Pengantar Statistik untuk Sains Data

Menggunakan R

true true true

2025-07-03

Contents

Kata Pengantar

Selamat datang di "Pengantar Statisika (Statistika Dasar) untuk pembalajaran Sains Data"! Buku ini dirancang sebagai panduan komprehensif bagi siapa saja yang ingin memahami konsep dasar statistik yang esensial dalam dunia sains data. Statistik membentuk dasar dari banyak analisis data yang kita lakukan setiap hari, dan pemahaman yang kuat tentang dasar-dasar ini sangat penting bagi siapa saja yang ingin sukses di bidang ini.

Di era big data saat ini, kemampuan untuk menganalisis dan menginterpretasikan data adalah keterampilan yang sangat bernilai. Sains data adalah bidang yang berkembang pesat, dan pemahaman dasar tentang statistik menjadi semakin penting. Buku ini bertujuan untuk memberikan pemahaman yang jelas dan menyeluruh tentang konsep dasar statistik yang diperlukan untuk menganalisis data secara efektif dan membuat keputusan yang berdasarkan data.

Sasaran Pembaca

Buku ini dirancang untuk memenuhi kebutuhan beberapa kelompok pembaca, yaitu:

- **Mahasiswa:** Buku ini sangat sesuai untuk mahasiswa yang mengambil mata kuliah statistik dasar atau sains data. Menyediakan landasan yang kokoh dalam konsep-konsep statistik yang fundamental.
- **Profesional:** Buku ini juga berguna bagi para praktisi di bidang sains data, menawarkan referensi yang cepat dan menyeluruh mengenai statistik dasar. Cocok untuk mereka yang ingin mengasah keterampilan analisis data.
- **Pemula:** Bagi individu yang baru memulai dalam statistik atau sains data, buku ini memberikan pengantar yang jelas serta menyeluruh mengenai statistik dasar.

Dengan pendekatan ini, diharapkan buku ini dapat memenuhi berbagai kebutuhan pembaca dan memberikan manfaat yang substansial.

Tentang Penulis

Bakti Siregar M.Sc., CDS

Bakti bekerja sebagai Dosen di Program Sains Data ITSB. Beliau meraih gelar Magister dari Departemen Matematika Terapan di National Sun Yat Sen University, Taiwan. Selain mengajar, Bakti juga bekerja sebagai Data Scientist Freelance untuk perusahaan-perusahaan terkemuka seperti JNE, Samora Group, Pertamina, dan PT. Green City Traffic. Beliau memiliki antusiasme khusus dalam mengerjakan proyek (pengajaran) di bidang Big Data Analytics, Machine Learning, Optimisasi, dan Analisis Deret Waktu dalam bidang keuangan dan investasi. Keahlian utama beliau terletak pada bahasa pemrograman statistik seperti R Studio dan Python. Beliau juga berpengalaman dalam menerapkan sistem basis data seperti MySQL/NoSQL untuk manajemen data dan mahir dalam menggunakan alat Big Data seperti Spark dan Hadoop. Beberapa proyek beliau dapat dilihat di tautan berikut: Rpubs, Github, Website, dan Kaggle.

Andi Pujo Rahadi, S.T., M.Sc.

Beliau adalah Dosen Sains Data di ITSB dengan keahlian dalam Data Science, Python, Machine Learning, dan Computer Vision. Beliau juga menjabat sebagai IT Manager di ITSB. Beliau meraih gelar sarjana dari Institut Teknologi Pembangunan Surabaya pada tahun 2004 dan gelar Magister dari Universitas Gadjah Mada pada tahun 2016. Sejak tahun 2016, beliau telah menjadi dosen di berbagai universitas swasta dan negeri terkemuka. Selain tugas mengajarnya, beliau juga merupakan Senior Data Scientist di sebuah startup dan firma konsultan IT nasional. Beliau telah mengerjakan berbagai proyek sains data, termasuk pemodelan panen, deteksi objek, pengelompokan pelanggan, dan prediksi churn bank. Beliau mahir dalam berbagai alat machine learning dan Python, seperti TensorFlow, PyTorch, OpenCV, Pandas, dan AWS SageMaker.

Monica M. Manurung, M.Kom

Beliau juga merupakan Dosen Sains Data di ITSB dengan keahlian dalam Analisis/Desain Sistem dan Analisis Data. Beliau meraih gelar sarjana dari Universitas Gunadarma pada tahun 1996 dan gelar Magister dari Universitas Amikom pada tahun 2016. Saat ini, beliau sedang melanjutkan studi di Chung Yuan Christian University. Beliau berdedikasi untuk memajukan pendidikan dan pengetahuan, serta pengembangan diri untuk menjadi pendidik yang lebih baik dan berkualitas.

Ucapan Terima Kasih

Proses penulisan eBook ini tidak akan mungkin terjadi tanpa dukungan dari berbagai pihak. Saya ingin mengucapkan terima kasih kepada:

- **Keluarga Saya** atas dukungan moral dan dorongan yang tak tergoyahkan.
- **Rekan Kerja dan Kolaborator** yang telah memberikan umpan balik, saran, dan kritik yang konstruktif.
- **Institusi dan Organisasi**, terutama **ITSB** yang telah menyediakan sumber daya dan fasilitas yang diperlukan selama proses penelitian dan penulisan.

Saya berharap eBook ini dapat menjadi referensi yang berguna, memberikan inspirasi dan pengetahuan baru bagi pembaca. Saya juga berharap eBook ini memenuhi harapan dan kebutuhan pembaca serta bahwa pengetahuan yang dibagikan bermanfaat untuk semua.

Umpan Balik & Saran

Umpan balik dan saran Anda sangat berharga bagi kami untuk meningkatkan eBook ini di masa depan. Pembaca/pengguna yang ingin memberikan umpan balik dan saran dipersilakan untuk melakukannya melalui informasi kontak di bawah ini!

Email:

- dsciencelabs@outlook.com
- siregarbakti@gmail.com
- siregarbakti@itsb.ac.id

Chapter 1

Pengenalan R & Rstudio

R dan RStudio adalah aplikasi open source yang digunakan secara masif dalam dunia big data dan sains data. Kombinasi keduanya memungkinkan pengguna untuk melakukan analisis data dan visualisasi data yang kompleks dengan efisien dan mudah digunakan.

Kedua aplikasi ini adalah contoh aplikasi open source, yang berarti mereka dapat digunakan, dimodifikasi, dan didistribusikan secara bebas. Informasi lebih lanjut tentang apa itu aplikasi open source dapat ditemukan pada artikel berikut: Apa Sih Aplikasi Open Source Itu?.

1.1 Sejarah Singkat R

Bahasa pemrograman R mulai dikembangkan pada awal 1990-an oleh Ross Ihaka dan Robert Gentleman di University of Auckland, Selandia Baru. Tujuannya adalah menciptakan alat analisis data yang lebih baik daripada bahasa statistik lain seperti S. R dirilis pada tahun 1995 dan menarik perhatian komunitas statistik.

Sebagai bahasa open source, R berkembang pesat dengan kontribusi global. CRAN (Comprehensive R Archive Network), didirikan pada 1997, menyediakan

ribuan paket komunitas yang memperluas fungsi R. Popularitas R meningkat sejak awal 2000-an, meluas ke industri dan akademisi.

1.2 Tentang Rstudio

Diluncurkan pada 21 Februari 2011. RStudio didirikan oleh J.J. Allaire, yang juga dikenal karena perannya dalam pengembangan teknologi web awal seperti ColdFusion. RStudio telah berkembang menjadi salah satu IDE yang paling populer khususnya untuk bahasa pemrograman R, menawarkan banyak fitur yang memudahkan pengguna melakukan analisis data, pengembangan kode, dan pembuatan dokumentasi dinamis menggunakan R Markdown.

Berikut adalah penggunaan singkat berbagai bahasa pemrograman yang dapat digunakan di RStudio:

- **R**: Bahasa utama untuk analisis data.
- **Python**: Melalui `reticulate` untuk analisis data.
- **SQL**: Dengan paket `DBI` untuk kueri database.
- **Stan**: Melalui `rstan` untuk pemodelan Bayesian.
- **Julia**: Dengan `JuliaCall` untuk komputasi cepat.
- **Shell (Bash)**: Untuk perintah sistem di terminal.
- **HTML/CSS/JavaScript**: Dalam R Markdown untuk dokumen web.

1.3 Instalasi R dan RStudio

1.3.1 Langkah 1: Unduh dan Instal R

Untuk Unduh R:

- Kunjungi situs CRAN R.
- Pilih “Download R for Windows” (atau sesuai sistem operasi Anda).
- Klik “base” untuk mendapatkan versi terbaru.
- Unduh file installer sesuai arsitektur komputer (32-bit atau 64-bit).

Untuk Instal R:

- Jalankan file installer yang diunduh.
- Ikuti petunjuk di layar untuk instalasi.

- Pilih direktori instalasi jika diperlukan.
- Klik "Finish" setelah instalasi selesai.

Catatan: Pastikan **R** sudah terinstal dengan benar sebelum melanjutkan ke instalasi **RStudio**.

1.3.2 Langkah 2: Unduh dan Instal RStudio

Untuk Unduh RStudio:

- Kunjungi situs RStudio.
- Pilih versi "RStudio Desktop".
- Unduh versi gratis ("RStudio Desktop Open Source License") atau versi berbayar, sesuai kebutuhan Anda.

Untuk Instal RStudio:

- Jalankan file installer yang diunduh.
- Ikuti petunjuk di layar untuk instalasi.
- Pilih direktori instalasi jika diperlukan.
- Klik "Finish" setelah instalasi selesai.

1.3.3 Langkah 3: Verifikasi Instalasi

Untuk R:

- Buka R dari menu Start atau desktop.
- Ketik `version` di console dan tekan Enter untuk memeriksa versi R yang terinstal.
- Pastikan versi R yang tertera adalah yang terbaru.

Untuk RStudio:

- Buka RStudio dari menu Start atau desktop.
- Periksa apakah RStudio dapat terhubung dengan instalasi R.
- Jalankan beberapa perintah dasar di console RStudio, seperti `2 + 2`, untuk memastikan bahwa RStudio berfungsi dengan benar.

1.4 Video Instalasi R & RStudio

1.5 Popularitas Bahasa R

Bahasa R dikenal luas di kalangan data scientist dan peneliti data. Berikut adalah beberapa alasan utama mengapa R sangat banyak digunakan:

1.5.1 Analisis Statistik dan Big Data

R sangat efisien dalam analisis statistik dan big data, berkat banyaknya paket dan library yang mendukung berbagai jenis analisis data.

1.5.2 Fleksibilitas dan Kompatibilitas

R fleksibel dan kompatibel dengan berbagai platform, memudahkan integrasi dengan software lain.

1.5.3 Komunitas Aktif

R memiliki komunitas pengguna yang besar dan aktif, yang menyediakan banyak sumber daya untuk belajar dan berbagi pengetahuan.

- **R Project**: Situs resmi untuk R. Lihat di sini
- **R Project Mailing Lists**: Berbagai daftar email untuk tetap terinformasi tentang kegiatan terkait R. Daftar R-announce adalah titik awal yang baik, yang akan memberikan pembaruan tentang rilis terbaru perangkat lunak R. Lihat di sini
- **Twitter #rstats**: Banyak pengguna R yang aktif di Twitter, dan mereka dapat ditemukan di sana. Lihat di sini
- **Tidy Tuesday**: Proyek online mingguan yang berfokus pada pemahaman cara merangkum, mengatur, dan membuat grafik yang bermakna dengan data open source. Proyek-proyek yang telah dilakukan oleh orang lain dapat dilihat dengan mengikuti #tidytuesday di Twitter. Lihat di sini
- **R-Ladies**: Grup global yang didedikasikan untuk mempromosikan kesetaraan gender di komunitas R. Mereka memiliki daftar sumber daya yang lengkap untuk belajar dan menyelenggarakan acara edukasi serta networking. Lihat di sini
- **R-Podcast**: Podcast berkala dengan saran praktis untuk menggunakan R, serta berita terbaru tentang R. Lihat di sini
- **R-Bloggers**: Situs blog di mana para penulis dapat memposting contoh kode, analisis data, dan visualisasi. Lihat di sini

1.5.4 Open Source

Sebagai software open source, R dapat digunakan dan dikembangkan secara bebas, sangat menarik bagi peneliti dengan anggaran terbatas.

1.5.5 Visualisasi Data

R unggul dalam visualisasi data, memungkinkan penyajian data kompleks dalam bentuk yang mudah dipahami dan menarik.

1.5.6 Cocok untuk Big Data dan Machine Learning

Dengan perkembangan big data dan machine learning, R terus berkembang dan beradaptasi, menyediakan tool reliabel untuk menangani tantangan data di era modern.

1.6 Menggunakan R

Untuk memulai menggunakan R secara efektif, ikuti langkah-langkah berikut:

1.6.1 Membuka R atau RStudio

- **R:** Jika Anda hanya menginstal R, Anda dapat membuka aplikasi R dari menu Start atau desktop. Ini akan membuka konsol R, tempat Anda dapat mengetik perintah secara langsung.
- **RStudio:** Jika Anda menggunakan RStudio, buka aplikasi RStudio dari menu Start atau desktop. RStudio menyediakan antarmuka grafis yang lebih ramah dibandingkan dengan konsol R.

1.6.2 Menulis dan Menjalankan Kode di RStudio

- **Tab Console:**
  - Di dalam RStudio, Anda dapat menulis perintah langsung di tab "Console". Ini adalah area di mana Anda dapat mengetik perintah R dan melihat hasilnya secara langsung.
  - Contoh: Ketik `print("Hello, World!")` di konsol dan tekan Enter untuk menampilkan pesan "Hello, World!".
- **Tab Script:**
  - Untuk menulis dan menyimpan serangkaian perintah, buka tab "File" > "New File" > "R Script" dari menu RStudio.
  - Di jendela skrip yang terbuka, ketik perintah R yang ingin Anda simpan dan jalankan. Misalnya:

```r
# Skrip R sederhana
x <- 10
y <- 5
hasil <- x + y
print(hasil)
```

  - Simpan skrip dengan mengklik "File" > "Save" dan beri nama file dengan ekstensi `.R`, misalnya `analisis_sederhana.R`.

1.6.3 Menginstal dan Memuat Paket

- **Menginstal Paket:**
  - Untuk menambahkan fungsionalitas ke R, Anda perlu menginstal paket tambahan. Gunakan perintah `install.packages("nama_paket")` di konsol.
  - Contoh: Untuk menginstal paket ggplot2, ketik:

    ```r
    install.packages("ggplot2")
    ```

  - Tunggu hingga proses instalasi selesai. Paket tersebut akan diunduh dan dipasang ke sistem Anda.
- **Memuat Paket:**
  - Setelah paket diinstal, Anda harus memuatnya ke sesi R saat ini dengan perintah `library(nama_paket)`.
  - Contoh: Untuk menggunakan paket ggplot2, ketik:

    ```r
    library(ggplot2)
    ```

  - Paket yang dimuat akan menyediakan fungsi dan data tambahan yang dapat Anda gunakan dalam analisis Anda.

1.6.4 Mengakses Dokumentasi

- **Mendapatkan Bantuan untuk Fungsi:**
  - R memiliki sistem dokumentasi bawaan. Untuk mendapatkan informasi tentang fungsi tertentu, gunakan perintah `help(nama_fungsi)` atau `?nama_fungsi`.
  - Contoh: Untuk mendapatkan dokumentasi tentang fungsi `plot`, ketik:

    ```r
    help(plot)
    ```

    atau

```
?plot
```

- Dokumentasi ini akan memberikan penjelasan mengenai cara penggunaan fungsi, argumen yang diterima, dan contoh penggunaannya.
- **Mengakses Vignettes:**
  - Banyak paket juga menyediakan vignettes, yaitu dokumentasi tambahan yang lebih mendalam tentang bagaimana menggunakan paket tersebut. Untuk melihat vignettes, gunakan perintah `vignette("nama_vignette")`.
  - Contoh: Jika Anda menggunakan paket ggplot2 dan ingin melihat vignettes terkait, ketik:

    ```
    vignette("ggplot2")
    ```

Dengan mengikuti panduan rinci ini, Anda dapat memanfaatkan R dan RStudio secara lebih efektif untuk analisis data dan pemrograman statistik.

Chapter 2

Konsep Dasar Statistik

2.1 Definisi dan Aspek

Statistik adalah cabang ilmu yang berkaitan dengan pengumpulan, analisis, interpretasi, dan presentasi data. Statistik menggunakan metode matematis dan algoritmik untuk mengelola data sehingga informasi yang diperoleh dapat digunakan untuk membuat keputusan yang lebih baik.

2.1.1 Aspek Utama Statistik

- **Pengumpulan Data**: Proses mengumpulkan data dari berbagai sumber melalui survei, eksperimen, atau observasi. Metode ini dapat mencakup sampling (pengambilan sampel) atau pengumpulan data lengkap dari populasi.

- **Analisis Data**: Teknik-teknik statistik digunakan untuk menganalisis data, termasuk deskriptif statistik seperti mean (rata-rata), median, mode, dan deviasi standar, serta inferensial statistik seperti uji hipotesis dan analisis regresi.

- **Interpretasi Data**: Menafsirkan hasil analisis untuk memahami apa yang data tersebut katakan tentang fenomena yang sedang dipelajari. Ini melibatkan penarikan kesimpulan yang dapat menjelaskan pola, tren, atau hubungan dalam data.

- **Presentasi Data**: Menyajikan hasil analisis dalam format yang mudah dipahami, seperti tabel, grafik, atau visualisasi lainnya, sehingga informasi dapat disampaikan secara efektif kepada audiens.

2.1.2 Pentingnya Statistik

Statistik sangat penting dalam berbagai aspek kehidupan dan pekerjaan karena beberapa alasan berikut:

- **Pengambilan Keputusan**: Statistik memungkinkan individu dan organisasi untuk membuat keputusan yang lebih baik dan lebih terinformasi. Dengan data yang tepat dan analisis yang benar, keputusan dapat didasarkan pada bukti empiris daripada asumsi atau spekulasi.
- **Identifikasi Pola dan Tren**: Melalui analisis statistik, kita dapat mengidentifikasi pola dan tren dalam data yang mungkin tidak terlihat dengan kasat mata. Ini bisa termasuk tren pasar, pola perilaku konsumen, atau hubungan antara variabel.
- **Uji Hipotesis**: Statistik memungkinkan kita untuk menguji hipotesis atau teori dengan menggunakan data. Uji hipotesis seperti t-test atau ANOVA dapat menentukan apakah perbedaan antara kelompok atau variabel signifikan secara statistik.
- **Prediksi dan Perencanaan**: Dengan menggunakan model statistik, kita dapat membuat prediksi tentang kejadian di masa depan berdasarkan data historis. Ini sangat berguna dalam perencanaan bisnis, peramalan ekonomi, dan analisis risiko.
- **Evaluasi Program dan Kebijakan**: Statistik digunakan untuk mengevaluasi efektivitas program atau kebijakan. Dengan menganalisis data sebelum dan sesudah implementasi, kita dapat menilai apakah perubahan yang dilakukan memiliki dampak yang diinginkan.
- **Penelitian dan Pengembangan**: Dalam penelitian ilmiah dan pengembangan teknologi, statistik memainkan peran kunci dalam merancang eksperimen, menganalisis hasil, dan menarik kesimpulan yang valid.
- **Kesehatan dan Kedokteran**: Statistik digunakan untuk analisis epidemiologi, uji klinis, dan penelitian medis. Ini membantu dalam memahami prevalensi penyakit, efektivitas pengobatan, dan faktor risiko.

2.1.3 Contoh Aplikasi Statistik

- **Ekonomi**: Menganalisis data pasar untuk memprediksi tren ekonomi dan perencanaan investasi.
- **Pendidikan**: Menilai efektivitas metode pengajaran atau program pendidikan.
- **Pemasaran**: Menggunakan analisis data untuk memahami perilaku konsumen dan mengembangkan strategi pemasaran yang efektif.
- **Kesehatan**: Menganalisis data kesehatan untuk memahami pola penyakit dan efektivitas intervensi medis.

Statistik adalah alat yang sangat penting untuk menjawab pertanyaan kompleks, memahami dunia di sekitar kita, dan membuat keputusan yang lebih baik berdasarkan data yang ada.

2.2 Jenis Statistika

Statistika terbagi menjadi dua jenis utama, yaitu statistika deskriptif dan statistika inferensial. Keduanya memiliki tujuan dan teknik yang berbeda dalam menganalisis data.

2.2.1 Statistika Deskriptif

Statistika deskriptif melibatkan metode yang digunakan untuk menggambarkan atau meringkas data dari sampel atau populasi tanpa membuat inferensi atau generalisasi tentang data tersebut. Tujuan utama statistika deskriptif adalah untuk menyajikan data dalam bentuk yang mudah dipahami dan memberikan ringkasan yang berguna tentang karakteristik data.

Teknik-teknik utama dalam statistika deskriptif meliputi:

1. **Ukuran Pemusatan Data**:
   - **Mean (Rata-rata)**: Jumlah nilai data dibagi dengan jumlah data. Ini memberikan nilai rata-rata dari data.
   - **Median**: Nilai tengah dari data ketika data diurutkan. Median berguna untuk data yang tidak terdistribusi normal dan dapat memberikan gambaran yang lebih baik tentang pusat data.
   - **Mode**: Nilai yang paling sering muncul dalam data. Mode berguna untuk data kualitatif dan data kuantitatif.
2. **Ukuran Dispersion**:
   - **Range (Rentang)**: Selisih antara nilai maksimum dan nilai minimum. Rentang memberikan gambaran kasar tentang sebaran data.
   - **Variance**: Rata-rata dari kuadrat deviasi setiap nilai dari mean. Variance mengukur seberapa jauh nilai-nilai data menyebar dari rata-rata.
   - **Standard Deviation (Deviasi Standar)**: Akar kuadrat dari variance. Deviasi standar memberikan ukuran penyebaran data yang lebih mudah dipahami daripada variance.
3. **Visualisasi Data**:
   - **Histogram**: Grafik yang menunjukkan distribusi frekuensi data dalam interval tertentu.
   - **Box Plot**: Grafik yang menggambarkan distribusi data melalui kuartil dan deteksi outlier.
   - **Bar Chart dan Pie Chart**: Grafik yang digunakan untuk menampilkan data kategorikal.

2.2.2 Statistika Inferensial

Statistika inferensial melibatkan metode yang digunakan untuk membuat generalisasi atau inferensi tentang populasi berdasarkan data sampel. Teknik-teknik dalam statistika inferensial digunakan untuk menguji hipotesis, membuat prediksi, dan menentukan hubungan antara variabel.

Teknik-teknik utama dalam statistika inferensial meliputi:

- **Uji Hipotesis**:
  - **Uji t**: Digunakan untuk membandingkan rata-rata dari dua kelompok atau sampel.
  - **Uji ANOVA (Analysis of Variance)**: Digunakan untuk membandingkan rata-rata antara lebih dari dua kelompok.
  - **Uji Chi-Square**: Digunakan untuk menguji hubungan antara dua variabel kategorikal.
- **Interval Kepercayaan**:
  - Interval kepercayaan memberikan rentang nilai yang mungkin mengandung parameter populasi. Misalnya, interval kepercayaan 95% menunjukkan bahwa ada 95% kemungkinan bahwa parameter populasi berada dalam rentang tersebut.
- **Regresi dan Korelasi**:
  - **Regresi**: Menggunakan data untuk memodelkan hubungan antara variabel dependen dan satu atau lebih variabel independen. Contohnya adalah regresi linier untuk memprediksi nilai berdasarkan variabel lain.
  - **Korelasi**: Mengukur kekuatan dan arah hubungan linear antara dua variabel. Koefisien korelasi dapat menunjukkan apakah ada hubungan positif atau negatif antara variabel.
- **Sampling dan Estimasi**:
  - **Sampling**: Proses pemilihan sampel dari populasi untuk analisis. Teknik sampling yang baik memastikan bahwa sampel representatif terhadap populasi.
  - **Estimasi**: Menggunakan data sampel untuk memperkirakan parameter populasi, seperti mean atau proporsi.

Statistika deskriptif dan inferensial saling melengkapi dalam analisis data. Statistika deskriptif memberikan gambaran umum tentang data, sedangkan statistika inferensial memungkinkan penarikan kesimpulan dan pembuatan keputusan berdasarkan data tersebut.

2.3 Jenis Data: Kualitatif vs. Kuantitatif

Dalam statistika, data dibagi menjadi dua kategori utama: kualitatif dan kuantitatif. Memahami perbedaan antara kedua jenis data ini penting untuk memilih metode analisis yang tepat.

2.3.1 Data Kualitatif

Data kualitatif, juga dikenal sebagai data kategorikal, adalah data yang menggambarkan kualitas atau karakteristik non-numerik. Data ini sering kali digunakan untuk mengklasifikasikan atau mengelompokkan objek, individu, atau peristiwa berdasarkan atribut atau kategori tertentu.

Jenis-jenis data kualitatif meliputi:

1. **Data Nominal**:
   - **Definisi**: Data nominal adalah data yang terdiri dari kategori yang tidak memiliki urutan atau hierarki. Kategori ini hanya berbeda satu sama lain dan tidak dapat diurutkan.
   - **Contoh**: Jenis kelamin (pria/wanita), warna mata (biru, coklat, hijau), dan jenis kendaraan (mobil, motor, sepeda).
2. **Data Ordinal**:
   - **Definisi**: Data ordinal adalah data yang memiliki urutan atau hierarki, tetapi jarak antara kategori tidak terukur secara tepat. Data ini mengindikasikan posisi relatif tetapi tidak memberikan informasi tentang seberapa besar perbedaan antar kategori.
   - **Contoh**: Tingkat kepuasan (sangat puas, puas, tidak puas), pangkat militer (letnan, kapten, kolonel), dan tingkat pendidikan (SMA, S1, S2).

2.3.2 Data Kuantitatif

Data kuantitatif, juga dikenal sebagai data numerik, adalah data yang dapat diukur dan dihitung. Data ini terdiri dari angka dan memungkinkan perhitungan matematis untuk analisis statistik.

Jenis-jenis data kuantitatif meliputi:

1. **Data Diskret**:
   - **Definisi**: Data diskret adalah data yang hanya dapat mengambil nilai tertentu dalam rentang yang terpisah atau terhitung. Data ini biasanya berupa angka bulat.
   - **Contoh**: Jumlah anak dalam keluarga, jumlah mobil yang dimiliki seseorang, dan hasil lemparan dadu.
2. **Data Kontinu**:
   - **Definisi**: Data kontinu adalah data yang dapat mengambil nilai dalam rentang yang kontinu atau tak terhingga. Data ini dapat diukur dengan presisi yang lebih tinggi dan sering kali termasuk angka desimal.
   - **Contoh**: Tinggi badan, berat badan, dan suhu.

2.3.3 Data Kualitatif vs Kuantitatif

- **Jenis Informasi**:

  - Data kualitatif memberikan informasi tentang kategori atau atribut non-numerik.
  - Data kuantitatif memberikan informasi tentang ukuran atau jumlah yang dapat diukur.
- **Metode Analisis**:
  - Data kualitatif biasanya dianalisis menggunakan teknik seperti frekuensi, proporsi, dan analisis kategori.
  - Data kuantitatif biasanya dianalisis menggunakan teknik seperti statistik deskriptif, uji hipotesis, regresi, dan analisis varians.
- **Penggunaan**:
  - Data kualitatif sering digunakan dalam penelitian sosial dan psikologi untuk memahami perilaku dan karakteristik individu.
  - Data kuantitatif sering digunakan dalam penelitian ilmiah dan bisnis untuk analisis numerik dan pengambilan keputusan berdasarkan angka.

Memahami jenis data yang kamu miliki akan membantu dalam memilih metode analisis yang tepat dan mendapatkan wawasan yang lebih baik dari data yang tersedia.

2.4 Tingkatan Pengukuran

Dalam statistika, tingkatan pengukuran menggambarkan cara data diukur dan dikelompokkan. Tingkatan pengukuran ini penting karena menentukan jenis analisis statistik yang dapat dilakukan pada data. Ada empat tingkatan pengukuran utama: nominal, ordinal, interval, dan rasio.

2.4.1 Pengukuran Nominal

Definisi: Pengukuran nominal adalah tingkatan pengukuran yang menggunakan kategori untuk mengklasifikasikan data tanpa urutan atau hierarki. Data nominal hanya membedakan kategori atau kelompok.

Karakteristik:

- **Kategori**: Data dikategorikan tanpa urutan yang spesifik.
- **Operasi Statistik**: Frekuensi, proporsi.

Contoh:

- Jenis kelamin (pria/wanita)
- Warna mata (biru, coklat, hijau)
- Jenis kendaraan (mobil, motor, sepeda)

2.4.2 Pengukuran Ordinal

Definisi: Pengukuran ordinal adalah tingkatan pengukuran yang menggunakan kategori dengan urutan atau hierarki. Data ordinal menunjukkan urutan tetapi

tidak memberikan informasi tentang jarak antar kategori.

Karakteristik:

- **Urutan**: Data memiliki urutan yang jelas.
- **Operasi Statistik**: Frekuensi, median, peringkat.

Contoh:

- Tingkat kepuasan (sangat puas, puas, tidak puas)
- Peringkat dalam kompetisi (juara pertama, kedua, ketiga)
- Tingkat pendidikan (SMA, S1, S2)

2.4.3 Pengukuran Interval

Definisi: Pengukuran interval adalah tingkatan pengukuran yang tidak hanya memiliki urutan tetapi juga jarak yang terukur antara nilai-nilai. Namun, tidak ada titik nol mutlak dalam pengukuran interval.

Karakteristik:

- **Urutan dan Jarak**: Data memiliki urutan dan jarak yang terukur, tetapi tidak ada titik nol mutlak.
- **Operasi Statistik**: Mean, median, deviasi standar, uji t.

Contoh:

- Suhu dalam Celsius atau Fahrenheit (selisih 10°C menunjukkan perbedaan suhu yang sama dari 20°C ke 30°C seperti dari 30°C ke 40°C)
- Tahun kalender (selisih tahun 1990 dan 2000 sama dengan selisih tahun 2000 dan 2010)

2.4.4 Pengukuran Rasio

Definisi: Pengukuran rasio adalah tingkatan pengukuran yang memiliki urutan, jarak yang terukur, dan titik nol mutlak. Data rasio memungkinkan perbandingan relatif dan operasi matematis yang lebih lengkap.

Karakteristik:

- **Urutan, Jarak, dan Titik Nol**: Data memiliki urutan, jarak terukur, dan titik nol mutlak yang berarti tidak ada nilai negatif.
- **Operasi Statistik**: Mean, median, deviasi standar, rasio, perbandingan.

Contoh:

- Tinggi badan (dapat memiliki nilai nol mutlak)
- Berat badan (dapat memiliki nilai nol mutlak)
- Pendapatan (dapat memiliki nilai nol mutlak dan perbandingan langsung)

Memahami tingkatan pengukuran membantu dalam memilih teknik analisis statistik yang sesuai dan menginterpretasikan hasil data dengan benar. Setiap

tingkatan memberikan informasi berbeda dan memungkinkan berbagai jenis analisis yang berbeda.

2.5 Video

2.5.1 Pengenalan Statistics

2.5.2 Nominal, Ordinal, Interval & Ratio Data

Chapter 3

Pengumpulan Data

Pengumpulan data merupakan tahap penting dalam penelitian, di mana peneliti mengumpulkan informasi untuk menjawab pertanyaan penelitian atau menguji hipotesis. Kualitas hasil penelitian dipengaruhi oleh metode pengumpulan data yang digunakan, baik itu data primer maupun sekunder, serta pendekatan kualitatif dan kuantitatif.

3.1 Data Primer

Data primer adalah data yang dikumpulkan langsung dari sumbernya. Metode ini digunakan ketika data yang diperlukan belum tersedia.

3.1.1 Survei

Survei (Kuesioner) adalah metode yang melibatkan serangkaian pertanyaan kepada responden. Kuesioner dapat berupa pertanyaan tertutup (pilihan ganda) atau terbuka.

Contoh: Perusahaan e-commerce melakukan survei online kepada 1.000 pelanggan untuk mengetahui kepuasan terhadap layanan pengiriman, dengan pertanyaan seperti:

- Bagaimana Anda menilai kecepatan pengiriman barang?
- Apakah Anda puas dengan kualitas layanan?

Aspek	Deskripsi
Keunggulan	- Menjangkau banyak orang dengan cepat. - Biaya efisien, terutama untuk survei online.
Kelemahan	- Kualitas data tergantung pada pertanyaan. - Responden bisa tidak jujur.

3.1.2 Wawancara

Wawancara adalah metode pengumpulan data melalui percakapan langsung. Ini bisa terstruktur (pertanyaan tetap) atau semi-terstruktur (fleksibel).

Contoh: Peneliti mewawancarai petani untuk mengetahui dampak perubahan iklim terhadap produksi pertanian.

Aspek	Deskripsi
Keunggulan	- Menggali informasi mendalam. - Klarifikasi jawaban yang tidak jelas.
Kelemahan	- Memakan waktu dan tenaga. - Hasil bisa dipengaruhi interaksi pewawancara.

3.1.3 Observasi

Observasi adalah metode dengan mengamati subjek atau objek tanpa interaksi langsung.

Contoh: Manajer toko mengamati perilaku pelanggan untuk melihat pola pembelian dan menemukan bahwa pelanggan berbelanja lebih banyak pada sore hari.

Aspek	Deskripsi
Keunggulan	- Data lebih objektif. - Digunakan ketika responden tidak bisa menjawab.
Kelemahan	- Hanya untuk kejadian yang terlihat. - Bisa dipengaruhi bias pengamat.

3.1.4 Eksperimen

Eksperimen melibatkan kontrol variabel untuk melihat pengaruhnya terhadap variabel lain.

Contoh: Perusahaan kosmetik menguji efektivitas produk baru pada 100 orang selama 4 minggu.

Aspek	Deskripsi
Keunggulan	- Menguji hubungan sebab-akibat. - Hasil lebih dapat diandalkan.
Kelemahan	- Sulit dilakukan di luar laboratorium. - Memerlukan biaya dan waktu yang besar.

3.2 Data Sekunder

Data sekunder adalah data yang sudah dikumpulkan oleh pihak lain, tersedia dalam dokumen, laporan, atau basis data.

3.2.1 Sumber Tertulis

Pengumpulan data dari sumber tertulis melibatkan dokumen yang sudah ada, seperti laporan tahunan atau artikel.

Contoh: Analis keuangan menggunakan laporan tahunan perusahaan untuk menganalisis kinerja keuangan selama 5 tahun.

Aspek	Deskripsi
Keunggulan	- Tidak memerlukan banyak waktu dan biaya. - Lebih mudah diakses.
Kelemahan	- Data bisa tidak relevan atau tidak lengkap. - Kualitas data bervariasi.

3.2.2 Database Publik

Data sekunder juga bisa diperoleh dari basis data publik seperti data sensus.

Contoh: Peneliti sosial menggunakan data sensus dari Badan Pusat Statistik untuk meneliti pola migrasi di Indonesia.

Aspek	Deskripsi
Keunggulan	- Data terstruktur dan siap digunakan. - Cakupan luas untuk analisis komparatif.
Kelemahan	- Ketersediaan data bisa terbatas. - Mungkin tidak sesuai dengan kebutuhan penelitian.

3.3 Reliabilitas dan Validitas

Reliabilitas dan validitas adalah dua konsep penting yang menentukan kualitas informasi yang diperoleh dalam penelitian.

3.3.1 Reliabilitas Data

Reliabilitas adalah konsistensi pengukuran dari waktu ke waktu. Data yang reliabel memberikan hasil yang sama dalam kondisi yang sama.

Jenis-Jenis Reliabilitas:

1. **Test-Retest**: Mengukur konsistensi hasil saat pengukuran diulang.

2. **Inter-Rater**: Mengukur konsistensi hasil antara beberapa penilai.
3. **Internal**: Mengukur konsistensi antara item dalam instrumen pengukuran.

3.3.2 Validitas Data

Validitas mengukur sejauh mana instrumen mengukur apa yang seharusnya diukur.

Jenis-Jenis Validitas:

1. **Isi**: Mengukur sejauh mana item-item mencakup aspek konsep yang diukur.
2. **Kriteria**: Mengukur korelasi hasil instrumen dengan alat ukur lain yang valid.
3. **Konstruk**: Mengukur sejauh mana instrumen mengukur konstruk teoretis yang dimaksud.

3.4 Rangkuman

Aspek	Bisnis	Kesehatan	Keuangan	Sains Data
Pengumpulan Data	Kuesioner online, wawancara	Rekam medis, wawancara pasien	Sistem akuntansi, laporan keuangan	Sumber data terbuka, API, pengumpulan data manual
Reliabilitas	Hasil survei konsisten jika diulang	Data diperiksa berkala untuk kesalahan pengkodean	Proses pengumpulan data harus konsisten	Data diuji konsistensi sebelum analisis
Validitas	Kuesioner mencakup semua aspek kepuasan pelanggan	Instrumen diuji untuk mengukur kecemasan	Data mencerminkan aktivitas keuangan yang sebenarnya	Data relevan dan representatif
Kendala	Responden bisa tidak jujur	Data tidak lengkap mengurangi validitas	Ketidakakuratan pencatatan menurunkan kualitas	Variasi metode pengumpulan dari berbagai sumber

3.5 Video

3.5.1 What is Data Collection

3.5.2 What is Data Quality?

3.6 Latihan

- Kuesioner penelitian sosial
- Wawancara mendalam dalam penelitian kualitatif
- Observasi perilaku konsumen
- Uji klinis obat baru
- Studi kasus perusahaan untuk strategi bisnis
- Menggunakan data sensus untuk penelitian demografis

Part I

Statistika Desktiptif

Chapter 4

Penyajian Data

Penyajian Data adalah proses pengorganisasian, visualisasi, dan interpretasi data agar lebih mudah dipahami dan dapat diambil kesimpulan. Dalam konteks analisis data, penyajian dilakukan dengan menggunakan berbagai alat dan metode visualisasi seperti tabel, grafik, dan diagram, tergantung pada jenis datanya (kualitatif atau kuantitatif).

4.1 Memuat Dataset

Pertama-tama, kita memuat dataset (skincare) yang akan digunakan dalam analisis ini.

```
# Memuat dataset dari CSV
skincare = read.csv("Data/skincare.csv", sep = ";")

# Menampilkan data awal dengan kable
head(skincare)
```

```
##   ID_Responden Jenis_Kelamin Usia Pendapatan Preferensi_Produk
## 1            1        Wanita   31    2268166          Haircare
## 2            2          Pria   21    2345555            Makeup
## 3            3        Wanita   56    5512972          Haircare
## 4            4        Wanita   18    2280645            Makeup
## 5            5          Pria   51    5246913            Makeup
## 6            6        Wanita   40    4040500          Skincare
##   Frekuensi_Pembelian Cara_Pembelian Tingkat_Kepuasan Tanggal_Pembelian
## 1                   3         Online                1        2021-01-01
## 2                   5         Online                2        2021-01-02
## 3                   2        Offline                1        2021-01-03
## 4                   5        Offline                3        2021-01-04
## 5                   2        Offline                4        2021-01-05
```

```
## 6                   2        Offline                 4        2021-01-06
##   Jumlah_Pembelian
## 1                2
## 2                3
## 3                9
## 4                3
## 5                7
## 6                6
```

4.2 Data Kualitatif

Data kualitatif biasanya menggambarkan kategori atau kelompok, seperti jenis kelamin, preferensi produk, atau cara pembelian. Penyajian data kualitatif bisa dilakukan dengan:

4.2.1 Tabel Distribusi Frekuensi

Misalkan anda ingin menampilkan kolom Preferensi_Produk: dapat dilakukan dengan cara menggunakan table() dan kemudian mengonversinya ke dalam data frame.

```r
# Memeriksa nama kolom dalam dataset
# names(skincare)

# Menghitung frekuensi
frekuensi <- table(skincare$Preferensi_Produk)

# Mengonversi tabel frekuensi ke dalam data frame
table_distribusi <- as.data.frame(frekuensi)

# Mengganti nama kolom secara manual
colnames(table_distribusi) <- c("Kategori Produk", "Frekuensi")

# Menampilkan tabel distribusi
table_distribusi
```

```
##   Kategori Produk Frekuensi
## 1        Haircare       391
## 2          Makeup       339
## 3        Skincare       365
```

Cara yang lebih sederhana untuk membuat tabel distribusi dan langsung memberi nama kolom adalah dengan menggunakan `dplyr`. Berikut ini adalah contohnya:

```r
# Memuat library dplyr
library(dplyr)
```

```r
# Membuat tabel distribusi menggunakan kolom yang benar
table_distribusi <- skincare %>%
  count(Preferensi_Produk, name = "Frekuensi") %>%
  rename("Kategori Produk" = Preferensi_Produk) %>%
  arrange(desc(Frekuensi))  # Mengurutkan dari yang terbesar ke terkecil
# Menampilkan tabel distribusi
table_distribusi
```

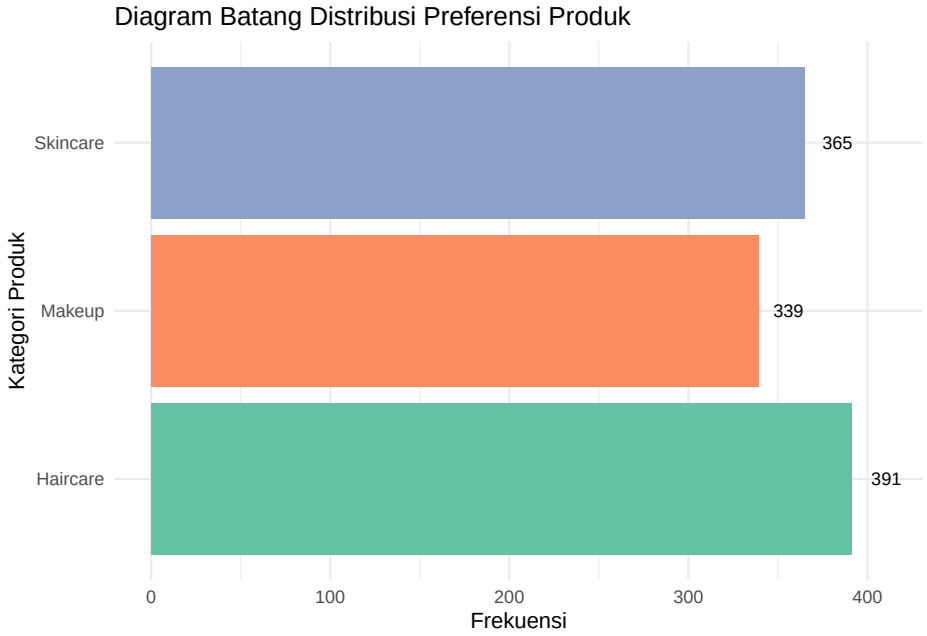
```
##   Kategori Produk Frekuensi
## 1        Haircare       391
## 2        Skincare       365
## 3          Makeup       339
```

4.2.2 Diagram Batang

Diagram Batang adalah jenis grafik yang digunakan untuk menampilkan dan membandingkan frekuensi atau jumlah dari kategori yang berbeda. Dalam diagram batang, setiap kategori diwakili oleh sebuah batang, dan panjang atau tinggi batang tersebut mencerminkan nilai yang terkait, seperti jumlah frekuensi atau proporsi.

Untuk membuat diagram batang dari data yang telah Anda siapkan menggunakan `dplyr`, Anda dapat melanjutkan dengan menggunakan `ggplot2`. Berikut adalah langkah-langkah lengkapnya, mulai dari menghitung frekuensi hingga membuat diagram batang.

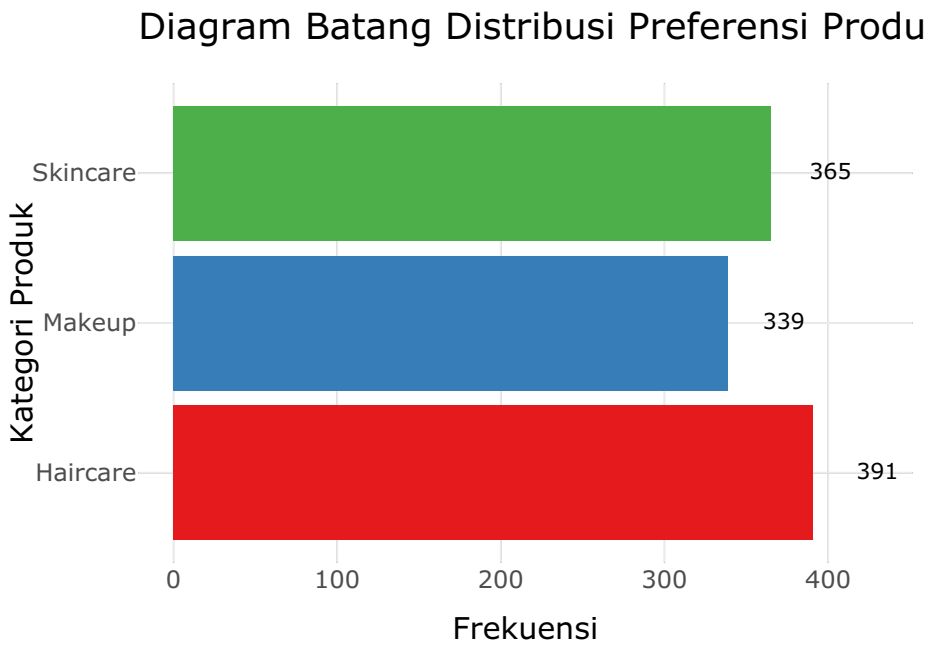
```r
library(ggplot2)
# Membuat diagram batang
ggplot(table_distribusi, aes(x = `Kategori Produk`,
                             y = Frekuensi,
                             fill = `Kategori Produk`)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = Frekuensi),
            position = position_stack(vjust = 1.05), # Mengatur posisi label
            color = "black", size = 3) +             # atur warna & ukuran label
  labs(title = "Diagram Batang Distribusi Preferensi Produk",
       x = "Kategori Produk",
       y = "Frekuensi") +
  scale_fill_brewer(palette = "Set2") +   # Menggunakan palet warna
  theme_minimal() +
  theme(legend.position = "none") +       # Menyembunyikan legenda
  coord_flip()                            # diagram menjadi horizontal
```



Jika anda ingin mengonversi visualisasi ggplot2 menjadi interaktif menggunakan plotly, Anda bisa menggunakan fungsi ggplotly() dari paket plotly. Ini memungkinkan Anda mengubah diagram batang statis dari ggplot2 menjadi grafik interaktif. Pastikan Anda sudah memuat library plotly sebelum menjalankan kode ini.

```
# Memuat library yang diperlukan
library(ggplot2)
library(plotly)

# Membuat diagram batang dengan ggplot2
p <- ggplot(table_distribusi, aes(x = `Kategori Produk`,
                                  y = Frekuensi,
                                  fill = `Kategori Produk`)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = Frekuensi),
            position = position_stack(vjust = 1.1),
            color = "black", size = 3) +  # posisi,warna dan label
  labs(title = "Diagram Batang Distribusi Preferensi Produk",
       x = "Kategori Produk",
       y = "Frekuensi") +
  scale_fill_brewer(palette = "Set1") +  # Menggunakan palet warna
  theme_minimal() +
  theme(legend.position = "none") +      # Menyembunyikan legenda
  coord_flip()                           # diagram horizontal
```

```
# Mengonversi ggplot menjadi plotly untuk interaktif
plotly_plot <- ggplotly(p)

# Menampilkan plot interaktif
plotly_plot
```



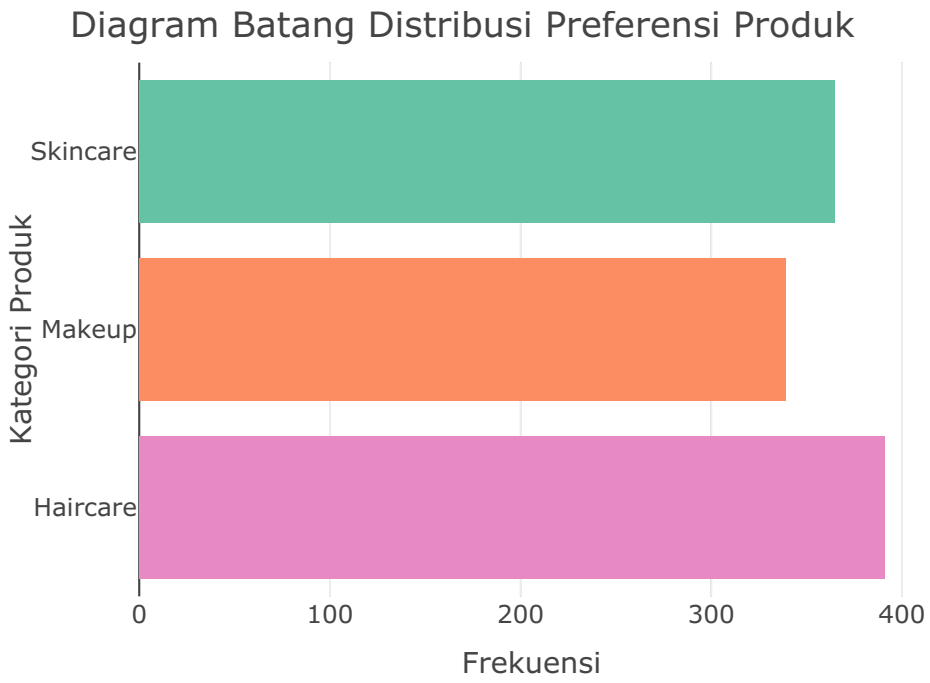
Cara lainnya untuk membuat diagram batang interaktif langsung dengan plotly adalah menggunakan fungsi plot_ly() tanpa terlebih dahulu membuat grafik dengan ggplot2. Anda bisa membuat grafik langsung dari data yang sudah diolah dalam table_distribusi.

```
# Memuat library yang diperlukan
library(plotly)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
# Menambahkan warna kustom berbeda untuk setiap kategori
colors <- c("#e78ac3","#66c2a5", "#fc8d62", "#a6d854", "#ffd92f")

# Menggunakan palet warna dari viridis
# colors <- viridis(n = nrow(table_distribusi), option = "D")

# Membuat diagram batang horizontal dengan plotly dan warna berbeda
plot_ly(data = table_distribusi,
        x = ~Frekuensi,
```

```
        y = ~`Kategori Produk`,
        type = 'bar',
        orientation = 'h',  # Membuat diagram batang horizontal
        marker = list(color = colors)) %>%
  layout(title = "Diagram Batang Distribusi Preferensi Produk",
         xaxis = list(title = "Frekuensi"),
         yaxis = list(title = "Kategori Produk"),
         showlegend = FALSE)  # Menyembunyikan legenda
```



Struktur Diagram Batang

- **Sumbu X (horizontal):** Mewakili kategori yang akan dibandingkan.
- **Sumbu Y (vertikal):** Mewakili nilai frekuensi atau jumlah untuk setiap kategori.
- **Batang:** Tinggi atau panjang batang menunjukkan seberapa besar nilai dari kategori tersebut.

Penerapan Diagram Batang

Diagram batang digunakan untuk:

- **Membandingkan Kategori:** Membandingkan frekuensi atau nilai antar kategori, seperti penjualan produk.
- **Data Kategorikal:** Menampilkan distribusi frekuensi data yang bersifat kategorikal.

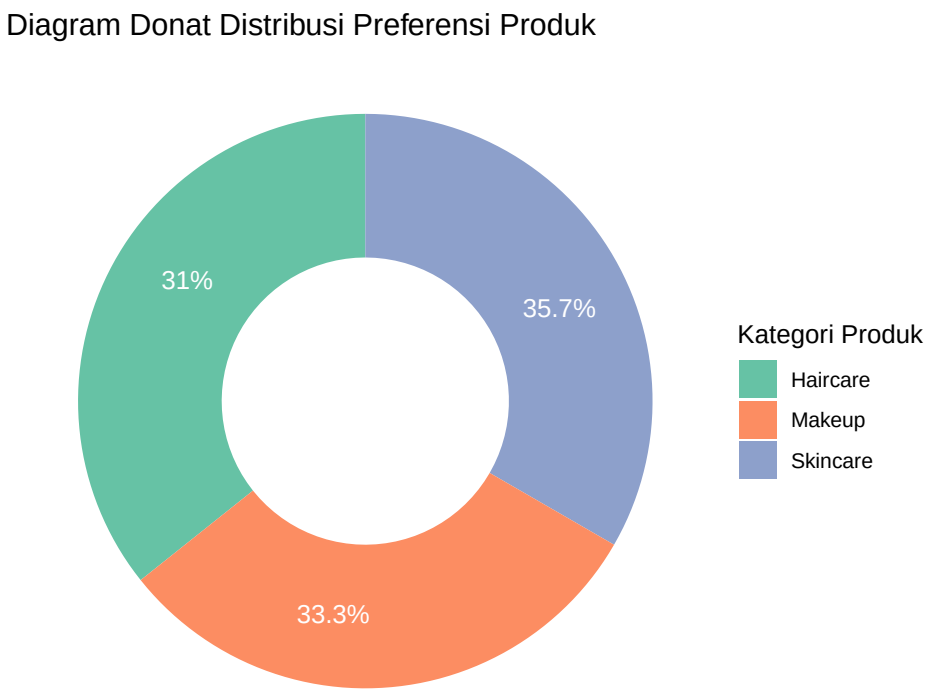
- **Hasil Survei:** Menyajikan pilihan responden dalam survei.
- **Perubahan dari Waktu ke Waktu:** Melacak perubahan kategori dari waktu ke waktu, seperti dalam diagram batang kelompok.
- **Menyoroti Tren:** Mengidentifikasi pola atau tren dalam data.

4.2.3 Diagram Lingkaran

Diagram lingkaran (pie chart) adalah visualisasi data yang menunjukkan proporsi atau persentase dari suatu total. Setiap bagian dari lingkaran mewakili kontribusi kategori tertentu terhadap keseluruhan.

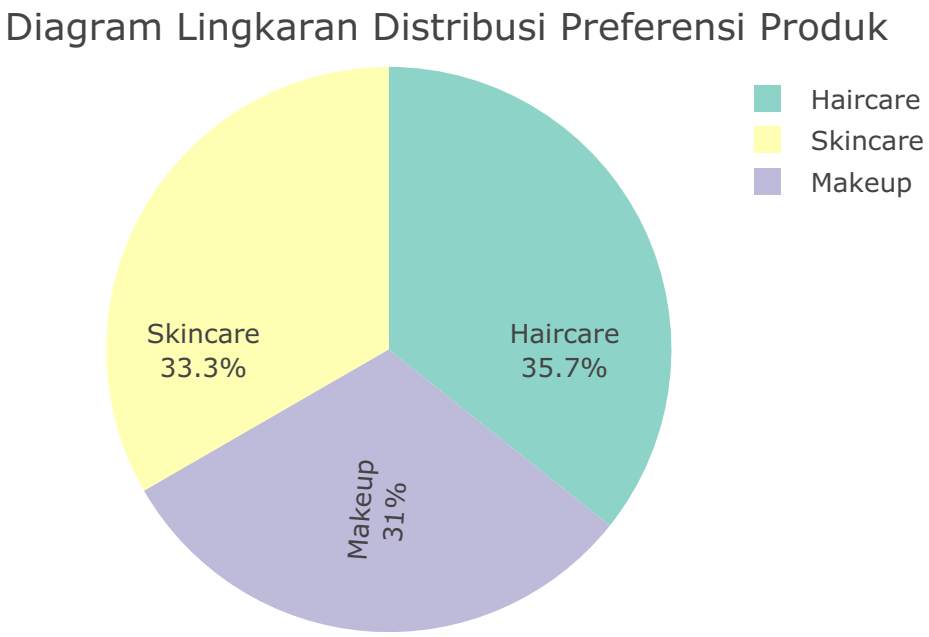
Berikut adalah cara membuat diagram lingkaran sederhana menggunakan ggplot2 untuk data table_distribusi, yang menunjukkan distribusi kategori produk:

```
# Memuat library yang diperlukan
library(ggplot2)

# Menghitung posisi label
table_distribusi <- table_distribusi %>%
  # Menghitung persentase dan posisi label di tengah-tengah segmen
  dplyr::mutate(Proporsi = Frekuensi / sum(Frekuensi) * 100,
                Posisi = cumsum(Frekuensi) - Frekuensi / 2)

# Membuat diagram donat dengan label
ggplot(table_distribusi, aes(x = 2,
                             y = Frekuensi,
                             fill = `Kategori Produk`)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Diagram Donat Distribusi Preferensi Produk") +
  theme_void() +
  xlim(0.5, 2.5) +
  scale_fill_brewer(palette = "Set2") +
  # Menambahkan label persentase
  geom_text(aes(y = Posisi, label = paste0(round(Proporsi, 1), "%")),
            color = "white", size = 4)   # Warna dan ukuran label
```



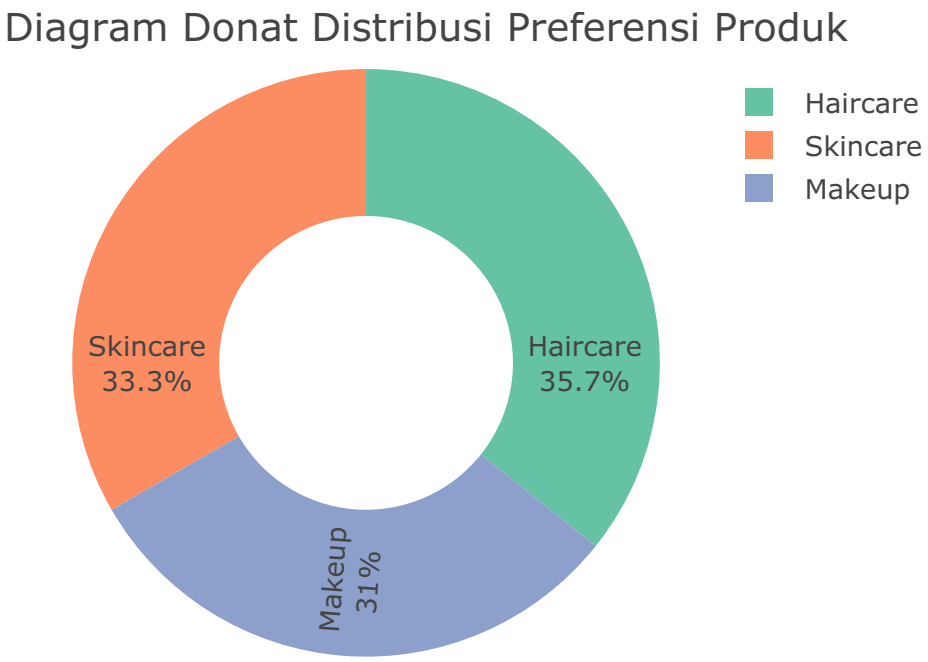
Berikut adalah konversi diagram lingkaran dari ggplot2 ke plotly. Kode ini akan menghasilkan diagram lingkaran interaktif dengan data yang sama

```
# Memuat library yang diperlukan
library(plotly)
library(RColorBrewer)

# Membuat diagram lingkaran dengan plotly
plot_ly(table_distribusi,
        labels = ~`Kategori Produk`,
        values = ~Frekuensi,
        type = 'pie',
        textinfo = 'label+percent',       # label dan persentase
        insidetextorientation = 'radial', # teks di dalam irisan
        marker = list(colors = brewer.pal(n = nrow(table_distribusi),
                                          name = "Set3"))) %>%
  layout(title = "Diagram Lingkaran Distribusi Preferensi Produk",
         showlegend = TRUE)               # Menampilkan legenda
```



Untuk membuat diagram donat (donut chart) menggunakan plotly, Anda bisa menggunakan fungsi plot_ly() dan mengatur parameter untuk menciptakan efek donat. Berikut adalah langkah-langkahnya:

```
# Memuat library yang diperlukan
library(plotly)

# Membuat diagram donat dengan plotly
plot_ly(table_distribusi,
        labels = ~`Kategori Produk`,
        values = ~Frekuensi,
        type = 'pie',
        textinfo = 'label+percent',         # label dan persentase
        insidetextorientation = 'radial',   # teks di dalam irisan
        hole = 0.5,                         # Mengatur ukuran lubang
        marker = list(colors = RColorBrewer::brewer.pal(n = nrow(table_distribusi),
                                                        name = "Set2"))) %>%
  layout(title = "Diagram Donat Distribusi Preferensi Produk",
         showlegend = TRUE)                 # Menampilkan legenda
```



Struktur Diagram Lingkaran

- **Lingkaran Utama:** Merepresentasikan total keseluruhan data.
- **Irisan (Slices):** Bagian dari lingkaran yang menunjukkan proporsi setiap kategori.
- **Label:** Menyediakan nama kategori dan seringkali menyertakan persentase atau nilai.
- **Legends:** Menjelaskan warna atau pola untuk membedakan kategori.
- **Warna:** Setiap irisan diberi warna berbeda untuk meningkatkan keterbacaan.
- **Total:** Kadang-kadang ditampilkan di tengah lingkaran untuk menunjukkan keseluruhan (100%)

Penerapan Diagram Lingkaran

Diagram lingkaran digunakan untuk:

- **Menunjukkan Proporsi:** Visualisasi proporsi setiap kategori terhadap total.
- **Data Kategorikal:** Menampilkan perbandingan antar kategori.
- **Hasil Survei:** Menunjukkan pilihan responden.
- **Analisis Sederhana:** Memudahkan analisis tanpa detail rumit.
- **Membandingkan Kategori Kecil:** Cocok untuk kategori yang relatif sedikit.

4.3 Data Kuantitatif

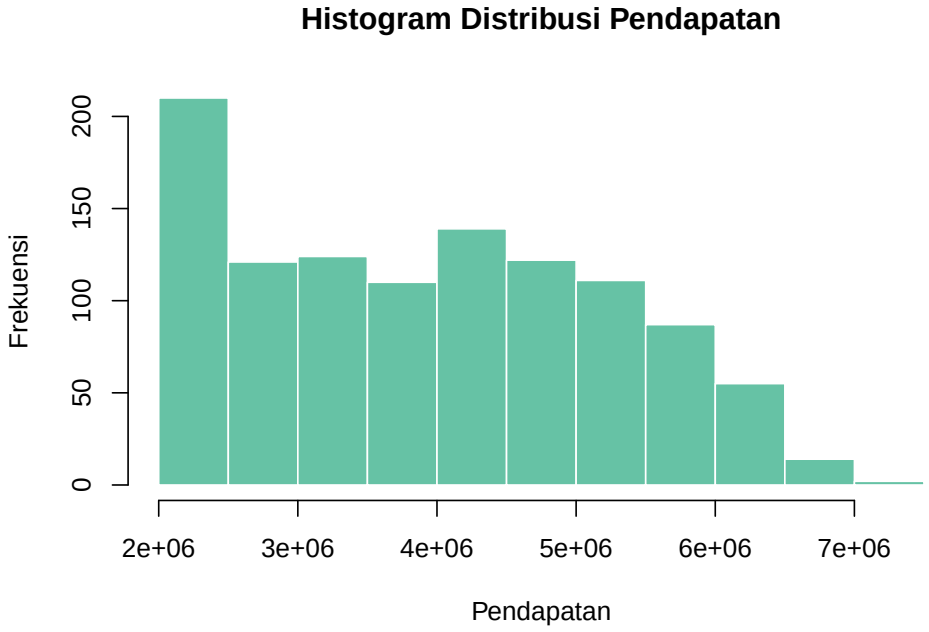
Data kuantitatif menggambarkan variabel numerik seperti usia, pendapatan, dan jumlah pembelian. Penyajian data kuantitatif dapat dilakukan dengan:

4.3.1 Diagram Histogram

Histogram adalah representasi grafis dari distribusi data numerik yang berbentuk batang (bar chart), di mana sumbu horizontal (x-axis) menunjukkan rentang nilai atau interval (bin), dan sumbu vertikal (y-axis) menunjukkan frekuensi atau jumlah data dalam setiap rentang.

Digunakan untuk menunjukkan distribusi data. Berikut adalah contoh pembuatan diagram histogram menggunakan data dari dataset skincare. Saya akan menggunakan kolom yang umum seperti `Pendapatan`. Jika kolom yang dimaksud berbeda, silakan ganti sesuai kebutuhan.

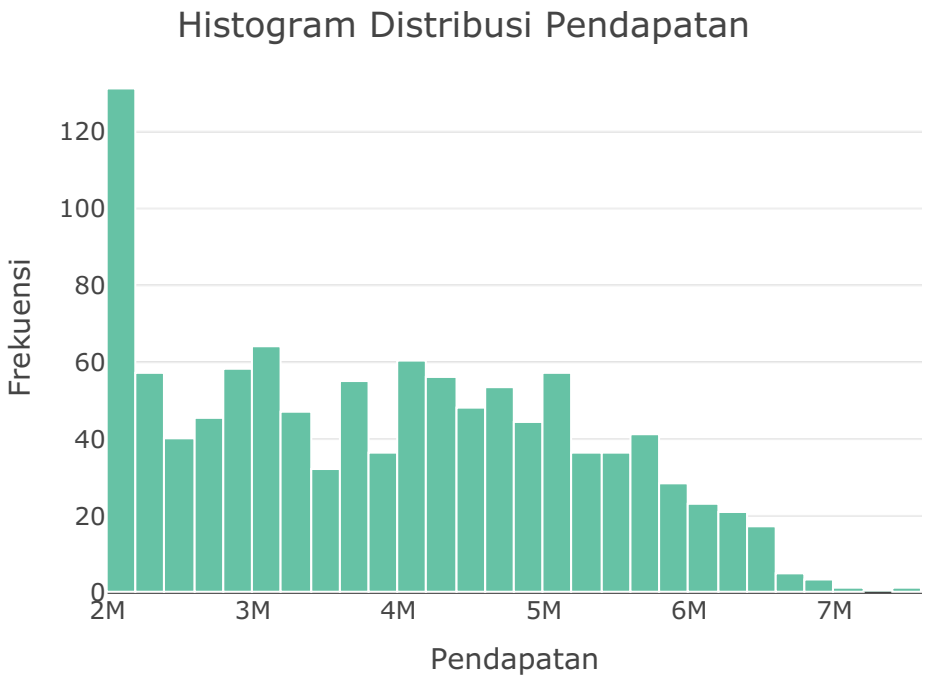
```r
# Membuat histogram menggunakan base R
hist(skincare$Pendapatan,
     main = "Histogram Distribusi Pendapatan",
     xlab = "Pendapatan",
     ylab = "Frekuensi",
     col = "#66c2a5",
     border = "white")  # Mengatur warna dan border
```



Untuk membuat histogram menggunakan plotly, Anda dapat menggunakan fungsi plot_ly() yang menyediakan opsi interaktif untuk visualisasi. Berikut

adalah cara membuat histogram distribusi Pendapatan dari dataset skincare dengan plotly.

```
# Memuat library yang diperlukan
library(plotly)

# Membuat histogram dengan variasi warna
plot_ly(data = skincare,
        x = ~Pendapatan,
        type = "histogram",
        marker = list(color = RColorBrewer::brewer.pal(n = 3, name = "Set2")[1],
                      line = list(color = "white", width = 1))) %>%
  layout(title = "Histogram Distribusi Pendapatan",
         xaxis = list(title = "Pendapatan"),
         yaxis = list(title = "Frekuensi"))
```



Struktur Histogram {-}

Histogram adalah grafik batang yang menunjukkan distribusi frekuensi data. Komponennya meliputi:

- **Sumbu X:** Interval kelas data.
- **Sumbu Y:** Frekuensi atau jumlah data dalam setiap interval.
- **Batang:** Tinggi batang mewakili frekuensi data dalam interval tersebut.

Histogram membantu memvisualisasikan pola distribusi data numerik, seperti simetri atau kecondongan (skewness).

Penerapan Histogram

Histogram digunakan ketika ingin:

- Melihat distribusi frekuensi dari data numerik.
- Mengetahui pola distribusi data (misalnya normal, miring, atau bimodal).
- Menganalisis data dalam kelompok atau interval tertentu.
- Mengidentifikasi outlier atau data yang menyimpang.

4.3.2 Diagram Garis

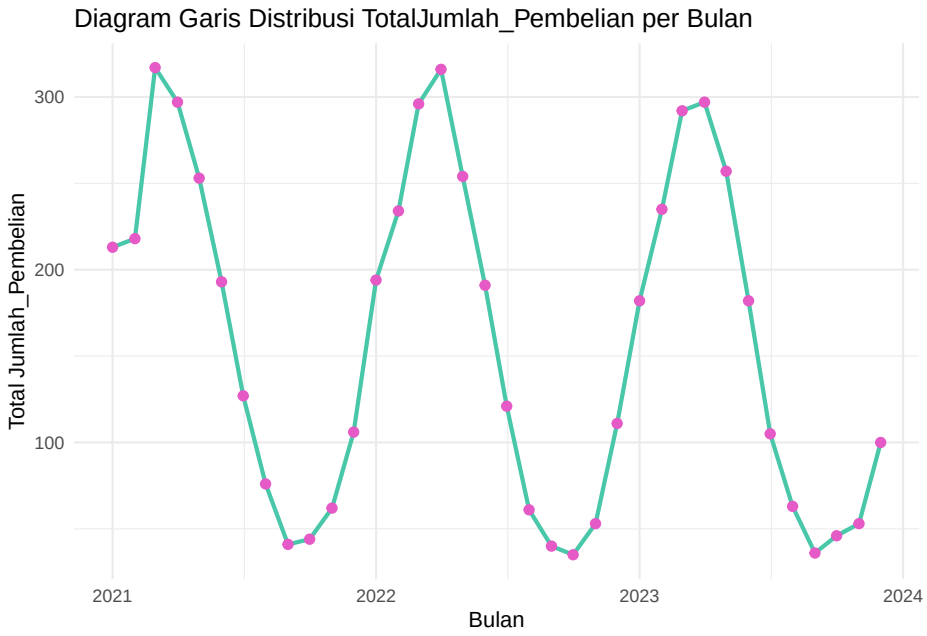
Diagram garis adalah jenis grafik yang digunakan untuk menunjukkan perubahan nilai dari satu atau lebih variabel seiring waktu. Grafik ini sangat berguna untuk mengidentifikasi tren dan pola dalam data, terutama jika data tersebut bersifat numerik dan berurutan.

Misalkan kita memiliki dataset skincare dengan kolom Tanggal dan Pendapatan. Berikut adalah contoh bagaimana cara membuat diagram garis dengan ggplot2:

```
# Memuat library yang diperlukan
library(dplyr)
library(ggplot2)

# Mengonversi kolom Tanggal menjadi tipe Date jika belum
skincare$Tanggal <- as.Date(skincare$Tanggal)

# Mengelompokkan data berdasarkan bulan dan tahun
data_bulanan <- skincare %>%
  mutate(Bulan = format(Tanggal, "%Y-%m")) %>%   # format "YYYY-MM"
  group_by(Bulan) %>%                            # Mengelompokkan
  summarise(Total_Jumlah_Pembelian = sum(Jumlah_Pembelian, na.rm = TRUE))
# Membuat diagram garis untuk pendapatan bulanan
ggplot(data = data_bulanan, aes(x = as.Date(paste0(Bulan, "-01")), y = Total_Jumlah_Pembelian)) +
  geom_line(color = "#49c7a9", size = 1) +      # warna dan ukuran garis
  geom_point(color = "#e55ac6", size = 2) +     # titik pada data
  labs(title = "Diagram Garis Distribusi TotalJumlah_Pembelian per Bulan",
       x = "Bulan",
       y = "Total Jumlah_Pembelian") +
  theme_minimal()  # Menggunakan tema minimal
```

```
## Warning: Using `size` aesthetic for lines was
## deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every
## 8 hours.
## Call
## `lifecycle::last_lifecycle_warnings()`
## to see where this warning was
```

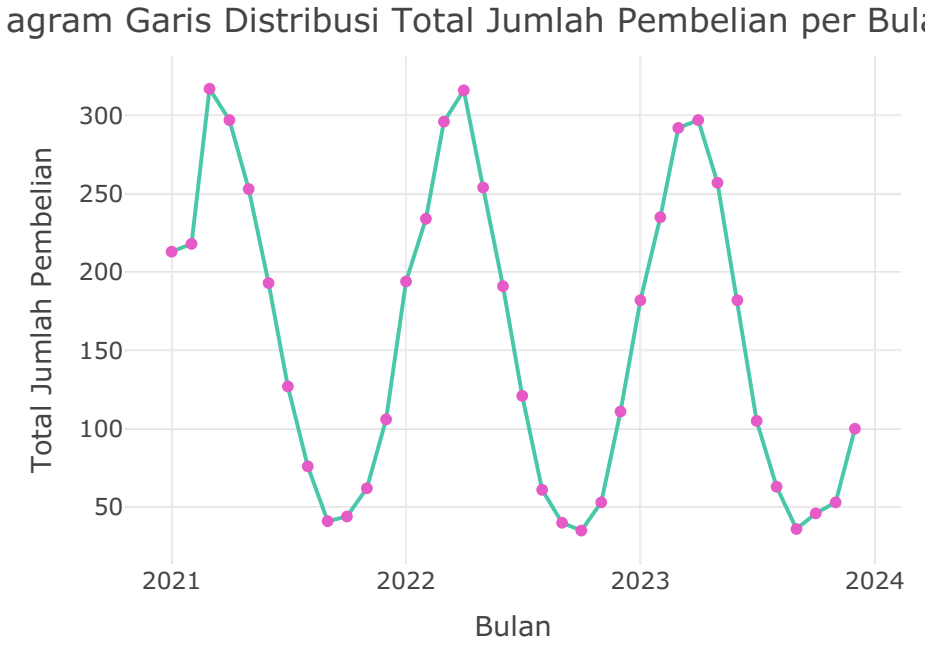
```
## generated.
```



Untuk membuat diagram garis yang menunjukkan total pendapatan per hari menggunakan Plotly, Anda dapat mengikuti langkah-langkah berikut. Kode di bawah ini mengelompokkan data berdasarkan tanggal dan menghitung total pendapatan setiap hari, kemudian memvisualisasikannya menggunakan Plotly.

```r
# Memuat library yang diperlukan
library(dplyr)
library(plotly)

# Mengonversi kolom Tanggal menjadi tipe Date jika belum
skincare$Tanggal <- as.Date(skincare$Tanggal)
# Mengelompokkan data berdasarkan bulan dan tahun
data_bulanan <- skincare %>%
  mutate(Bulan = format(Tanggal, "%Y-%m")) %>%
  group_by(Bulan) %>%
  summarise(Total_Jumlah_Pembelian = sum(Jumlah_Pembelian, na.rm = TRUE))
# Membuat plot interaktif menggunakan Plotly
plot <- plot_ly(data = data_bulanan,
                x = ~as.Date(paste0(Bulan, "-01")),
                y = ~Total_Jumlah_Pembelian,
                type = 'scatter',
                mode = 'lines+markers',
                line = list(color = "#49c7a9", width = 2),
                marker = list(color = "#e55ac6", size = 6)) %>%
```

```
  layout(title = "Diagram Garis Distribusi Total Jumlah Pembelian per Bulan",
         xaxis = list(title = "Bulan"),
         yaxis = list(title = "Total Jumlah Pembelian"),
         showlegend = FALSE)

# Tampilkan plot
plot
```



Struktur Diagram Garis

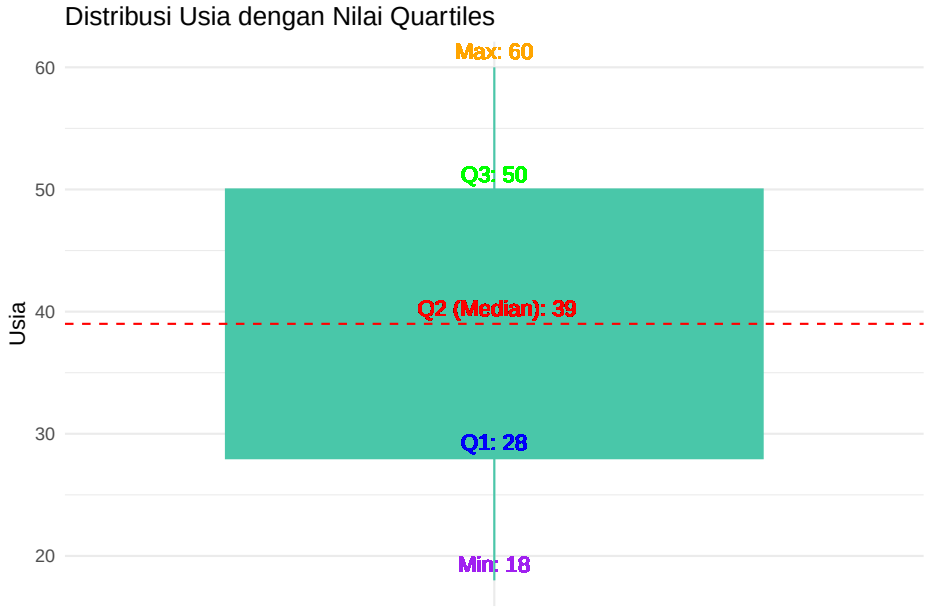
- **Sumbu:** X (Waktu) dan Y (Nilai).
- **Titik Data:** Menunjukkan nilai untuk kombinasi X dan Y.
- **Garis:** Menghubungkan titik data untuk menunjukkan tren.
- **Label:** Judul, label sumbu, dan label data (opsional).
- **Legends:** Menunjukkan kategori jika ada lebih dari satu garis.
- **Tema:** Mengatur tampilan (warna, gaya).

Penerapan Diagaram Garis

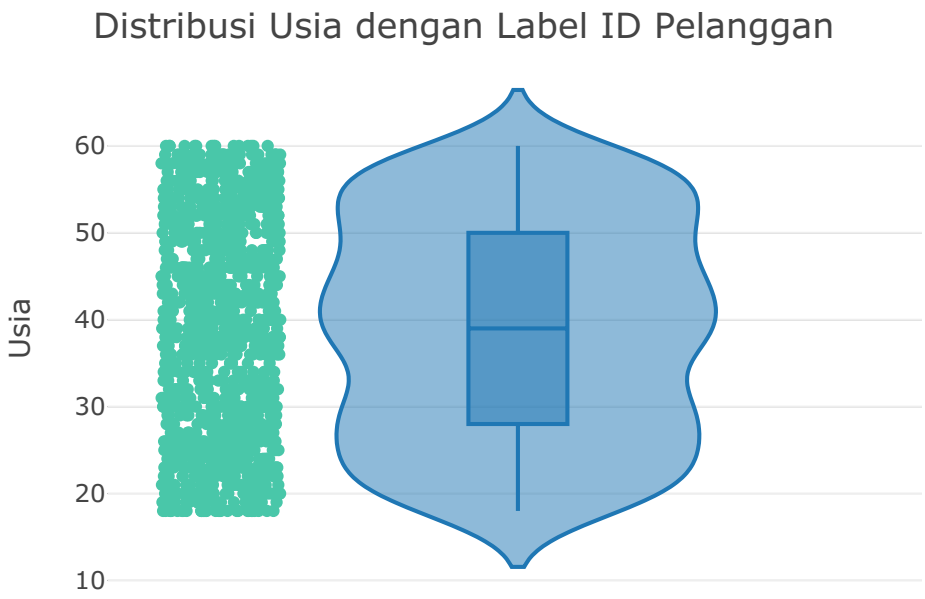
- Menunjukkan tren waktu.
- Membandingkan data dengan rentang waktu sama.
- Menampilkan hubungan antara dua variabel.

4.3.3 Diagram Boxplot

```r
# Memuat library yang diperlukan
library(ggplot2)
library(ggrepel)  # Library untuk menghindari label yang saling bertumpukan
library(dplyr)    # Library untuk manipulasi data

# Menghitung nilai quartiles dan statistik lainnya
stat_summary <- skincare %>%
  summarise(
    Min = min(Usia, na.rm = TRUE),
    Q1 = quantile(Usia, 0.25, na.rm = TRUE),
    Median = median(Usia, na.rm = TRUE),
    Q3 = quantile(Usia, 0.75, na.rm = TRUE),
    Max = max(Usia, na.rm = TRUE)
  )

# Membuat Boxplot untuk Usia dan menambahkan nilai quartiles
boxplot <- ggplot(skincare, aes(x = "", y = Usia)) +  # Dummy x axis
  geom_boxplot(outlier.shape = NA, fill = "#49c7a9", color = "#49c7a9") +
  labs(title = "Distribusi Usia dengan Nilai Quartiles",
       y = "Usia",
       x = NULL) +
  theme_minimal()

# Menambahkan garis median ke plot
boxplot <- boxplot +
  geom_hline(yintercept = stat_summary$Median,
             linetype = "dashed", color = "red") +  # Garis median
  geom_text(aes(x = 1, y = stat_summary$Median,
                label = paste(" Q2 (Median):", stat_summary$Median)),
            vjust = -0.5, color = "red") +
  geom_text(aes(x = 1, y = stat_summary$Q1,
                label = paste("Q1:", stat_summary$Q1)), vjust = -0.5, color = "blue") +
  geom_text(aes(x = 1, y = stat_summary$Q3,
                label = paste("Q3:", stat_summary$Q3)), vjust = -0.5, color = "green") +
  geom_text(aes(x = 1, y = stat_summary$Min,
                label = paste("Min:", stat_summary$Min)), vjust = -0.5, color = "purple") +
  geom_text(aes(x = 1, y = stat_summary$Max,
                label = paste("Max:", stat_summary$Max)), vjust = -0.5, color = "orange")

# Tampilkan plot
print(boxplot)
```




```r
# Memuat library yang diperlukan
library(plotly)

# Membuat Violin Plot untuk Usia dengan label ID Pelanggan
violin_plot <- plot_ly(
  data = skincare,
  y = ~Usia,                                  # Hanya Usia untuk sumbu y
  type = 'violin',                            # Tipe plot violin
  box = list(visible = TRUE),                 # Menampilkan boxplot di dalam violin
  points = "all",                             # Menampilkan semua poin
  jitter = 0.3,                               # Menambahkan sedikit jitter pada poin
  text = ~paste("ID Pelanggan:", ID_Responden,
                "<br>Usia:", Usia),           # Menampilkan ID Pelanggan dan Usia
  hoverinfo = "text",                         # Menampilkan label pada hover
  marker = list(color = "#49c7a9"),           # Warna marker
  fillcolor = I("rgba(73, 199, 169, 0.5)")# Warna fill violin
) %>%
  layout(
    title = "Distribusi Usia dengan Label ID Pelanggan",
    yaxis = list(title = "Usia"),
    xaxis = list(showticklabels = FALSE), # Tidak menampilkan label x-axis
    showlegend = FALSE
  )

# Tampilkan plot
violin_plot
```



4.4 Multivariat Data

Data multivariat adalah data yang melibatkan lebih dari dua variabel pada waktu yang sama. Penyajian data multivariat sering menggunakan:

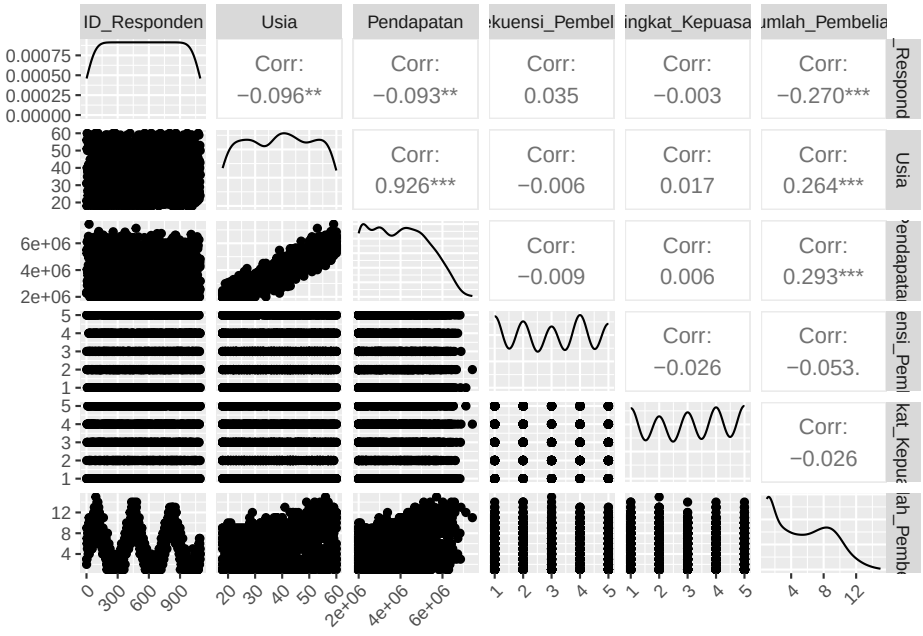
4.4.1 Scatter Plot Matrix

Menampilkan hubungan antara banyak variabel numerik.

```
library(dplyr)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
# Seleksi variabel numerik
numerical_data <- skincare %>% select_if(is.numeric)

# Membuat scatter plot matrix dengan GGally
ggpairs(numerical_data) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

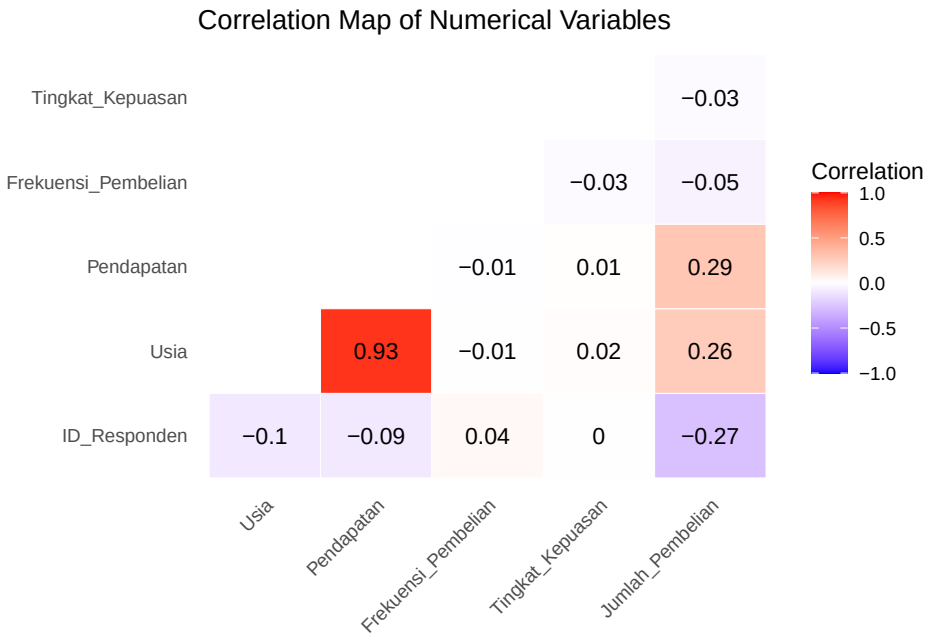


4.4.2 Heatmap

Menampilkan korelasi antar variabel dalam bentuk visual.

```
library(ggplot2)
library(reshape2)
library(dplyr)
# Seleksi variabel numerik
numerical_data <- skincare %>% select_if(is.numeric)

# Menghitung matriks korelasi
cor_matrix <- cor(numerical_data)

# Mengubah matriks korelasi menjadi format long
cor_matrix_melted <- melt(cor_matrix)

# Filter hanya segitiga atas (untuk pasangan unik)
cor_matrix_melted <- cor_matrix_melted %>%
  filter(as.numeric(Var2) < as.numeric(Var1))

# Membuat heatmap korelasi dengan ggplot2
ggplot(cor_matrix_melted, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1, 1),
                       name = "Correlation") +
```

```
  geom_text(aes(label = round(value, 2)), color = "black", size = 4) +  # Menambahkan
  theme_minimal() +
  labs(title = "Correlation Map of Numerical Variables", x = "", y = "") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(panel.grid.major = element_blank())  # Menghilangkan grid
```

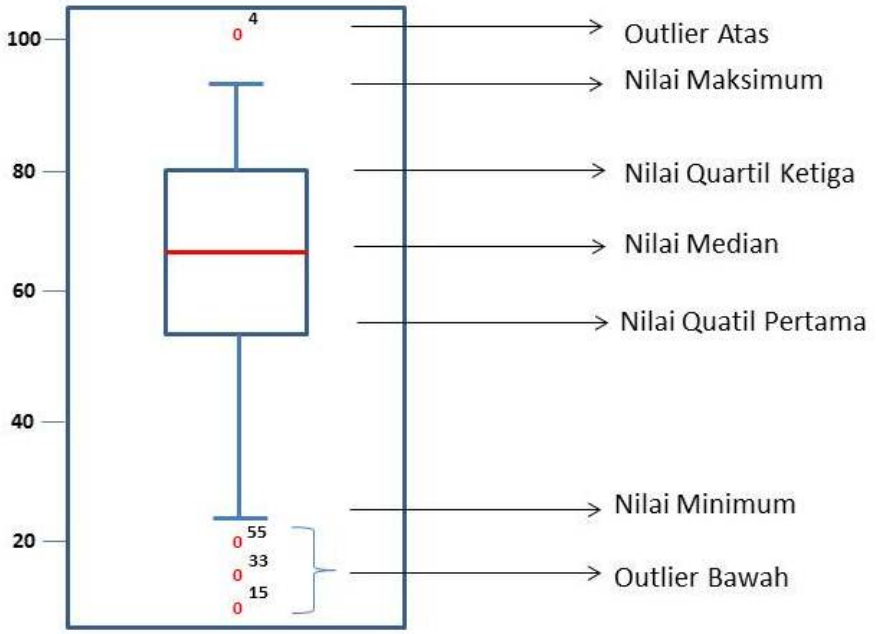


Chapter 5

Ukuran Pemusatan Data

Dalam era data yang terus berkembang, memahami ukuran pemusatan menjadi keterampilan penting dalam analisis data. Ukuran seperti mean, median, dan modus membantu menggambarkan karakteristik utama distribusi data dari rata-rata nilai hingga pola frekuensi tertinggi.

Bab ini memberikan pengantar konsep ukuran pemusatan dalam sains data, menggabungkan teori dan aplikasi praktis. Pembahasan meliputi peran ukuran pemusatan dalam eksplorasi data, studi kasus pada data dunia nyata, dan kaitannya dengan visualisasi seperti histogram densitas dan boxplot.



5.1 Definisi dan Konsep

Ukuran pemusatan adalah metode statistik yang digunakan untuk menentukan nilai yang mewakili pusat dari kumpulan data. Nilai ini membantu menggambarkan karakteristik umum data secara keseluruhan dan memberikan informasi tentang bagaimana data terdistribusi di sekitar titik tertentu.

5.2 Peran Ukuran Pemusatan

Ukuran pemusatan memiliki peran penting dalam statistik dan analisis data, terutama dalam menyederhanakan dan menyimpulkan informasi dari kumpulan data. Berikut adalah beberapa peran utama ukuran pemusatan data:

Peran	**Penjelasan**
Penyederhanaan	Ukuran pemusatan (mean, median, modus) memberikan gambaran umum data besar dengan satu nilai representatif.
Perbandingan	Memungkinkan perbandingan karakteristik utama antara beberapa kelompok data, seperti pendapatan antar wilayah.

Peran	Penjelasan
Identifikasi Pola & Tren	Membantu mengenali kecenderungan atau pola data, berguna untuk memprediksi tren atau perubahan di masa depan.
Informasi Dasar	Menjadi dasar analisis statistik lanjutan, seperti perhitungan varians dan deviasi standar.
Pengambilan Keputusan	Digunakan untuk mendukung keputusan berbasis data, misalnya dalam menentukan strategi pemasaran berdasarkan rata-rata penjualan.
Mengatasi Keragaman	Median dan modus memberikan informasi yang lebih tepat daripada mean ketika data memiliki variasi besar atau outliers.

5.3 Mean (Rata-rata)

Mean (atau rata-rata) adalah ukuran pemusatan yang paling umum digunakan dalam statistik. Mean dihitung dengan menjumlahkan semua nilai dalam suatu kumpulan data, lalu membaginya dengan jumlah data yang ada. Mean memberikan gambaran umum tentang posisi pusat dari data.

Rumus untuk menghitung mean adalah sebagai berikut:

$$\text{Mean} = \frac{\sum X}{n}$$

dimana:

- $\sum X$ adalah jumlah dari semua nilai data.
- n adalah jumlah data.

Langkah-langkah untuk menghitung mean:

- Jumlahkan semua nilai dalam data.
- Bagi hasil jumlah dengan banyaknya data.

Misalkan kita memiliki data sebagai berikut: 4, 5, 7, 8, 9.

- Jumlahkan semua nilai:

$$4 + 5 + 7 + 8 + 9 = 33$$

- Bagi dengan jumlah data ($n = 5$):

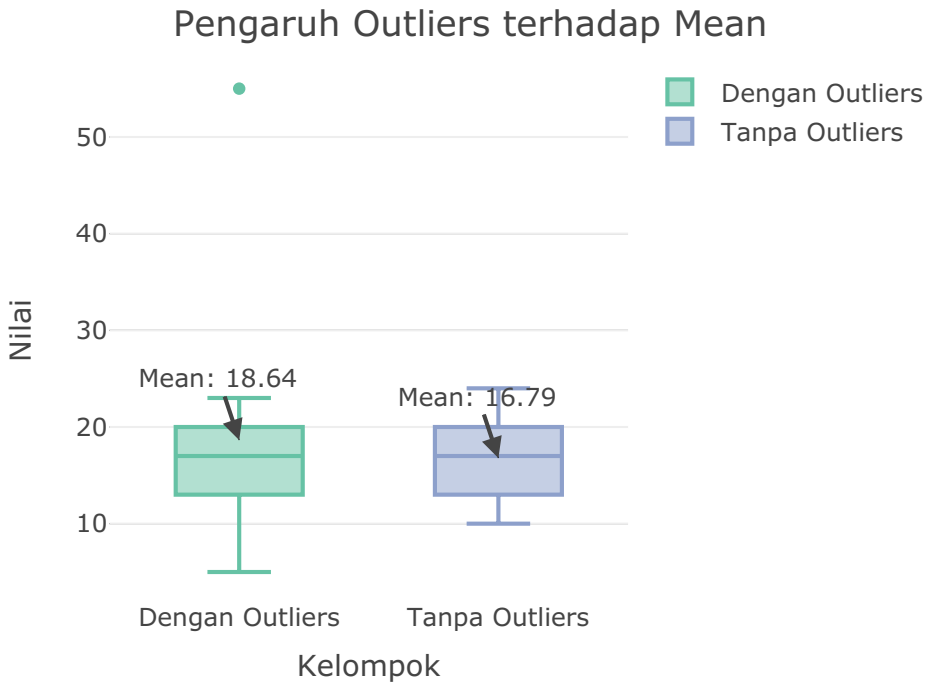
$$\text{Mean} = \frac{33}{5} = 6.6$$

Jadi, mean dari data tersebut adalah 6.6.

5.3.1 Mean dalam Boxplot

```
# Memuat library
library(plotly)

# Data: dua skenario, satu dengan outliers, satu tanpa outliers
data_dengan_outliers <- c(5, 11, 12, 13, 14, 15, 16, 18,18, 19, 20, 22, 23, 55)  # Deng
data_tanpa_outliers <- c(10, 11, 12, 13, 14, 15, 16, 18,18, 19, 20, 22, 23, 24)

# Menghitung rata-rata untuk kedua dataset
mean_dengan_outliers <- mean(data_dengan_outliers)
mean_tanpa_outliers <- mean(data_tanpa_outliers)

# Menggabungkan data ke dalam satu data frame untuk visualisasi
data <- data.frame(
  Nilai = c(data_dengan_outliers, data_tanpa_outliers),
  Kelompok = rep(c("Dengan Outliers", "Tanpa Outliers"),
                 times = c(length(data_dengan_outliers), length(data_tanpa_outliers)))
)

# Membuat boxplot menggunakan Plotly dengan outliers ditampilkan
plot <- plot_ly(
  data,
  y = ~Nilai,
  color = ~Kelompok,
  type = "box",
  boxpoints = "outliers"  # Menampilkan titik outliers
) %>%
  layout(
    title = "Pengaruh Outliers terhadap Mean",
    yaxis = list(title = "Nilai"),
    xaxis = list(title = "Kelompok"),
    annotations = list(
      list(
        x = "Dengan Outliers",
        y = mean_dengan_outliers,
        text = paste("Mean:", round(mean_dengan_outliers, 2)),
        showarrow = TRUE,
        arrowhead = 2
      ),
      list(
        x = "Tanpa Outliers",
        y = mean_tanpa_outliers,
        text = paste("Mean:", round(mean_tanpa_outliers, 2)),
        showarrow = TRUE,
```

```
        arrowhead = 2
      )
    )
  )

# Menampilkan plot
plot
```

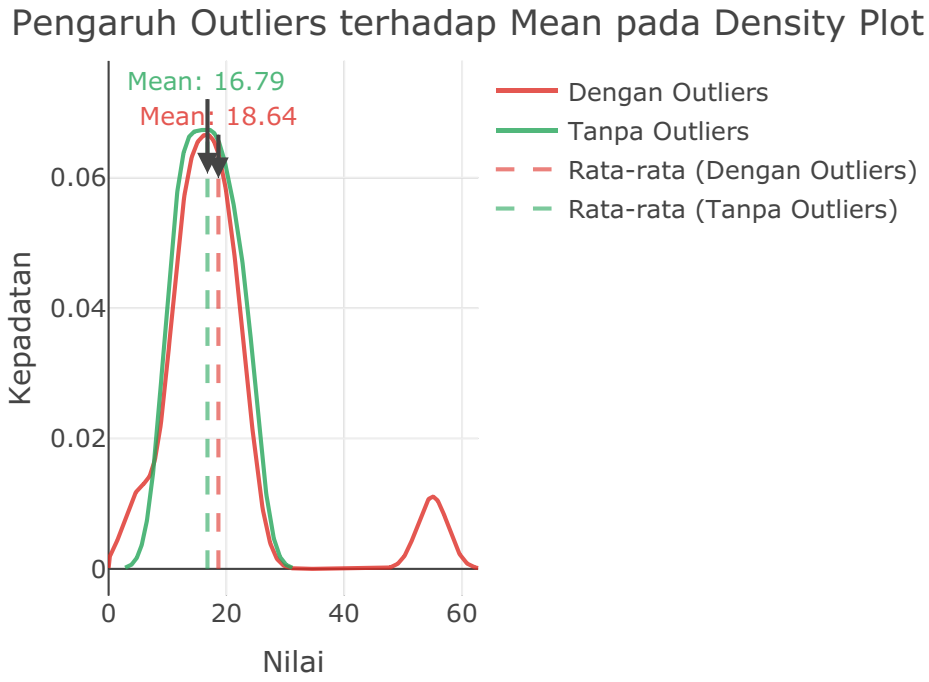


5.3.2 Mean dalam Histogram

```
# Memuat library
library(plotly)

# Data: dua skenario, satu dengan outliers, satu tanpa outliers
data_dengan_outliers <- c(5, 11, 12, 13, 14, 15, 16, 18,18, 19, 20, 22, 23, 55)  # Dengan outlier
data_tanpa_outliers <- c(10, 11, 12, 13, 14, 15, 16, 18,18, 19, 20, 22, 23, 24)          # Tanpa

# Membuat density plot untuk masing-masing dataset
density_dengan_outliers <- density(data_dengan_outliers)
density_tanpa_outliers <- density(data_tanpa_outliers)

# Pastikan tidak ada nilai negatif di x dan y
density_dengan_outliers$x <- pmax(0, density_dengan_outliers$x)
```

```r
density_tanpa_outliers$x <- pmax(0, density_tanpa_outliers$x)

# Menghitung rata-rata
mean_dengan_outliers <- mean(data_dengan_outliers)
mean_tanpa_outliers <- mean(data_tanpa_outliers)

# Membuat plot menggunakan Plotly
plot <- plot_ly() %>%
  # Menambahkan density plot untuk dataset dengan outliers
  add_trace(
    x = ~density_dengan_outliers$x,
    y = ~density_dengan_outliers$y,
    type = 'scatter',
    mode = 'lines',
    name = "Dengan Outliers",
    line = list(color = 'rgba(222, 45, 38, 0.8)', width = 2)
  ) %>%
  # Menambahkan density plot untuk dataset tanpa outliers
  add_trace(
    x = ~density_tanpa_outliers$x,
    y = ~density_tanpa_outliers$y,
    type = 'scatter',
    mode = 'lines',
    name = "Tanpa Outliers",
    line = list(color = 'rgba(38, 166, 91, 0.8)', width = 2)
  ) %>%
  # Menambahkan garis rata-rata untuk dataset dengan outliers
  add_trace(
    x = c(mean_dengan_outliers, mean_dengan_outliers),
    y = c(0, max(density_dengan_outliers$y)),
    type = "scatter",
    mode = "lines",
    name = "Rata-rata (Dengan Outliers)",
    line = list(color = 'rgba(222, 45, 38, 0.6)', dash = 'dash')
  ) %>%
  # Menambahkan garis rata-rata untuk dataset tanpa outliers
  add_trace(
    x = c(mean_tanpa_outliers, mean_tanpa_outliers),
    y = c(0, max(density_tanpa_outliers$y)),
    type = "scatter",
    mode = "lines",
    name = "Rata-rata (Tanpa Outliers)",
    line = list(color = 'rgba(38, 166, 91, 0.6)', dash = 'dash')
  ) %>%
  layout(
```

```
    title = "Pengaruh Outliers terhadap Mean pada Density Plot",
    xaxis = list(title = "Nilai"),
    yaxis = list(title = "Kepadatan"),
    annotations = list(
      # Anotasi untuk rata-rata dataset dengan outliers
      list(
        x = mean_dengan_outliers,
        y = max(density_dengan_outliers$y) * 0.9,
        text = paste("Mean:", round(mean_dengan_outliers, 2)),
        showarrow = TRUE,
        arrowhead = 2,
        ax = 0,
        ay = -30,  # Posisi teks sedikit lebih tinggi dari garis
        font = list(color = 'rgba(222, 45, 38, 0.8)', size = 12)
      ),
      # Anotasi untuk rata-rata dataset tanpa outliers
      list(
        x = mean_tanpa_outliers,
        y = max(density_tanpa_outliers$y) * 0.9,
        text = paste("Mean:", round(mean_tanpa_outliers, 2)),
        showarrow = TRUE,
        arrowhead = 2,
        ax = 0,
        ay = -45,  # Posisi teks sedikit lebih tinggi dari garis
        font = list(color = 'rgba(38, 166, 91, 0.8)', size = 12)
      )
    )
  )

# Menampilkan plot
plot
```



5.4 Median

Median adalah nilai tengah dalam suatu kumpulan data yang telah diurutkan. Jika data terdiri dari jumlah yang ganjil, median adalah nilai yang tepat berada di tengah. Jika jumlah data genap, median dihitung sebagai rata-rata dari dua nilai tengah yang berurutan.

Rumus dan Cara Menghitung Median:

- **Langkah pertama**: Urutkan data dari yang terkecil hingga yang terbesar.
- **Langkah kedua**: Tentukan posisi median:
  - Jika jumlah data **ganjil**, median adalah nilai di posisi tengah.
    - Posisi median =
$$\frac{n+1}{2}$$
  - Jika jumlah data **genap**, median adalah rata-rata dari dua nilai tengah.
    - Posisi median =
$$\frac{n}{2}$$
dan
$$\frac{n}{2}+1$$

* Median =

$$\frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}$$

Contoh Perhitungan Median untuk Data Ganjil dan Genap

1. Contoh Data Ganjil:

$$5, 10, 12, 13, 15$$

- Urutkan data:

$$5, 10, 12, 13, 15$$

- Jumlah data = 5 (ganjil)
- Posisi median =

$$\frac{5+1}{2} = 3$$

- Jadi, median adalah nilai ke-3, yaitu **12**.

2. Contoh Data Genap:

$$7, 10, 12, 13, 14, 15$$

- Urutkan data:

$$7, 10, 12, 13, 14, 15$$

- Jumlah data = 6 (genap)
- Posisi median:

$$\frac{6}{2} = 3$$

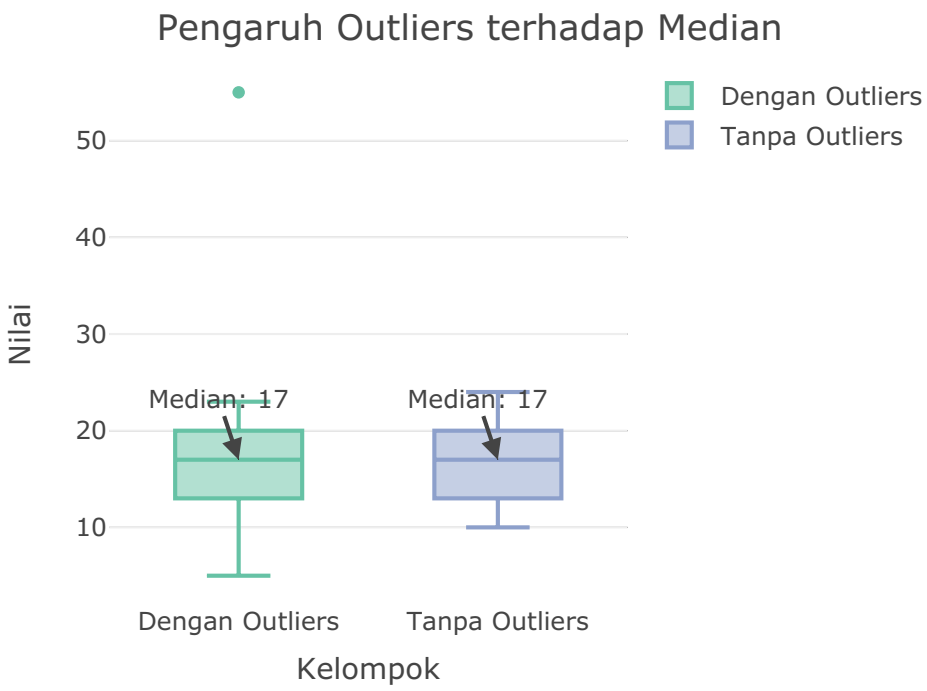
dan

$$\frac{6}{2} + 1 = 4$$

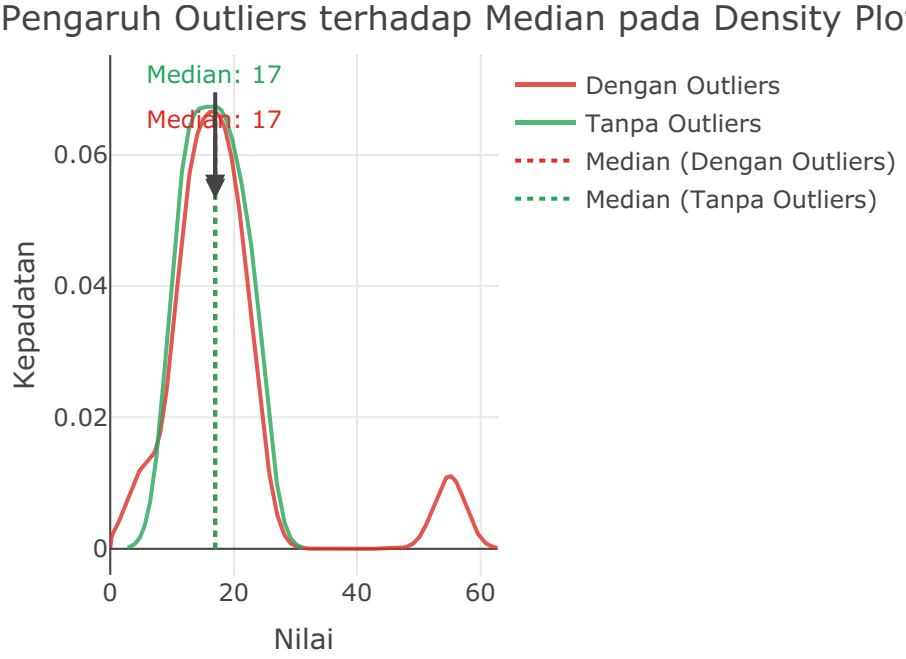
- Median =

$$\frac{12+13}{2} = 12.5$$

5.4.1 Median dalam Boxplot



5.4.2 Median dalam Histogram



5.5 Modus

Modus adalah nilai yang paling sering muncul dalam sebuah dataset. Modus digunakan untuk mengetahui nilai yang paling dominan atau paling sering terjadi dalam suatu kumpulan data. Modus bisa ditemukan dalam data kuantitatif maupun kategorikal.

Untuk mengidentifikasi modus dalam data, kita perlu mencari nilai yang memiliki frekuensi tertinggi. Untuk data numerik, modus bisa dihitung menggunakan frekuensi kemunculan masing-masing angka, sementara untuk data kategorikal, modus adalah kategori yang paling sering muncul.

Misalnya, kita memiliki data pengukuran tinggi badan berikut:

$$5, 10, 12, 13, 15, 15, 16, 17, 20, 25, 28, 30, 35$$

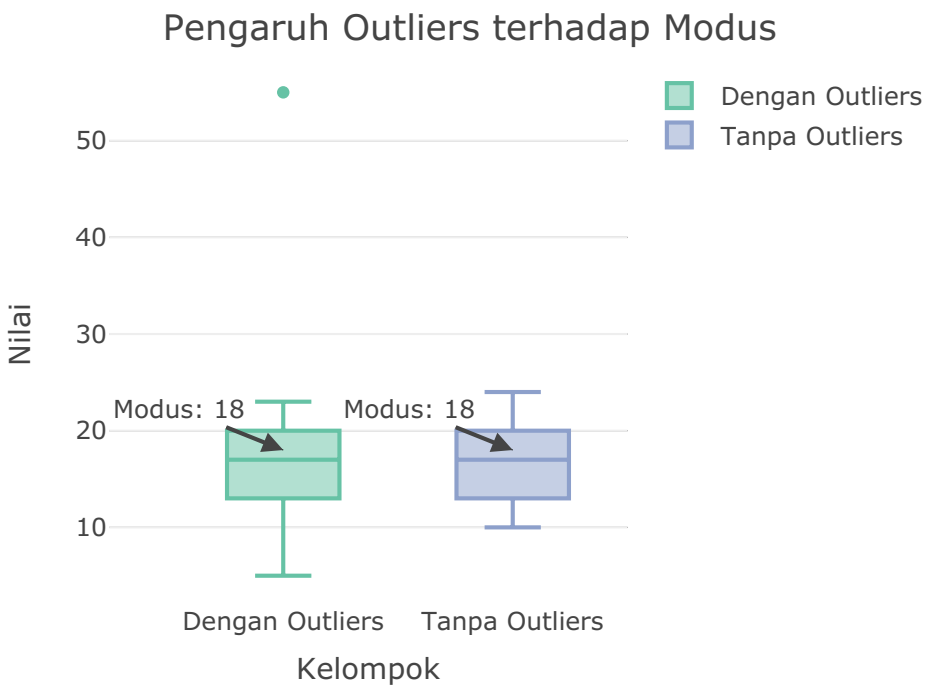
Untuk mencari modusnya, kita melihat nilai yang muncul paling sering. Dalam data di atas, angka 15 muncul tiga kali, sementara angka lainnya hanya muncul dua kali atau kurang. Oleh karena itu, modus dari data ini adalah **15**.

Jika kita memiliki data kategorikal seperti warna favorit:

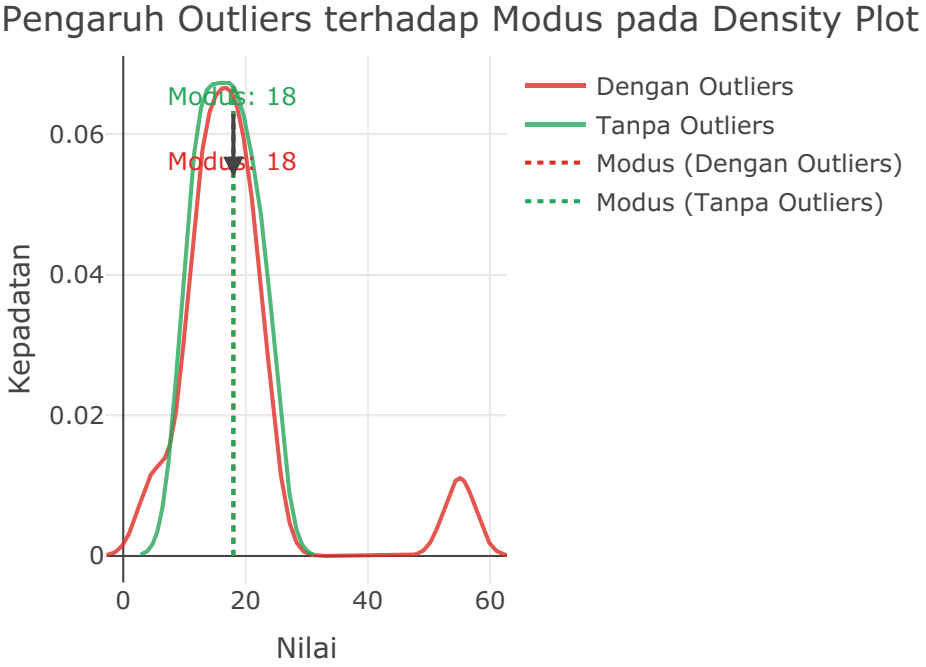
$$Merah, Biru, Merah, Hijau, Merah, Biru$$

Maka, modusnya adalah **Merah**, karena warna tersebut paling sering muncul.

5.5.1 Modus dalam Boxplot



5.5.2 Modus dalam Histogram



5.6 Perbandingan Mean, Median, dan Modus

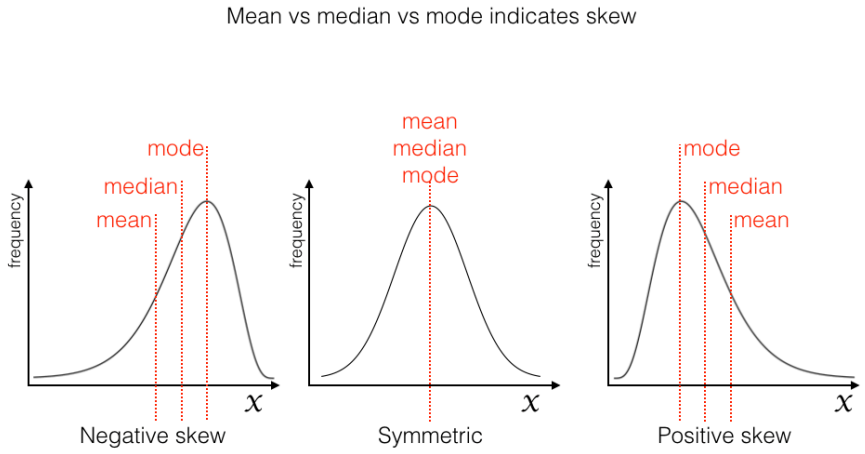
Tabel berikut merangkum kelebihan, kekurangan, dan aplikasi utama dari **mean**, **median**, dan **modus**:

Aspek	Mean	Median	Modus
Definisi	Rata-rata aritmatika dari semua nilai dalam dataset.	Nilai tengah dari data yang diurutkan.	Nilai yang paling sering muncul dalam dataset.
Kelebihan	- Menggunakan semua data, mencerminkan keseluruhan dataset.- Cocok untuk data interval dan rasio.	- Tidak terpengaruh oleh outlier.- Cocok untuk data ordinal.	- Relevan untuk data kategorikal.- Mudah dihitung.

Aspek	Mean	Median	Modus
Kekurangan	Rentan terhadap outlier (nilai ekstrem).- Tidak cocok untuk data distribusi tidak normal.	- Tidak mencakup keseluruhan informasi dataset.- Kurang stabil untuk data kecil.	- Tidak selalu ada modus (jika nilai sama frekuensinya).- Tidak menggunakan semua informasi dataset.
Penggunaan Utama	Data simetris tanpa outlier.- Analisis kuantitatif seperti ekonomi atau keuangan.	- Data dengan distribusi tidak normal atau outlier.	- Data kategorikal seperti frekuensi preferensi pelanggan.
Ketahanan Terhadap Outlier	Sangat rentan terhadap outlier.	Tidak dipengaruhi oleh outlier.	Tidak dipengaruhi oleh outlier.
Contoh Aplikasi	- Menghitung rata-rata nilai ujian.- Rata-rata gaji karyawan.	- Median pendapatan rumah tangga di wilayah tidak merata.	- Menentukan ukuran pakaian yang paling sering dibeli.

Catatan:

- **Mean** cocok digunakan untuk data yang terdistribusi normal.
- **Median** sering digunakan untuk menggambarkan data yang tidak simetris.
- **Modus** sangat berguna untuk data kategorikal atau nominal.



5.7 Praktikum 1

Buatkanlah penjelasan secara manual dan visualisasi ukuran Pemusatan untuk Data Kelompok, dengan sub-topik sebagai berikut:

5.7.1 Mean untuk Data Kelompok

5.7.2 Median untuk Data Kelompok

5.7.3 Modus untuk Data Kelompok

5.8 Praktikum 2

Carilah contoh sederhana yang menggukanakn Ukuran Pemusatan dalam Studi Kasus sebagai berikut:

5.8.1 Bisnis

5.8.2 Kesehatan

5.8.3 Pendidikan

Chapter 6

Ukuran Penyebaran Data

Ukuran penyebaran data adalah alat statistik yang menggambarkan tingkat variasi atau distribusi data dalam suatu dataset, melengkapi ukuran pemusatan seperti mean dan median. Beberapa ukuran penyebaran yang umum meliputi range, variansi, simpangan baku, dan interquartile range (IQR). Ukuran ini penting untuk memahami konsistensi, kestabilan, serta risiko dalam data, sehingga membantu pengambilan keputusan yang lebih baik. Sementara ukuran pemusatan menunjukkan nilai representatif dari data, ukuran penyebaran memberikan konteks tentang seberapa jauh data menyimpang dari nilai pusat, menjelaskan variasi yang ada di dalam dataset.

6.1 Jangkauan (Range)

6.1.1 Definisi Jangkauan

Jangkauan adalah selisih antara nilai maksimum dan nilai minimum dalam suatu dataset. Ini memberikan gambaran umum tentang sebaran data.

6.1.2 Menghitung Jangkauan

Jangkauan dihitung dengan rumus:

$$\text{Range} = \text{Max} - \text{Min}$$

Contoh: Dari dataset {3, 7, 2, 9, 5}, jangkauannya adalah 9 - 2 = 7.

6.2 Jangkauan Antar Kuartil (IQR)

6.2.1 Definisi IQR

Jangkauan Antar Kuartil (Interquartile Range, IQR) adalah selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1). IQR mengukur sebaran data di bagian tengah, mengabaikan nilai ekstrem.

6.2.2 Menghitung Kuartil

- **Kuartil pertama (Q1)** adalah nilai yang memisahkan 25% data terendah.
- **Kuartil ketiga (Q3)** adalah nilai yang memisahkan 25% data tertinggi.
- **Median (Q2)** adalah nilai tengah dataset.

IQR dihitung dengan rumus:

$$IQR = Q_3 - Q_1$$

6.2.3 Interpretasi IQR

IQR menunjukkan rentang data antara kuartil pertama dan ketiga, menggambarkan konsentrasi 50% data di tengah. Semakin besar IQR, semakin tersebar data di bagian tengah.

6.2.4 IQR dalam Mendeteksi Pencilan

Pencilan dapat diidentifikasi dengan menggunakan IQR. Nilai di luar rentang:

$$Q_1 - 1.5 * IQR \text{dan} Q_3 + 1.5 * IQR$$

disebut sebagai pencilan. Jika data berada di luar rentang ini, dianggap sebagai pencilan.

Misalkan kita memiliki dataset berikut:

$$2, 4, 6, 8, 10, 12, 14, 18, 20, 22$$

Langkah-langkah untuk menghitung IQR dan mendeteksi pencilan:

1. Urutkan data:

$$2, 4, 6, 8, 10, 12, 14, 18, 20, 22$$

2. Cari Median (Q2)

Median (Q2) adalah nilai tengah dari dataset, yang dalam hal ini adalah rata-rata dari 10 dan 12, yaitu 11.

3. Cari Kuartil pertama (Q1) dan Kuartil ketiga (Q3):

- Q1 adalah median dari set data pertama {2, 4, 6, 8, 10}, yaitu 6.
- Q3 adalah median dari set data kedua {12, 14, 18, 20, 22}, yaitu 18.

4. Hitung IQR:

$$Q_R = Q_3 - Q_1 = 18 - 6 = 12$$

5. Tentukan batas pencilan:

- Batas bawah: `Q1 - 1.5 * IQR = 6 - 1.5 * 12 = 6 - 18 = -12`
- Batas atas: `Q3 + 1.5 * IQR = 18 + 1.5 * 12 = 18 + 18 = 36`

6. Identifikasi pencilan:

Data yang berada di luar rentang [-12, 36] dianggap sebagai pencilan. Dalam hal ini, semua data berada dalam rentang tersebut, sehingga **tidak ada pencilan**.

Jika data yang diberikan adalah `{2, 4, 6, 8, 10, 100, 12, 14, 18, 20, 22}`, maka angka 100 berada di luar batas atas 36 dan akan dianggap **sebagai pencilan**.

6.3 Varians

6.3.1 Definisi Varians

Varians adalah ukuran statistik yang menggambarkan sebaran atau penyebaran data dalam suatu set data. Varians menunjukkan seberapa jauh nilai-nilai data tersebar dari rata-rata (mean). Semakin besar varians, semakin tersebar data di sekitar rata-rata.

6.3.2 Varians Populasi & Sampel

Varians Populasi digunakan ketika kita menghitung varians dari seluruh populasi. Rumusnya adalah:

$$\sigma^2 = \frac{\sum(x_i - \mu)^2}{N}$$

Di mana:

- σ^2 = varians populasi
- x_i = setiap nilai dalam data
- μ = rata-rata populasi
- N = jumlah data dalam populasi

Varians Sampel digunakan ketika kita hanya memiliki sampel data dan ingin mengestimasi varians populasi. Rumusnya adalah:

$$s^2 = \frac{\Sigma(x_i - \bar{x})^2}{(n-1)}$$

Di mana:

- s^2 = varians sampel
- x_i = setiap nilai dalam sampel
- $\bar{x}$ = rata-rata sampel
- n = jumlah data dalam sampel

6.3.3 Contoh Perhitungan Varians

Misalkan kita memiliki data berikut:

$2, 4, 6, 8, 10$

1. **Cari Rata-rata ($\bar{x}$):**

$\bar{x} = (2 + 4 + 6 + 8 + 10)/5 = 30/5 = 6$

2. **Hitung selisih antara setiap nilai dengan rata-rata dan kuadratkan hasilnya:**

$(2-6)^2 = (-4)^2 = 16 (4-6)^2 = (-2)^2 = 4 (6-6)^2 = 0^2 = 0 (8-6)^2 = 2^2 = 4 (10-6)^2 = 4^2 = 16$

3. **Jumlahkan hasil kuadrat perbedaan:**

$16 + 4 + 0 + 4 + 16 = 40$

4. **Hitung Varians Sampel:**

$s^2 = 40/(5-1) = 40/4 = 10$

Jadi, **varians sampel** dari data tersebut adalah 10.

6.4 Standar Deviasi

6.4.1 Definisi Standar Deviasi

Standar deviasi adalah ukuran sebaran atau penyebaran data dalam suatu distribusi. Semakin kecil standar deviasi, semakin dekat nilai-nilai data ke rata-rata. Sebaliknya, semakin besar standar deviasi, semakin tersebar data tersebut dari rata-rata.

6.4.2 Varians & Standar Deviasi

Varians adalah kuadrat dari standar deviasi. Varians memberikan gambaran tentang seberapa besar data tersebar dari rata-rata, tetapi satuannya dalam kuadrat. Sedangkan, standar deviasi memberikan ukuran yang lebih mudah dipahami karena satuannya sama dengan satuan data aslinya.

6.4.3 Standar Deviasi Populasi & Sampel

Terdapat dua jenis standar deviasi, yaitu:

1. **Standar Deviasi Populasi**: Digunakan ketika data yang dimiliki mencakup seluruh populasi. Rumusnya adalah:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

   di mana:

   - N adalah jumlah data,
   - x_i adalah setiap nilai data,
   - μ adalah rata-rata populasi.

2. **Standar Deviasi Sampel**: Digunakan ketika data hanya merupakan sampel dari populasi. Rumusnya adalah:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

   di mana:

   - n adalah jumlah sampel,
   - x_i adalah setiap nilai data,
   - $\bar{x}$ adalah rata-rata sampel.

6.4.4 Interpretasi Standar Deviasi

Standar deviasi menggambarkan sejauh mana nilai-nilai data menyimpang dari rata-rata. Sebagai contoh:

- Standar deviasi yang kecil menunjukkan bahwa nilai data hampir semua berada di dekat rata-rata.
- Standar deviasi yang besar menunjukkan adanya penyebaran yang lebih luas antara nilai-nilai data.

6.4.5 Contoh Perhitungan Standar Deviasi

Misalkan kita memiliki data berikut:

$$10, 12, 23, 23, 16, 23, 21, 16$$

1. **Langkah 1: Hitung rata-rata (mean)**

$$\bar{x} = \frac{10 + 12 + 23 + 23 + 16 + 23 + 21 + 16}{8} = 17.5$$

2. **Langkah 2: Hitung varians**
   Varians adalah rata-rata kuadrat selisih antara setiap nilai dengan rata-rata:

$$\text{Varians} = \frac{(10-17.5)^2 + (12-17.5)^2 + \cdots + (16-17.5)^2}{8-1}$$

   Varians = 28.57

3. **Langkah 3: Hitung standar deviasi**
   Standar deviasi adalah akar kuadrat dari varians:

$$\sigma = \sqrt{28.57} = 5.34$$

6.5 Koefisien Variasi

6.5.1 Definisi Koefisien Variasi

Koefisien variasi (CV) adalah ukuran yang menggambarkan seberapa besar variasi atau penyebaran data relatif terhadap rata-rata. Koefisien variasi dihitung dengan membandingkan standar deviasi dengan rata-rata, yang memungkinkan perbandingan antara variabilitas dua atau lebih dataset meskipun satuannya berbeda.

6.5.2 Cara Menghitung Koefisien Variasi

Koefisien variasi dihitung menggunakan rumus berikut:

$$\text{CV} = \frac{\sigma}{\mu} \times 100\%$$

di mana:

- σ adalah standar deviasi,
- μ adalah rata-rata.

Dengan rumus ini, koefisien variasi memberikan hasil dalam bentuk persen, yang menunjukkan proporsi standar deviasi terhadap rata-rata.

6.5.3 Interpretasi Koefisien Variasi

Koefisien variasi memberikan informasi tentang variabilitas relatif data. Beberapa interpretasi umum dari CV adalah:

- **CV rendah**: Menunjukkan bahwa data lebih terpusat atau memiliki variasi kecil relatif terhadap rata-rata.
- **CV tinggi**: Menunjukkan bahwa data memiliki variasi yang lebih besar relatif terhadap rata-rata, yang mengindikasikan ketidakstabilan atau ketidakteraturan yang lebih tinggi.

Koefisien variasi sering digunakan untuk membandingkan variabilitas antara dataset dengan satuan yang berbeda, karena CV mengabaikan satuan unit dalam perhitungannya.

6.6 Rentang Semi-Interkuartil

6.6.1 Definisi Rentang Semi-Interkuartil

Rentang Semi-Interkuartil (Semi-Interquartile Range, SIQR) adalah ukuran statistik yang menggambarkan sebaran data dengan fokus pada kuartil pertama (Q1) dan kuartil ketiga (Q3). Rentang ini mengukur sebaran nilai tengah data dan mengabaikan nilai ekstrem (outlier). Rentang Semi-Interkuartil digunakan untuk memberikan gambaran yang lebih stabil tentang variabilitas data dibandingkan dengan rentang biasa.

6.6.2 Menghitung Rentang Semi-Interkuartil

Rentang Semi-Interkuartil dihitung dengan rumus:

$$\text{SIQR} = \frac{Q3 - Q1}{2}$$

di mana:

- $Q1$ adalah kuartil pertama (25% data di bawahnya),
- $Q3$ adalah kuartil ketiga (75% data di bawahnya).

Rentang Semi-Interkuartil mengukur jarak antara kuartil pertama dan kuartil ketiga, kemudian membaginya dengan dua untuk mendapatkan rentang pada bagian tengah data.

6.7 Analisis Penyebaran Data

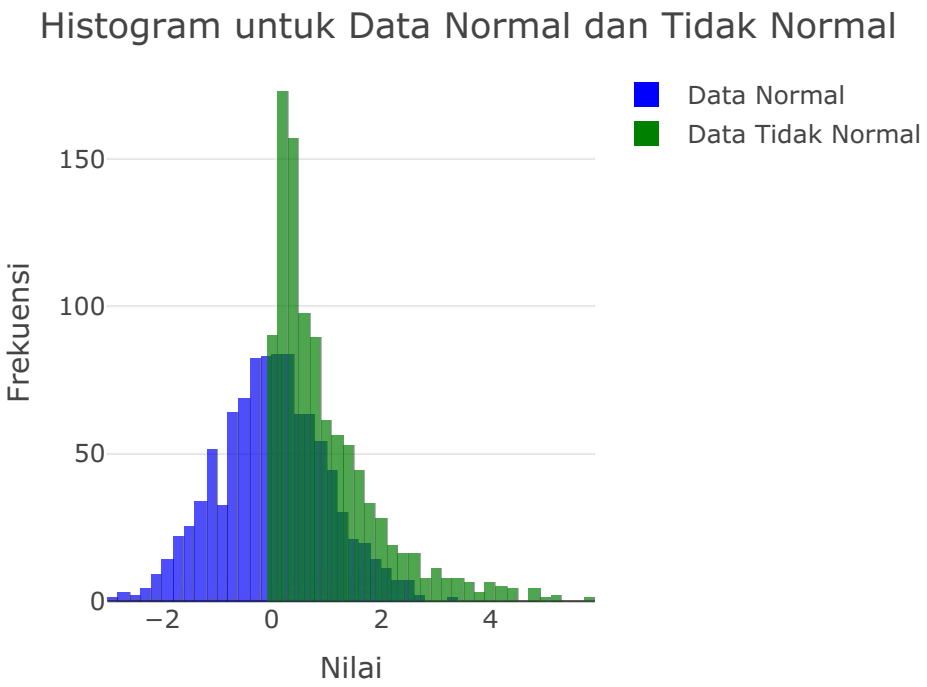
6.7.1 Histogram

Histogram adalah alat grafis yang digunakan untuk menunjukkan distribusi data numerik dengan membagi data ke dalam interval atau "bin". Setiap batang dalam histogram mewakili jumlah data dalam interval tersebut. Histogram memberikan gambaran visual tentang bentuk distribusi data, apakah simetris, skewed, atau memiliki banyak puncak (multimodal).

Histogram sangat berguna untuk:

- Menilai distribusi data,
- Mengidentifikasi outlier atau pencilan,
- Membandingkan distribusi beberapa kelompok data.

```r
library(plotly)

# Membuat data normal dan tidak normal
set.seed(123)
x_normal <- rnorm(1000)  # Data normal
x_non_normal <- rexp(1000)  # Data tidak normal (eksponensial)

# Membuat histogram untuk data normal dan tidak normal
plot_ly() %>%
  add_trace(
    x = x_normal,
    type = "histogram",
    name = "Data Normal",
    marker = list(color = 'blue', opacity = 0.7)
  ) %>%  # Histogram untuk data normal
  add_trace(
    x = x_non_normal,
    type = "histogram",
    name = "Data Tidak Normal",
    marker = list(color = 'green', opacity = 0.7)
  ) %>%  # Histogram untuk data tidak normal
  layout(
    title = "Histogram untuk Data Normal dan Tidak Normal",
    xaxis = list(title = "Nilai"),
    yaxis = list(title = "Frekuensi"),
    barmode = "overlay"  # Menampilkan histogram secara tumpang tindih
  )
```



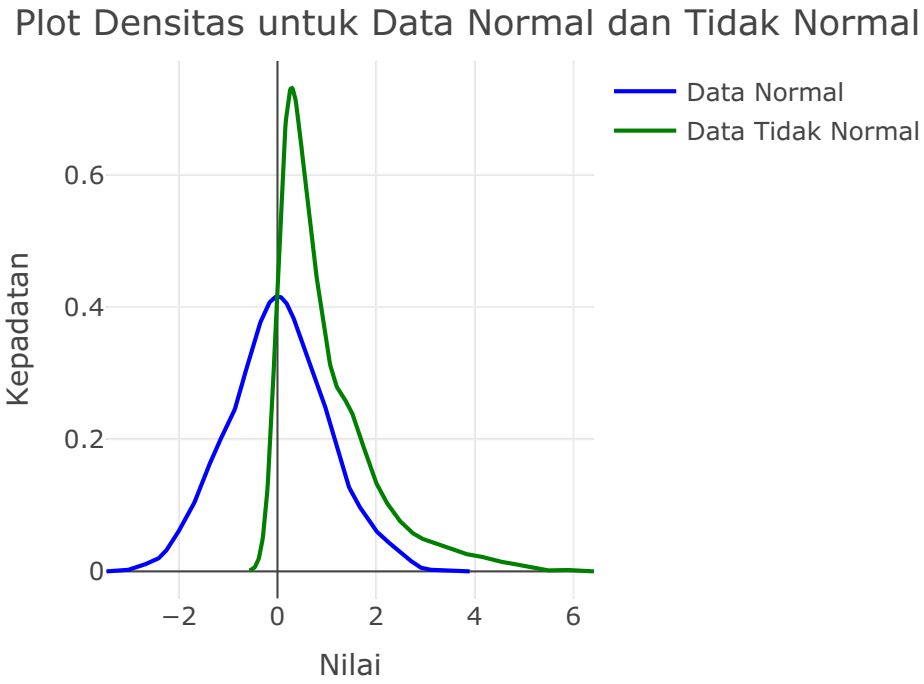
6.7.2 Densitas

Plot densitas adalah alat grafis yang digunakan untuk menggambarkan distribusi probabilitas dari sebuah dataset. Dengan menggunakan estimasi kepadatan kernel (Kernel Density Estimation atau KDE), plot densitas memperlihatkan bagaimana data tersebar sepanjang rentang nilai. Berbeda dengan histogram, plot densitas tidak terbatas pada interval atau "bin" tertentu, sehingga menghasilkan kurva yang lebih halus dan memberikan gambaran yang lebih jelas tentang distribusi data.

Plot densitas sangat berguna untuk:

- Mengidentifikasi bentuk distribusi: Apakah data simetris, miring (skewed), atau multimodal (memiliki lebih dari satu puncak).
- Menilai sebaran data: Menyediakan gambaran tentang bagaimana data tersebar di sepanjang sumbu nilai, apakah konsentrasi data lebih banyak di satu area atau tersebar merata.
- Membandingkan distribusi berbagai kelompok data: Memungkinkan perbandingan visual distribusi dari beberapa dataset, misalnya data normal vs. tidak normal, atau data dari kelompok yang berbeda.
- Mendeteksi outlier: Dengan melihat area yang jauh dari puncak distribusi, kita dapat mengidentifikasi potensi outlier atau nilai yang jarang terjadi.
- Keuntungan menggunakan plot densitas dibandingkan histogram adalah plot densitas memberikan representasi yang lebih halus dan lebih mudah untuk dibandingkan antar distribusi dataset.

```
library(plotly)

# Membuat data normal dan tidak normal
set.seed(123)
x_normal <- rnorm(1000)  # Data normal
x_non_normal <- rexp(1000)  # Data tidak normal (eksponensial)

# Menghitung densitas data
density_normal <- density(x_normal)
density_non_normal <- density(x_non_normal)

# Membuat plot densitas untuk data normal dan tidak normal
plot_ly() %>%
  add_trace(
    x = density_normal$x,
    y = density_normal$y,
    type = "scatter",
    mode = "lines",
    name = "Data Normal",
    line = list(color = 'blue', width = 2)
  ) %>%  # Plot densitas untuk data normal
  add_trace(
    x = density_non_normal$x,
    y = density_non_normal$y,
    type = "scatter",
    mode = "lines",
    name = "Data Tidak Normal",
    line = list(color = 'green', width = 2)
  ) %>%  # Plot densitas untuk data tidak normal
  layout(
    title = "Plot Densitas untuk Data Normal dan Tidak Normal",
    xaxis = list(title = "Nilai"),
    yaxis = list(title = "Kepadatan")
  )
```



6.7.3 Box Plot dan Penyebaran Data

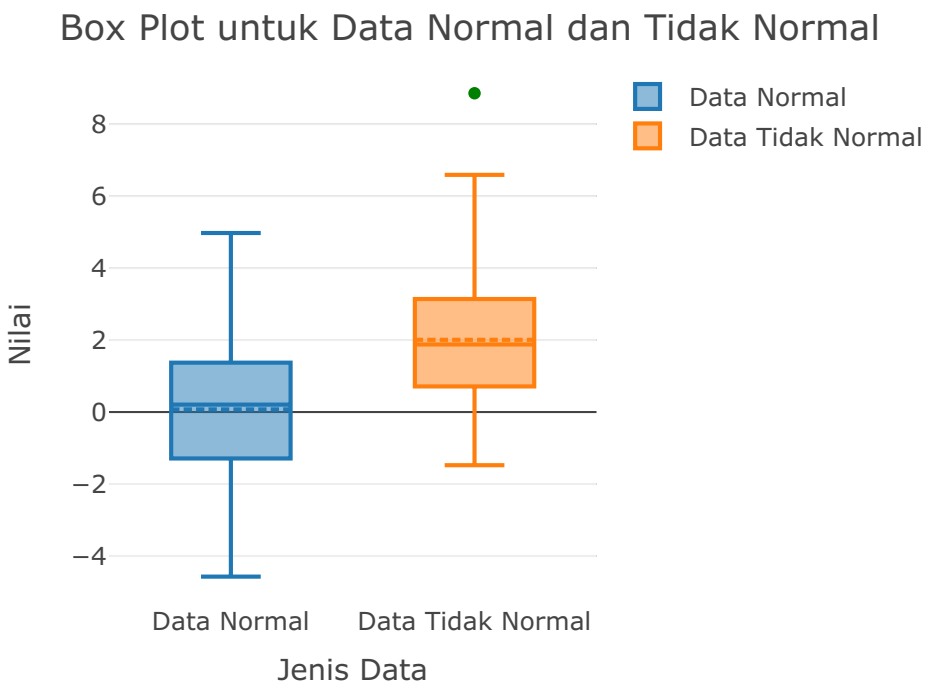
Box plot (atau diagram kotak) adalah alat yang digunakan untuk menggambarkan penyebaran data berdasarkan lima nilai utama: minimum, kuartil pertama (Q1), median (Q2), kuartil ketiga (Q3), dan maksimum. Box plot memberikan gambaran visual tentang distribusi data, serta adanya outlier.

Komponen utama box plot:

- **Kotak**: Mewakili interkuartil range (IQR) antara Q1 dan Q3.
- **Garis dalam kotak**: Menunjukkan median data.
- **Garis vertikal di luar kotak**: Menunjukkan rentang data (minimum dan maksimum).
- **Titik di luar garis vertikal**: Menunjukkan outlier (nilai yang jauh dari distribusi utama data).

```
library(plotly)

# Membuat data normal dan tidak normal
set.seed(123)
x_normal <- rnorm(100)  # Data normal
y_normal <- 2 * x_normal + rnorm(100)

x_non_normal <- rexp(100)  # Data tidak normal (eksponensial)
y_non_normal <- 2 * x_non_normal + rnorm(100)
```

```
# Membuat boxplot untuk data normal dan tidak normal
plot_ly() %>%
  add_trace(
    y = y_normal,
    type = "box",
    name = "Data Normal",
    boxmean = TRUE,
    marker = list(color = 'blue')
  ) %>%  # Boxplot untuk data normal
  add_trace(
    y = y_non_normal,
    type = "box",
    name = "Data Tidak Normal",
    boxmean = TRUE,
    marker = list(color = 'green')
  ) %>%  # Boxplot untuk data tidak normal
  layout(
    title = "Box Plot untuk Data Normal dan Tidak Normal",
    yaxis = list(title = "Nilai"),
    xaxis = list(title = "Jenis Data")
  )
```



Box plot sangat berguna untuk:

- Mengidentifikasi simetri distribusi data,
- Menilai penyebaran data,
- Menemukan outlier atau pencilan.

6.7.4 Scatter Plot & Variabilitas

Scatter plot adalah jenis grafik yang menunjukkan hubungan antara dua variabel numerik. Setiap titik dalam scatter plot mewakili pasangan nilai dari dua variabel tersebut. Scatter plot sangat berguna untuk mengidentifikasi pola hubungan, apakah ada korelasi antara variabel atau tidak.

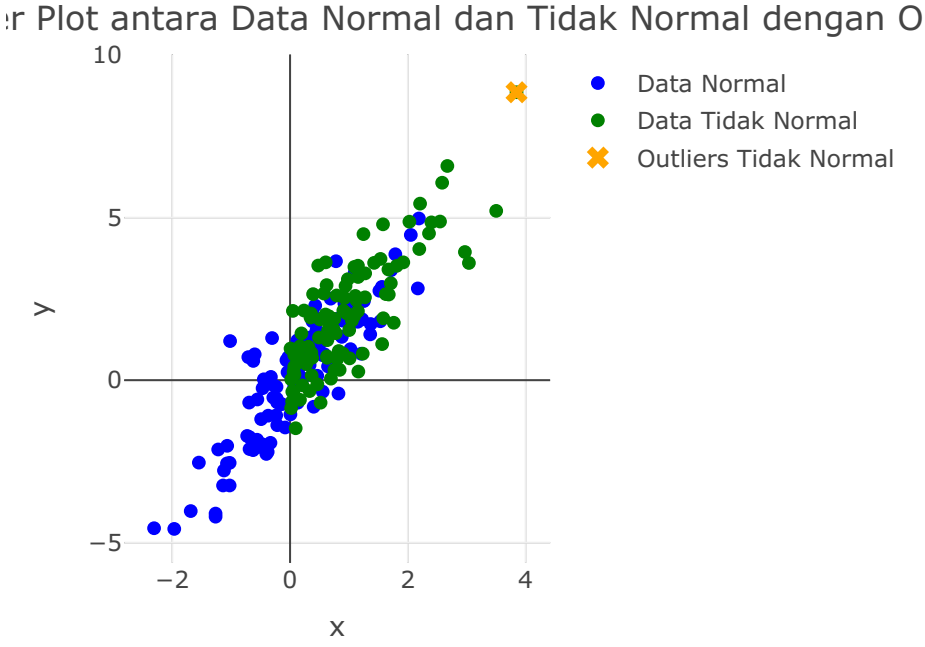
Scatter plot membantu dalam:

- Mengidentifikasi jenis hubungan antara dua variabel (positif, negatif, atau tidak ada hubungan),
- Menilai variabilitas atau penyebaran data pada kedua variabel,
- Mendeteksi outlier yang dapat mempengaruhi hubungan antara variabel.

Dengan scatter plot, kita dapat melihat apakah data mengikuti pola linier, eksponensial, atau pola yang lebih kompleks, serta mengevaluasi tingkat variabilitas pada setiap variabel.

```r
library(plotly)

# Membuat data normal dan tidak normal
set.seed(123)
x_normal <- rnorm(100)  # Data normal
y_normal <- 2 * x_normal + rnorm(100)

x_non_normal <- rexp(100)  # Data tidak normal (eksponensial)
y_non_normal <- 2 * x_non_normal + rnorm(100)

# Menghitung kuartil dan IQR untuk mendeteksi outlier (untuk kedua data)
Q1_normal <- quantile(y_normal, 0.25)
Q3_normal <- quantile(y_normal, 0.75)
IQR_normal <- Q3_normal - Q1_normal

Q1_non_normal <- quantile(y_non_normal, 0.25)
Q3_non_normal <- quantile(y_non_normal, 0.75)
IQR_non_normal <- Q3_non_normal - Q1_non_normal

# Menentukan batas atas dan bawah untuk outlier (untuk kedua data)
lower_bound_normal <- Q1_normal - 1.5 * IQR_normal
upper_bound_normal <- Q3_normal + 1.5 * IQR_normal

lower_bound_non_normal <- Q1_non_normal - 1.5 * IQR_non_normal
upper_bound_non_normal <- Q3_non_normal + 1.5 * IQR_non_normal
```

```r
# Menandai outlier
outliers_normal <- which(y_normal < lower_bound_normal | y_normal > upper_bound_normal)
outliers_non_normal <- which(y_non_normal < lower_bound_non_normal | y_non_normal > upp

# Membuat scatter plot untuk data normal dan tidak normal, dengan menandai outlier
plot_ly() %>%
  add_trace(x = x_normal, y = y_normal, type = "scatter", mode = "markers", name = "Dat
            marker = list(color = 'blue', size = 7)) %>%  # Data normal
  add_trace(x = x_normal[outliers_normal], y = y_normal[outliers_normal], type = "scatt
            name = "Outliers Normal",
            marker = list(color = 'red', size = 10, symbol = 'x')) %>%  # Outlier pada
  add_trace(x = x_non_normal, y = y_non_normal, type = "scatter", mode = "markers", nam
            marker = list(color = 'green', size = 7)) %>%  # Data tidak normal
  add_trace(x = x_non_normal[outliers_non_normal], y = y_non_normal[outliers_non_normal
            name = "Outliers Tidak Normal",
            marker = list(color = 'orange', size = 10, symbol = 'x')) %>%  # Outlier pa
  layout(title = "Scatter Plot antara Data Normal dan Tidak Normal dengan Outliers",
         xaxis = list(title = "x"),
         yaxis = list(title = "y"))
```



6.8 Studi Kasus 1

Dalam kasus ini, kita akan menganalisis waktu yang diperlukan oleh dua jenis mesin untuk menyelesaikan satu siklus produksi. Mesin A memiliki distribusi

waktu proses yang normal, sedangkan Mesin B memiliki distribusi waktu proses yang tidak normal (eksponensial). Tujuan analisis ini adalah untuk membandingkan distribusi waktu proses produksi antara kedua mesin menggunakan **Histogram**, **Plot Densitas**, dan **Box Plot**.

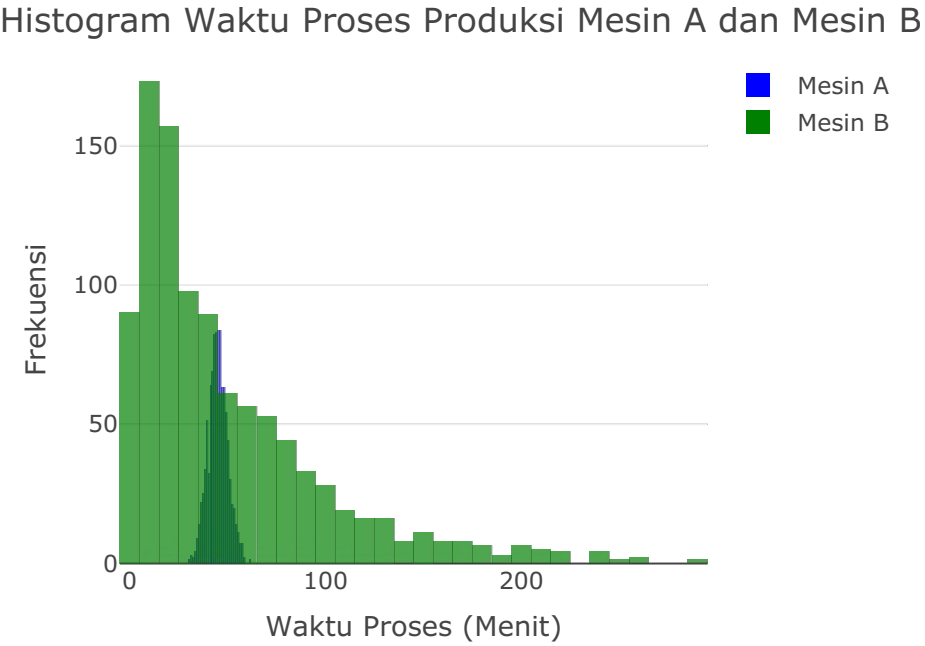
6.8.1 Membuat Data

```
library(plotly)

# Membuat data untuk Mesin A (distribusi normal) dan Mesin B (distribusi eksponensial)
set.seed(123)
mesin_A <- rnorm(1000, mean = 45, sd = 5)  # Mesin A (normal distribution)
mesin_B <- rexp(1000, rate = 1/50)  # Mesin B (exponential distribution)
```

6.8.2 Histogram Waktu Proses Produksi

Histogram digunakan untuk memvisualisasikan distribusi data dengan membaginya dalam interval tertentu. Berikut adalah histogram waktu proses produksi untuk Mesin A dan Mesin B.

```
# Membuat Histogram untuk Mesin A dan Mesin B
plot_ly() %>%
  add_trace(
    x = mesin_A,
    type = "histogram",
    name = "Mesin A",
    marker = list(color = 'blue', opacity = 0.7)
  ) %>%  # Histogram untuk Mesin A
  add_trace(
    x = mesin_B,
    type = "histogram",
    name = "Mesin B",
    marker = list(color = 'green', opacity = 0.7)
  ) %>%  # Histogram untuk Mesin B
  layout(
    title = "Histogram Waktu Proses Produksi Mesin A dan Mesin B",
    xaxis = list(title = "Waktu Proses (Menit)"),
    yaxis = list(title = "Frekuensi"),
    barmode = "overlay"  # Menampilkan histogram secara tumpang tindih
  )
```

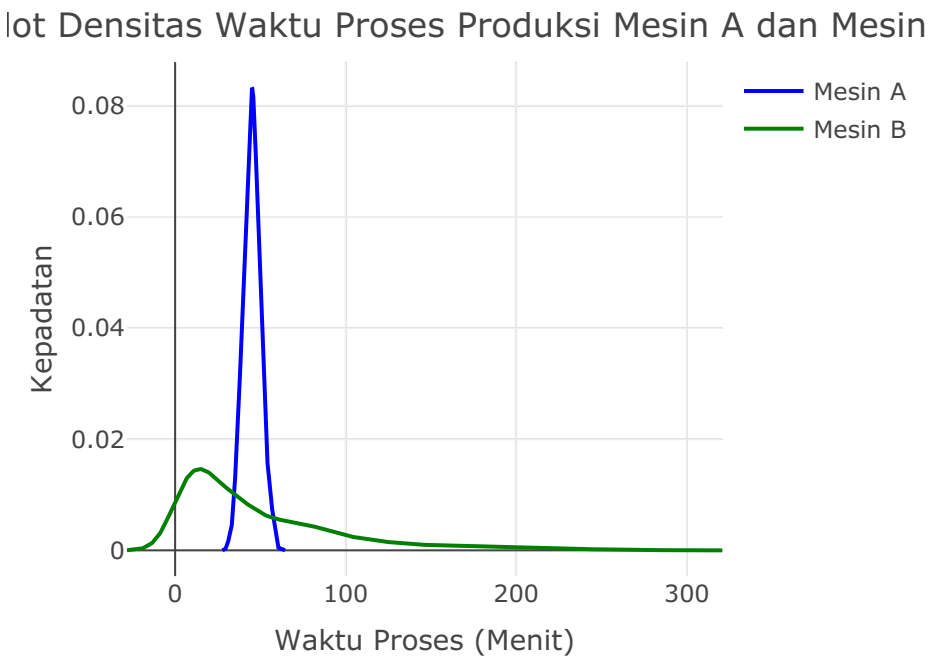


6.8.3 Plot Densitas Waktu Proses Produksi

Plot densitas memberikan gambaran lebih halus tentang distribusi data dengan estimasi kepadatan kernel. Berikut adalah plot densitas untuk Mesin A dan Mesin B.

```
# Menghitung densitas untuk Mesin A dan Mesin B
densitas_A <- density(mesin_A)
densitas_B <- density(mesin_B)

# Membuat Plot Densitas untuk Mesin A dan Mesin B
plot_ly() %>%
  add_trace(
    x = densitas_A$x,
    y = densitas_A$y,
    type = "scatter",
    mode = "lines",
    name = "Mesin A",
    line = list(color = 'blue', width = 2)
  ) %>%  # Plot densitas untuk Mesin A
  add_trace(
    x = densitas_B$x,
    y = densitas_B$y,
    type = "scatter",
    mode = "lines",
```

```
    name = "Mesin B",
    line = list(color = 'green', width = 2)
  ) %>%  # Plot densitas untuk Mesin B
  layout(
    title = "Plot Densitas Waktu Proses Produksi Mesin A dan Mesin B",
    xaxis = list(title = "Waktu Proses (Menit)"),
    yaxis = list(title = "Kepadatan")
  )
```

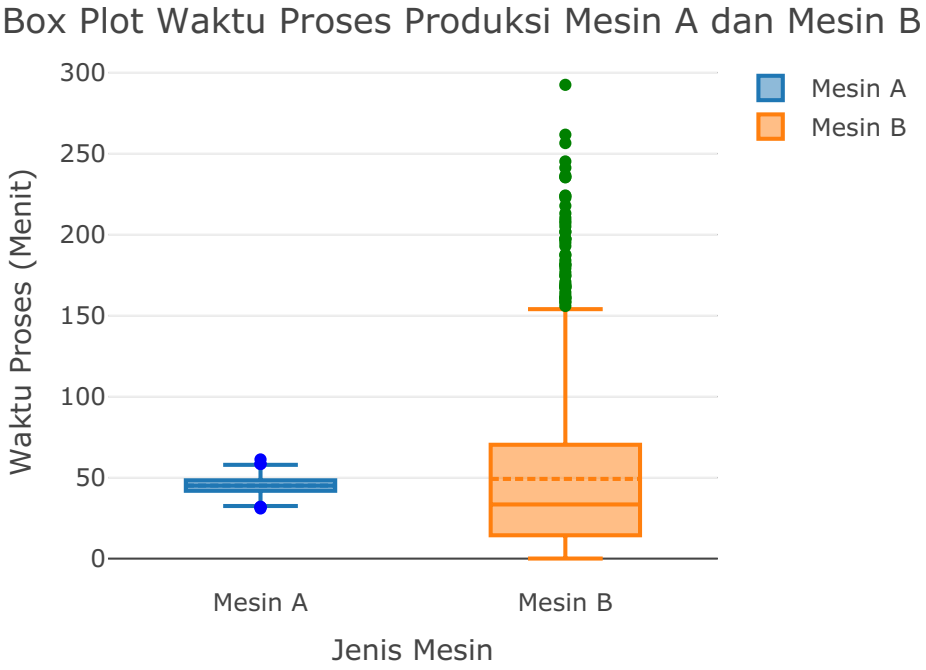


6.8.4 Box Plot Waktu Proses Produksi

Box plot digunakan untuk menggambarkan distribusi data dalam hal kuartil dan nilai pencilan (outlier). Berikut adalah box plot untuk waktu proses produksi Mesin A dan Mesin B.

```
# Membuat Box Plot untuk Mesin A dan Mesin B
plot_ly() %>%
  add_trace(
    y = mesin_A,
    type = "box",
    name = "Mesin A",
    boxmean = TRUE,
    marker = list(color = 'blue')
  ) %>%  # Boxplot untuk Mesin A
  add_trace(
```

```
    y = mesin_B,
    type = "box",
    name = "Mesin B",
    boxmean = TRUE,
    marker = list(color = 'green')
  ) %>%  # Boxplot untuk Mesin B
  layout(
    title = "Box Plot Waktu Proses Produksi Mesin A dan Mesin B",
    yaxis = list(title = "Waktu Proses (Menit)"),
    xaxis = list(title = "Jenis Mesin")
  )
```



6.9 Studi Kasus 2

Sebuah perusahaan logistik mengelola pengiriman barang ke empat wilayah (Utara, Selatan, Timur, dan Barat). Perusahaan ingin menganalisis efisiensi berdasarkan waktu pengiriman, jumlah barang yang dikirim, dan biaya pengiriman. Data selama seminggu terakhir adalah sebagai berikut:

Wilayah	Waktu Pengiriman (jam)	Jumlah Barang (unit)	Biaya Pengiriman (Rp/unit)
Utara	12, 14, 13, 16, 18	120, 110, 115, 130, 140	50,000, 55,000, 53,000, 52,000, 51,000
Selatan	22, 20, 24, 25, 23	150, 140, 145, 155, 160	40,000, 42,000, 41,000, 43,000, 44,000
Timur	10, 11, 12, 11, 10	80, 85, 83, 90, 88	60,000, 62,000, 61,000, 63,000, 64,000
Barat	18, 19, 20, 22, 21	100, 95, 105, 110, 108	45,000, 47,000, 46,000, 48,000, 50,000

6.9.1 Data Pengiriman Barang

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages -
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.4     v tidyr     1.3.1
## v readr     2.1.5
## -- Conflicts -------------------------
## x plotly::filter() masks dplyr::filter(), stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
# Data pengiriman barang
data <- data.frame(
  Wilayah = rep(c("Utara", "Selatan", "Timur", "Barat"), each = 5),
  Waktu_Pengiriman = c(12, 14, 13, 16, 18, 22, 20, 24, 25, 23, 10, 11, 12, 11, 10, 18, 19, 20, 22
  Jumlah_Barang = c(120, 110, 115, 130, 140, 150, 140, 145, 155, 160, 80, 85, 83, 90, 88, 100, 95
  Biaya_Per_Unit = c(50000, 55000, 53000, 52000, 51000, 40000, 42000, 41000, 43000, 44000, 60000,
)
```

6.9.2 Analisis Statistik

Rata-rata, median, dan simpangan baku untuk setiap wilayah:

```
stats <- data %>%
  group_by(Wilayah) %>%
  summarise(
    Rata_Rata_Waktu = mean(Waktu_Pengiriman),
```

```
    Median_Waktu = median(Waktu_Pengiriman),
    SD_Waktu = sd(Waktu_Pengiriman),
    Rata_Rata_Barang = mean(Jumlah_Barang),
    Median_Barang = median(Jumlah_Barang),
    SD_Barang = sd(Jumlah_Barang),
    Rata_Rata_Biaya = mean(Biaya_Per_Unit),
    Median_Biaya = median(Biaya_Per_Unit),
    SD_Biaya = sd(Biaya_Per_Unit)
  )
stats
```

```
## # A tibble: 4 x 10
##   Wilayah Rata_Rata_Waktu Median_Waktu SD_Waktu Rata_Rata_Barang Median_Barang
##   <chr>             <dbl>        <dbl>    <dbl>            <dbl>         <dbl>
## 1 Barat              20             20    1.58             104.            105
## 2 Selatan            22.8           23    1.92             150             150
## 3 Timur              10.8           11    0.837             85.2            85
## 4 Utara              14.6           14    2.41             123             120
## # i 4 more variables: SD_Barang <dbl>, Rata_Rata_Biaya <dbl>,
## #   Median_Biaya <dbl>, SD_Biaya <dbl>
```

6.9.3 Efisiensi Pengiriman

Hitung total biaya pengiriman dan efisiensi biaya:

```
data <- data %>%
  mutate(Biaya_Total = Jumlah_Barang * Biaya_Per_Unit)

efisiensi <- data %>%
  group_by(Wilayah) %>%
  summarise(
    Total_Biaya = sum(Biaya_Total),
    Efisiensi_Biaya = sum(Biaya_Total) / sum(Jumlah_Barang)
  )
efisiensi
```

```
## # A tibble: 4 x 3
##   Wilayah Total_Biaya Efisiensi_Biaya
##   <chr>         <dbl>           <dbl>
## 1 Barat      24475000          47249.
## 2 Selatan    31530000          42040
## 3 Timur      26435000          62054.
## 4 Utara      32045000          52106.
```

Wilayah dengan efisiensi biaya terbaik adalah wilayah dengan `Efisiensi_Biaya` **terendah**.

6.9.4 Persentase Pengiriman yang Melebihi Target

Hitung persentase pengiriman dengan waktu > 15 jam:

```
target <- data %>%
  group_by(Wilayah) %>%
  summarise(
    Persentase_Lambat = mean(Waktu_Pengiriman > 15) * 100
  )
target
```

```
## # A tibble: 4 x 2
##   Wilayah Persentase_Lambat
##   <chr>               <dbl>
## 1 Barat                 100
## 2 Selatan               100
## 3 Timur                   0
## 4 Utara                  40
```

6.10 Visualisasi Data

Scatter plot hubungan jumlah barang, waktu pengiriman, dan biaya pengiriman per unit:

```
library(plotly)

plot_3d <- plot_ly(
  data,
  x = ~Jumlah_Barang,
  y = ~Waktu_Pengiriman,
  z = ~Biaya_Per_Unit,
  type = 'scatter3d',
  mode = 'markers',
  color = ~Wilayah,
  size = ~Biaya_Per_Unit * 0.0001,  # Memperbesar ukuran bubble
  marker = list(
    size = 10,
    opacity = 1
  ),
  text = ~paste(
    "Wilayah:", Wilayah,
    "<br>Waktu Pengiriman:", Waktu_Pengiriman, "jam",
    "<br>Jumlah Barang:", Jumlah_Barang, "unit",
    "<br>Biaya per Unit: Rp", Biaya_Per_Unit
  )) %>%
  layout(
    title = "Analisis 3D Efisiensi Pengiriman Barang",
```

```
    scene = list(
      xaxis = list(
        title = "Jumlah Barang",
        titlefont = list(size = 12),
        tickfont = list(size = 10)
      ),
      yaxis = list(
        title = "Pengiriman (jam)",
        titlefont = list(size = 12),
        tickfont = list(size = 10)
      ),
      zaxis = list(
        title = "Biaya (Rp)",
        titlefont = list(size = 12),
        tickfont = list(size = 10)
      )
    ),
    legend = list(
      title = list(text = "Wilayah"),
      bgcolor = "rgba(255, 255, 255, 0.5)",
      bordercolor = "rgba(0, 0, 0, 0.5)",
      borderwidth = 1
    )
  )

plot_3d
```

```
## Warning: `line.width` does not currently support multiple values.
## Warning: `line.width` does not currently support multiple values.
## Warning: `line.width` does not currently support multiple values.
## Warning: `line.width` does not currently support multiple values.
```

6.11 Kesimpulan

Dari hasil analisis, kita dapat menyimpulkan bahwa:

- Mesin A memiliki distribusi waktu proses yang lebih simetris dengan sedikit pencilan, yang menunjukkan bahwa performa mesin lebih konsisten.
- Mesin B memiliki distribusi waktu proses yang lebih miring dan lebih banyak pencilan, yang mengindikasikan adanya variasi besar dalam waktu proses, dan mungkin dipengaruhi oleh faktor eksternal atau mesin yang tidak selalu berfungsi dengan optimal.

Dengan menggunakan histogram, plot densitas, dan box plot, kita dapat memvisualisasikan dan membandingkan distribusi waktu proses dari kedua mesin secara efektif.

6.12 Latihan 1

Sebuah perusahaan ingin memahami karakteristik penyebaran data hasil penjualan dari empat cabang (A, B, C, dan D) selama satu bulan terakhir. Data penjualan (dalam juta rupiah) dari keempat cabang tersebut adalah sebagai berikut:

- Cabang A: 50, 55, 60, 65, 70
- Cabang B: 40, 50, 60, 70, 80

- Cabang C: 30, 30, 35, 40, 45
- Cabang D: 70, 75, 80, 85, 90

1. Hitunglah rata-rata, median, dan standar deviasi untuk masing-masing cabang.
2. Cabang mana yang memiliki penyebaran data paling kecil? Jelaskan alasannya.
3. Jika target penjualan minimum adalah 50 juta rupiah, cabang mana saja yang gagal mencapai target di semua datanya?
4. Buatlah diagram kotak (box plot) untuk memvisualisasikan penyebaran data setiap cabang.
5. Jika Anda adalah manajer perusahaan, bagaimana Anda akan menggunakan informasi ini untuk merencanakan strategi peningkatan penjualan?

6.13 Latihan 2

Perusahaan XYZ mengelola pengiriman barang ke berbagai wilayah dengan menggunakan berbagai jenis transportasi. Setiap pengiriman melibatkan biaya transportasi, waktu yang dibutuhkan, dan jumlah barang yang dikirim. Berikut adalah data terkait pengiriman barang berdasarkan wilayah dan jenis barang:

Wilayah	Jenis Barang	Jumlah Barang (unit)	Waktu Pengiriman (jam)	Biaya per Unit (Rp)
Utara	Elektronik	200	5	15000
Selatan	Pakaian	150	8	8000
Timur	Makanan	180	6	10000
Barat	Peralatan	120	7	12000
Tengah	Elektronik	250	4	14000
Utara	Pakaian	300	9	8500
Selatan	Makanan	220	7	9500
Timur	Peralatan	140	5	11000
Barat	Elektronik	180	6	14500
Tengah	Pakaian	350	8	7800
Utara	Peralatan	170	4	12000
Selatan	Elektronik	250	6	16000
Timur	Pakaian	190	7	8200
Barat	Makanan	130	5	10500
Tengah	Peralatan	180	5	11500

1. **Analisis Efisiensi Pengiriman**:
    - Visualisasikan pengiriman barang berdasarkan jumlah barang, waktu pengiriman, dan biaya per unit dengan menggunakan plot 3D.
    - Tentukan wilayah mana yang memiliki efisiensi pengiriman terendah berdasarkan biaya per unit dan waktu pengiriman.

2. **Rekomendasi Operasional**:
    - Berdasarkan hasil analisis, wilayah mana yang memerlukan perhatian khusus untuk meningkatkan efisiensi pengiriman?
    - Apa rekomendasi untuk mengurangi biaya dan waktu pengiriman di wilayah tersebut?
3. **Kinerja Berdasarkan Jenis Barang**:

Analisis kinerja pengiriman berdasarkan jenis barang dan wilayah. Mana yang memiliki waktu pengiriman lebih cepat dan biaya per unit lebih rendah?

Part II

Teori Probabilitas

Chapter 7

Konsep Dasar Probabilitas

Probabilitas adalah cabang ilmu Matematika yang digunakan untuk mengukur ketidakpastian. Dalam sains data, probabilitas adalah dasar dari metode inferensi statistik, pengambilan keputusan, dan pembelajaran mesin. Berikut materi lengkapnya:

7.1 Ruang Sampel dan Kejadian

7.1.1 Definisi Ruang Sampel

Ruang sampel (*Sample Space*, S) adalah kumpulan semua kemungkinan hasil dari suatu percobaan acak.

- **Contoh 1**: Melempar satu dadu, $S = \{1, 2, 3, 4, 5, 6\}$.
- **Contoh 2**: Melempar dua koin, $S = \{\text{GG}, \text{GA}, \text{AG}, \text{AA}\}$ (G = Gambar, A = Angka).

7.1.2 Definisi Kejadian

Kejadian (A) adalah subset dari ruang sampel, yaitu himpunan hasil tertentu yang menjadi fokus analisis.

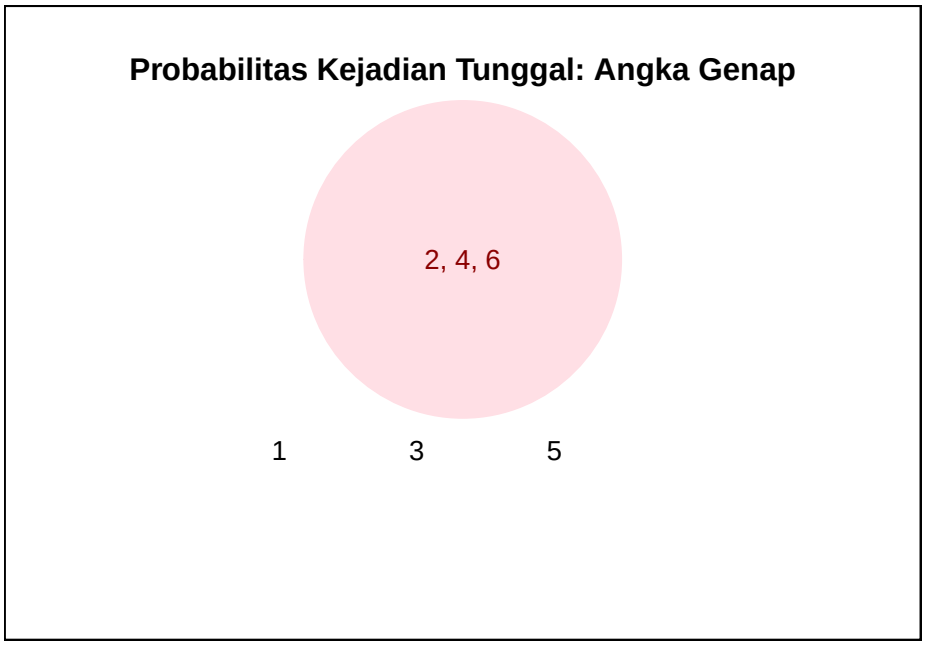
- **Contoh 1**: Kejadian A adalah "angka genap" saat melempar dadu, maka $\mathit{Genap} = \{2, 4, 6\}$.
- **Contoh 2**: Kejadian B adalah "dua koin menunjukkan B," maka $B = \{\text{KK}\}$.

7.2 Probabilitas Kejadian Tunggal

Jika semua hasil dalam ruang sampel memiliki peluang yang sama, probabilitas suatu kejadian A dihitung sebagai:

$$P(A) = \frac{\text{Jumlah hasil yang memenuhi kejadian } A}{\text{Jumlah total hasil dalam ruang sampel}}$$

Perhatikan diagram venn berikut:



7.2.1 Contoh 1: Lemparan Koin

Ruang sampel: $\Omega = \{G, A\}$ (Gambar atau Angka).
Kejadian A: Mendapatkan Gambar ($Gambar = \{G\}$).

Probabilitas:

$$P(A) = \frac{\text{Jumlah hasil yang mendukung } A}{\text{Jumlah total hasil dalam } \Omega} = \frac{1}{2} = 0.5$$

7.2.2 Contoh 2: Lemparan Dadu

Ruang sampel: $\Omega = \{1, 2, 3, 4, 5, 6\}$.
Kejadian A: Mendapatkan angka genap ($A = \{2, 4, 6\}$).

Probabilitas:

$$P(A) = \frac{\text{Jumlah hasil yang mendukung } A}{\text{Jumlah total hasil dalam } \Omega} = \frac{3}{6} = 0.5$$

7.2.3 Contoh 3: Undian

Sebuah kotak berisi 10 bola, terdiri dari 7 bola merah dan 3 bola biru.
Kejadian A: Memilih bola merah.

Probabilitas:

$$P(A) = \frac{\text{Jumlah bola merah}}{\text{Jumlah total bola}} = \frac{7}{10} = 0.7$$

7.3 Probabilitas Saling Eksklusif

Probabilitas gabungan dari dua kejadian saling eksklusif adalah peluang bahwa salah satu dari kedua kejadian tersebut terjadi, tetapi tanpa adanya irisan antara keduanya. Jika dua kejadian saling eksklusif, maka kemungkinan keduanya terjadi bersamaan adalah nol, yaitu:

$$P(A \cap B) = 0$$

Dalam hal ini, rumus untuk probabilitas gabungan $P(A \cup B)$ menjadi lebih sederhana:

$$P(A \cup B) = P(A) + P(B)$$

Mari kita ambil contoh pelemparan sebuah dadu. Misalkan kita memiliki dua kejadian:

- **Kejadian A**: Muncul angka 1 pada pelemparan dadu pertama.
- **Kejadian B**: Muncul angka 6 pada pelemparan dadu kedua.

Kedua kejadian ini saling eksklusif, karena angka 1 pada dadu pertama dan angka 6 pada dadu kedua tidak bisa muncul bersamaan pada satu pelemparan.

Langkah-langkah untuk menghitung probabilitas gabungan:

7.3.1 $P(A)$

Probabilitas kejadian A adalah muncul angka 1 pada pelemparan dadu pertama. Pada pelemparan dadu, peluang muncul angka 1 adalah:

$$P(A) = \frac{1}{6} \quad \text{(karena ada 6 sisi pada dadu)}$$

7.3.2 $P(B)$

Probabilitas kejadian B adalah muncul angka 6 pada pelemparan dadu kedua. Peluang muncul angka 6 adalah:

$$P(B) = \frac{1}{6} \quad \text{(karena ada 6 sisi pada dadu)}$$

7.3.3 $P(A \cup B)$

Karena kejadian A dan B **saling eksklusif**, maka $P(A \cap B) = 04$. **Maka**, probabilitas gabungan $P(A \cup B)$ adalah:

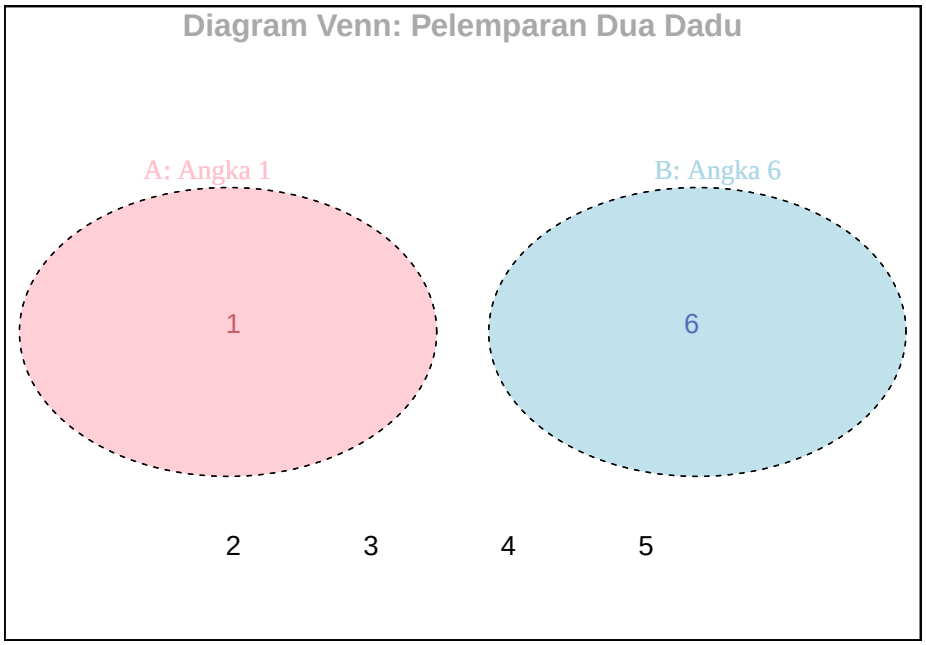
$$P(A \cup B) = P(A) + P(B) = \frac{1}{6} + \frac{1}{6} = \frac{2}{6} = \frac{1}{3}$$

Karena kejadian A dan B saling eksklusif, maka probabilitas gabungan $P(A \cup B)$ adalah:

$$P(A \cup B) = \frac{1}{3}$$

Artinya, peluang bahwa pada pelemparan dua dadu, muncul angka 1 pada dadu pertama **atau** angka 6 pada dadu kedua adalah $\frac{1}{3}$ atau sekitar 33.33%.

```
## Loading required package: futile.logger
```

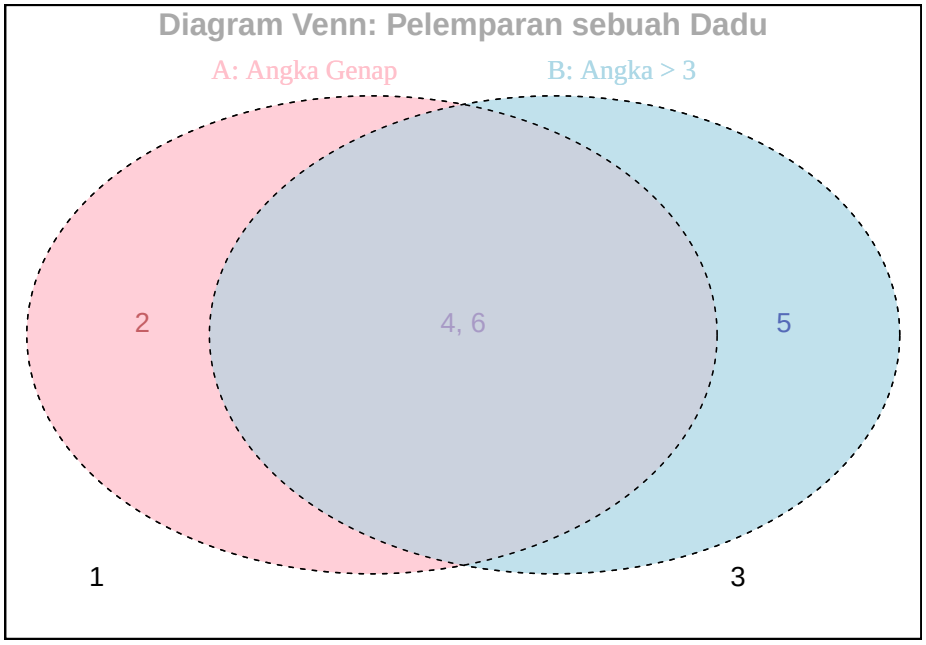


7.4 Probabilitas Tidak Saling Eksklusif

Probabilitas Gabungan (Tidak Saling Eksklusif) mencakup semua hasil dalam kejadian A, B, atau keduanya. Rumusnya:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Andaikan dilakukan pelemparan sebuah dadu, diperlihatkan dalam diagram sebagai berikut:



Sehingga diperoleh:

7.4.1 $P(A)$

Kejadian A terdiri dari angka genap $\{2, 4, 6\}$. Peluangnya adalah:

$$P(A) = \frac{\text{Jumlah elemen di A}}{\text{Jumlah total elemen}} = \frac{3}{6} = 0.5$$

7.4.2 $P(B)$

Kejadian B terdiri dari angka lebih dari 3 $\{4, 5, 6\}$. Peluangnya adalah:

$$P(B) = \frac{\text{Jumlah elemen di B}}{\text{Jumlah total elemen}} = \frac{3}{6} = 0.5$$

7.4.3 $P(A \cap B))$

Kejadian A dan B yang terjadi bersamaan adalah angka yang genap dan lebih dari 3, yaitu $\{4, 6\}$. Peluangnya adalah:

$$P(A \cap B) = \frac{\text{Jumlah elemen di irisan A dan B}}{\text{Jumlah total elemen}} = \frac{2}{6} = \frac{1}{3}$$

Maka, probabilitas gabungan dari A dan B adalah:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$P(A \cup B) = 0.5 + 0.5 - \frac{1}{3} = 1 - \frac{1}{3} = \frac{2}{3} \approx 0.6667$$

7.5 Probabilitas Bersyarat

Probabilitas bersyarat mengukur peluang kejadian A terjadi, dengan syarat bahwa kejadian B sudah terjadi. Probabilitas bersyarat ditulis sebagai $P(A \mid B)$ dan dihitung menggunakan rumus berikut:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Di sini:

- $P(A \mid B)$ adalah probabilitas kejadian A terjadi, dengan syarat kejadian B telah terjadi.
- $P(A \cap B)$ adalah probabilitas kejadian A dan B terjadi bersamaan (irisan).
- $P(B)$ adalah probabilitas kejadian B.

Misalkan kita melakukan pelemparan dua buah dadu, dan kita memiliki dua kejadian:

- Kejadian A: Muncul angka genap pada pelemparan dadu pertama.
- Kejadian B: Muncul angka lebih dari 3 pada pelemparan dadu kedua.

Kita ingin menghitung probabilitas bersyarat $P(A \mid B)$, yaitu peluang bahwa angka yang muncul pada pelemparan dadu pertama adalah genap, dengan syarat bahwa angka pada pelemparan dadu kedua lebih besar dari 3.

Langkah-langkah untuk menghitung probabilitas bersyarat:

7.5.1 $P(A \cap B)$

Kejadian $A \cap B$ adalah kejadian di mana dadu pertama menunjukkan angka genap dan dadu kedua menunjukkan angka lebih dari 3. Angka yang memenuhi kondisi ini adalah pasangan-pasangan berikut:

- (2, 4), (2, 5), (2, 6)
- (4, 4), (4, 5), (4, 6)
- (6, 4), (6, 5), (6, 6)

Total ada 9 pasangan yang memenuhi kondisi ini. Jadi, $P(A \cap B)$ adalah:

$$P(A \cap B) = \frac{9}{36} = \frac{1}{4}$$

7.5.2 $P(B)$

Kejadian B adalah angka lebih dari 3 pada dadu kedua, yang terdiri dari $\{4, 5, 6\}$. Jadi, terdapat 3 kemungkinan pada pelemparan dadu kedua, dan total kemungkinan pada pelemparan dua dadu adalah 36. Maka, $P(B)$ adalah:

$$P(B) = \frac{3 \times 6}{36} = \frac{18}{36} = \frac{1}{2}$$

7.5.3 $P(A \mid B)$

Menggunakan rumus probabilitas bersyarat:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{1}{4}}{\frac{1}{2}} = \frac{1}{2}$$

Probabilitas bersyarat $P(A \mid B)$ adalah $\frac{1}{2}$, atau 50%. Artinya, jika diketahui bahwa angka pada dadu kedua lebih dari 3, maka peluang angka genap muncul pada dadu pertama adalah 50%.

7.6 Probabilitas dalam Sains Data

7.6.1 Metode Pengambilan Sampel

Pada analisis data, penting untuk menentukan ukuran sampel yang diperlukan agar estimasi dari populasi memiliki tingkat keakuratan yang diinginkan. Ukuran sampel dapat dihitung menggunakan probabilitas, tingkat kepercayaan, dan margin of error.

Rumus yang digunakan untuk menghitung ukuran sampel pada populasi besar atau tak terbatas adalah:

$$n = \frac{Z^2 \times p \times (1-p)}{E^2}$$

Dimana:

- n = jumlah sampel yang diperlukan
- Z = nilai Z pada tingkat kepercayaan yang diinginkan (misalnya, untuk tingkat kepercayaan 95%, $Z = 1.96$)
- p = proporsi yang diharapkan (misalnya, $p = 0.5$ jika kita tidak tahu proporsi pasti)
- E = margin of error yang dapat diterima (misalnya, $E = 0.05$)

Jika ukuran populasi terbatas, rumus dapat disesuaikan dengan faktor koreksi:

$$n_{\text{adjusted}} = \frac{n}{1 + \frac{(n-1)}{N}}$$

Dimana:

- N = ukuran populasi

Misalkan Anda ingin melakukan survei pada populasi besar dengan tingkat kepercayaan 95% dan margin of error 5%. Anda memperkirakan proporsi dalam populasi adalah 50% (misalnya, 50% dari populasi menggunakan produk tertentu).

Untuk kasus ini, kita akan menghitung ukuran sampel yang diperlukan dengan menggunakan R untuk menghitung ukuran sampel. Berikut adalah kode R untuk menghitung ukuran sampel berdasarkan informasi di atas:

```
# Fungsi untuk menghitung ukuran sampel
sample_size <- function(Z, p, E) {
  n <- (Z^2 * p * (1 - p)) / E^2
  return(n)
}

# Parameter
Z <- 1.96    # Z untuk tingkat kepercayaan 95%
p <- 0.5     # Estimasi proporsi
E <- 0.05    # Margin of error 5%

# Hitung ukuran sampel
n <- sample_size(Z, p, E)
cat("Ukuran sampel yang diperlukan: ", ceiling(n), "\n")
```

```
## Ukuran sampel yang diperlukan:  385
```

Dari perhitungan di atas, ukuran sampel yang diperlukan adalah sekitar 384. Ini berarti Anda perlu mengambil sampel sebanyak 384 untuk mendapatkan estimasi dengan margin of error 5% dan tingkat kepercayaan 95%.

7.7 Studi Kasus 1

Penerapan Probabilitas dalam Prediksi Kualitas Produk:

Sebuah perusahaan manufaktur memproduksi barang elektronik dan ingin memprediksi apakah suatu produk akan cacat atau tidak. Data historis menunjukkan bahwa 5% dari produk yang diproduksi adalah cacat. Perusahaan menggunakan data tentang jenis komponen dan proses produksi untuk memprediksi cacat produk menggunakan teknik probabilitas.

7.7.1 Fitur Data

- **Komponen (C)**: Apakah komponen elektronik yang digunakan adalah berkualitas tinggi atau rendah.
- **Proses Produksi (P)**: Apakah proses produksi dilakukan di bawah standar atau sesuai standar.
- **Cacat (D)**: Status cacat produk (ya/tidak).

7.7.2 Data Historis (Contoh)

- Probabilitas produk cacat ($P(D = \text{Yes})$) = 5%
- Probabilitas produk tidak cacat ($P(D = \text{No})$) = 95%
- Probabilitas menggunakan komponen berkualitas rendah ($P(C = \text{Low})$) = 30%
- Probabilitas menggunakan komponen berkualitas tinggi ($P(C = \text{High})$) = 70%
- Probabilitas proses produksi di bawah standar ($P(P = \text{Below})$) = 40%
- Probabilitas proses produksi sesuai standar ($P(P = \text{Standard})$) = 60%

Bagaimana probabilitas bahwa suatu produk akan cacat ($D = \text{Yes}$), jika diketahui komponen yang digunakan berkualitas rendah dan proses produksi di bawah standar?

Gunakan **Teorema Bayes** untuk menghitung probabilitas bersyarat ini:

$$P(D = \text{Yes} \mid C = \text{Low}, P = \text{Below}) = \frac{P(C = \text{Low}, P = \text{Below} \mid D = \text{Yes}) \cdot P(D = \text{Yes})}{P(C = \text{Low}, P = \text{Below})}$$

7.8 Studi Kasus 2

Penerapan Probabilitas dalam Deteksi Penipuan Transaksi:

Sebuah perusahaan e-commerce ingin mendeteksi transaksi yang berpotensi penipuan. Berdasarkan data historis, 1% dari transaksi yang dilakukan adalah penipuan. Perusahaan ingin menggunakan fitur-fitur tertentu seperti **lokasi transaksi**, **jumlah pembelian**, dan **metode pembayaran** untuk memprediksi apakah suatu transaksi adalah penipuan atau tidak.

7.8.1 Fitur Data

- **Lokasi (L)**: Negara atau kota tempat transaksi dilakukan.
- **Jumlah Pembelian (A)**: Jumlah uang yang dibelanjakan.
- **Metode Pembayaran (M)**: Metode pembayaran yang digunakan (kartu kredit, dompet digital, dll).
- **Penipuan (F)**: Status transaksi apakah penipuan atau tidak.

7.8.2 Data Historis (Contoh)

- Probabilitas transaksi adalah penipuan ($P(F = \text{Fraud})) = 1\%$
- Probabilitas transaksi bukan penipuan ($P(F = \text{Not Fraud})) = 99\%$
- Probabilitas lokasi tertentu adalah di luar negeri ($P(L = \text{Foreign})) = 20\%$
- Probabilitas jumlah pembelian lebih dari 500(P(A = High)\$) = 10%
- Probabilitas menggunakan kartu kredit sebagai metode pembayaran ($P(M = \text{Credit Card})) = 50\%$

Bagaimana probabilitas bahwa suatu transaksi adalah penipuan ($F = \text{Fraud}$), jika diketahui transaksi dilakukan dari lokasi luar negeri, jumlah pembelian lebih dari \$500, dan metode pembayaran menggunakan kartu kredit?

Gunakan **Teorema Bayes** untuk menghitung probabilitas bersyarat ini:

$$P(F = \text{Fraud} \mid L = \text{Foreign}, A = \text{High}, M = \text{Credit Card}) = \frac{P(L = \text{Foreign}, A = \text{High}, M = \text{Credit Car}}{P(L = \text{Foreign}, A = \text{High}, M}$$

Chapter 8

Distribusi Probabilitas dan Sampling

Distribusi probabilitas adalah konsep fundamental dalam statistika yang menggambarkan bagaimana probabilitas suatu kejadian didistribusikan di antara semua hasil yang mungkin. Dalam sains data, distribusi probabilitas membantu memahami pola dan perilaku data, baik untuk tujuan deskriptif maupun inferensial.

8.1 Distribusi Diskrit

Distribusi diskrit digunakan untuk menggambarkan situasi di mana variabel hanya dapat memiliki nilai tertentu, seperti bilangan bulat. Dua distribusi diskrit yang sering digunakan adalah **Distribusi Binomial** dan **Distribusi Poisson**.

8.1.1 Distribusi Binomial

Distribusi Binomial digunakan untuk memodelkan jumlah keberhasilan dalam sejumlah percobaan independen, di mana setiap percobaan memiliki dua kemungkinan hasil: berhasil (*success*) atau gagal (*failure*).

Ciri-Ciri Distribusi Binomial

- **Percobaan Bernoulli**: Hasil dari setiap percobaan hanya terdiri dari dua kemungkinan (misalnya, sukses/gagal).
- **Probabilitas Tetap**: Probabilitas keberhasilan (p) tetap sama untuk setiap percobaan.

- **Independensi**: Hasil satu percobaan tidak memengaruhi percobaan lainnya.
- **Jumlah Percobaan Terbatas**: Sebanyak n percobaan dilakukan.

Fungsi Distribusi Probabilitas Binomial

Probabilitas k keberhasilan dari n percobaan dihitung menggunakan:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Di mana:

- $\binom{n}{k} = \frac{n!}{k!(n-k)!}$: Kombinasi n percobaan yang memilih k keberhasilan.
- p: Probabilitas keberhasilan dalam satu percobaan.
- $(1-p)$: Probabilitas kegagalan dalam satu percobaan.

Contoh Distribusi Probabilitas Binomial

Jika sebuah koin dilempar 10 kali ($n = 10$) dengan probabilitas munculnya sisi gambar $p = 0.5$, maka distribusi binomial dapat digunakan untuk memodelkan jumlah sisi gambar yang muncul. Untuk menghitung probabilitas binomial, kita harus terlebih dahulu menghitung nilai kombinasi $\binom{n}{k}$.

Kombinasi dihitung dengan rumus:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Untuk kasus ini, kita ingin menghitung probabilitas munculnya **5 sisi gambar** (jadi $k = 5$) dalam 10 lemparan koin.

Menggunakan rumus kombinasi:

$$\binom{10}{5} = \frac{10!}{5!(10-5)!} = \frac{10 \times 9 \times 8 \times 7 \times 6}{5 \times 4 \times 3 \times 2 \times 1} = 252$$

Setelah kita menghitung kombinasi, kita dapat menghitung probabilitas untuk mendapatkan **5 sisi gambar** menggunakan rumus distribusi binomial:

$$P(X = 5) = \binom{10}{5} (0.5)^5 (1-0.5)^{10-5}$$

Substitusikan nilai-nilai yang sudah diketahui:

$$P(X = 5) = 252 \times (0.5)^5 \times (0.5)^5$$

Sederhanakan perhitungan:

$$(0.5)^5 = \frac{1}{32}$$

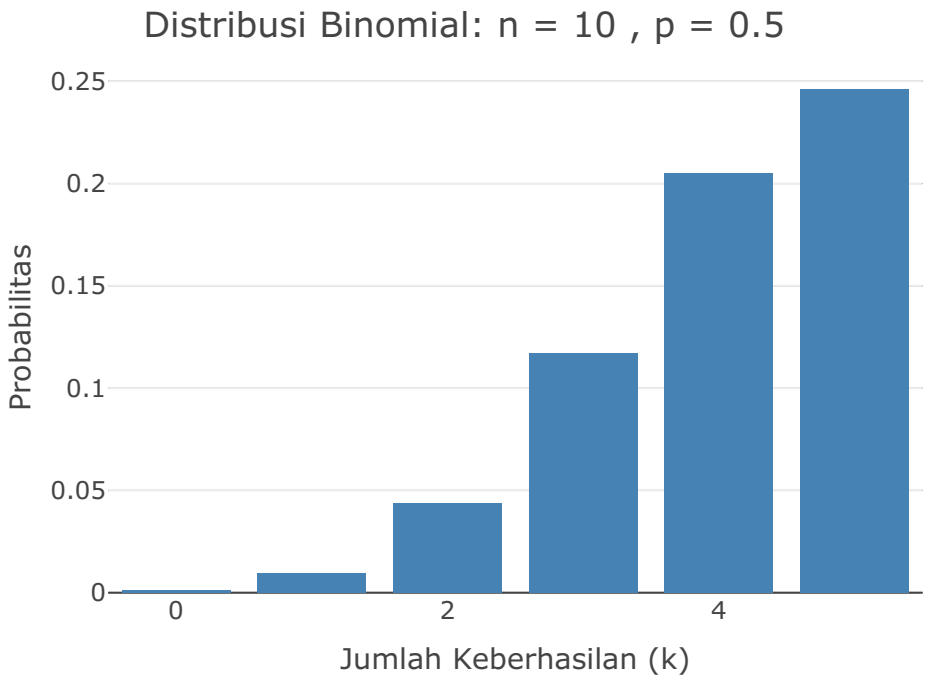
Sehingga:

$$P(X = 5) = 252 \times \frac{1}{32} \times \frac{1}{32} = 252 \times \frac{1}{1024} = 0.2461$$

```r
# Contoh Distribusi Binomial di R
n <- 10                       # banyaknya percobaaan
p <- 0.5                      # Probabilitas muncul gambar (1 kali pelemparan)
dbinom(5, size = n, prob = p) # Probabilitas muncul 5 gambar
```

```
## [1] 0.2460938
```

Visualisasi Distribusi Binomial



8.1.2 Distribusi Poisson

Distribusi Poisson digunakan untuk memodelkan jumlah kejadian dalam interval waktu atau ruang tertentu, dengan asumsi bahwa kejadian bersifat independen dan terjadi dengan rata-rata tetap.

Ciri-Ciri Distribusi Poisson

- Kejadian bersifat acak dan independen.
- Probabilitas terjadinya kejadian dalam interval kecil tetap konstan.
- Tidak ada dua kejadian yang terjadi pada waktu yang bersamaan (asumsi kelangkaan).

Fungsi Probabilitas Distribusi Poisson

Probabilitas terjadinya k kejadian dalam interval tertentu dihitung menggunakan:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Di mana:

- λ: Rata-rata jumlah kejadian dalam interval tertentu.
- k: Jumlah kejadian yang diinginkan.
- e: Bilangan Euler (≈ 2.718).

Contoh Distribusi Poisson

Misalkan rata-rata jumlah kendaraan yang melewati sebuah jalan tol adalah 3 kendaraan per menit ($\lambda = 3$). Distribusi Poisson dapat digunakan untuk menghitung probabilitas bahwa tepat 5 kendaraan akan melewati jalan tersebut dalam satu menit.

Dalam masalah ini, kita tahu bahwa:

- $\lambda = 3$ (rata-rata kendaraan per menit),
- $k = 5$ (jumlah kendaraan yang kita ingin hitung probabilitasnya),
- $e \approx 2.718$.

Kita akan menghitung probabilitas bahwa tepat 5 kendaraan melewati jalan tol dalam satu menit menggunakan rumus distribusi Poisson. Substitusikan nilai-nilai tersebut ke dalam rumus:

$$P(X = 5) = \frac{3^5 e^{-3}}{5!}$$

Sehingga probabilitasnya:

$$P(X = 5) = \frac{243 \times 0.0498}{120} \approx \frac{12.1044}{120} \approx 0.1009$$

Berikut adalah perhitungan probabilitas $P(X = 5)$ untuk distribusi Poisson menggunakan R:

```
# Parameter distribusi Poisson
lambda <- 3  # Rata-rata kejadian
k <- 5       # Jumlah kejadian yang dihitung

# Menghitung probabilitas
prob <- dpois(k, lambda)
prob
```

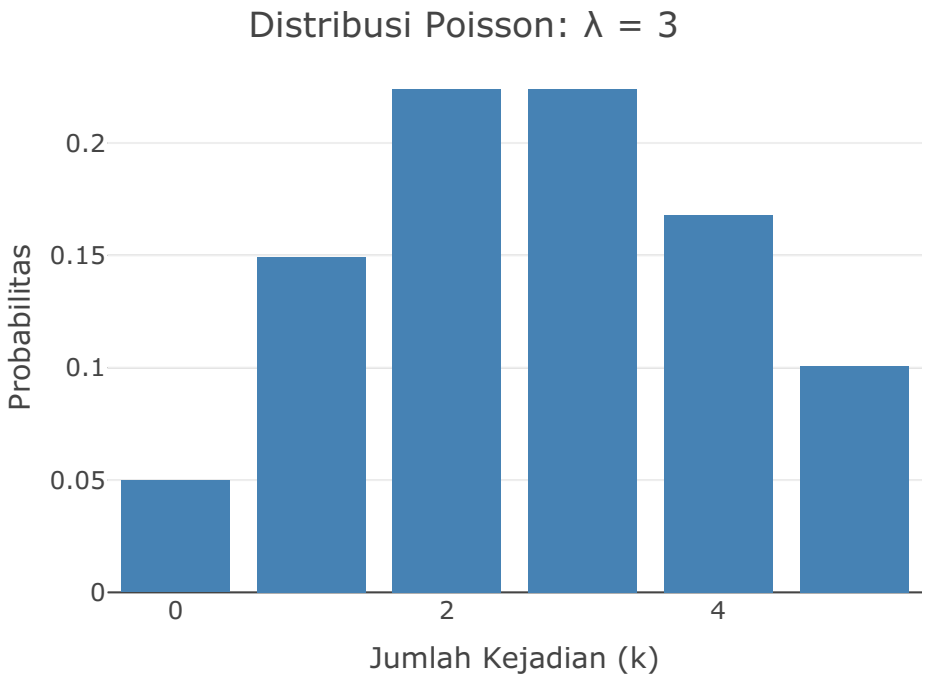
```
## [1] 0.1008188
```

Probabilitas bahwa tepat **5 kendaraan** akan melewati jalan tol dalam satu menit adalah sekitar **0.1009** atau **10.09%**.

Visualisasi Distribusi Poisson

```
# Memuat library yang diperlukan
library(plotly)

# Fungsi untuk menghitung distribusi Poisson
generate_poisson_plot <- function(lambda, k) {
  k_values <- 0:k
  probs <- dpois(k_values, lambda)

  # Membuat plot interaktif menggunakan plotly
  plot_ly(
    x = k_values,
    y = probs,
    type = 'bar',
    marker = list(color = 'steelblue')
  ) %>%
    layout(
      title = paste("Distribusi Poisson:  =", lambda),
      xaxis = list(title = 'Jumlah Kejadian (k)'),
      yaxis = list(title = 'Probabilitas')
    )
}
```

```r
# Definisikan parameter untuk distribusi Poisson
lambda <- 3   # Rata-rata kejadian per interval waktu
k <- 5        # Ruang sampel (jumlah kejadian yang dihitung)

# Menampilkan plot distribusi Poisson
generate_poisson_plot(lambda, k)
```



8.2 Distribusi Kontinu

Distribusi kontinu digunakan untuk menggambarkan situasi di mana variabel dapat memiliki nilai-nilai yang tidak terbatas, yang bisa berupa bilangan real.

8.2.1 Distribusi Uniform

Distribusi Uniform digunakan untuk memodelkan kejadian yang memiliki probabilitas yang sama untuk setiap nilai dalam rentang tertentu. Dalam distribusi ini, setiap nilai dalam interval memiliki peluang yang sama untuk terjadi.

Ciri-Ciri Distribusi Uniform

- **Probabilitas Konstan**: Setiap nilai dalam interval memiliki probabilitas yang sama untuk terjadi.

- **Batas Bawah dan Atas**: Distribusi ini hanya terdefinisi dalam rentang tertentu, yaitu antara batas bawah a dan batas atas b.
- **Keberagaman Nilai yang Sama**: Tidak ada nilai yang lebih mungkin terjadi daripada nilai lainnya dalam interval tersebut.

Fungsi Probabilitas Distribusi Uniform

Untuk distribusi Uniform Kontinu, probabilitas bahwa suatu variabel acak X berada dalam rentang $[a, b]$ dapat dihitung menggunakan fungsi kepadatan probabilitas (PDF) berikut:

$$f(x) = \frac{1}{b-a} \quad \text{untuk} \quad a \leq x \leq b$$

Di mana:

- a: Batas bawah dari distribusi.
- b: Batas atas dari distribusi.

Contoh Distribusi Uniform

Misalkan suatu eksperimen menghasilkan angka acak antara 0 dan 10. Dalam hal ini, distribusi angka yang dihasilkan adalah distribusi Uniform dengan batas bawah $a = 0$ dan batas atas $b = 10$. Kita dapat menghitung probabilitas bahwa angka yang dihasilkan berada dalam interval $[3, 7]$.

Untuk menghitung probabilitas tersebut, kita dapat menggunakan fungsi distribusi kumulatif (CDF) untuk distribusi Uniform:

$$P(3 \leq X \leq 7) = F(7) - F(3)$$

Dimana fungsi distribusi kumulatifnya adalah:

$$F(x) = \frac{x-a}{b-a}$$

Substitusi $x = 7$, $a = 0$, dan $b = 10$:

$$F(7) = \frac{7-0}{10-0} = 0.7$$

Substitusi $x = 3$, $a = 0$, dan $b = 10$:

$$F(3) = \frac{3-0}{10-0} = 0.3$$

Jadi, probabilitas bahwa angka yang dihasilkan berada dalam interval $[3, 7]$ adalah:

$$P(3 \leq X \leq 7) = F(7) - F(3) = 0.7 - 0.3 = 0.4$$

Berikut adalah cara menghitung probabilitas untuk interval tertentu menggunakan distribusi Uniform di R:

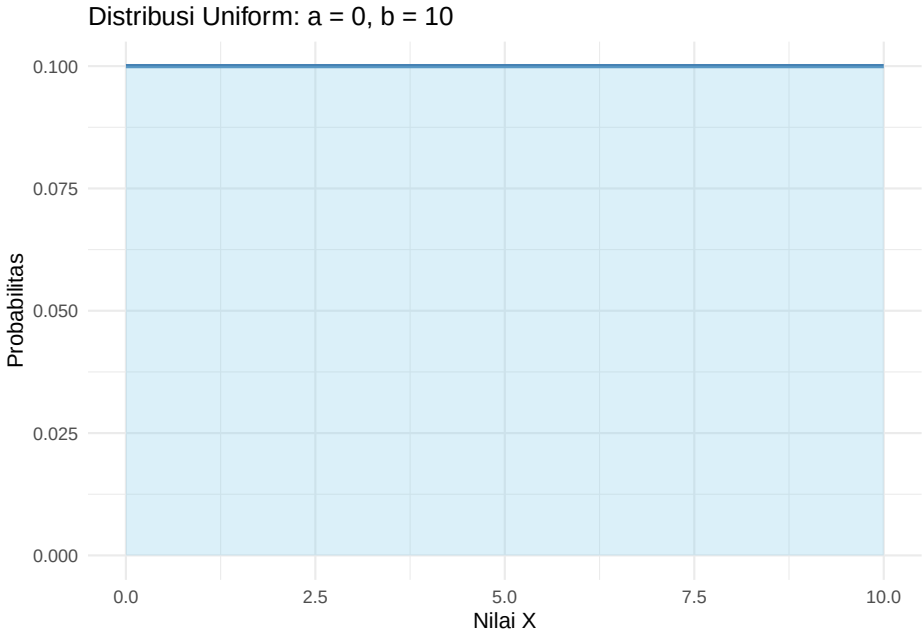
```
# Parameter distribusi Uniform
a <- 0   # Batas bawah
b <- 10  # Batas atas

# Menghitung probabilitas bahwa X berada dalam interval [3, 7]
probabilitas <- punif(7, min = a, max = b) - punif(3, min = a, max = b)
probabilitas
```

```
## [1] 0.4
```

Visualisasi Distribusi Uniform

Berikut adalah visualisasi distribusi Uniform dengan batas bawah $a = 0$ dan batas atas $b = 10$:

```
# Memuat library yang diperlukan
library(ggplot2)

# Parameter distribusi Uniform
a <- 0   # Batas bawah
b <- 10  # Batas atas

# Membuat vektor untuk nilai x
x_vals <- seq(a, b, length.out = 100)

# Menghitung probabilitas (fungsi kepadatan)
y_vals <- rep(1 / (b - a), length(x_vals))

# Membuat plot
ggplot(data = data.frame(x = x_vals, y = y_vals), aes(x = x, y = y)) +
  geom_line(color = 'steelblue', size = 1) +
  geom_ribbon(aes(ymin = 0, ymax = y), fill = 'skyblue', alpha = 0.3) +
  labs(
    title = "Distribusi Uniform: a = 0, b = 10",
    x = "Nilai X",
    y = "Probabilitas"
  ) +
  theme_minimal()
```



8.2.2 Distribusi Normal

Distribusi Normal adalah distribusi kontinu yang sering digunakan dalam statistik untuk menggambarkan distribusi variabel yang terdistribusi secara simetris di sekitar nilai rata-rata. Distribusi ini sangat penting karena banyak fenomena alam dan sosial yang mengikuti pola distribusi normal.

Ciri-Ciri Distribusi Normal

- **Simetris**: Distribusi normal bersifat simetris di sekitar nilai rata-rata (μ).
- **Bentuk Lonceng**: Bentuk distribusi normal adalah lonceng, dengan puncak di rata-rata dan merata di kedua sisi.
- **Rata-rata, Median, dan Modus Sama**: Nilai rata-rata (μ), median, dan modus dari distribusi normal adalah sama.
- **Asimtotik**: Kurva normal semakin mendekati sumbu horizontal, tetapi tidak pernah menyentuhnya.
- **Parameter**: Distribusi normal sepenuhnya ditentukan oleh dua parameter:
  - μ: Rata-rata (mean) dari distribusi.
  - σ: Simpangan baku (standar deviasi), yang mengukur sebaran data.

Fungsi Kepadatan Probabilitas Distribusi Normal

Fungsi kepadatan probabilitas distribusi normal diberikan oleh rumus:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Di mana: - x: Variabel acak yang terdistribusi normal. - μ: Rata-rata dari distribusi. - σ: Simpangan baku dari distribusi. - exp adalah fungsi eksponensial.

Contoh Distribusi Normal

Misalkan tinggi badan siswa di sebuah sekolah terdistribusi normal dengan rata-rata (μ) 170 cm dan simpangan baku (σ) 10 cm. Jika kita ingin menghitung probabilitas bahwa seorang siswa memiliki tinggi badan antara 160 cm dan 180 cm, kita dapat menghitungnya menggunakan distribusi normal.

Untuk menghitung probabilitas ini, kita akan menggunakan fungsi distribusi kumulatif normal. Misalkan kita ingin menghitung probabilitas:

$$P(160 \leq X \leq 180)$$

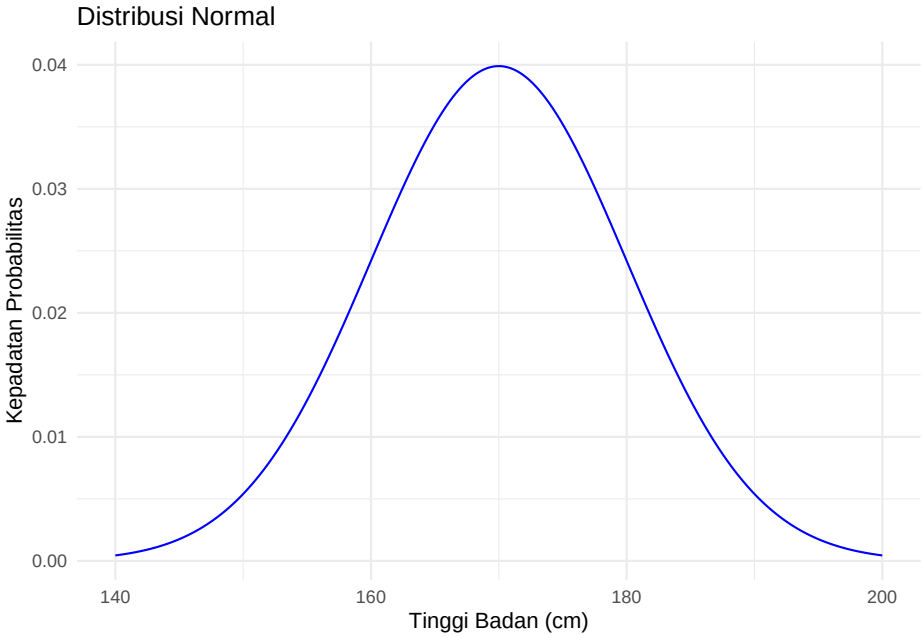
Ini dihitung dengan mencari selisih antara probabilitas kumulatif di 180 dan 160.

```
# Menggunakan distribusi normal untuk menghitung probabilitas
mu <- 170  # rata-rata
sigma <- 10  # simpangan baku
prob <- pnorm(180, mean = mu, sd = sigma) - pnorm(160, mean = mu, sd = sigma)
prob
```

```
## [1] 0.6826895
```

Visualisasi Distribusi Normal

```
# Memuat library yang diperlukan
library(ggplot2)

# Membuat data untuk distribusi normal
x_values <- seq(140, 200, length.out = 1000)
y_values <- dnorm(x_values, mean = mu, sd = sigma)

# Membuat plot distribusi normal
ggplot(data.frame(x = x_values, y = y_values), aes(x = x, y = y)) +
  geom_line(color = "blue") +
  ggtitle("Distribusi Normal") +
  xlab("Tinggi Badan (cm)") +
  ylab("Kepadatan Probabilitas") +
  theme_minimal()
```



8.2.3 Distribusi Eksponensial

Distribusi Eksponensial digunakan untuk memodelkan waktu antara kejadian-kejadian dalam suatu proses Poisson. Distribusi ini sering digunakan untuk menggambarkan waktu tunggu atau durasi sampai terjadinya kejadian pertama.

Ciri-Ciri Distribusi Eksponensial

- **Memoryless**: Probabilitas kejadian berikutnya tidak tergantung pada waktu yang telah berlalu.
- **Parameter Tunggal**: λ adalah rata-rata laju kejadian per satuan waktu.
- **Model Waktu Tunggu**: Digunakan untuk memodelkan waktu tunggu atau durasi.

Fungsi Kepadatan Probabilitas

Fungsi kepadatan probabilitas distribusi eksponensial diberikan oleh:

$$f(x) = \lambda \exp(-\lambda x),\ x \geq 0$$

Contoh Distribusi Eksponensial

Misalkan sebuah mesin memiliki waktu antar kegagalan yang terdistribusi eksponensial dengan rata-rata waktu antar kegagalan **2 jam** ($\lambda = \frac{1}{2}$). Jika kita

ingin menghitung probabilitas bahwa mesin gagal dalam waktu kurang dari **1 jam**, kita dapat menggunakan fungsi distribusi kumulatif:

$$P(X \leq x) = 1 - \exp(-\lambda x)$$

Substitusikan nilai-nilai yang diketahui:

$$P(X \leq 1) = 1 - \exp\left(-\frac{1}{2} \cdot 1\right)$$

Hasilnya:

$$P(X \leq 1) = 1 - \exp(-0.5) \approx 1 - 0.6065 = 0.3935$$

Implementasi di R:

```
# Parameter distribusi eksponensial
lambda <- 1 / 2  # Laju kejadian (1/2 kejadian per jam)
x <- 1           # Waktu yang dihitung (1 jam)

# Probabilitas mesin gagal dalam waktu kurang dari 1 jam
prob <- pexp(x, rate = lambda)
prob
```

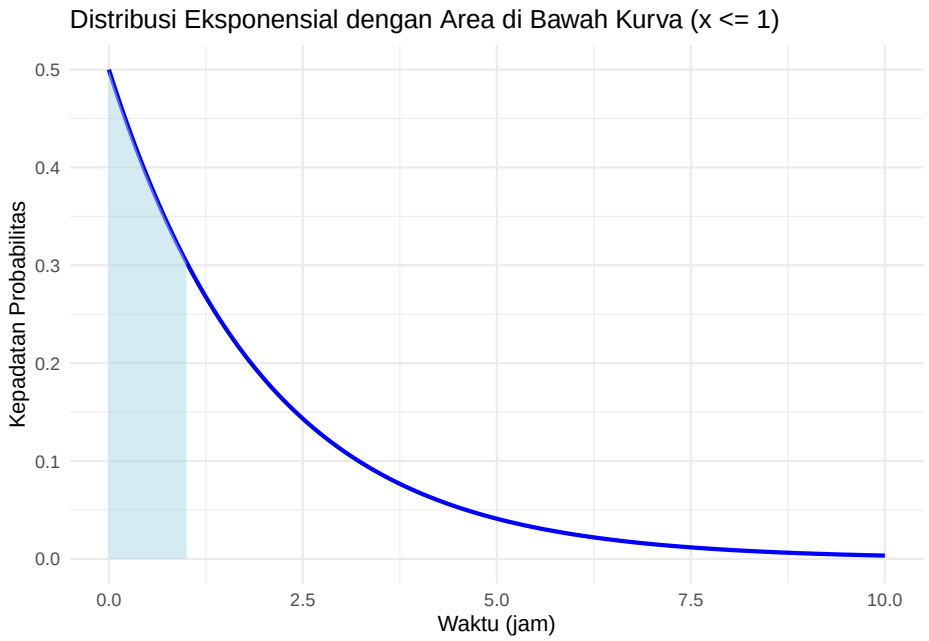
```
## [1] 0.3934693
```

Visualisasi Distribusi Eksponensial

Kita dapat memvisualisasikan distribusi eksponensial untuk menunjukkan probabilitas.

```
# Memuat library
library(ggplot2)

# Membuat data untuk distribusi eksponensial
x_values <- seq(0, 10, length.out = 1000)
y_values <- dexp(x_values, rate = lambda)

# Data untuk area di bawah kurva (x <= 1)
x_fill <- seq(0, 1, length.out = 100)
y_fill <- dexp(x_fill, rate = lambda)

# Membuat plot
ggplot(data.frame(x = x_values, y = y_values), aes(x = x, y = y)) +
  geom_line(color = "blue", linewidth = 1) +  # Mengganti size dengan linewidth
  geom_area(data = data.frame(x = x_fill, y = y_fill), aes(x = x, y = y), fill = "ligh
```

```
  ggtitle("Distribusi Eksponensial dengan Area di Bawah Kurva (x   1)") +
  xlab("Waktu (jam)") +
  ylab("Kepadatan Probabilitas") +
  theme_minimal()
```



8.2.4 Distribusi Beta

Distribusi Beta adalah distribusi kontinu yang digunakan untuk memodelkan variabel acak yang terletak dalam interval $[0, 1]$. Distribusi ini sering digunakan dalam analisis Bayesian dan untuk memodelkan proporsi atau probabilitas.

Ciri-Ciri Distribusi Beta

- **Interval** $[0, 1]$: Variabel acak yang terdistribusi Beta selalu berada dalam rentang $[0, 1]$.
- **Bergantung pada dua parameter**: Distribusi Beta memiliki dua parameter, yaitu α (alpha) dan β (beta), yang mempengaruhi bentuk distribusi.
- **Bentuk Distribusi Fleksibel**: Bentuk distribusi Beta sangat fleksibel, dapat berupa distribusi yang lebih mirip distribusi uniform, lebih mirip distribusi normal, atau distribusi berbentuk U tergantung pada nilai parameter α dan β. Hubungan antara kedua parameter ini mempengaruhi bentuk distribusi secara signifikan.
  - Ketika $\alpha = \beta$: distribusi Beta akan simetris.

- $\alpha > \beta$: distribusi beta akan condong ke kanan [1].
- Ketika $\alpha < \beta$: distribusi Beta akan condong ke kiri [0].

Fungsi Probabilitas Distribusi Beta

Fungsi kepadatan probabilitas (PDF) dari distribusi Beta adalah:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \quad \text{untuk} \quad 0 \leq x \leq 1$$

Di mana:

- α: Parameter pertama (shape parameter).
- β: Parameter kedua (shape parameter).
- $B(\alpha, \beta)$: Fungsi Beta, yang merupakan normalisasi konstanta yang memastikan bahwa total probabilitas sama dengan 1.

Contoh Distribusi Beta

Untuk distribusi Beta dengan parameter $\alpha = 2$ dan $\beta = 5$, distribusi ini cenderung memiliki penyebaran lebih besar ke arah 0 (kiri), karena $\alpha < \beta$. Ini berarti bahwa probabilitasnya lebih besar di sisi kiri distribusi, tetapi nilai kumulatif untuk interval $[0.2, 0.8]$ lebih kecil daripada yang mungkin Anda perkirakan jika melihat distribusi seragam.

Dengan parameter $\alpha = 2$ dan $\beta = 5$, kita dapat mengharapkan hasil seperti berikut:

- $F(0.8) \approx 0.8316$ (CDF pada $x = 0.8$)
- $F(0.2) \approx 0.1779$ (CDF pada $x = 0.2$)

Sehingga,

$$P(0.2 \leq X \leq 0.8) = F(0.8) - F(0.2) \approx 0.8316 - 0.1779 = 0.65376$$

Jadi, hasil 0.65376 atau sekitar 65.38% adalah probabilitas yang benar untuk distribusi Beta dengan parameter $\alpha = 2$ dan $\beta = 5$ dalam interval $[0.2, 0.8]$.

Distribusi Beta Menggunakan R

Berikut adalah cara menghitung probabilitas untuk interval tertentu menggunakan distribusi Beta di R:

```
# Parameter distribusi Beta
alpha <- 2  # Parameter alpha
beta <- 5   # Parameter beta

# Menghitung probabilitas bahwa X berada dalam interval [0.2, 0.8]
```

```r
probabilitas <- pbeta(0.8, alpha, beta) - pbeta(0.2, alpha, beta)
probabilitas
```

```
## [1] 0.65376
```

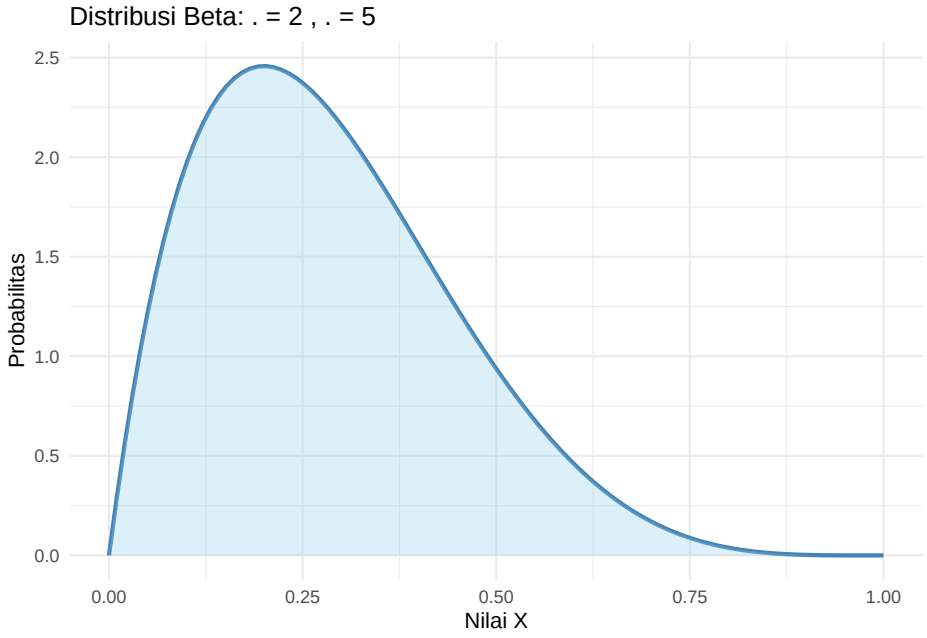
Probabilitas bahwa proporsi keberhasilan berada dalam interval $[0.2, 0.8]$ adalah 0.8833 atau 88.33%.

Dengan $I_x(\alpha, \beta)$ adalah fungsi distribusi kumulatif Beta yang terintegrasi. Menggunakan parameter $\alpha = 2$ dan $\beta = 5$, kita dapat menghitung probabilitas ini menggunakan fungsi CDF dari distribusi Beta.

Visualisasi Distribusi Beta

Berikut adalah visualisasi distribusi Beta dengan parameter $\alpha = 2$ dan $\beta = 5$:

```r
# Memuat library yang diperlukan
library(ggplot2)

# Parameter distribusi Beta
alpha <- 2  # Parameter alpha
beta <- 5   # Parameter beta

# Membuat vektor untuk nilai x
x_vals <- seq(0, 1, length.out = 100)

# Menghitung probabilitas (fungsi kepadatan)
y_vals <- dbeta(x_vals, alpha, beta)

# Membuat plot
ggplot(data = data.frame(x = x_vals, y = y_vals), aes(x = x, y = y)) +
  geom_line(color = 'steelblue', size = 1) +
  geom_area(fill = 'skyblue', alpha = 0.3) +
  labs(
    title = paste("Distribusi Beta:  =", alpha, ",  =", beta),
    x = "Nilai X",
    y = "Probabilitas"
  ) +
  theme_minimal()
```



8.2.5 Distribusi Gamma

Distribusi Gamma adalah distribusi probabilitas kontinu yang sering digunakan untuk memodelkan waktu tunggu atau durasi dalam proses yang melibatkan beberapa kejadian yang terjadi secara berturut-turut. Distribusi ini sering digunakan dalam analisis risiko, waktu tunggu, dan banyak aplikasi lainnya yang melibatkan waktu atau durasi.

Ciri-Ciri Distribusi Gamma

- **Tipe Distribusi Kontinu**: Distribusi ini digunakan untuk variabel kontinu, khususnya untuk waktu atau durasi.
- **Parameter**: Distribusi Gamma memiliki dua parameter utama:
  - **Shape parameter (α)**: Menentukan bentuk distribusi.
  - **Rate parameter (β)** atau **Scale parameter (θ)**: Menentukan lebar distribusi.
- **Generalization of Exponential Distribution**: Jika $\alpha = 1$, maka distribusi Gamma menjadi distribusi eksponensial.

Fungsi Kepadatan Probabilitas

Fungsi kepadatan probabilitas (PDF) distribusi Gamma diberikan oleh rumus:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1} \exp(-x/\beta)}{\beta^{\alpha} \Gamma(\alpha)}, \quad x \geq 0$$

Di mana:

- x: Variabel acak.
- α: Parameter shape (bentuk).
- β: Parameter scale (skala).
- $\Gamma(\alpha)$: Fungsi Gamma, yang merupakan generalisasi dari faktorial.

Contoh Distribusi Gamma

Misalkan waktu yang dibutuhkan untuk 3 kejadian dalam suatu proses Poisson mengikuti distribusi Gamma dengan parameter $\alpha = 3$ dan $\beta = 2$. Kita ingin menghitung probabilitas bahwa waktu total kejadian tersebut lebih kecil dari 5.

Probabilitas ini dihitung dengan menggunakan fungsi distribusi kumulatif (CDF) dari distribusi Gamma:

$$P(X \leq 5) = F(5) = \int_0^5 \frac{x^{\alpha-1} \exp(-x/\beta)}{\beta^\alpha \Gamma(\alpha)} dx$$

Substitusi nilai-nilai yang diketahui:

- $\alpha = 3$
- $\beta = 2$
- $x = 5$

Dengan menggunakan perangkat lunak atau tabel distribusi Gamma, kita dapat menghitung nilai $F(5)$.

Implementasi di R:

```
# Parameter distribusi Gamma
alpha <- 3  # Shape parameter
beta <- 2   # Scale parameter
x <- 5      # Waktu yang dihitung (5)

# Probabilitas waktu total lebih kecil dari 5
prob <- pgamma(x, shape = alpha, scale = beta)
prob
```

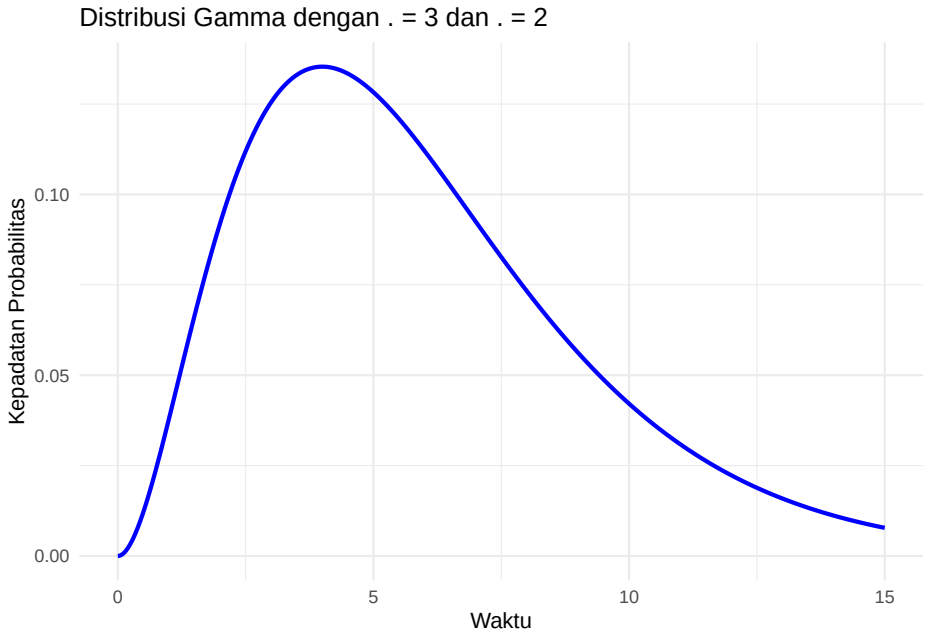
```
## [1] 0.4561869
```

Visualisasi Distribusi Gamma

Untuk memvisualisasikan distribusi Gamma, kita dapat membuat grafik kepadatan probabilitasnya.

```
# Memuat library
library(ggplot2)

# Membuat data untuk distribusi Gamma
```

```r
x_values <- seq(0, 15, length.out = 1000)
y_values <- dgamma(x_values, shape = alpha, scale = beta)

# Membuat plot
ggplot(data.frame(x = x_values, y = y_values), aes(x = x, y = y)) +
  geom_line(color = "blue", linewidth = 1) +  # Mengganti size dengan linewidth
  ggtitle("Distribusi Gamma dengan   = 3 dan   = 2") +
  xlab("Waktu") +
  ylab("Kepadatan Probabilitas") +
  theme_minimal()
```



8.2.6 Distribusi Chi-Square

Distribusi Chi-Square adalah distribusi probabilitas yang digunakan untuk menguji hipotesis tentang varians dari suatu populasi atau untuk menguji kesesuaian antara frekuensi yang diamati dengan yang diharapkan. Distribusi ini adalah kasus khusus dari distribusi Gamma, dan sering digunakan dalam uji statistik seperti uji chi-square untuk independensi atau kecocokan.

Ciri-Ciri Distribusi Chi-Square

- **Distribusi Kontinu**: Distribusi Chi-Square adalah distribusi kontinu yang hanya bernilai positif.
- **Derajat Kebebasan (Degrees of Freedom, df)**: Distribusi Chi-Square tergantung pada parameter derajat kebebasan, yang biasanya

digunakan untuk menggambarkan jumlah variabel bebas dalam suatu sampel atau eksperimen.
- **Tidak Simetris**: Distribusi Chi-Square memiliki bentuk yang condong ke kanan, dan semakin banyak derajat kebebasan, semakin mendekati bentuk distribusi normal.

Fungsi Kepadatan Probabilitas

Fungsi kepadatan probabilitas (PDF) distribusi Chi-Square dengan k derajat kebebasan adalah:

$$f(x;k) = \frac{x^{(k/2)-1}\exp(-x/2)}{2^{k/2}\Gamma(k/2)}, \quad x \geq 0$$

Di mana:

- x: Variabel acak.
- k: Derajat kebebasan.
- $\Gamma(k/2)$: Fungsi Gamma, yang digunakan untuk normalisasi.

Contoh Distribusi Chi-Square

Misalkan kita memiliki sampel yang terdiri dari 10 pengukuran, dan kita ingin menguji apakah varians sampel tersebut sesuai dengan nilai yang diharapkan. Untuk menguji hipotesis tersebut, kita menggunakan distribusi Chi-Square dengan 9 derajat kebebasan ($df = n - 1$), di mana n adalah ukuran sampel.

Misalnya, kita ingin menghitung probabilitas bahwa nilai Chi-Square yang dihitung adalah kurang dari 15 dengan 9 derajat kebebasan.

Probabilitas ini dihitung dengan menggunakan fungsi distribusi kumulatif (CDF) dari distribusi Chi-Square:

$$P(X \leq 15) = F(15;9)$$

Implementasi di R:

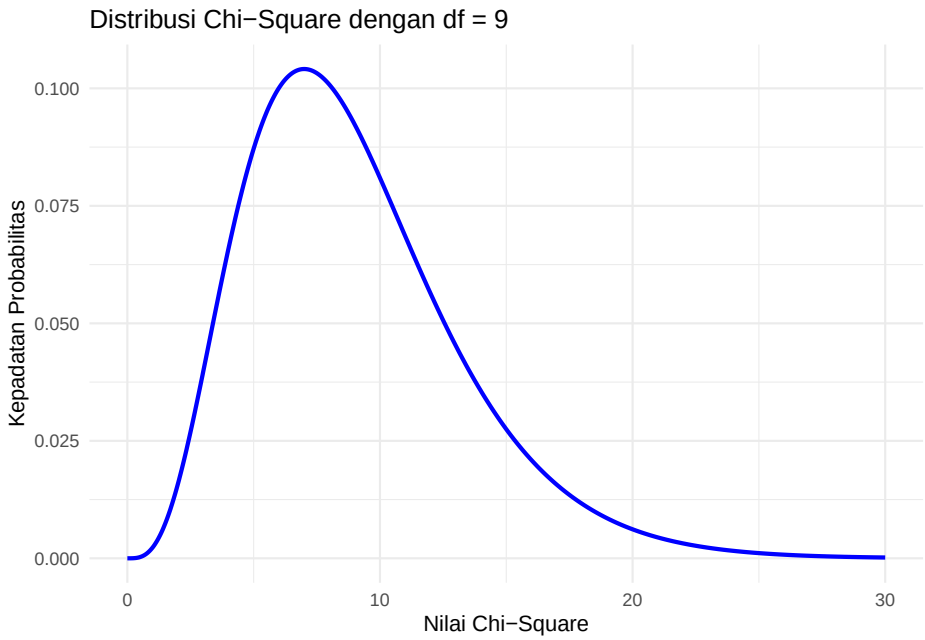
```
# Parameter distribusi Chi-Square
df <- 9  # Derajat kebebasan
x <- 15  # Nilai yang dihitung

# Probabilitas nilai Chi-Square lebih kecil dari 15 dengan 9 derajat kebebasan
prob <- pchisq(x, df)
prob
```

```
## [1] 0.909064
```

Visualisasi Distribusi Chi-Square

Untuk memvisualisasikan distribusi Chi-Square, kita dapat membuat grafik kepadatan probabilitasnya.

```
# Memuat library
library(ggplot2)

# Membuat data untuk distribusi Chi-Square
x_values <- seq(0, 30, length.out = 1000)
y_values <- dchisq(x_values, df = df)

# Membuat plot
ggplot(data.frame(x = x_values, y = y_values), aes(x = x, y = y)) +
  geom_line(color = "blue", linewidth = 1) +  # Mengganti size dengan linewidth
  ggtitle("Distribusi Chi-Square dengan df = 9") +
  xlab("Nilai Chi-Square") +
  ylab("Kepadatan Probabilitas") +
  theme_minimal()
```



Penggunaan Distribusi Chi-Square

Distribusi Chi-Square sering digunakan dalam berbagai uji statistik, di antaranya:

- Uji Kesesuaian (Goodness of Fit Test): Menguji apakah distribusi sampel sesuai dengan distribusi teoritis.

- Uji Independen: Digunakan untuk menguji apakah dua variabel kategorikal independen satu sama lain dalam uji tabel kontingensi.
- Uji Homogenitas: Digunakan untuk menguji apakah distribusi dari satu variabel kategorikal sama antara beberapa kelompok.

Dengan memahami distribusi Chi-Square, kita dapat lebih mudah melakukan analisis statistik, seperti menguji hipotesis tentang varians dan independensi antar variabel.

8.2.7 Distribusi t-Student

Distribusi t-Student adalah distribusi probabilitas yang digunakan untuk mengestimasi rata-rata populasi ketika ukuran sampel kecil dan varians populasi tidak diketahui. Distribusi ini sering digunakan dalam uji hipotesis, terutama dalam uji t untuk sampel kecil.

Ciri-Ciri Distribusi t-Student

- **Bentuk**: Distribusi t-Student mirip dengan distribusi normal, namun lebih lebar di bagian ekor, yang mencerminkan variabilitas lebih besar pada sampel kecil.
- **Parameter**: Distribusi t-Student hanya memiliki satu parameter, yaitu **derajat kebebasan (df)**, yang terkait dengan ukuran sampel. Semakin besar derajat kebebasan, distribusi t-Student semakin mendekati distribusi normal.
- **Puncak**: Memiliki puncak yang lebih tinggi dan lebar dibandingkan dengan distribusi normal.
- **Penggunaan**: Digunakan ketika ukuran sampel kecil (biasanya n < 30) dan populasi memiliki distribusi normal atau mendekati normal.

Fungsi Kepadatan Probabilitas (PDF)

Fungsi kepadatan probabilitas distribusi t-Student dengan derajat kebebasan ν diberikan oleh:

$$f(x;\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \quad x \in (-\infty,\infty)$$

Di mana:

- $\Gamma(\cdot)$ adalah fungsi Gamma, yang memperluas fungsi faktorial ke bilangan real.
- ν adalah derajat kebebasan (df).

Contoh Distribusi t-Student

Misalkan kita ingin menguji apakah rata-rata hasil ujian suatu kelompok siswa berbeda dari nilai 75. Kita menggunakan sampel dengan ukuran $n = 10$ dan rata-rata sampel $\bar{x} = 72$ dengan deviasi standar sampel $s = 8$.

Untuk uji hipotesis, kita dapat menggunakan distribusi t-Student dengan derajat kebebasan $\nu = n - 1 = 9$.

Menghitung Probabilitas t-Student

Misalkan kita ingin menghitung probabilitas bahwa nilai t lebih kecil dari 2.26 untuk derajat kebebasan $\nu = 9$:

$$P(T \leq 2.26) = F(2.26; 9)$$

Implementasi di R:

```
# Derajat kebebasan
df <- 9  # Derajat kebebasan

# Nilai t yang dihitung
t_value <- 2.26

# Probabilitas distribusi t-Student
prob <- pt(t_value, df)
prob
```

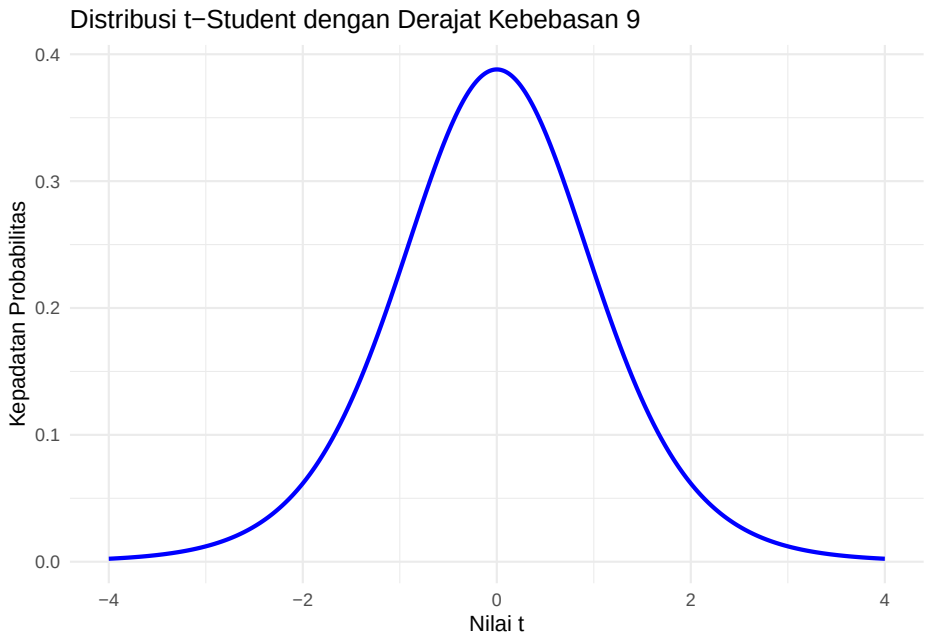
```
## [1] 0.9749117
```

Visualisasi Distribusi t-Student

Untuk memvisualisasikan distribusi t-Student dengan derajat kebebasan 9, kita dapat menggunakan grafik berikut:

```
# Memuat library ggplot2 untuk visualisasi
library(ggplot2)

# Membuat data untuk distribusi t-Student
x_values <- seq(-4, 4, length.out = 1000)
y_values <- dt(x_values, df)

# Membuat plot distribusi t-Student
ggplot(data.frame(x = x_values, y = y_values), aes(x = x, y = y)) +
  geom_line(color = "blue", size = 1) +
  ggtitle("Distribusi t-Student dengan Derajat Kebebasan 9") +
  xlab("Nilai t") +
```

```
  ylab("Kepadatan Probabilitas") +
  theme_minimal()
```



Distribusi t-Student adalah distribusi yang penting untuk digunakan dalam uji hipotesis, terutama ketika kita bekerja dengan sampel kecil dan varians populasi tidak diketahui. Distribusi ini lebih lebar di bagian ekor dibandingkan distribusi normal, yang mencerminkan ketidakpastian yang lebih besar dalam estimasi ketika ukuran sampel kecil. Seiring bertambahnya ukuran sampel, distribusi t-Student mendekati distribusi normal.

8.2.8 Distribusi Weibull

Distribusi Weibull adalah distribusi probabilitas yang digunakan untuk memodelkan data durasi hidup atau waktu kegagalan, dan sering digunakan dalam analisis kegagalan dan reliabilitas. Distribusi ini dapat digunakan untuk menggambarkan waktu hingga kegagalan sistem atau komponen dalam berbagai bidang, seperti rekayasa, pengolahan, dan analisis statistik.

Ciri-Ciri Distribusi Weibull

- **Fleksibilitas**: Distribusi Weibull memiliki dua parameter, yaitu **skala** $ $ dan **bentuk** k, yang memungkinkan distribusi ini menyesuaikan diri dengan berbagai jenis data.
  - $k = 1$: Distribusi eksponensial (bila waktu kegagalan tidak bergantung pada waktu yang telah berlalu).

  - $k > 1$: Distribusi yang menggambarkan proses kegagalan yang semakin lambat seiring waktu (biasa digunakan dalam model keandalan).
  - $k < 1$: Distribusi yang menggambarkan proses kegagalan yang lebih cepat seiring waktu.
- **Parameter**:
  - **Skala** λ: Menentukan skala atau rentang waktu kejadian.
  - **Bentuk** k: Menentukan bentuk distribusi dan mempengaruhi seberapa cepat atau lambatnya kegagalan terjadi.

Fungsi Kepadatan Probabilitas (PDF)

Fungsi kepadatan probabilitas (PDF) dari distribusi Weibull diberikan oleh:

$$f(x; k, \lambda) = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, \quad x \geq 0$$

Di mana:

- k: Parameter bentuk (shape).
- λ: Parameter skala (scale).
- x: Waktu atau durasi kegagalan.

Fungsi Distribusi Kumulatif (CDF)

Fungsi distribusi kumulatif (CDF) dari distribusi Weibull adalah:

$$F(x; k, \lambda) = 1 - e^{-(x/\lambda)^k}, \quad x \geq 0$$

Contoh Distribusi Weibull

Misalkan kita memiliki data waktu kegagalan mesin dengan distribusi Weibull, di mana $k = 1.5$ (parameter bentuk) dan $\lambda = 3$ (parameter skala). Kita ingin menghitung probabilitas bahwa waktu kegagalan mesin berada di bawah 2 jam.

Untuk menghitung probabilitas ini, kita menggunakan fungsi distribusi kumulatif (CDF) dari distribusi Weibull:

$$P(X \leq 2) = F(2; 1.5, 3)$$

Implementasi di R:

```
# Parameter distribusi Weibull
shape <- 1.5  # Parameter bentuk (k)
scale <- 3    # Parameter skala ( )
x <- 2        # Waktu yang dihitung (2 jam)
```

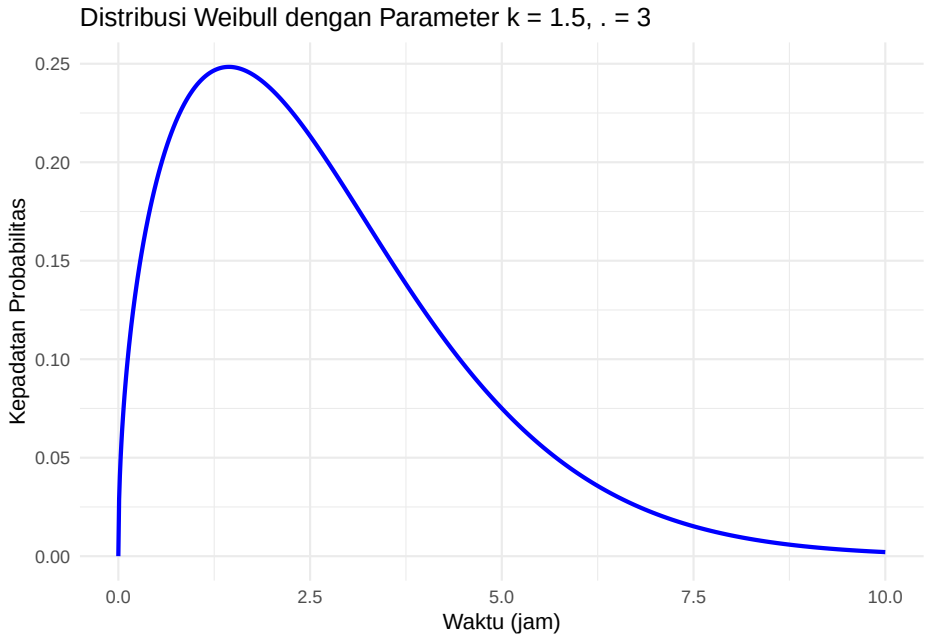
```r
# Menghitung probabilitas
prob <- pweibull(x, shape, scale)
prob
```

```
## [1] 0.4197702
```

Visualisasi Distribusi Weibull

Untuk memvisualisasikan distribusi Weibull dengan parameter $k = 1.5$ dan $\lambda = 3$, kita dapat membuat grafik distribusi sebagai berikut:

```r
# Memuat library ggplot2 untuk visualisasi
library(ggplot2)

# Membuat data untuk distribusi Weibull
x_values <- seq(0, 10, length.out = 1000)
y_values <- dweibull(x_values, shape, scale)

# Membuat plot distribusi Weibull
ggplot(data.frame(x = x_values, y = y_values), aes(x = x, y = y)) +
  geom_line(color = "blue", size = 1) +
  ggtitle("Distribusi Weibull dengan Parameter k = 1.5,   = 3") +
  xlab("Waktu (jam)") +
  ylab("Kepadatan Probabilitas") +
  theme_minimal()
```



Distribusi Weibull sangat berguna dalam analisis reliabilitas dan durasi hidup. Fleksibilitasnya dalam parameter bentuk memungkinkan distribusi ini untuk memodelkan berbagai jenis data kegagalan, baik yang memiliki laju kegagalan konstan, semakin lambat, atau semakin cepat seiring waktu. Distribusi ini sering digunakan dalam bidang pengolahan, rekayasa, dan penelitian ketahanan material.

8.2.9 Distribusi Log-Normal

Distribusi Log-Normal adalah distribusi probabilitas yang digunakan untuk memodelkan variabel acak yang nilainya terdistribusi secara logaritmik. Jika suatu variabel acak X terdistribusi log-normal, maka logaritma dari X terdistribusi normal. Distribusi ini sering digunakan untuk memodelkan data yang tidak terdistribusi normal tetapi memiliki nilai yang positif dan cenderung berskala lebih besar.

Ciri-Ciri Distribusi Log-Normal

- **Distribusi Positif**: Semua nilai dalam distribusi log-normal adalah positif ($X > 0$).
- **Transformasi Logaritmik**: Jika $X \sim \text{Log-Normal}(\mu, \sigma^2)$, maka $Y = \ln(X) \sim \text{Normal}(\mu, \sigma^2)$, dengan μ dan σ adalah parameter dari distribusi normal yang mendasari (mean dari logaritma X),
- **Penggunaan untuk Data Skala Besar**: Distribusi log-normal sering digunakan untuk data yang memiliki rentang nilai yang luas, seperti pengukuran harga saham, pendapatan, atau waktu hidup perangkat.

Fungsi Kepadatan Probabilitas (PDF)

Fungsi kepadatan probabilitas dari distribusi log-normal adalah:

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right), \quad x > 0$$

Di mana:

- μ adalah parameter rata-rata dari distribusi normal yang mendasari (mean dari logaritma X),
- σ adalah parameter standar deviasi dari distribusi normal yang mendasari (standard deviation dari logaritma X),
- x adalah variabel acak yang terdistribusi log-normal.

Fungsi Distribusi Kumulatif (CDF)

Fungsi distribusi kumulatif (CDF) dari distribusi log-normal adalah:

$$F(x;\mu,\sigma) = P(X \leq x) = \Phi\left(\frac{\ln(x) - \mu}{\sigma}\right)$$

Di mana Φ adalah fungsi distribusi kumulatif dari distribusi normal standar (mean = 0, standar deviasi = 1).

Contoh Distribusi Log-Normal

Misalkan pendapatan tahunan seseorang mengikuti distribusi log-normal dengan parameter $\mu = 3$ dan $\sigma = 1.5$. Kita ingin menghitung probabilitas bahwa pendapatan tahunan X berada dalam rentang $[1000, 10000]$.

Probabilitas ini dapat dihitung dengan menggunakan fungsi distribusi kumulatif (CDF) dari distribusi log-normal:

$$P(1000 \leq X \leq 10000) = F(10000) - F(1000)$$

Implementasi di R:

```
# Parameter distribusi Log-Normal
mu <- 3      # Parameter rata-rata logaritma
sigma <- 1.5 # Parameter standar deviasi logaritma
x1 <- 1000   # Batas bawah rentang
x2 <- 10000  # Batas atas rentang

# Menghitung probabilitas bahwa X berada dalam rentang [1000, 10000]
probabilitas <- plnorm(x2, mu, sigma) - plnorm(x1, mu, sigma)
probabilitas
```

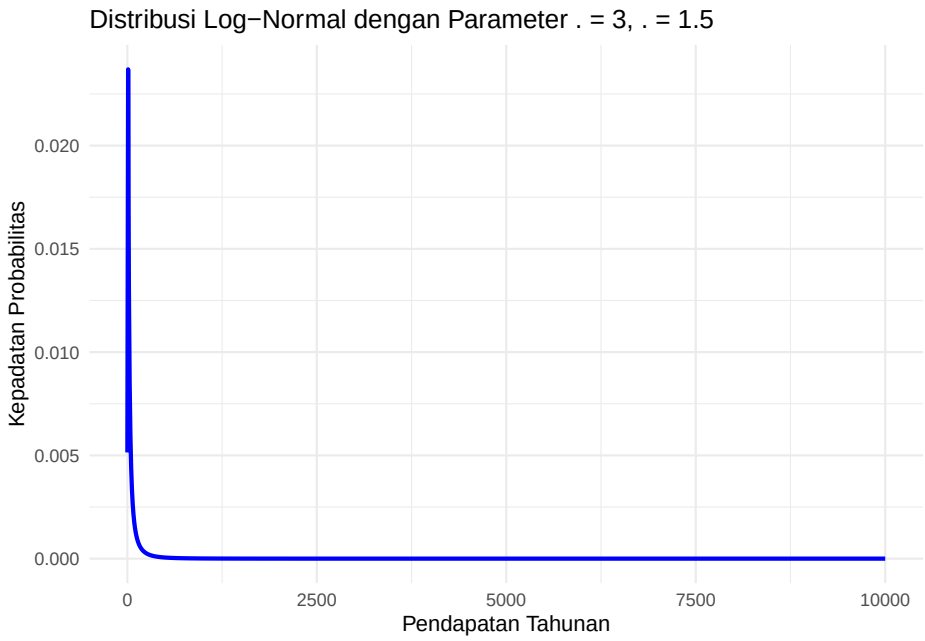
```
## [1] 0.004574084
```

Visualisasi Distribusi Log-Normal

Untuk memvisualisasikan distribusi log-normal, kita dapat membuat grafik dari fungsi kepadatan probabilitas (PDF) untuk parameter $\mu = 3$ dan $\sigma = 1.5$:

```
library(ggplot2)

# Membuat data untuk distribusi log-normal
x_values <- seq(0.1, 10000, length.out = 1000)
y_values <- dlnorm(x_values, mu, sigma)

# Membuat plot distribusi log-normal
ggplot(data.frame(x = x_values, y = y_values), aes(x = x, y = y)) +
  geom_line(color = "blue", size = 1) +
  ggtitle("Distribusi Log-Normal dengan Parameter   = 3,   = 1.5") +
  xlab("Pendapatan Tahunan") +
```

```
  ylab("Kepadatan Probabilitas") +
  theme_minimal()
```



Distribusi log-normal adalah distribusi yang sangat berguna untuk memodelkan data yang terdistribusi secara positif dan memiliki variabilitas yang besar. Distribusi ini banyak digunakan dalam bidang keuangan, ekologi, dan ilmu sosial untuk memodelkan data seperti pendapatan, harga saham, dan durasi hidup sistem atau perangkat. Keunggulan dari distribusi log-normal adalah kemampuannya untuk menggambarkan fenomena yang memiliki distribusi dengan "skewed" atau penyimpangan yang tinggi ke nilai yang lebih besar.

8.2.10 Distribusi Cauchy

Distribusi Cauchy adalah distribusi probabilitas yang memiliki sifat distribusi yang sangat lebar dan "heavy-tailed". Distribusi ini sering digunakan untuk memodelkan data yang memiliki nilai ekstrim atau outliers yang sangat signifikan, yang tidak dapat dijelaskan dengan distribusi normal.

Distribusi Cauchy dikenal dengan sifat-sifatnya yang tidak memiliki momen (mean dan variance) yang terdefinisi dengan baik. Meskipun demikian, distribusi Cauchy banyak digunakan dalam berbagai bidang, seperti dalam statistik robust dan fisika.

Ciri-Ciri Distribusi Cauchy

- **Heavy-Tailed**: Distribusi Cauchy memiliki ekor yang sangat lebar, sehingga lebih banyak nilai ekstrem (outliers) yang mungkin terjadi dibandingkan distribusi normal.
- **Tidak Memiliki Momen**: Distribusi Cauchy tidak memiliki rata-rata (μ) atau varians (σ^2) yang terdefinisi, karena integral untuk momen pertama dan kedua tidak konvergen.
- **Simetri**: Distribusi Cauchy adalah distribusi simetris, seperti distribusi normal, namun dengan ekor yang lebih lebar.

Fungsi Kepadatan Probabilitas (PDF)

Fungsi kepadatan probabilitas untuk distribusi Cauchy diberikan oleh:

$$f(x; x_0, \gamma) = \frac{1}{\pi\gamma \left(1 + \left(\frac{x - x_0}{\gamma}\right)^2\right)}$$

Di mana:

- x_0 adalah lokasi (atau parameter lokasi) dari distribusi Cauchy (biasanya ini adalah nilai puncak distribusi),
- γ adalah parameter skala yang mengontrol lebar distribusi (lebih besar γ, distribusi lebih lebar),
- x adalah variabel acak yang terdistribusi Cauchy.

Fungsi Distribusi Kumulatif (CDF)

Fungsi distribusi kumulatif (CDF) dari distribusi Cauchy diberikan oleh:

$$F(x; x_0, \gamma) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x - x_0}{\gamma}\right)$$

Contoh Distribusi Cauchy

Misalkan kita ingin memodelkan distribusi data yang sangat "heavy-tailed" dengan parameter $ x_0 = 0 $ dan $ = 1 $. Kita ingin menghitung probabilitas bahwa variabel acak $ X $ berada dalam interval $ [-2, 2] $.

Probabilitas ini dihitung menggunakan fungsi distribusi kumulatif (CDF) dari distribusi Cauchy:

$$P(-2 \leq X \leq 2) = F(2) - F(-2)$$

Substitusikan nilai-nilai tersebut ke dalam CDF:

$$F(2) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{2-0}{1}\right) = \frac{1}{2} + \frac{1}{\pi} \arctan(2)$$

$$F(-2) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{-2-0}{1}\right) = \frac{1}{2} + \frac{1}{\pi} \arctan(-2)$$

Hasilnya adalah:

$$P(-2 \leq X \leq 2) = F(2) - F(-2) = \left(\frac{1}{2} + \frac{1}{\pi} \arctan(2)\right) - \left(\frac{1}{2} + \frac{1}{\pi} \arctan(-2)\right)$$

Dengan perhitungan numerik:

$$P(-2 \leq X \leq 2) \approx 0.732$$

Implementasi di R:

```
# Parameter distribusi Cauchy
x0 <- 0  # Parameter lokasi
gamma <- 1  # Parameter skala
x1 <- -2   # Batas bawah rentang
x2 <- 2    # Batas atas rentang

# Menghitung probabilitas bahwa X berada dalam rentang [-2, 2]
probabilitas <- pcauchy(x2, location = x0, scale = gamma) - pcauchy(x1, location = x0,
probabilitas
```

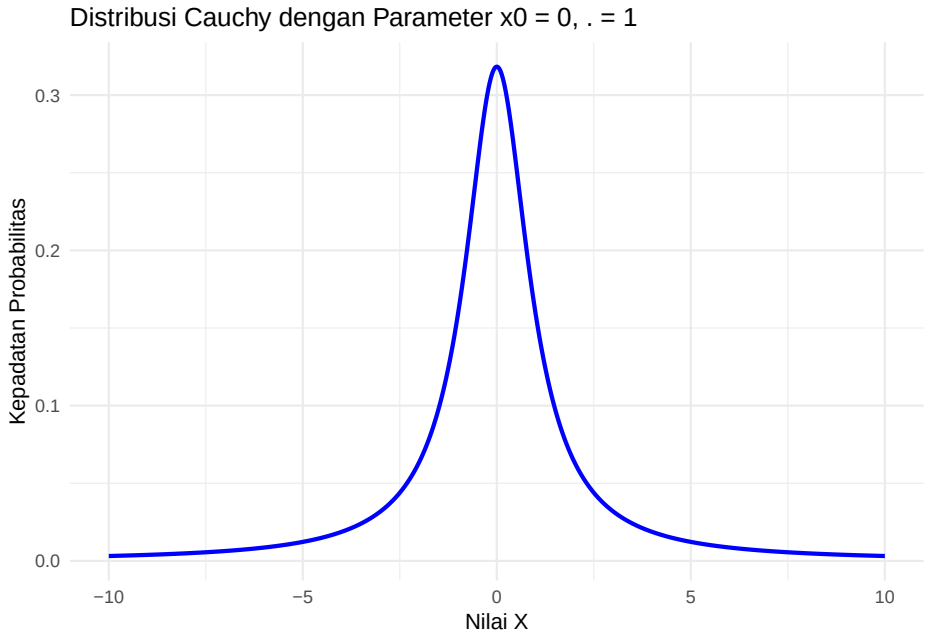
```
## [1] 0.7048328
```

Visualisasi Distribusi Cauchy

Untuk memvisualisasikan distribusi Cauchy, kita dapat membuat grafik dari fungsi kepadatan probabilitas (PDF) untuk parameter $x_0 = 0$ dan $\gamma = 1$:

```
# Memuat library ggplot2 untuk visualisasi
library(ggplot2)

# Membuat data untuk distribusi Cauchy
x_values <- seq(-10, 10, length.out = 1000)
y_values <- dcauchy(x_values, location = x0, scale = gamma)

# Membuat plot distribusi Cauchy
ggplot(data.frame(x = x_values, y = y_values), aes(x = x, y = y)) +
  geom_line(color = "blue", size = 1) +
  ggtitle("Distribusi Cauchy dengan Parameter x0 = 0,   = 1") +
```

```
  xlab("Nilai X") +
  ylab("Kepadatan Probabilitas") +
  theme_minimal()
```



Distribusi Cauchy adalah distribusi dengan ekor yang sangat lebar dan sering digunakan untuk memodelkan data yang memiliki nilai ekstrim atau outliers yang signifikan. Salah satu ciri utama dari distribusi ini adalah tidak adanya momen yang terdefinisi, sehingga rata-rata dan variansinya tidak ada. Meskipun begitu, distribusi Cauchy sangat berguna dalam statistik robust dan dalam memodelkan fenomena dengan data yang sangat tersebar.

8.3 Terapan Distribusi Probabilitas

Tabel berikut menunjukkan berbagai distribusi probabilitas, penerapan utamanya, contoh aplikasi, dan contoh perhitungannya:

Distribusi	Penerapan Utama	Contoh	Contoh Perhitungan
Normal	Psikologi, ekonomi, teknik: Memodelkan data yang terdistribusi secara simetris.	Menghitung probabilitas tinggi badan dalam populasi tertentu.	$P(X \leq 170)$ untuk $\mu = 165$, $\sigma = 10$: $\Phi((170 - 165)/10) = \Phi(0.5) \approx 0.6915$.

Distribusi	Penerapan Utama	Contoh	Contoh Perhitungan
Eksponensial	Teknik, pemeliharaan: Memodelkan waktu antar kegagalan atau kejadian dalam proses Poisson.	Memodelkan waktu tunggu kereta di stasiun.	$P(X \leq 2)$ untuk $\lambda = 0.5$: $1 - e^{-0.5 \cdot 2} = 1 - e^{-1} \approx 0.6321$.
Poisson	Antrian, analisis kejadian langka: Memodelkan jumlah kejadian dalam interval waktu tertentu.	Menghitung jumlah kendaraan yang melewati tol dalam 1 menit.	$P(X = 3)$ untuk $\lambda = 2$: $\frac{2^3 e^{-2}}{3!} = \frac{8 \cdot 0.1353}{6} \approx 0.1804$.
Binomial	Statistik, biologi: Memodelkan hasil eksperimen dengan dua kemungkinan hasil (sukses/gagal).	Menghitung probabilitas mendapatkan 5 kepala dalam 10 lemparan koin.	$P(X = 5)$ untuk $n = 10$, $p = 0.5$: $\binom{10}{5}(0.5)^5(0.5)^5 = 0.2461$.
Chi-Square	Statistik inferensial: Memodelkan distribusi varians sampel, digunakan dalam uji kesesuaian dan independensi.	Menguji apakah distribusi pengunjung toko sesuai dengan harapan.	$P(X \leq 15)$ untuk $df = 9$: $F(15; 9) \approx 0.901$.
Student-t	Statistik inferensial: Membandingkan rata-rata dua kelompok, terutama jika ukuran sampel kecil.	Menguji apakah rata-rata skor tes siswa berbeda antara dua kelas.	$P(T \leq 2)$ untuk $df = 10$: $F(2; 10) \approx 0.963$.
Gamma	Teknik, ekonofisika: Memodelkan waktu hingga sejumlah kejadian terjadi.	Memodelkan waktu yang dibutuhkan untuk menyelesaikan proyek tertentu.	$P(X \leq 3)$ untuk $\alpha = 2$, $\beta = 1$: $\int_0^3 \frac{x^{2-1} e^{-x}}{\Gamma(2)} dx \approx 0.800$.
Beta	Statistik Bayesian: Memodelkan distribusi probabilitas untuk proporsi.	Memodelkan kemungkinan keberhasilan kampanye pemasaran.	$P(0.2 \leq X \leq 0.8)$ untuk $\alpha = 2$, $\beta = 5$: $I_{0.8}(2, 5) - I_{0.2}(2, 5) \approx 0.654$.
Weibull	Pemeliharaan, keandalan: Memodelkan umur atau waktu sampai kegagalan.	Memodelkan waktu kegagalan komponen mesin.	$P(X \leq 2)$ untuk $\lambda = 1$, $k = 2$: $1 - e^{-(2/1)^2} = 1 - e^{-4} \approx 0.9817$.

Distribusi	Penerapan Utama	Contoh	Contoh Perhitungan
Log-Normal	Ekonomi, biologi: Memodelkan distribusi variabel positif dengan penyebaran asimetris.	Memodelkan harga saham atau distribusi waktu penyembuhan pasien.	$P(X \leq 5)$ untuk $\mu = 1$, $\sigma = 0.5$: $F(\ln(5); 1, 0.5) \approx 0.841$.
Cauchy	Fisika, optik: Memodelkan distribusi data dengan ekor tebal atau outlier yang signifikan.	Menganalisis distribusi lintasan cahaya di sekitar sumber energi besar.	$P(X \leq 1)$ untuk $x_0 = 0$, $\gamma = 1$: $\frac{1}{\pi} \arctan((1-0)/1) + \frac{1}{2} \approx 0.75$.

8.4 Jenis Metode Sampling

8.4.1 Metode Sampling Acak

Sampling acak adalah metode pengambilan sampel di mana setiap elemen dalam populasi memiliki peluang yang sama untuk dipilih. Metode ini dianggap sebagai metode yang paling adil karena tidak ada bias dalam proses pemilihan.

Misalkan Anda memiliki populasi sebanyak 100 orang, dan Anda ingin memilih 10 orang secara acak. Anda dapat menggunakan fungsi berikut di R untuk melakukan sampling acak:

```
# Populasi
populasi <- 1:100

# Mengambil sampel acak sebanyak 10 orang
set.seed(123) # Untuk reproduksi hasil
sampel_acak <- sample(populasi, 10)
sampel_acak
```

```
##  [1] 31 79 51 14 67 42 50 43 97 25
```

8.4.2 Metode Sampling Berstrata

Sampling berstrata melibatkan pembagian populasi menjadi beberapa kelompok atau strata berdasarkan karakteristik tertentu, seperti usia, jenis kelamin, atau lokasi geografis. Sampel kemudian diambil dari setiap strata.

Misalkan populasi dibagi menjadi tiga strata berdasarkan usia: anak-anak, dewasa, dan lansia. Anda ingin mengambil 5 sampel dari masing-masing strata:

```r
# Populasi dengan strata
populasi <- data.frame(
  ID = 1:30,
  Usia = c(rep("Anak", 10), rep("Dewasa", 10), rep("Lansia", 10))
)

# Mengambil 5 sampel dari setiap strata
library(dplyr)

set.seed(123)
sampel_strata <- populasi %>%
  group_by(Usia) %>%
  sample_n(5)
sampel_strata
```

```
## # A tibble: 15 x 2
## # Groups:   Usia [3]
##       ID Usia
##    <int> <chr>
##  1     3 Anak
##  2    10 Anak
##  3     2 Anak
##  4     8 Anak
##  5     6 Anak
##  6    15 Dewasa
##  7    14 Dewasa
##  8    16 Dewasa
##  9    18 Dewasa
## 10    11 Dewasa
## 11    30 Lansia
## 12    25 Lansia
## 13    23 Lansia
## 14    28 Lansia
## 15    21 Lansia
```

8.4.3 Metode Sampling Klaster

Sampling klaster melibatkan pembagian populasi menjadi kelompok atau klaster berdasarkan lokasi geografis atau kelompok alami lainnya. Selanjutnya, seluruh klaster dipilih secara acak, dan semua elemen dalam klaster yang dipilih dijadikan sampel.

Misalkan populasi dibagi menjadi 5 klaster berdasarkan wilayah geografis, dan Anda ingin memilih 2 klaster secara acak:

```r
# Populasi dengan klaster
populasi <- data.frame(
  ID = 1:50,
  Klaster = rep(1:5, each = 10)
)

# Memilih 2 klaster secara acak
set.seed(123)
klaster_terpilih <- sample(unique(populasi$Klaster), 2)

# Mengambil semua elemen dari klaster yang terpilih
sampel_klaster <- populasi %>%
  filter(Klaster %in% klaster_terpilih)
sampel_klaster
```

```
##    ID Klaster
## 1  11       2
## 2  12       2
## 3  13       2
## 4  14       2
## 5  15       2
## 6  16       2
## 7  17       2
## 8  18       2
## 9  19       2
## 10 20       2
## 11 21       3
## 12 22       3
## 13 23       3
## 14 24       3
## 15 25       3
## 16 26       3
## 17 27       3
## 18 28       3
## 19 29       3
## 20 30       3
```

8.5 Distribusi Sampling dari Rata-rata Sampel

Distribusi sampling dari rata-rata sampel menggambarkan distribusi dari rata-rata sampel yang diambil dari populasi tertentu. Distribusi ini penting dalam statistik karena membantu kita memahami bagaimana rata-rata sampel berperilaku dalam hubungannya dengan parameter populasi.

8.5.1 Karakteristik Utama:

- **Rata-rata**: Rata-rata distribusi sampling ($\mu_{\bar{X}}$) sama dengan rata-rata populasi (μ):

$$\mu_{\bar{X}} = \mu$$

- **Simpangan baku**: Simpangan baku distribusi sampling ($\sigma_{\bar{X}}$) dikenal sebagai *standard error* dan dihitung sebagai:

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$$

  Di mana σ adalah simpangan baku populasi dan n adalah ukuran sampel.

- **Bentuk**: Jika ukuran sampel cukup besar, distribusi rata-rata sampel mendekati distribusi normal, terlepas dari bentuk distribusi populasi.

8.5.2 Teorema Limit Tengah

Teorema Limit Tengah (Central Limit Theorem, CLT) menyatakan bahwa:

> Untuk ukuran sampel yang cukup besar ($n \geq 30$), distribusi rata-rata sampel dari populasi apapun akan mendekati distribusi normal, dengan rata-rata μ dan simpangan baku $\sigma/\sqrt{n}$.

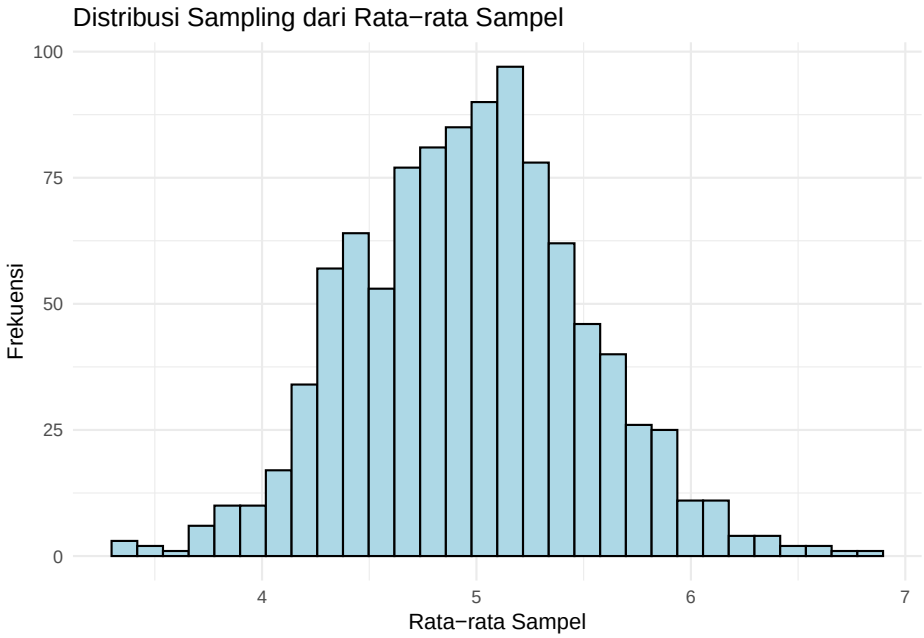
Teorema ini sangat penting karena memungkinkan kita menggunakan pendekatan normal untuk analisis statistik bahkan jika populasi awal tidak berdistribusi normal.

8.5.3 Ilustrasi dalam R

Misalkan kita memiliki populasi berdistribusi uniform dan ingin memeriksa bagaimana rata-rata sampel berperilaku saat ukuran sampel meningkat.

```
# Populasi berdistribusi uniform
set.seed(123)
populasi <- runif(10000, min = 0, max = 10)

# Ukuran sampel dan jumlah simulasi
n <- 30  # Ukuran sampel
simulasi <- 1000  # Jumlah sampel

# Mengambil rata-rata sampel
rata_rata_sampel <- replicate(simulasi, mean(sample(populasi, n, replace = TRUE)))

# Plot distribusi rata-rata sampel
library(ggplot2)
ggplot(data.frame(rata_rata_sampel), aes(x = rata_rata_sampel)) +
  geom_histogram(bins = 30, color = "black", fill = "lightblue") +
```

```
  ggtitle("Distribusi Sampling dari Rata-rata Sampel") +
  xlab("Rata-rata Sampel") +
  ylab("Frekuensi") +
  theme_minimal()
```



8.6 Perhitungan Probabilitas Menggunakan Teorema Limit Tengah

Teorema Limit Tengah memungkinkan kita untuk menghitung probabilitas rata-rata sampel berada dalam interval tertentu. Berikut adalah contoh perhitungannya:

Misalkan kita memiliki populasi dengan rata-rata $\mu = 50$, simpangan baku $\sigma = 10$, dan ukuran sampel $n = 25$. Probabilitas rata-rata sampel berada dalam interval $[48, 52]$ dapat dihitung sebagai berikut:

```
# Parameter
mu <- 50  # Rata-rata populasi
sigma <- 10  # Simpangan baku populasi
n <- 25  # Ukuran sampel
sigma_xbar <- sigma / sqrt(n)  # Standard error

# Probabilitas
lower <- 48
```

```
upper <- 52
prob <- pnorm(upper, mean = mu, sd = sigma_xbar) - pnorm(lower, mean = mu, sd = sigma_x
prob
```

```
## [1] 0.6826895
```

Part III

Statistika Inferensial

Chapter 9

Pengujian Hipotesis

Statistika inferensial adalah cabang statistik yang menganalisis data sampel untuk membuat kesimpulan tentang populasi, meliputi estimasi parameter dan pengujian hipotesis. Pengujian hipotesis dimulai dengan merumuskan hipotesis nol (H_0) sebagai klaim awal dan hipotesis alternatif (H_a) sebagai tandingan, diikuti oleh penentuan tingkat signifikansi α, perhitungan statistik uji, dan nilai-p *(p-value)*. Jika nilai-p lebih kecil dari α, H_0 ditolak, menunjukkan bukti signifikan mendukung H_a. Metode ini sering digunakan untuk menguji rata-rata, proporsi, atau hubungan antar variabel, dengan aplikasi dalam berbagai bidang seperti riset pemasaran, pengendalian mutu, dan pengambilan keputusan berbasis data.

9.1 Hipotesis Nol dan Alternatif

H_0 adalah pernyataan awal yang menyatakan tidak ada efek, perbedaan, atau hubungan antara variabel yang diuji. Biasanya, H_0 dirumuskan untuk mempertahankan asumsi awal, misalnya: "Rata-rata penghasilan karyawan = Rp5 juta". Sebaliknya, H_a adalah pernyataan tandingan yang menyatakan adanya efek, perbedaan, atau hubungan, misalnya: "Rata-rata penghasilan karyawan $\neq$ Rp5 juta". Dalam pengujian hipotesis, fokus utama adalah menilai apakah data cukup kuat untuk menolak H_0 dan mendukung H_a. Pengujian ini dilakukan dengan menggunakan statistik uji dan membandingkan nilai probabilitas (p-value) terhadap tingkat signifikansi (α).

9.2 Kesalahan Tipe I dan Tipe II

Dalam pengujian hipotesis, terdapat dua jenis kesalahan yang dapat terjadi:

1. **Kesalahan Tipe I (α):**
   Terjadi ketika hipotesis nol (H_0) ditolak padahal sebenarnya H_0 benar.

Kesalahan ini sering disebut **false positive** dan tingkat kejadiannya diwakili oleh tingkat signifikansi (α), misalnya 0,05 (5%).

2. **Kesalahan Tipe II (β):**
   Terjadi ketika hipotesis nol (H_0) tidak ditolak padahal sebenarnya hipotesis alternatif (H_a) benar. Kesalahan ini dikenal sebagai **false negative**, dan probabilitas untuk tidak melakukan kesalahan tipe II disebut **kekuatan uji** atau **power of the test** ($1 - \beta$).

Keputusan	H_0 **Benar**	H_a **Benar**
Terima H_0	Benar (tidak ada kesalahan)	Kesalahan Tipe II (β)
Tolak H_0	Kesalahan Tipe I (α)	Benar (keputusan tepat)

Kesalahan ini dapat dikontrol dengan pemilihan tingkat signifikansi (α) yang sesuai dan memastikan ukuran sampel memadai untuk meningkatkan kekuatan uji ($1 - \beta$).

9.3 Nilai p dan Tingkat Signifikansi

Nilai-p adalah probabilitas yang menunjukkan seberapa konsisten data sampel dengan H_0. Nilai ini menggambarkan kemungkinan mendapatkan hasil sampel yang ekstrem atau lebih ekstrem dari hasil yang diamati, dengan asumsi bahwa H_0 benar. Semakin kecil nilai-p, semakin kuat bukti untuk menolak H_0.

Tingkat Signifikansi α adalah batas probabilitas yang telah ditentukan sebelumnya untuk memutuskan apakah H_0 dapat ditolak. Umumnya, $\alpha = 0.05$ (5%) digunakan, tetapi dalam beberapa kasus tertentu dapat menggunakan $\alpha = 0.01$ (1%) atau $\alpha = 0.10$ (10%).

Aturan Keputusan:

- Jika $p \leq \alpha$: Tolak H_0 (data memberikan bukti signifikan mendukung H_a).
- Jika $p > \alpha$: Gagal menolak H_0 (data tidak cukup untuk mendukung H_a).

9.4 Uji Z

Uji Z adalah uji statistik yang digunakan untuk membandingkan nilai rata-rata sampel dengan nilai rata-rata populasi ketika varians populasi diketahui atau ukuran sampel cukup besar. Uji ini umumnya digunakan dalam analisis dua sisi, di mana tujuan utamanya adalah untuk menguji apakah suatu sampel berasal dari populasi dengan rata-rata tertentu.

9.4.1 Ketentuan Uji Z

- Sampel memiliki distribusi normal atau ukuran sampel besar (biasanya $n > 30$).
- Varians atau deviasi standar populasi diketahui atau diperkirakan dengan baik.

9.4.2 Formula Uji Z

$$Z = \frac{\bar{X} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$$

Keterangan:

- $\bar{X}$ = Rata-rata sampel
- μ_0 = Rata-rata populasi yang diuji
- σ = Deviasi standar populasi
- n = Ukuran sampel

9.4.3 Langkah-langkah Uji Z

Misalkan kita ingin menguji apakah rata-rata penghasilan karyawan berbeda dari Rp5 juta, dengan ukuran sampel $n = 100$, deviasi standar populasi $\sigma = 1$ juta, dan rata-rata sampel $\bar{X} = 4.9$ juta. Interpretasi Uji Z dengan contoh Data, sebagai berikut:

1. **Hipotesis**:
   - H_0: Rata-rata penghasilan karyawan adalah Rp5 juta $\mu = 5$.
   - H_a: Rata-rata penghasilan karyawan berbeda dari Rp5 juta $\mu \neq 5$.
2. **Data dan Parameter**:
   - Ukuran sampel (n): 100
   - Rata-rata sampel ($\bar{X}$): Rp4.9 juta
   - Rata-rata populasi (μ): Rp5 juta
   - Deviasi standar populasi (σ): Rp1 juta
3. **Statistik Uji** Z: Rumus:
   $$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$
   Substitusi nilai:
   $$Z = \frac{4.9 - 5}{1/\sqrt{100}} = \frac{-0.1}{0.1} = -1$$
4. **Tingkat Signifikansi** (α):
   - Diasumsikan $\alpha = 0.05$ (tingkat signifikansi 5%).

5. **Nilai Kritikal**: Untuk uji dua sisi, nilai kritikal Z-kritis pada $\alpha = 0.05$ adalah:
$$Z_{\text{kritis}} = \pm 1.96$$

6. **Keputusan Uji**:
   - Karena nilai $Z = -1$ berada di dalam rentang -1.96 hingga 1.96, maka kita **tidak dapat menolak** H_0.
   - Dengan kata lain, tidak terdapat bukti signifikan bahwa rata-rata penghasilan karyawan berbeda dari Rp5 juta.

7. **Nilai-p**: Nilai-p dihitung berdasarkan area di bawah kurva distribusi normal standar untuk $|Z| = 1$:
$$p = 2 \times \text{P}(Z > 1) \approx 2 \times 0.1587 = 0.3174$$
   Karena $p > \alpha$, hasilnya konsisten dengan keputusan di atas H_0 tidak ditolak).

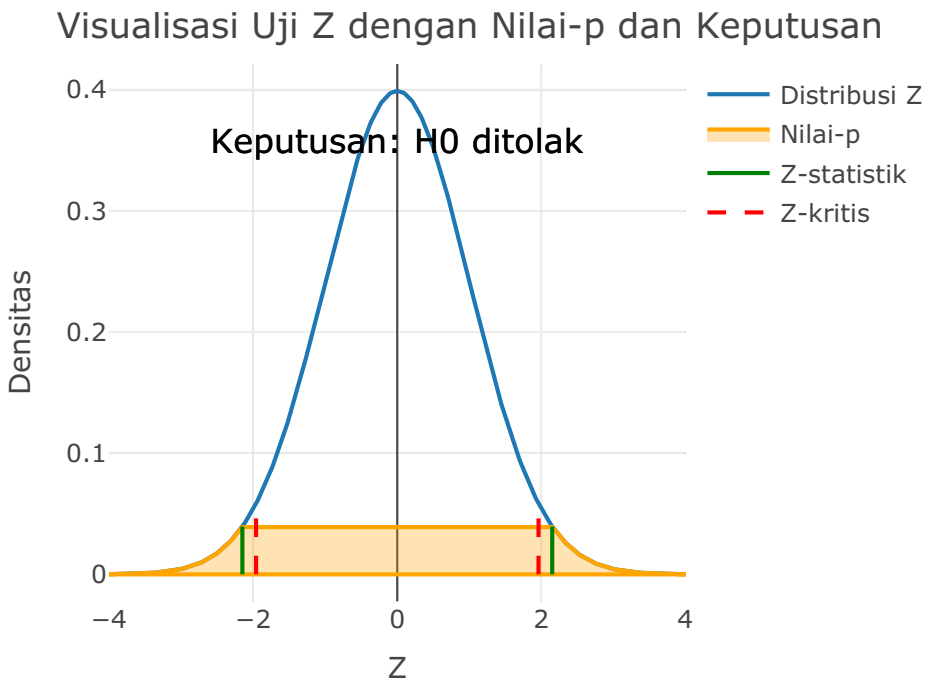
Kesimpulan:

- Berdasarkan data sampel, tidak terdapat cukup bukti untuk menyatakan bahwa rata-rata penghasilan karyawan berbeda secara signifikan dari Rp5 juta pada tingkat signifikansi 5%.
- **Keputusan**: H_0 **diterima.**

Berikut adalah kode R untuk membuat visualisasi interaktif menggunakan Plotly. Distribusi normal ini akan menampilkan area p-value berdasarkan nilai statistik Z.

```
library(plotly)

# Parameter uji Z
z_stat <- 2.15                  # Contoh nilai Z-statistik
p_value <- 2 * (1 - pnorm(abs(z_stat))) # Nilai p untuk uji dua sisi
alpha <- 0.05                   # Tingkat signifikansi

# Distribusi normal standar
x <- seq(-4, 4, length = 500)
z_dist <- dnorm(x)

# Filter data untuk area p-value
x_p <- c(x[x <= -abs(z_stat)], x[x >= abs(z_stat)])
y_p <- c(dnorm(x[x <= -abs(z_stat)]), dnorm(x[x >= abs(z_stat)]))

# Batas kritis Z
critical_z <- qnorm(1 - alpha / 2)

# Keputusan uji hipotesis
```

```
keputusan <- ifelse(p_value < alpha, "H0 ditolak", "H0 diterima")

# Plot distribusi normal
fig <- plot_ly(x = x, y = z_dist, type = 'scatter', mode = 'lines', name = "Distribusi Z") %>%
  # Area untuk Nilai-p
  add_trace(x = c(x_p, rev(x_p)),
            y = c(y_p, rep(0, length(y_p))),
            fill = 'toself',
            name = "Nilai-p",
            fillcolor = 'rgba(255, 165, 0, 0.3)',
            line = list(color = "orange")) %>%
  # Garis untuk Z-statistik
  add_segments(x = z_stat, xend = z_stat, y = 0, yend = dnorm(z_stat),
               line = list(color = "green"), name = "Z-statistik") %>%
  add_segments(x = -z_stat, xend = -z_stat, y = 0, yend = dnorm(-z_stat),
               line = list(color = "green"), showlegend = FALSE) %>%
  # Garis kritis
  add_segments(x = critical_z, xend = critical_z, y = 0, yend = dnorm(critical_z),
               line = list(dash = "dash", color = "red"), name = "Z-kritis") %>%
  add_segments(x = -critical_z, xend = -critical_z, y = 0, yend = dnorm(-critical_z),
               line = list(dash = "dash", color = "red"), showlegend = FALSE) %>%
  # Tambahkan anotasi untuk keputusan
  add_annotations(
    x = 0, y = max(z_dist) * 0.9,
    text = paste("Keputusan:", keputusan),
    showarrow = FALSE,
    font = list(size = 16, color = "black")
  ) %>%
  layout(
    title = "Visualisasi Uji Z dengan Nilai-p dan Keputusan",
    xaxis = list(title = "Z"),
    yaxis = list(title = "Densitas"),
    showlegend = TRUE
  )

fig
```



9.5 Uji T (t-test)

Uji t adalah uji statistik yang digunakan untuk menguji perbedaan rata-rata antara dua grup atau membandingkan rata-rata sampel dengan suatu nilai tertentu ketika varians populasi tidak diketahui atau ukuran sampel kecil (biasanya $n < 30$.

Uji t menggunakan distribusi t-student dan biasanya digunakan untuk situasi berikut:

1. Menguji apakah rata-rata satu sampel berbeda dengan suatu nilai (sampel tunggal).
2. Menguji apakah dua sampel independen memiliki rata-rata yang berbeda.
3. Menguji apakah dua sampel yang berhubungan memiliki perbedaan rata-rata.

9.5.1 Jenis-jenis Uji T

1. **Uji t Satu Sampel (One-sample t-test)**: Menguji apakah rata-rata suatu sampel berbeda dari nilai yang diketahui (misalnya rata-rata populasi).

$$t = \frac{\bar{X} - \mu}{\frac{s}{\sqrt{n}}}$$

Keterangan:

- $\bar{X}$: Rata-rata sampel
- μ: Nilai rata-rata yang diuji (biasanya rata-rata populasi)
- s: Deviasi standar sampel
- n: Ukuran sampel

2. **Uji t Dua Sampel Independen (Two-sample t-test)**:

   Menguji apakah dua grup sampel independen memiliki rata-rata yang berbeda.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

   Keterangan:

   - $\bar{X}_1$ dan $\bar{X}_2$: Rata-rata sampel grup 1 dan 2
   - s_1^2 dan s_2^2: Varians sampel grup 1 dan 2
   - n_1 dan n_2: Ukuran sampel grup 1 dan 2

3. **Uji t Sampel Berpasangan (Paired sample t-test)**: Menguji apakah ada perbedaan rata-rata dalam dua kondisi yang berhubungan, misalnya sebelum dan sesudah suatu perlakuan pada kelompok yang sama.

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}$$

   Keterangan:

   - $\bar{d}$: Rata-rata selisih
   - s_d: Deviasi standar selisih
   - n: Ukuran sampel

9.5.2 Langkah-langkah Uji t

1. **Formulasi Hipotesis**:
   - H_0: Rata-rata μ = nilai yang diuji (misalnya rata-rata populasi atau perbedaan antara dua grup = 0).
   - Hipotesis Alternatif H_a: Rata-rata $\mu \neq$ nilai yang diuji atau perbedaan antara dua grup $\neg 0$.
2. **Hitung Nilai t** menggunakan rumus sesuai dengan jenis uji t yang dilakukan.
3. **Bandingkan Nilai t dengan t-Tabel** untuk menentukan apakah nilai t terletak di area kritis. (Atau bandingkan nilai-p dengan tingkat signifikansi α).
4. **Keputusan**:

- Jika nilai t lebih besar dari t-kritis (untuk uji dua sisi) atau nilai-p lebih kecil dari α (misalnya 0.05), maka H_0 ditolak.

9.5.3 Uji t Satu Sampel

Apakah Rata-rata Penghasilan Berbeda dari Rp5 Juta?

1. $H_0 : \mu = 5$, Rata-rata penghasilan karyawan sama dengan Rp5 juta.
2. $H_a : \mu \neq 5$, Rata-rata penghasilan karyawan berbeda dari Rp5 juta.

Uji ini **two-tailed test**.

```
# Data penghasilan karyawan (dalam juta)
penghasilan <- c(4.8, 5.2, 5.1, 5.0, 4.9, 5.3, 4.7, 5.4, 4.6, 5.5)

# Uji t: Rata-rata penghasilan dibandingkan dengan Rp5 juta
t_test_result <- t.test(penghasilan, mu = 5)  # Uji hipotesis: rata-rata = 5 juta
print(t_test_result)
```

```
##
##  One Sample t-test
##
## data:  penghasilan
## t = 0.52223, df = 9, p-value = 0.6141
## alternative hypothesis: true mean is not equal to 5
## 95 percent confidence interval:
##  4.833415 5.266585
## sample estimates:
## mean of x
##      5.05
```

```
# Ekstrak hasil
t_statistic <- t_test_result$statistic       # Nilai t
p_value <- t_test_result$p.value             # P-value
confidence_interval <- t_test_result$conf.int # Confidence interval
mean_sample <- t_test_result$estimate        # Rata-rata sampel
df <- t_test_result$parameter                # Derajat kebebasan
```

Penjelasan Hasil:

- Karena $t = 0.5222$ jauh dari nilai kritis yang signifikan (ditentukan oleh derajat kebebasan dan tingkat signifikansi), ini menunjukkan **tidak ada perbedaan signifikan**.
- p-value = 0.6141 lebih besar dari 0.05, sehingga kita gagal menolak H_0. Tidak ada cukup bukti untuk menyimpulkan bahwa rata-rata penghasilan berbeda dari Rp5 juta.
- Confidence Interval (CI): Dengan tingkat kepercayaan 95%, rata-rata penghasilan karyawan diperkirakan antara 4.8334, 5.2666 juta.
- Rata-rata Sampel: adalah 5.05 juta.

2. Uji t Dua Sampel Independen (Two-Sample t-test)

Misalkan kita ingin menguji apakah rata-rata skor ujian antara dua grup (Grup A dan Grup B) berbeda.

```
# Data skor ujian
grup_A <- c(80, 85, 82, 88)
grup_B <- c(75, 70, 78, 72)

# Uji t Dua Sampel Independen
t_test_result <- t.test(grup_A, grup_B)  # Uji hipotesis: perbedaan rata-rata
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  grup_A and grup_B
## t = 4.0406, df = 6, p-value = 0.006798
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   3.944202 16.055798
## sample estimates:
## mean of x mean of y
##     83.75     73.75
```

3. Uji t Sampel Berpasangan (Paired Sample t-test):

Menggunakan data sebelum dan sesudah perlakuan pada sampel yang sama.

```
# Data sebelum dan sesudah perlakuan
sebelum <- c(85, 88, 90, 91)
sesudah <- c(90, 92, 91, 95)

# Uji t Sampel Berpasangan
paired_t_test_result <- t.test(sebelum, sesudah, paired = TRUE)  # Uji hipotesis: rata-rata selis
print(paired_t_test_result)
```

```
##
##  Paired t-test
##
## data:  sebelum and sesudah
## t = -4.0415, df = 3, p-value = 0.02726
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -6.2560793 -0.7439207
## sample estimates:
## mean difference
##            -3.5
```

Kesimpulan:

- Uji t digunakan untuk menguji perbedaan rata-rata antara satu atau lebih sampel.
- Uji t Satu Sampel digunakan untuk membandingkan rata-rata sampel dengan nilai tertentu.
- Uji t Dua Sampel digunakan untuk menguji perbedaan rata-rata antara dua grup sampel.
- Uji t Sampel Berpasangan digunakan untuk menguji perbedaan rata-rata dalam dua kondisi yang berhubungan.

Chapter 10

Korelasi dan Regresi

Chapter 11

Uji Non-Parametrik

11.1 Uji Tanda, Uji Wilcoxon Signed-Rank

11.2 Uji Mann-Whitney

11.3 Uji Kruskal-Wallis

Chapter 12

Terapan Statistika

12.1 Studi Kasus dalam Berbagai Bidang

(misalnya, Kesehatan, Bisnis, Ilmu Sosial)

12.2 Proyek Analisis Data Dunia Nyata