukan sekadar angka, tetapi sumber informasi yang dapat membantu pengambilan keputusan yang lebih tepat dan akurat.

Dalam konteks pemecahan masalah, berpikir statistik melibatkan beberapa prinsip dasar:

1. Memahami Variabilitas – Data selalu menunjukkan variasi. Mengidentifikasi pola dan penyebab variasi membantu kita memahami fenomena yang sedang diteliti.
2. Menggunakan Sampel untuk Menyimpulkan Populasi – Tidak selalu mungkin mengamati seluruh populasi, sehingga sampel digunakan untuk membuat kesimpulan yang sah.
3. Membuat Keputusan Berdasarkan Data – Keputusan dibuat dengan mempertimbangkan bukti statistik, bukan hanya intuisi atau asumsi.
4. Iterasi dan Evaluasi – Analisis data sering bersifat iteratif: kita terus memeriksa asumsi, mengevaluasi hasil, dan memperbaiki pendekatan sesuai kebutuhan.

Dengan mengembangkan kemampuan berpikir statistik, mahasiswa dan praktisi dapat menganalisis masalah secara lebih kritis, menyusun strategi berbasis bukti, dan meminimalkan kesalahan dalam pengambilan keputusan. Dalam era data saat ini, kemampuan ini menjadi salah satu kompetensi kunci dalam sains, teknologi, dan komputasi.

2.5 Skala Pengukuran Variabel

2.5.1 Pengertian Variabel dan Skala Pengukuran

Dalam statistika dan analisis data, variabel didefinisikan sebagai karakteristik atau atribut dari suatu objek, individu, atau fenomena yang dapat diamati dan diukur. Fokus utama dalam desain penelitian dan analisis statistik terletak pada kajian terhadap variabel, karena seluruh tahapan pengolahan data—mulai dari pengumpulan, penyajian, hingga analisis—sangat ditentukan oleh bagaimana variabel tersebut didefinisikan dan diukur.

Pengukuran variabel adalah proses pemberian nilai atau atribut pada suatu objek. Skala pengukuran variabel merupakan seperangkat aturan atau prosedur dalam pemberian nilai, simbol, atau angka terhadap suatu objek atau fenomena berdasarkan karakteristik tertentu. Proses ini dikenal sebagai definisi operasional variabel, yaitu penjabaran tentang bagaimana suatu konsep yang bersifat abstrak diukur secara konkret dan dapat diobservasi.

Ketidaktepatan dalam menentukan skala pengukuran variabel dapat menimbulkan konsekuensi metodologis yang signifikan, antara lain dalam pemilihan metode analisis statistik, penentuan jenis visualisasi data, serta pemilihan struktur data yang digunakan dalam perangkat lunak statistik, seperti R. Oleh karena itu, pemahaman yang tepat mengenai skala pengukuran variabel menjadi fondasi yang penting sebelum mempelajari pemrograman R dan struktur data dalam R.

Menurut Stanley Smith Stevens (1946), skala pengukuran variabel dalam statistika diklasifikasikan ke dalam empat jenis, yaitu skala nominal, skala ordinal, skala interval, dan skala rasio. Keempat jenis skala pengukuran tersebut tersusun secara hierarkis, di mana setiap tingkat skala yang lebih tinggi mencakup seluruh karakteristik skala pada tingkat di bawahnya. Skala nominal merupakan tingkat pengukuran paling dasar, sedangkan skala rasio merupakan tingkat pengukuran paling tinggi karena memiliki karakteristik yang paling lengkap.

Fokus utama dalam desain penelitian dan analisis statistik adalah kajian terhadap variabel. Ketika peneliti bermaksud mempelajari suatu fenomena, langkah awal yang harus dilakukan adalah mendefinisikan fenomena tersebut secara jelas, yaitu dengan menentukan variabel-variabel yang akan diamati serta menetapkan bagaimana variabel tersebut diukur. Proses penetapan cara pengukuran variabel ini dikenal sebagai definisi operasional.

Pemahaman yang baik terhadap konsep variabel dan skala pengukuran merupakan prasyarat penting untuk memahami suatu fenomena secara ilmiah. Ketidakjelasan dalam menentukan cara pengukuran variabel yang diteliti dapat menyebabkan kesulitan dalam merancang desain penelitian yang tepat, sekaligus menimbulkan kekeliruan dalam pemilihan prosedur analisis statistik yang sesuai.

Setelah variabel penelitian ditentukan, langkah berikutnya adalah menetapkan cara pengukurannya. Pengukuran merupakan dasar dari kegiatan penelitian, karena seluruh proses penelitian dimulai dengan mengukur objek yang akan dipelajari. Secara sederhana, pengukuran dapat dipahami sebagai proses pemberian angka atau kode pada suatu objek berdasarkan aturan tertentu.

Tabel~2 menunjukkan contoh dataset yang terdiri atas beberapa variabel dengan skala pengukuran yang berbeda, mulai dari nominal hingga ordinal. Dataset ini digunakan sebagai ilustrasi dalam menjelaskan konsep skala data serta penerapannya dalam analisis statistika.

Tabel 2: Contoh Data Karakteristik Individu dan Hasil Evaluasi

No.	Nama	Jenis Kelamin	Warna Kulit	Perilaku / Sikap	Suhu Tubuh (°C)	Berat Badan	Ujian	Peringkat
1	Barb	P	Hitam	80	36	60	100	1
2	Chris	L	Coklat	48	35	65	96	2.5
3	Bonnie	P	Putih	74	36	55	96	2.5
4	Robert	L	Kuning	35	37	57	93	4
5	Jim	L	Merah	79	35	70	92	5
6	Tina	P	Putih	60	34	45	89	7
7	Ron	L	Hitam	40	36	67	89	7
8	Jeff	L	Coklat	56	37	58	89	7
9	Brenda	P	Coklat	74	35	50	88	9
10	Mark	L	Putih	56	37	100	82	10
11	Mike	L	Kuning	65	36	90	75	11
Skala Data		Nominal	Nominal	Interval	Interval	Rasio	Rasio	Ordinal

Perhatikan nilai/data yang terdapat pada Variabel Jenis Kelamin. Apakah anda bisa membedakan antara jenis kelamin laki-laki dan perempuan? Tentu Anda bisa! Barb adalah perempuan sedangkan Cris laki-laki. Disini kita bisa menentukan antara yang jenis kelaminnya sama (=) dan yang berbeda (). Dapatkah Anda mengurutkan atau membuat rangking? $L > P$ atau $L < P$? Tentu kita tidak dapat membuat peringkatnya! Demikian juga dengan warna kulit, disini kita hanya bisa membedakan tanpa bisa membuat peringkatnya!

Kita hanya bisa membedakan atau mengkatagorikan nilai/kode dari variabel tersebut, namun kita tidak mungkin merangkingnya. Skala pengukuran demikian dinamakan Nominal.

Sekarang perhatikan variabel Huruf Mutu. Barb dan Chris memperoleh huruf mutu yang sama, yaitu A, sedangkan Tina memperoleh huruf mutu B. Pada variabel ini, kita tidak hanya dapat membedakan nilai yang sama dan berbeda, tetapi juga dapat mengurutkannya, di mana nilai A lebih baik daripada B. Hal yang sama berlaku pada variabel Peringkat, di mana peringkat 1 lebih baik daripada peringkat 11. Variabel yang dapat dibedakan dan diurutkan seperti ini menggunakan skala ordinal.

Selanjutnya, perhatikan variabel Perilaku dan Suhu Tubuh. Selain dapat dibedakan dan diurutkan, kedua variabel ini juga memiliki jarak (interval) yang sama antar nilainya. Sebagai contoh, selisih nilai perilaku 65 dan 60 sama dengan selisih antara 60 dan 55, yaitu 5 poin. Demikian pula, selisih suhu 35 °C dan 36 °C sama dengan selisih 100 °C dan 101 °C, yaitu 1 °C. Oleh karena itu, variabel perilaku dan suhu tubuh termasuk dalam skala interval.

Berbeda dengan skala interval, pada skala rasio nilai variabel memiliki nol mutlak, sehingga memungkinkan dilakukan perbandingan rasio. Contohnya pada variabel Berat Badan, berat Mike (90 kg) dua kali lebih besar dibandingkan Tina (45 kg). Hal ini tidak dapat dilakukan pada variabel perilaku maupun suhu Celsius, karena keduanya tidak memiliki nol mutlak. Variabel yang memenuhi karakteristik tersebut, seperti berat badan dan nilai ujian, termasuk dalam skala rasio.

Tabel 3: Ringkasan Skala Pengukuran Variabel

No.	Skala	Karakteristik	Contoh Variabel	Operator Aritmetika
1	Nominal	Hanya klasifikasi/kategori, tidak dapat diurutkan	Jenis kelamin, warna kulit, agama, golongan darah	=, ≠
2	Ordinal	Dapat diurutkan/peringkat, tetapi jarak antar nilai tidak bermakna	Status sosial ekonomi, peringkat kelas, tingkat keparahan penyakit	=, ≠, <, >
3	Interval	Dapat diurutkan dan mengukur perbedaan/interval antar nilai; tidak memiliki nol mutlak	Suhu (°C/°F), tingkat kecerdasan (IQ)	=, ≠, <, >, +, -
4	Rasio	Memiliki semua karakteristik interval dan nol mutlak sehingga memungkinkan perbandingan rasio	Berat badan, tinggi badan, usia, suhu (Kelvin)	=, ≠, <, >, +, -, ×, ÷

Skala pengukuran variabel penting untuk penentuan uji statistik yang sesuai: skala nominal dan ordinal hanya bisa menggunakan uji statistik non parametrik, sedangkan skala interval dan rasio bisa menggunakan statistik parametrik.

2.6 Komputasi Statistik untuk Analisis Data

Lingkungan komputasi statistik adalah platform perangkat lunak dan alat analisis yang memungkinkan peneliti dan praktisi untuk mengelola, memproses, menganalisis, dan memvisualisasikan data secara efisien. Dalam era data modern, di mana jumlah data semakin besar dan kompleks, lingkungan komputasi statistik menjadi fondasi penting untuk pengolahan data yang cepat, akurat, dan reproducible.

Beberapa peran utama lingkungan komputasi statistik antara lain:

1. Pemrosesan Data yang Efisien – Memudahkan pengolahan dataset besar, termasuk pemberian data, transformasi, dan penggabungan dari berbagai sumber.
2. Analisis Statistik dan Pemodelan – Mendukung berbagai metode statistik, mulai dari deskriptif, inferensial, hingga pemodelan lanjutan seperti regresi, clustering, dan machine learning.
3. Visualisasi Data – Menyediakan alat untuk membuat grafik dan diagram interaktif yang membantu interpretasi dan komunikasi hasil analisis.
4. Reproducibility dan Otomatisasi – Kode dan skrip analisis memungkinkan analisis dapat diulang dan diperiksa secara sistematis, sehingga meningkatkan kredibilitas hasil penelitian.

Dengan memanfaatkan lingkungan komputasi statistik seperti R, Python, atau MATLAB, mahasiswa dan peneliti dapat mengubah data mentah menjadi informasi yang bermakna, mendukung pengambilan keputusan berbasis bukti, dan mempercepat proses inovasi dalam sains dan teknologi modern.

Praktikum :

Saat mempelajari R, dunia data menjadi lebih dari sekadar angka. R adalah alat yang memungkinkan untuk mengurai kompleksitas data, menemukan pola tersembunyi, dan menghasilkan wawasan yang mendalam, sehingga mendukung pengambilan keputusan yang cerdas.

2.7 R dan RStudio

Sebelum melakukan analisis data, langkah pertama yang krusial adalah mempersiapkan lingkungan kerja (environment). Terdapat dua komponen utama yang perlu dipasang: R sebagai bahasa pemrograman dan mesin komputasi, serta RStudio sebagai Integrated Development Environment (IDE) yang memudahkan pengguna dalam menulis kode dan mengelola proyek.

Catatan penting

Selalu pastikan untuk memasang R terlebih dahulu sebelum memasang RStudio. Hal ini bertujuan agar RStudio dapat mendeteksi keberadaan bahasa R di dalam sistem secara otomatis saat dijalankan pertama kali.

2.7.1 Instalasi R dan RStudio di Windows

R didistribusikan melalui jaringan situs web yang disebut CRAN (Comprehensive R Archive Network). Ikuti langkah-langkah berikut:

- Akses Situs CRAN: Buka peramban dan kunjungi laman <https://cran.r-project.org/>.
- Pilih Sistem Operasi: Pilih tautan yang sesuai dengan sistem operasi Anda (Download R for Windows, macOS, atau Linux).

- Unduh Berkas Instalasi: Untuk pengguna Windows, pilih sub-direktori “base” lalu klik “Download R x.x.x for Windows”.
- Proses Instalasi: Jalankan berkas .exe atau .pkg yang telah diunduh. Ikuti instruksi pada wisaya instalasi (installation wizard) dengan memilih opsi standar hingga selesai.

2.7.2 Mengunduh dan Memasang RStudio

Setelah R terpasang, langkah selanjutnya adalah memasang RStudio agar antarmuka kerja menjadi lebih intuitif.

- Akses Situs Posit: Kunjungi laman <https://posit.co/download/rstudio-desktop/>.
- Pilih Versi Gratis: Pilih RStudio Desktop (Free) dan klik tombol unduh.
- Identifikasi Versi OS: Situs biasanya akan secara otomatis menyarankan versi yang sesuai dengan sistem operasi Anda. Klik tombol “Install RStudio”.
- Eksekusi Instalasi: Jalankan berkas instalasi dan ikuti prosedur pemasangan hingga selesai.

2.7.3 Verifikasi Instalasi

Untuk memastikan kedua perangkat lunak telah terintegrasi dengan benar, ikuti langkah verifikasi berikut:

- Buka aplikasi RStudio.
- Pada panel Console, ketikkan perintah berikut dan tekan Enter:

```
1 version
```

- Jika sistem menampilkan detail versi R (misalnya “R version 4.x.x”), maka instalasi telah berhasil dan perangkat siap digunakan untuk analisis statistik.

2.7.4 Mengenal Antarmuka RStudio

Setelah proses instalasi selesai dan aplikasi dijalankan, Anda akan melihat antarmuka RStudio yang terbagi menjadi empat panel utama. Memahami fungsi setiap panel adalah kunci efisiensi dalam bekerja dengan R.

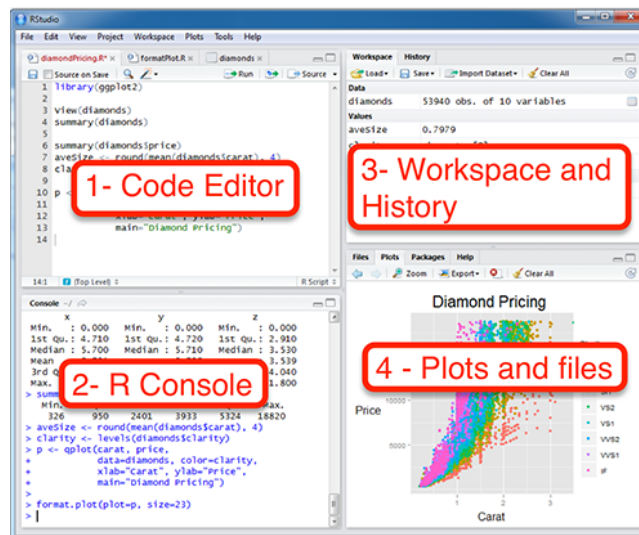
Tabel 4: Panel Utama pada Antarmuka RStudio

No.	Nama Panel	Fungsi Utama	Posisi
1	Source Editor	Tempat menulis, mengedit, dan menyimpan skrip R. Kode yang ditulis pada panel ini tidak langsung dijalankan sebelum diperintahkan secara eksplisit oleh pengguna.	Kiri Atas
2	Console	Merupakan inti (jantung) dari R. Pada panel ini, perintah R dieksekusi dan hasil perhitungan, pesan sistem, maupun kesalahan (error) ditampilkan secara langsung.	Kiri Bawah

Bersambung ke halaman berikutnya

No.	Nama Panel	Fungsi Utama	Posisi
3	Environment / History	Menampilkan daftar objek seperti data, variabel, dan fungsi yang sedang aktif di memori. Tab <i>History</i> mencatat riwayat perintah R yang telah dijalankan.	Kanan Atas
4	Files / Plots / Packages / Help	Panel multifungsi untuk mengelola berkas, menampilkan grafik hasil visualisasi data, mengelola paket (library) R, serta mengakses dokumentasi dan bantuan.	Kanan Bawah

Sebagai pelengkap Tabel 4, Gambar 1 memperlihatkan tampilan antarmuka utama RStudio secara visual. Gambar tersebut menunjukkan pembagian layar RStudio ke dalam empat panel utama sebagaimana telah dijelaskan sebelumnya, yaitu Source Editor, Console, Environment/History, serta Files/Plots/Packages/Help. Dengan memahami keterkaitan antara posisi panel pada gambar dan fungsi masing-masing panel pada tabel, pengguna diharapkan dapat lebih cepat beradaptasi dan bekerja secara efisien dalam lingkungan RStudio, khususnya saat menulis skrip, mengeksekusi perintah, serta menganalisis dan memvisualisasikan data.



Gambar 1: Tampilan Antarmuka RStudio

(a) Code Editor: Lembar Kerja Skrip

Panel ini berfungsi seperti aplikasi pengolah kata, namun khusus untuk kode pemrograman. Menggunakan Source Editor sangat disarankan dibandingkan mengetik langsung di Console, karena Anda dapat memperbaiki kesalahan, menyimpan pekerjaan dalam format .R, dan menjalankan ulang seluruh analisis di masa mendatang.

(b) Console: Interaksi Langsung

Jika Anda ingin melakukan uji coba cepat atau perhitungan sederhana, Anda dapat mengetik-kannya langsung di sini. Setiap perintah yang diawali dengan simbol > menunjukkan bahwa R siap menerima instruksi baru.

(c) Environment: Manajemen Objek

Panel Environment menampilkan seluruh objek yang sedang aktif di memori (workspace), seperti data, variabel, dan fungsi yang dibuat atau diimpor selama sesi kerja R. Panel ini membantu pengguna memantau objek yang digunakan tanpa harus memanggilnya kembali melalui Console. Workspace bersifat sementara selama sesi berjalan dan dapat disimpan untuk digunakan kembali pada sesi berikutnya.

Tab History mencatat seluruh perintah R yang telah dijalankan. Fitur ini memudahkan pengguna untuk meninjau, mengulang, atau mendokumentasikan proses analisis data secara lebih efisien dan sistematis.

(d) Panel Output (Files, Plots, Help)

Tab Plots menampilkan hasil visualisasi data secara otomatis setiap kali perintah pembuatan grafik dijalankan. Tab Files digunakan untuk menelusuri dan mengelola berkas pada direktori kerja, sehingga memudahkan pengaturan data dan hasil analisis.

(e) Help: Fitur krusial bagi pemula. Jika Anda bingung dengan fungsi tertentu, Anda dapat mengetikkan tanda tanya diikuti nama fungsi di Console (contoh: `?mean`), dan penjelasannya akan muncul di sini.

Pada tahap awal praktikum, mahasiswa diharapkan melakukan pengenalan terhadap lingkungan kerja R yang digunakan dalam proses analisis data. Langkah-langkah yang perlu dilakukan adalah sebagai berikut:

1. Jalankan aplikasi RStudio sebagai lingkungan pengembangan terintegrasi (Integrated Development Environment/IDE) untuk bahasa pemrograman R.
2. Untuk mengetahui direktori kerja (working directory) yang sedang aktif, ketikkan perintah berikut pada panel Console RStudio:

```
1 getwd()
```

Setelah perintah `getwd()` dijalankan, R akan menampilkan alamat direktori kerja (working directory) yang sedang aktif. Direktori kerja ini merupakan lokasi default yang digunakan R untuk membaca berkas masukan (input) dan menyimpan berkas keluaran (output), seperti skrip R, data, maupun hasil analisis.

Pemahaman terhadap working directory menjadi penting karena seluruh operasi terkait berkas di R, seperti pemanggilan data menggunakan fungsi `read.csv()` atau penyimpanan hasil analisis dengan fungsi `write.csv()`, secara otomatis akan mengacu pada direktori tersebut apabila tidak ditentukan lokasi lain secara eksplisit.

Apabila direktori kerja yang aktif belum sesuai dengan lokasi penyimpanan berkas praktikum, mahasiswa dapat mengatur ulang working directory melalui menu: Session > Set Working Directory > Choose Directory. Jika menggunakan R Console cukup menuliskan:

```
1 setwd('path')
```

2.8 Pembuatan dan Eksekusi Skrip R

Pada tahap ini, dilakukan praktik pembuatan skrip R sederhana serta eksekusi perintah yang terdapat di dalamnya untuk memahami proses pembentukan objek dan pengelolaan workspace.

1. Buat sebuah berkas skrip baru dengan memilih menu File > New File > R Script pada RStudio.

2. Pada berkas skrip yang telah dibuat, tuliskan perintah berikut untuk mendefinisikan sebuah vektor numerik:

```
1 pengamatan <- c(10, 20, 30, 40)
```

3. Simpan berkas skrip tersebut dengan nama Latihan1_NamaMahasiswa.R dengan menekan tombol Ctrl + S.
4. Blok (seleksi) baris perintah yang telah dituliskan, kemudian jalankan perintah tersebut dengan menekan tombol Run atau menggunakan pintasan keyboard Ctrl + Enter.
5. Amati panel Environment. Objek bernama pengamatan akan muncul sebagai bagian dari workspace aktif, yang menandakan bahwa objek tersebut telah berhasil dibuat dan disimpan dalam memori R.
6. Selanjutnya, untuk menghitung nilai rata-rata dari data yang terdapat dalam objek pengamatan, ketikkan perintah berikut pada panel Console:

```
1 mean(pengamatan)
```

7. Identifikasi empat panel utama yang tampil pada antarmuka RStudio, yaitu Source, Console, Environment, dan Packages, serta jelaskan secara singkat fungsi masing-masing panel tersebut.

2.9 Objek di R

R telah mengimplementasikan konsep pemrograman berorientasi objek (object-oriented programming). Dalam R, seluruh proses pengolahan data berfokus pada kelas (class) dan objek (object). Semua entitas yang digunakan di R –seperti vektor, faktor, matriks, array, data frame, list, maupun fungsi– dipandang dan diperlakukan sebagai objek. Oleh karena itu, setiap hasil kerja di R dapat disimpan dalam bentuk objek dengan cara mendefinisikannya terlebih dahulu. Objek yang telah dibuat selanjutnya dapat digunakan kembali dengan memanggil nama objek tersebut.

Untuk mendeklarasikan sebuah objek, pengguna perlu menetapkan nama objek. R memberikan fleksibilitas dalam penamaan objek, namun terdapat beberapa aturan dasar yang perlu diperhatikan agar kode dapat dijalankan dengan benar dan mudah dibaca, yaitu sebagai berikut:

1. Nama objek dapat menggunakan huruf besar (A–Z) dan huruf kecil (a–z).
2. Nama objek dapat mengandung angka (0–9), serta karakter titik (.) dan garis bawah (_).
3. Nama objek tidak boleh mengandung spasi. Sebagai contoh, penulisan tingkat income tidak valid dan sebaiknya diganti menjadi tingkat_income.
4. Nama objek bersifat case sensitive, artinya R membedakan antara huruf besar dan huruf kecil. Sebagai contoh, tingkat_income dan tingkat_Income merupakan dua objek yang berbeda.
5. Penamaan objek sebaiknya diawali dengan huruf. Meskipun R mengizinkan penggunaan awalan titik (.) atau garis bawah (_), seperti .tingkatincome atau _tingkatincome, penulisan tersebut tidak disarankan.
6. Sebaiknya hindari penggunaan nama yang sama dengan perintah atau kata kunci yang telah didefinisikan dalam R (reserved words), seperti if, else, repeat, while, function, for, in, next, break, TRUE, FALSE, NULL, NA, NA_integer_, dan sebagainya.

Selanjutnya, dalam buku ini istilah objek akan disebut sebagai variabel. R akan menyimpan variabel tersebut sebagai objek di dalam memori, sehingga pengguna dapat melakukan berbagai operasi terhadapnya, seperti penjumlahan, pengurangan, perkalian, maupun operasi lainnya sesuai dengan kebutuhan

analisis data.

Variabel memungkinkan pengguna untuk memanggil kembali, memanipulasi, dan mengelola data secara fleksibel selama analisis. R memberikan fleksibilitas tinggi dalam pemberian nama variabel, namun terdapat beberapa aturan yang perlu diperhatikan agar penamaan variabel benar dan tidak menimbulkan kesalahan saat eksekusi.

Pemahaman mengenai variabel tidak dapat dipisahkan dari tipe data yang disimpannya. Setiap variabel menyimpan data dengan tipe tertentu, seperti numerik, karakter, logika, vektor, matriks, daftar (list), atau data frame. Pemilihan tipe data yang tepat sangat penting karena menentukan jenis operasi dan analisis yang dapat diterapkan pada variabel tersebut. Topik Tipe Data akan dibahas lebih lanjut pada Bab 3: Tipe dan Struktur Data di R.

Untuk memperjelas konsep variabel dan tipe data tersebut, berikut ditunjukkan contoh sederhana penggunaan variabel dalam R. Contoh ini memperlihatkan bagaimana sebuah nilai disimpan ke dalam variabel dan kemudian digunakan dalam operasi aritmatika.

```
1 myData <- 345
2 5*myData
3 [1] 1725
```

1. Baris ke-1, kita membuat sebuah variabel (objek) bernama myData dan memberinya nilai 345 (“berikan nilai 345 ke dalam variabel myData”). Tanda <- digunakan untuk melakukan assignment, yang artinya memberikan nilai tertentu ke sebuah variabel. Dalam hal ini, myData akan menyimpan nilai 345.
2. Baris ke-2, operasi perkalian antara angka 5 dan nilai yang tersimpan dalam variabel myData. Karena myData bernilai 345, maka operasi ini setara dengan $5 * 345$. Hasil dari operasi perkalian ini adalah 1725.
3. R menampilkan output dalam bentuk vektor, dalam hal ini merupakan vektor dengan satu elemen. Tanda [1] di depan angka 1725 menunjukkan bahwa ini adalah elemen pertama dari hasil yang ditampilkan.

Perintah untuk memunculkan dan menghapus variabel (object) di R, dengan mengetikkan perintah:

```
1 ls()
2 rm(list = ls())
```

1. Perintah ls() dalam R digunakan untuk menampilkan daftar semua objek yang saat ini berada di lingkungan kerja. Ini mencakup variabel, fungsi, data frame, dan struktur data lainnya yang telah dibuat atau impor selama sesi R Anda. Dengan menggunakan ls(), Anda dapat dengan mudah melihat apa saja yang sudah ada di environment R. Perintah ini membantu mengelola dan melacak objek yang digunakan.
2. Perintah rm(list = ls()) dalam R digunakan untuk menghapus semua objek yang ada di lingkungan kerja.

Objek R adalah wadah penyimpanan data atau hasil proses dalam R yang berada di dalam workspace dan dapat digunakan kembali dalam analisis selanjutnya. Objek tersebut diakses melalui variabel, yaitu nama atau simbol yang mereferensikan objek tertentu di dalam memori R. Seluruh objek yang aktif akan tersimpan dalam workspace dan dapat diamati melalui panel Environment pada RStudio. Sementara itu, riwayat perintah yang telah dijalankan untuk membentuk atau memanipulasi objek dapat ditinjau melalui panel History, yang memudahkan penelusuran kembali proses analisis yang telah dilakukan

2.9.1 Karakteristik dan Keunggulan Bahasa R

Seperti bahasa pemrograman lainnya, R memiliki sintaks dan konsep yang unik. Beberapa karakteristik utama dari bahasa R antara lain:

1. Case-sensitive
R peka terhadap perbedaan huruf besar dan kecil, sehingga penulisan sintaks harus diperhatikan dengan cermat.
2. Berbasis objek
Semua elemen dalam R diperlakukan sebagai objek, mirip dengan bahasa berbasis objek seperti Java dan Python, tetapi dengan konsep yang lebih sederhana.
3. Bahasa terinterpretasi
Kode R dapat dijalankan langsung tanpa perlu dikompilasi ke bahasa mesin terlebih dahulu.
4. Mendukung kontrol alur program
R memiliki fitur loop, decision making, serta berbagai operator, seperti aritmatika dan logika.
5. Fleksibel dalam pengolahan data
R mendukung impor dan ekspor berbagai format file (CSV, Excel, SQL, dll.), serta fungsi pengolahan data seperti eksplorasi, transformasi, dan pembersihan data.
6. Dapat diperluas dengan pustaka tambahan
R memiliki ekosistem paket yang kaya, memungkinkan pengguna menambah fitur melalui library dari CRAN atau GitHub.
7. Kaya akan fitur visualisasi data
Paket seperti ggplot2 dan ggvis memungkinkan pembuatan grafik yang informatif dan menarik untuk analisis data.

Selain karakteristiknya, R juga memiliki berbagai fitur yang menjadikannya salah satu pilihan utama dalam analisis data:

1. Manajemen Data
R mendukung berbagai fungsi pengolahan data, termasuk impor data dari berbagai format, pemeriksaan kualitas data, penggabungan dataset, serta penyusunan ulang data agar siap dianalisis.
2. Statistik Deskriptif R menyediakan fungsi statistik ringkasan, seperti mean, median, standard deviation, skewness, dan quantiles, yang membantu dalam memahami distribusi variabel.
3. Uji Statistik
Berbagai metode pengujian, seperti uji rata-rata, uji proporsi, uji hubungan antar variabel, dan uji distribusi data, tersedia secara langsung di R.
4. Visualisasi dan Pemrograman

Dengan pustaka seperti ggplot2, pengguna dapat membuat visualisasi data yang menarik, sementara paket seperti tidyverse mempermudah manipulasi data.

5. Ekosistem Paket yang Kaya

R memiliki ribuan paket yang mendukung berbagai bidang, mulai dari ekonometrika, keuangan, analisis deret waktu, bioinformatika, hingga machine learning. Komunitas global yang aktif terus mengembangkan ekosistem ini.

2.10 Paket R

Paket R atau R package merupakan koleksi dari fungsi R yang mengandung kode dan contoh sampel data. Tersimpan di direktori "library" di program R. Pada program bawaannya R telah terinstall beberapa package.

Salah satu alasan mengapa R menjadi populer adalah ketersediaan berbagai fungsi tambahan dalam bundle yang disebut dengan paket (packages). Paket merupakan kumpulan fungsi (kode/perintah) R yang dapat mencakup data dan dikompilasi dalam format standar. Direktori tempat paket disimpan dalam komputer disebut library. Paket digunakan untuk melakukan analisis tertentu serta menambah fungsionalitas atau kemampuan R.

Saat pertama kali menginstal R, hanya beberapa paket yang terpasang secara otomatis. Instalasi standar R akan memuat berbagai library dasar, seperti base, datasets, graphics, utils, dan stats. Sementara itu, paket tambahan yang dikembangkan oleh komunitas pengguna R harus diinstal secara terpisah sesuai dengan kebutuhan analisis.

Hingga saat ini, terdapat lebih dari 6.000 paket R yang tersedia di repository CRAN atau dapat diakses melalui tautan ini. Setiap paket R disertai dengan informasi yang mencakup jenis paket, tanggal rilis, nama pengembang, deskripsi singkat, fungsi serta kegunaannya, serta sumber kode dan sintaks yang digunakan. Jumlah paket R terus bertambah secara eksponensial seiring dengan kontribusi dari komunitas pengguna.

Secara umum, paket dalam R dibagi menjadi dua jenis utama:

1. Base packages

paket yang telah dimuat secara otomatis saat instalasi R. Untuk melihat daftar base packages, gunakan perintah:

```
1 getOption("defaultPackages")
```

2. Contributed packages

Paket tambahan yang perlu diunduh dan diinstal secara manual. Untuk menginstal paket tambahan, gunakan perintah berikut di console editor:

```
1 install.packages("nama_paket")
```


Setelah perintah dijalankan, R akan secara otomatis mengunduh dan menginstal paket tersebut dari situs CRAN. Pastikan perangkat terhubung ke internet saat melakukan instalasi. Setelah paket terinstal, kita dapat menggunakannya dalam R dengan menjalankan perintah berikut:

```
1 library(nama_paket)
```

Untuk mengetahui lokasi direktori tempat paket R tersimpan, gunakan perintah:

```
1 libPaths()
```

2.11 Memulai Bahasa R

Bahasa R disebut sebagai *Interpreted Language*, artinya setiap perintah akan langsung dieksekusi baris demi baris sesaat setelah pengguna menekan tombol **enter** (). Jadi tidak diperlukan proses kompilasi dari bahasa R ke kode mesin. Dalam kondisi normal, setiap baris kode R digunakan untuk menuliskan satu perintah. Setiap baris perintah yang dapat dieksekusi atau dijalankan disebut sebagai statement atau instruksi.

R juga memungkinkan penulisan beberapa instruksi pada satu baris yang sama. Dalam kasus ini, setiap instruksi dipisahkan oleh tanda titik koma (;). Pendekatan ini dapat digunakan untuk menyederhanakan skrip yang singkat atau menggabungkan perintah terkait dalam satu baris. Sebagai contoh, perhatikan kode berikut:

```
1 x <- 10; y <- 20; z <- x + y
```

Pada contoh di atas, terdapat tiga instruksi yang dituliskan dalam satu baris:

- Membuat objek x dengan nilai 10.
- Membuat objek y dengan nilai 20.
- Membuat objek z sebagai hasil penjumlahan x dan y.

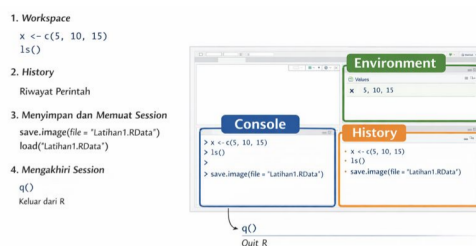
Meskipun memungkinkan, penggunaan banyak instruksi dalam satu baris sebaiknya dibatasi agar skrip tetap mudah dibaca dan dipelihara, terutama dalam praktikum atau proyek analisis data yang kompleks.

2.11.1 R Session

Menjalankan atau menggunakan R dapat dilakukan dengan dua cara. Kita bisa mengetikkan baris perintah di layar di dalam “R-session”, atau kita bisa menyimpan perintah sebagai file “script” dan menjalankan seluruh file di dalam R.

Sebuah **R session** terdiri dari langkah memulai R, menggunakan R untuk bekerja atau analisa data dan kemudian menutup atau keluar dari R. Saat memulai R dengan mengetikkan “R” di terminal atau klik ikon R di menu program, R akan menampilkan prompt yang dilambangkan dengan simbol “>”.

Prompt (R console) menunjukkan di mana kalian akan mengetik perintah untuk R. Bahasa R disebut sebagai *Interpreted Language*, artinya setiap perintah akan langsung dieksekusi baris demi baris sesaat setelah pengguna menekan tombol **enter**. Jadi tidak diperlukan proses proses kompilasi dari bahasa R ke kode mesin.



Gambar 2: Session di R

Untuk bekerja dengan R, kalian cukup memasukkan perintah ke dalam R console dan diikuti dengan menekan tombol enter (lihat Gambar 2). Dalam kondisi normal, satu baris hanya untuk satu perintah. Setiap baris perintah yang dapat dieksekusi atau dijalankan disebut sebagai **statement** atau instruksi. Namun R memungkinkan merangkai beberapa instruksi pada sebuah garis, di mana masing-masing instruksi dipisahkan dengan tanda semi colon (";").

2.11.2 Demo di R

R menyediakan demo program agar pengguna baru memiliki insight tentang berbagai fitur dan fungsi yang tersedia dalam bahasa pemrograman ini, serta cara menggunakannya untuk analisis data dan visualisasi. Berikut demo R yang dapat dicoba:

1. Jalankan R/ RStudio
2. Ketikkan di console perintah berikut. Perhatikan, catat dan berikan penjelasan secukupnya.

```
1 getOption("defaultPackages")      #list default paket
2 citation()                        #sitasi
3 demo()                            #list demo, keluar dari list ketikkan q
4 demo(graphics)                   #demo program
5 example(lm)                      #melihat contoh dengan nama fungsi 'lm'
6 example(mean)
```

Penjelasan:

1. Symbol "#" merupakan penanda komentar. Komentar adalah teks apapun yang diawali dengan tanda "#", digunakan untuk memberikan catatan kepada pembaca kode dan tidak akan dijalankan oleh R.
2. Fungsi example() memberi penjelasan untuk perintah atau fungsi tertentu. Misalnya ingin memahami fungsi built-in mean() dapat melihat contoh dengan mengetikkan example('mean').
3. Fungsi demo() hampir mirip dengan fungsi example(). Fungsi demo() memberi contoh R script.

2.11.3 R Sebagai Kalkulator

Fungsi paling sederhana dari R adalah sebagai kalkulator. Tidak hanya sesederhana itu, R juga dapat menyelesaikan problematika matematika seperti persamaan kuadrat, matriks, kalkulus, trigonometri dan sebagainya. R sangat membantu jika tidak ingin bersusah payah menggunakan cara-cara manual dalam menyelesaikan problematika matematika. Contoh-contoh berikut adalah operasi-operasi matematika yang disertai dengan komentar.

Anda ketikkan atau input kode berikut: ¹.

```
1 100.1 + 234.9 + 12.01
2 2+2
3 sqrt(256)
4 log10(100)*cos(pi)
5 cumsum(c(2,3,4,5,6))
6 170166719 %% 31079
7 48:(14*3)
```

¹Perlu diperhatikan bahwa fungsi atau kode pada lingkungan R adalah case sensitif, artinya penggunaan huruf besar atau kecil akan sangat berpengaruh

```
8 48:14*3
9 log(10)
10 log10(10)
11 exp(2)
12 abs(-4)
```

1. Persamaan liner dengan dua variabel:
$$\begin{cases} 5x - 4y = -10 & (a) \\ -x + y = 2 & (b) \end{cases}$$

Hitung nilai x dan y ! Untuk menjawab pertanyaan ini, digunakan persamaan:

$$Ax = B \Rightarrow x = A^{-1}B$$

```
1 A <- matrix(c(5,4,-1,1),2,2, byrow=TRUE)
2 b <- c(-10,2)
3 print(A)
4 print(b)
5 solve(A,b)
```

2. Persamaan liner dengan tiga variabel:
$$\begin{cases} x + 2y + 3z = 20 & (a) \\ 2x + 2y + 3z = 100 & (b) \\ 3x + 2y + 8z = 200 & (c) \end{cases}$$

Hitung nilai x , y , dan z ! Rumus yang digunakan adalah:

$$Ax = B \Rightarrow x = A^{-1}B$$

Persamaan tersebut dapat dihitung menggunakan R:

```
1 # create matrix A and B using given equations
2 A <- rbind(c(1, 2, 3),
3           c(2, 2, 3),
4           c(3, 2, 8))
5 b <- c(20, 100, 200)
6
7 solve(A, b)
```

2.12 Instruksi Terpandu

1. Membuat Skrip Baru
 - Buka RStudio.
 - Pilih menu File > New File > R Script untuk membuat berkas skrip baru.
 - Simpan skrip tersebut dengan nama Latihan2_NamaAnda.R menggunakan pintasan Ctrl + S atau melalui menu File > Save As.
 - Pastikan lokasi penyimpanan skrip sesuai dengan working directory yang aktif.
2. Menuliskan Perintah R

```
1 # 1. Menggunakan R sebagai kalkulator
2 15 + 25 * 2
3
4 # 2. Membuat variabel (Assignment)
```

```

5 nilai_hadir <- 90
6 nilai_uts <- 80
7
8 # 3. Operasi antar variabel
9 nilai_akhir <- (nilai_hadir + nilai_uts) / 2
10
11 # 4. Menampilkan hasil ke Console
12 nilai_akhir

```

```

1 # Membuat vektor sudut dari 0 hingga 2*pi
2 x <- seq(0, 2*pi, length.out = 100)
3
4 # Menghitung nilai sinus dan cosinus
5 y_sin <- sin(x)
6 y_cos <- cos(x)
7
8 # Membuat grafik
9 plot(x, y_sin, type = "l", col = "blue", lwd = 2,
10      ylim = c(-1, 1), xlab = "Sudut (radian)", ylab = "Nilai",
11      main = "Grafik Sinus dan Cosinus")
12
13 # Menambahkan kurva cosinus
14 lines(x, y_cos, col = "red", lwd = 2)
15
16 # Menambahkan legenda
17 legend("topright", legend = c("sin(x)", "cos(x)"),
18      col = c("blue", "red"), lwd = 2)

```

```

1 # Membuat vektor suhu dalam Celcius
2 celcius <- c(0, 10, 20, 30, 40)
3
4 # Menghitung konversi ke Fahrenheit
5 fahrenheit <- 9/5 * celcius + 32
6
7 # Membuat data frame
8 conversion <- data.frame(Celcius = celcius, Fahrenheit = fahrenheit)
9
10 # Menampilkan tabel
11 print(conversion)

```

3. Menjalankan Skrip

- Blok seluruh perintah atau sebagian perintah, lalu tekan tombol Run atau gunakan pintasan Ctrl + Enter.
- Perhatikan panel Environment untuk memastikan variabel tersimpan di workspace.
- Lihat panel Console untuk melihat hasil eksekusi perintah, baik yang berupa nilai variabel maupun hasil perhitungan langsung.

2.13 Latihan Mandiri

Soal 1

Jika Anda ingin membuat grafik dari sekumpulan data, di panel manakah hasil visualisasi tersebut akan muncul?

- A. Console
- B. Plots
- C. Environment
- D. Files

Soal 2

Tuliskan perintah **R** untuk membuat sebuah variabel bernama `usia` yang berisi angka umur Anda saat ini.

Soal 3

Apa yang akan terjadi jika Anda mengetikkan perintah `?sum` di bagian Console?

- A. Menghapus semua data
- B. Menampilkan panduan bantuan tentang cara menjumlahkan data
- C. Menutup aplikasi RStudio
- D. Menghitung total seluruh variabel di Environment

Soal 4

Apa kegunaan utama dari tanda pagar (`#`) di dalam skrip R?

- A. Untuk membuat judul besar
- B. Untuk memberikan komentar atau penjelasan yang tidak akan diproses oleh R
- C. Sebagai tanda perkalian
- D. Untuk menghapus baris kode

Soal 5

Manakah perintah R berikut yang **benar** untuk menampilkan isi dari sebuah variabel bernama `data` di Console?

- A. `print(data)`
- B. `show(data)`
- C. `view(data)`
- D. `open(data)`

Praktikum:

Bagian ini merupakan kelanjutan dari pembahasan instalasi dan pengenalan antarmuka R dan RStudio. Setelah mahasiswa berhasil memasang R dan RStudio serta memahami komponen dasar lingkungan kerja RStudio, langkah berikutnya adalah mulai berinteraksi langsung dengan R melalui perintah (command). Praktikum ini dirancang sebagai fondasi awal agar mahasiswa terbiasa menulis, menjalankan, dan memahami perintah R sebelum masuk ke konsep probabilitas dan statistika yang lebih formal.

Fokus utama pada tahap awal ini adalah:

- Menjalankan perintah dasar R.
- Memahami cara R melakukan perhitungan dan menyimpan nilai.
- Mengenal struktur data dan ringkasan data menggunakan fungsi bawaan R.

2.14 Menjalankan Perintah Dasar R

R adalah bahasa pemrograman berbasis ekspresi. Artinya, setiap perintah yang diketikkan di Console akan dievaluasi dan menghasilkan keluaran (output). Pada tahap awal, mahasiswa perlu memahami bahwa R dapat digunakan seperti kalkulator ilmiah sekaligus alat analisis data.

Untuk memudahkan pemahaman, konsep operasi aritmatika dasar dan assignment (pemberian nilai ke variabel) dirangkum dalam bentuk tabel berikut. Tabel ini dirancang agar mahasiswa dapat langsung mengaitkan konsep matematika, sintaks R, dan tujuan penggunaannya dalam analisis statistika.

Tabel 5: Operasi Aritmatika Dasar dan Assignment Variabel dalam R

No.	Konsep	Operator / Sintaks R	Penjelasan	Contoh Aktivitas Praktikum	Tujuan Pembelajaran
1	Penjumlahan	+	Digunakan untuk menjumlahkan dua atau lebih nilai numerik.	Menjumlahkan dua nilai hasil pengukuran.	Memahami operasi dasar dan notasi R.
2	Pengurangan	-	Digunakan untuk menghitung selisih dua nilai.	Menghitung selisih nilai maksimum dan minimum.	Melatih interpretasi hasil perhitungan.
3	Perkalian	*	Digunakan untuk operasi perkalian numerik.	Mengalikan konstanta dengan suatu nilai pengamatan.	Memahami operasi skalar dalam R.
4	Pembagian	/	Digunakan untuk membagi suatu nilai dengan nilai lainnya.	Menghitung rata-rata secara manual (jumlah dibagi banyak data).	Menghubungkan konsep statistika dan komputasi.

Bersambung ke halaman berikutnya

No.	Konsep	Operator / Sintaks R	Penjelasan	Contoh Aktivitas Praktikum	Tujuan Pembelajaran
5	Pemangkatan	<code>^</code>	Digunakan untuk operasi perpangkatan.	Menghitung kuadrat atau pangkat dua suatu nilai.	Mengenal ekspresi matematika dalam R.
6	Prioritas Operasi	Kombinasi operator	R mengikuti aturan prioritas operasi matematika (pemangkatan, perkalian/pembagian, penjumlahan/pengurangan).	Membandingkan hasil perhitungan dengan dan tanpa tanda kurung.	Memahami cara R mengevaluasi ekspresi.
7	Assignment Variabel	<code><-</code>	Menyimpan hasil perhitungan ke dalam suatu variabel.	Menyimpan hasil rata-rata ke dalam variabel.	Memahami konsep penyimpanan nilai.
8	Variabel Dinamis	Tanpa deklarasi tipe	Tipe data variabel ditentukan otomatis oleh R.	Mengubah isi variabel dengan nilai berbeda.	Mengenal fleksibilitas bahasa R.
9	Penamaan Variabel	Nama bermakna	Nama variabel sebaiknya merepresentasikan isi data.	Menggunakan nama variabel sesuai konteks data.	Melatih praktik pemrograman yang baik.
10	Lingkungan Kerja	Panel <i>Environment</i>	Variabel yang dibuat akan tampil pada panel Environment di RStudio.	Mengamati perubahan Environment setelah assignment.	Memahami alur kerja di RStudio.

Berikut contoh terintegrasi Operasi Aritmatika Dasar dan Assignment Variabel yang kontekstual dengan mahasiswa Teknologi Rekayasa Perangkat Lunak (TRPL).

```

1 # Penjumlahan, pembagian, assignment variabel.
2 # Menghitung Rata-rata Waktu Eksekusi Program
3 t1 <- 120
4 t2 <- 135
5 t3 <- 125
6
7 rata_waktu <- (t1 + t2 + t3) / 3
8 rata_waktu
9
10 # Menghitung Beban CPU Rata-rata
11 cpu1 <- 45
12 cpu2 <- 55

```

```

13 cpu3 <- 50
14
15 cpu_rata <- (cpu1 + cpu2 + cpu3) / 3
16 cpu_rata
17
18 # Menghitung Total dan Rata-rata Baris Kode (LOC)
19 modul_A <- 350
20 modul_B <- 420
21 modul_C <- 380
22
23 total_LOC <- modul_A + modul_B + modul_C
24 rata_LOC <- total_LOC / 3
25
26 total_LOC
27 rata_LOC
28
29 # Perhitungan Estimasi Biaya Komputasi Cloud
30 jam_server <- 120
31 tarif_per_jam <- 1500
32
33 biaya_total <- jam_server * tarif_per_jam
34 biaya_total
35
36 # Penilaian proyek berdasarkan bobot komponen.
37 kode <- 85
38 dokumentasi <- 80
39 presentasi <- 90
40
41 skor_akhir <- (0.5 * kode) + (0.3 * dokumentasi) + (0.2 * presentasi)
42 skor_akhir

```

2.15 Fungsi Dasar di R

R menyediakan banyak fungsi bawaan (built-in functions) untuk operasi matematika dan statistika dasar. Fungsi merupakan komponen inti dalam R, karena hampir seluruh analisis data dilakukan melalui pemanggilan fungsi.

Tabel 6: Fungsi Dasar Matematika dan Statistika pada R

No.	Jenis Fungsi	Fungsi R	Kegunaan Utama
1	Matematika	<code>sqrt()</code>	Menghitung akar kuadrat dari suatu nilai numerik.
2	Matematika	<code>log()</code>	Melakukan transformasi logaritma (logaritma natural secara default).
3	Matematika	<code>abs()</code>	Menghitung nilai mutlak dari suatu bilangan atau selisih nilai.
4	Statistika	<code>min()</code>	Menentukan nilai minimum dari sekumpulan data numerik.
<i>Bersambung ke halaman berikutnya</i>			

No.	Jenis Fungsi	Fungsi R	Kegunaan Utama
5	Statistika	<code>max()</code>	Menentukan nilai maksimum dari sekumpulan data numerik.
6	Statistika	<code>mean()</code>	Menghitung nilai rata-rata (mean) dari data numerik.
7	Statistika	<code>summary()</code>	Menampilkan ringkasan statistik deskriptif data, meliputi nilai minimum, kuartil pertama (Q1), median, mean, kuartil ketiga (Q3), dan nilai maksimum.

Beberapa karakteristik fungsi di R yang perlu dipahami: - Fungsi memiliki nama dan argumen. - Argumen dapat diisi secara berurutan atau dengan menyebutkan nama argumennya. - Fungsi dapat menghasilkan satu atau lebih nilai keluaran.

Pada tahap awal, mahasiswa dikenalkan pada fungsi-fungsi dasar seperti: - Fungsi matematika (misalnya akar, logaritma, nilai mutlak). - Fungsi statistika sederhana (misalnya nilai minimum, maksimum, dan rata-rata).

```

1  # menghitung akar kuadrat
2  waktu <- 144
3  akar_waktu <- sqrt(waktu)
4  akar_waktu
5
6  # Logaritma Ukuran File
7  ukuran_file <- 2048
8  log_file <- log(ukuran_file)
9  log_file
10
11 # Nilai Mutlak Selisih Waktu Respons
12 respon_v1 <- 320
13 respon_v2 <- 290
14
15 selisih_respon <- abs(respon_v1 - respon_v2)
16 selisih_respon
17
18 # Fungsi Statistika Sederhana
19 memori <- c(512, 480, 530, 500)
20
21 min_memori <- min(memori)
22 max_memori <- max(memori)
23
24 min_memori
25 max_memori
26
27 # Rata-rata Jumlah Baris Kode per Modul
28 baris_kode <- c(350, 420, 380, 400)
29
30 rata_kode <- mean(baris_kode)
31 rata_kode

```



```

1 round(3.1415, digits = 2)
2 100:130
3 t <- c(74, 31, 95, 61, 76, 34, 23, 54, 96)
4 mean(t)
5 median(t)
6 sort(t)
7 quantile(t)
8 summary(t)

```

```

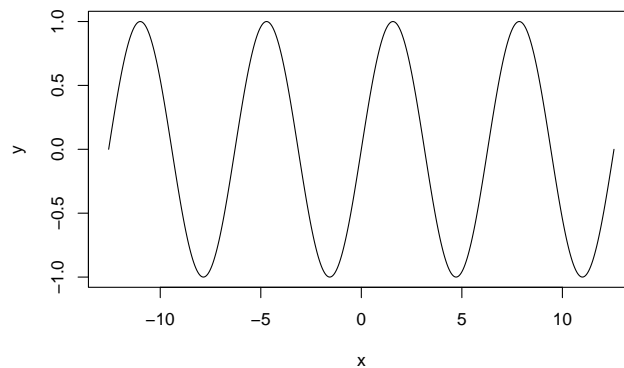
1 x <- seq(1, 20, 0.1)
2 y <- sin(x)
3 plot(x,y)

```

```

1 x<-seq(-4*pi,4*pi,length=500)
2 y<-sin(x)
3 plot(x,y,type="l")

```



```

1 x<-c(7,5,4)
2 x1<-seq(10)
3 x2<-seq(0,1,by=0.1)
4 x3<-rep(1,3)
5 x4<-c(rep(1,3),rep(2,2),rep(-1,4))
6 x5<-rep("Small",3)
7 rep(x,c(2,3,5))
8 rep(9, times=5)
9 rep(5:7, times=3)
10 rep(5:7, each=3)

```

2.16 Kesimpulan

Bab ini telah membahas fondasi konseptual dan teknis dalam pengantar probabilitas dan statistika, dengan penekanan pada peran statistika sebagai alat utama dalam analisis data kuantitatif di bidang sains, teknologi, dan komputasi. Statistika dipahami tidak sekadar sebagai kumpulan teknik perhitungan, melainkan sebagai kerangka berpikir sistematis yang memungkinkan data mentah diolah menjadi informasi yang bermakna dan relevan untuk pengambilan keputusan berbasis bukti.

Melalui pembahasan mengenai statistika deskriptif dan statistika inferensial, mahasiswa diperkenalkan pada dua pilar utama analisis statistik. Statistika deskriptif berfungsi untuk merangkum dan menyajikan data, sedangkan statistika inferensial memungkinkan penarikan kesimpulan dan prediksi terhadap populasi berdasarkan data sampel. Pemahaman terhadap perbedaan dan keterkaitan kedua pendekatan ini menjadi prasyarat penting sebelum mempelajari metode analisis statistik yang lebih lanjut.

Bab ini juga menekankan pentingnya berpikir statistik (*statistical thinking*) dalam pemecahan masalah berbasis data. Mahasiswa diarahkan untuk memahami variabilitas data, keterbatasan sampel, serta pentingnya pengambilan keputusan yang didasarkan pada analisis objektif, bukan semata-mata intuisi. Pendekatan ini diharapkan mampu membentuk pola pikir kritis dan analitis dalam menghadapi permasalahan nyata yang melibatkan data.

Selain aspek konseptual, bab ini memperkenalkan lingkungan komputasi statistik sebagai sarana pendukung utama analisis data modern. Pemahaman terhadap skala pengukuran variabel—nominal, ordinal, interval, dan rasio—ditekankan sebagai dasar metodologis yang menentukan pemilihan teknik analisis statistik yang tepat. Kesalahan dalam memahami skala pengukuran dapat berdampak langsung pada validitas hasil analisis dan interpretasi data.

Pengenalan R dan RStudio sebagai lingkungan komputasi statistik memberikan bekal teknis awal bagi mahasiswa untuk mengimplementasikan konsep statistika secara praktis. Melalui latihan eksplorasi antarmuka, penulisan skrip sederhana, serta evaluasi pemahaman, mahasiswa diharapkan mampu mengoperasikan RStudio secara mandiri dan memahami alur kerja analisis data yang sistematis dan dapat direproduksi.

Secara keseluruhan, capaian pembelajaran pada bab ini mendukung prinsip *Outcome-Based Education* dengan membangun kesiapan konseptual dan keterampilan dasar komputasi statistik. Kompetensi yang diperoleh pada bab ini menjadi landasan penting untuk mempelajari topik probabilitas, distribusi peluang, inferensi statistik, dan pemodelan data pada bab-bab selanjutnya.

Pustaka

Intro to R Part 24: Hypothesis Testing, a. URL <https://kaggle.com/code/hamelg/intro-to-r-part-24-hypothesis-testing>.

Stats and R, b. URL <https://statsandr.com/>.

Jane M Horgan, editor. *Probability with R*. Wiley, 1 edition, January 2020. ISBN 978-1-119-53694-9 978-1-119-53696-3. doi: 10.1002/9781119536963. URL <https://onlinelibrary.wiley.com/doi/book/10.1002/9781119536963>.

Elinor Jones, Simon Harden, and Michael J Crawley. *The R book*. John Wiley & Sons, 2022.

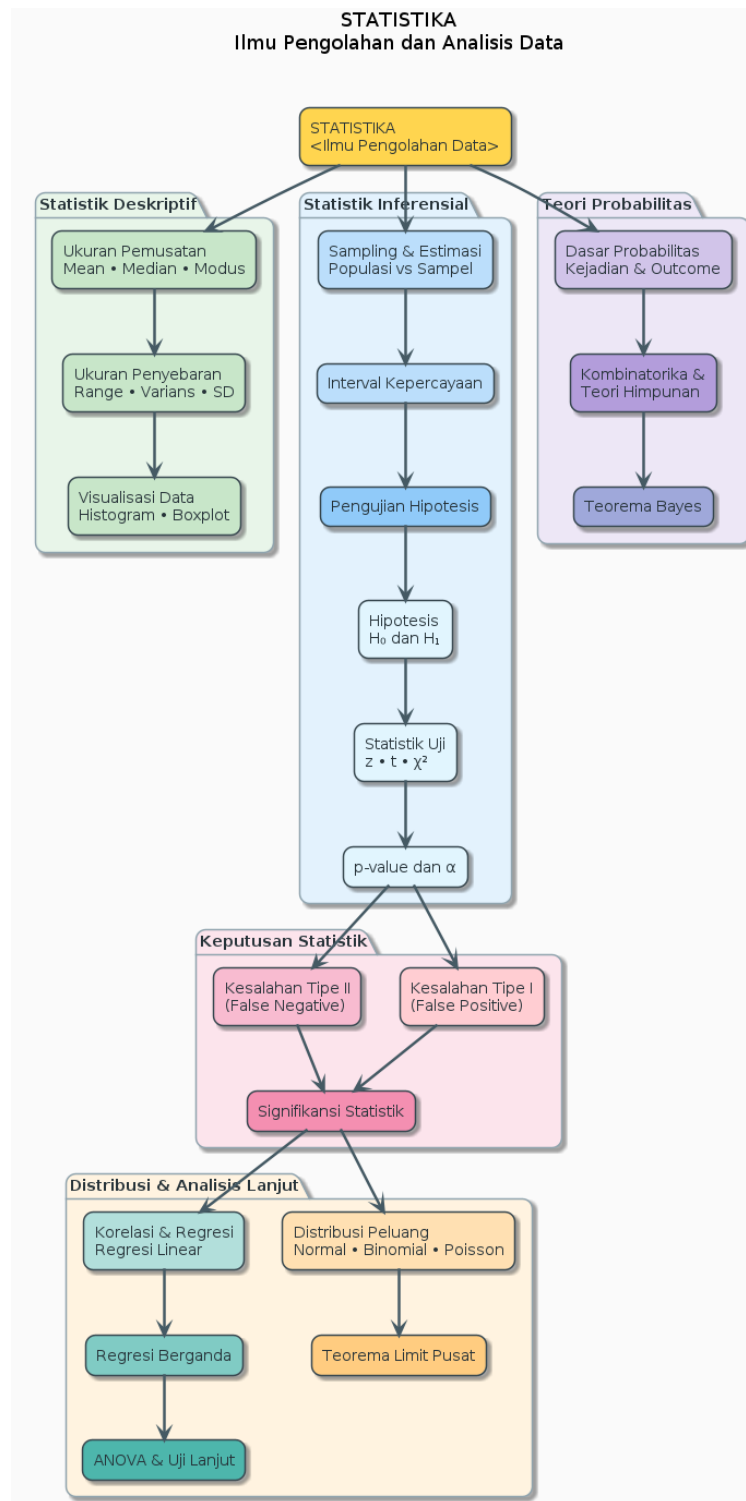
Danielle Navarro. Introduction to probability. URL <https://kpu.pressbooks.pub/learningstatistics/chapter/introduction-to-probability/>. Book Title: Learning Statistics with R.

Måns Thulin. *Modern Statistics with R*. URL https://www-modernstatisticswithr-com.translate.googleusercontent.com/translate?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc.

Mark PJ Van der Loo. *Learning RStudio for R statistical computing*. Packt Publishing Ltd, 2012.

Hadley Wickham, Garrett Grolemund, et al. *R for data science*, volume 2. O'Reilly Sebastopol, 2017.

Leland Wilkinson. *ggplot2: elegant graphics for data analysis by wickham, h.*, 2011.



Gambar 3: Diagram Hirarki Konsep Statistika