LAPORAN
PRAKTIKUM PEMODELAN STATISTIKA
MODUL 2

Disusun oleh :

Nama : Fidelia Ping

NIM : 245410012

Kelas : Informatika 1

PROGRAM STUDI INFORMATIKA

PROGRAM SARJANA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

YOGYAKARTA

2025

MODUL 2
PENGELOLAAN DATA DENGAN R

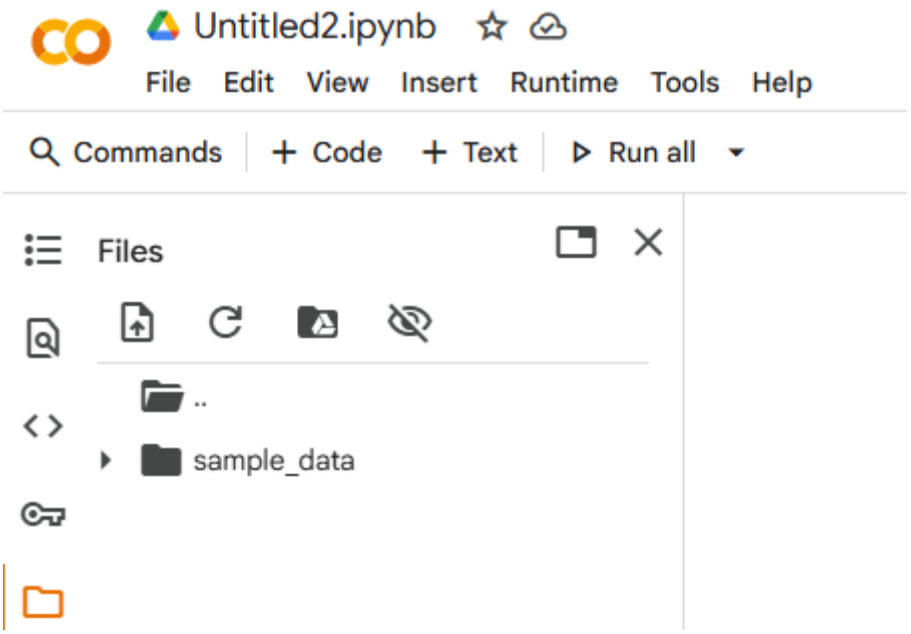
A. PEMBAHASAN PRAKTIK

a. Praktik impor file dari upload lokal di google colab

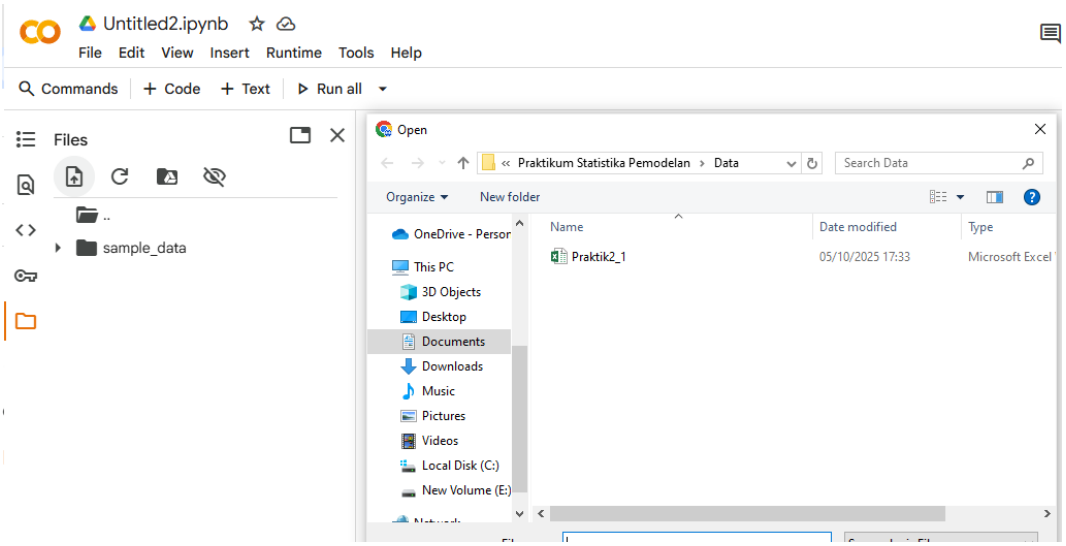
- Buat file dalam Excel dari data dibawah, simpan di komputer lokal, misal diberi nama Praktik2_1

Nama	IPK
Andi	3.6
Budi	3.4
Cici	3.3
Dodi	2.9
Ina	2.8
Rudi	3.0
Rini	2.4
Yani	1.9
Yuli	2.5
Val	2.8

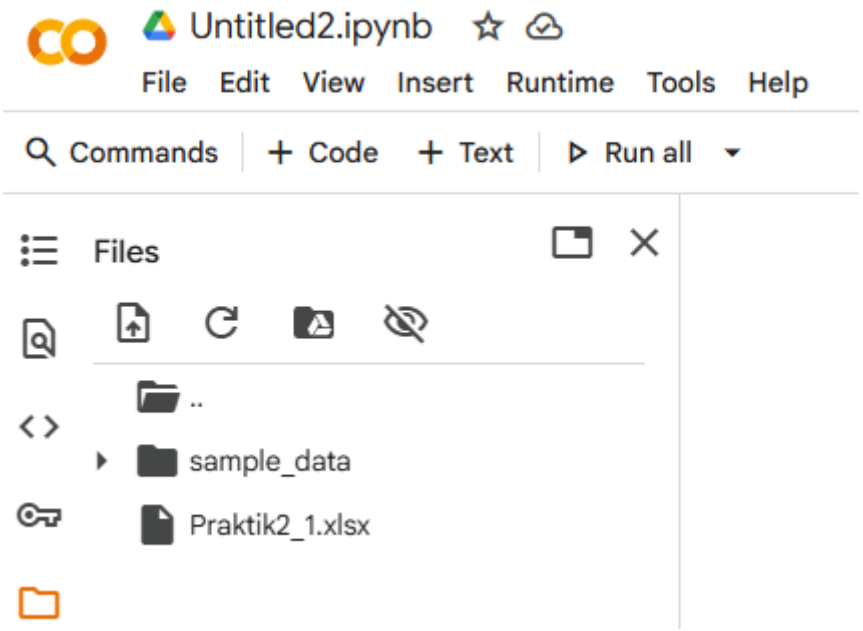
- Masuk di Google Colab dengan login di akun anda
- Untuk mengunggah file Praktik2_1.xlsx, klik ikon folder di sidebar kiri Colab (panel Files).



- Klik ikon upload (kertas dengan panah ke atas) dan pilih file dari komputer.



– File akan tersimpan sementara di direktori /content.



– Untuk mengimpor file excel yang diunggah, sebelumnya copi path file yang akan diimport, gunakan perintah

```
# Praktik Impor File Excel di Google Colab

# 1. Instal dan muat paket readxl
cat("=== Menginstal dan Memuat Paket readxl ===\n")
  install.packages("readxl")
library(readxl)

# 2. Membaca file Excel
# Menggunakan read_excel() dari paket readxl
cat("\n=== Membaca File Excel ===\n")
data <- read_excel("/content/Praktik2_1.xlsx", sheet = 1)
print(data)
```

Pembahasan: Bagian pertama dari kode ini memastikan bahwa R memiliki semua alat yang dibutuhkan untuk membaca file Excel. cat("=== Menginstal dan Memuat Paket readxl ===\n") Fungsi cat() digunakan untuk menampilkan output ke konsol. Ini seperti fungsi print(), tetapi seringkali lebih cocok untuk menampilkan pesan yang diformat. Output yang dihasilkan adalah: === Menginstal dan Memuat Paket readxl === install.packages("readxl") Ini adalah perintah penting yang digunakan untuk mengunduh dan memasang paket (package) readxl dari repositori CRAN. Paket readxl adalah pustaka spesifik di R yang dirancang untuk membaca data dari file Microsoft Excel (.xlsx dan .xls). Di Google Colab, perintah ini akan menginstal paket. Pada bagian cat("\n===

Membaca File Excel ===\n") Sama seperti sebelumnya, ini hanya menampilkan pesan informatif ke konsol. Outputnya adalah: === Membaca File Excel === data <- read_excel("/content/Praktik2_1.xlsx", sheet = 1) Ini adalah perintah utama untuk mengimpor data. Mari kita pecah: read_excel(): Fungsi ini adalah bagian dari paket readxl yang sudah kita muat. "/content/Praktik2_1.xlsx": Ini adalah jalur (path) file. Di Google Colab, file yang diunggah ke panel "Files" akan disimpan sementara di direktori /content/. Jadi, ini adalah alamat di mana R akan menemukan file. sheet = 1: Parameter ini menentukan lembar kerja (sheet) mana di dalam file Excel yang akan dibaca. Angka 1 berarti R akan membaca lembar kerja pertama. data <- ...: Hasil dari fungsi read_excel() (yaitu, data dari Excel) akan disimpan ke dalam variabel bernama data. Variabel ini sekarang berisi sebuah data frame di R, yang merupakan struktur data berbentuk tabel. print(data) Perintah ini menampilkan isi dari variabel data ke konsol.

Output

```
=== Membaca File Excel ===
# A tibble: 10 × 2
   Nama  IPK
   <chr> <chr>
 1 Andi  3.6
 2 Budi  3.4
 3 Cici  3.3
 4 Dodi  2.9
 5 Ina   2.8
 6 Rudi  3.0
 7 Rini  2.4
 8 Yani  1.9
 9 Yuli  2.5
10 Val   2.8
```

Pembahasan: dapat dilihat outputnya menampilkan table dari file excel ke konsol dengan 10 baris dan 2 kolom (Nama dan IPK) telah dibuat. R secara otomatis mengenali tipe data: <chr> untuk karakter (nama) dan <dbl> untuk ganda (angka desimal, IPK).

b. Praktik memeriksa data

```
# 1. Membuat dataset
data <- data.frame(
  id = c(1, 1, 2, 3, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
         19, 20, 21, 22, 23, 24, 25, 26, 27, 28),
  umur = c(25, 25, 30, 22, 22, 35, 28, 40, 33, 27, 29, NA, 31, 26, 38, 45, 23,
           29, 34, 36, 41, 27, 30, 32, NA, 28, 39, 24, 31, 50),
  pendapatan = c(5000, 5000, 6000, 4500, 4500, 7000, 5500, 8000, 6200, NA,
                 5100, 4900, 7500, 5200, 6800, 9000, 4700, 5300, 6100, 7200,
                 NA, 5400, 6500, 4800, 5100, 5900, 7800, 4600, 6700, 15000),
  kategori = c("PNS", "PNS", "Swasta", "Wirausaha", "Wirausaha", "PNS",
               "Swasta", "PNS", "Wirausaha", "Swasta", "PNS", "Swasta",
               "PNS", "Wirausaha", "Swasta", "PNS", "Wirausaha", "Swasta",
               "PNS", "Swasta", "Wirausaha", "PNS", "Swasta", "Wirausaha",
               "PNS", "Swasta", "PNS", "Wirausaha", "Swasta", "PNS")
)

# 2. Memeriksa Data

# a. Melihat data secara umum
cat("=== a. Melihat Data Secara Umum ===\n")
cat("Semua data:\n")
```

```
print(data)
cat("\nHead (6 baris pertama):\n")
print(head(data))
cat("\nTail (6 baris terakhir):\n")
print(tail(data))


# b. Memeriksa struktur data
cat("\n=== b. Memeriksa Struktur Data ===\n")
str(data)

# c. Memeriksa dimensi data
cat("\n=== c. Memeriksa Dimensi Data ===\n")
cat("Dimensi (baris, kolom):", dim(data), "\n")
cat("Jumlah baris:", nrow(data), "\n")
cat("Jumlah kolom:", ncol(data), "\n")
cat("Nama kolom:", names(data), "\n")

# d. Ringkasan statistik
cat("\n=== d. Ringkasan Statistik ===\n")
summary(data)

# e. Mengecek nilai hilang
cat("\n=== e. Mengecek Nilai Hilang ===\n")
cat("Jumlah NA per kolom:\n")
print(colSums(is.na(data)))
cat("Jumlah baris lengkap (tanpa NA):", sum(complete.cases(data)), "\n")

# f. Mengecek duplikat
cat("\n=== f. Mengecek Duplikat ===\n")
cat("Jumlah baris duplikat:", sum(duplicated(data)), "\n")
cat("Baris duplikat:\n")
print(data[duplicated(data), ])

# g. Mengecek outlier pada pendapatan
cat("\n=== g. Mengecek Outlier ===\n")
Q1 <- quantile(data$pendapatan, 0.25, na.rm = TRUE)
Q3 <- quantile(data$pendapatan, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
cat("Batas bawah (Q1 - 1.5 * IQR):", lower_bound, "\n")
cat("Batas atas (Q3 + 1.5 * IQR):", upper_bound, "\n")
outliers <- data$pendapatan[data$pendapatan < lower_bound |
                            data$pendapatan > upper_bound & !is.na(data$pendapatan)]
cat("Nilai outlier pada pendapatan:", outliers, "\n")
```

Pembahasan: pada praktik ini terdapat 2 bagian utama yang pertama membuat Dataset dan memeriksa kualitas data pada bagian pertama Kode ini dimulai dengan membuat sebuah data frame. Di R, data frame adalah struktur data berbentuk tabel yang paling sering digunakan, mirip seperti spreadsheet. data <- data.frame(...): Perintah ini membuat sebuah data frame baru dan menyimpannya ke dalam variabel bernama data. Di dalam data.frame() ini untuk mendefinisikan empat kolom: id, umur,

pendapatan, dan kategori. Setiap kolom dibuat dari sebuah vektor (kumpulan nilai) yang disatukan dengan fungsi c(). Terdapat beberapa nilai NA di kolom umur dan pendapatan. NA (Not Available) adalah cara R untuk merepresentasikan data yang hilang atau tidak ada. Pada bagian ke 2 kode tersebut Setelah data frame dibuat, kode ini menjalankan serangkaian perintah untuk memeriksa kualitas dan karakteristik datanya.

Output

```
=== a. Melihat Data Secara Umum ===
Semua data:
   id umur pendapatan  kategori
1   1   25       5000       PNS
2   1   25       5000       PNS
3   2   30       6000    Swasta
4   3   22       4500 Wirausaha
5   3   22       4500 Wirausaha
6   4   35       7000       PNS
7   5   28       5500    Swasta
8   6   40       8000       PNS
9   7   33       6200 Wirausaha
10  8   27         NA    Swasta
11  9   29       5100       PNS
12 10   NA       4900    Swasta
13 11   31       7500       PNS
14 12   26       5200 Wirausaha
15 13   38       6800    Swasta
16 14   45       9000       PNS
17 15   23       4700 Wirausaha
18 16   29       5300    Swasta
19 17   34       6100       PNS
20 18   36       7200    Swasta
21 19   41         NA Wirausaha
22 20   27       5400       PNS
23 21   30       6500    Swasta
24 22   32       4800 Wirausaha
25 23   NA       5100       PNS
26 24   28       5900    Swasta
27 25   39       7800       PNS
28 26   24       4600 Wirausaha
29 27   31       6700    Swasta
30 28   50      15000       PNS

Head (6 baris pertama):
  id umur pendapatan  kategori
1  1   25       5000       PNS
2  1   25       5000       PNS
3  2   30       6000    Swasta
4  3   22       4500 Wirausaha
5  3   22       4500 Wirausaha
6  4   35       7000       PNS

Tail (6 baris terakhir):
   id umur pendapatan  kategori
25 23   NA       5100       PNS
26 24   28       5900    Swasta
27 25   39       7800       PNS
28 26   24       4600 Wirausaha
29 27   31       6700    Swasta
30 28   50      15000       PNS

=== b. Memeriksa Struktur Data ===
'data.frame':   30 obs. of  4 variables:
 $ id        : num  1 1 2 3 3 4 5 6 7 8 ...
 $ umur      : num  25 25 30 22 22 35 28 40 33 27 ...
 $ pendapatan: num  5000 5000 6000 4500 4500 7000 5500 8000 6200 NA ...
```

```
=== c. Memeriksa Dimensi Data ===
Dimensi (baris, kolom): 30 4
Jumlah baris: 30
Jumlah kolom: 4
Nama kolom: id umur pendapatan kategori

=== d. Ringkasan Statistik ===
       id             umur         pendapatan      kategori
 Min.   : 1.00   Min.   :22.00   Min.   : 4500   Length:30
 1st Qu.: 6.25   1st Qu.:26.75   1st Qu.: 5000   Class :character
 Median :13.50   Median :30.00   Median : 5700   Mode  :character
 Mean   :13.67   Mean   :31.43   Mean   : 6261
 3rd Qu.:20.75   3rd Qu.:35.25   3rd Qu.: 6850
 Max.   :28.00   Max.   :50.00   Max.   :15000
                 NA's   :2       NA's   :2

=== e. Mengecek Nilai Hilang ===
Jumlah NA per kolom:
        id       umur pendapatan   kategori
         0          2          2          0
Jumlah baris lengkap (tanpa NA): 26

=== f. Mengecek Duplikat ===
Jumlah baris duplikat: 2
Baris duplikat:
  id umur pendapatan  kategori
2  1   25       5000       PNS
5  3   22       4500 Wirausaha

=== g. Mengecek Outlier ===
Batas bawah (Q1 - 1.5 * IQR): 2225
Batas atas (Q3 + 1.5 * IQR): 9625
Nilai outlier pada pendapatan: NA NA 15000
```

Pembahasan:

a. Melihat Data Secara Umum

cat("=== a. Melihat Data Secara Umum ===\n"): Ini hanya untuk menampilkan pesan di konsol. print(data): Perintah ini menampilkan seluruh data frame di layar. Outputnya adalah tabel lengkap dari 30 baris dan 4 kolom yang dibuat. print(head(data)) dan print(tail(data)): Fungsi ini sangat berguna untuk melihat gambaran cepat data tanpa mencetak semuanya. head() menampilkan 6 baris pertama, sedangkan tail() menampilkan 6 baris terakhir.

b. Memeriksa Struktur Data

cat("\n=== b. Memeriksa Struktur Data ===\n"): Pesan lagi. str(data): Fungsi str() (singkatan dari structure) adalah salah satu alat terbaik untuk memahami data. Outputnya akan menunjukkan: Tipe objeknya ('data.frame'). Jumlah observasi (baris) dan variabel (kolom) (30 obs. of 4 variables). Nama setiap kolom dan tipe datanya (misalnya, int untuk integer, num untuk numerik, chr untuk karakter).

c. Memeriksa Dimensi Data

cat("\n=== c. Memeriksa Dimensi Data ===\n"): Pesan. dim(data): Mengembalikan jumlah baris dan kolom sebagai dua angka. Outputnya adalah 30 4. nrow(data): Mengembalikan jumlah baris, yaitu 30. ncol(data): Mengembalikan jumlah kolom, yaitu 4. names(data): Mengembalikan nama-nama kolom ("id" "umur" "pendapatan" "kategori").

d. Ringkasan Statistik

summary(data): Perintah ini memberikan ringkasan statistik deskriptif untuk setiap kolom. Untuk kolom numerik (id, umur, pendapatan), ia menampilkan nilai minimum, Q1, median, rata-rata, Q3, dan nilai maksimum. Untuk kolom karakter (kategori), ia menampilkan frekuensi dari nilai-nilai unik. Ini membantu untuk melihat sebaran data secara sekilas.

e. Mengecek Nilai Hilang

print(colSums(is.na(data))): Ini adalah cara pintar untuk menghitung NA per kolom. is.na(data) membuat sebuah tabel logis (TRUE/FALSE) di mana TRUE menunjukkan adanya NA. colSums() menjumlahkan nilai TRUE di setiap kolom (di R, TRUE dihitung sebagai 1 dan FALSE sebagai 0). Outputnya akan menunjukkan: id = 0, umur = 2, pendapatan = 2, dan kategori = 0. Ini berarti ada dua nilai hilang di kolom umur dan dua di kolom pendapatan. sum(complete.cases(data)): Menghitung berapa banyak baris yang tidak memiliki NA sama sekali. Outputnya adalah 26, karena 4 baris mengandung NA (2 di umur dan 2 di pendapatan).

f. Mengecek Duplikat

sum(duplicated(data)): Menghitung jumlah baris duplikat. duplicated() akan mengembalikan TRUE untuk baris yang merupakan salinan dari baris yang sudah ada. Outputnya adalah 2. print(data[duplicated(data),]): Menampilkan baris-baris yang duplikat. Outputnya akan menunjukkan baris kedua (id 1) dan baris kelima (id 3), mengonfirmasi bahwa mereka adalah duplikat.

g. Mengecek Outlier

Kode ini menggunakan metode Interquartile Range (IQR) untuk mendeteksi outlier pada kolom pendapatan. Q1 <- quantile(data$pendapatan, 0.25, na.rm = TRUE): Menghitung kuartil pertama (Q1), yang merupakan 25% nilai terendah. na.rm = TRUE memastikan nilai NA diabaikan dalam perhitungan. Q3 <- quantile(data$pendapatan, 0.75, na.rm = TRUE): Menghitung kuartil ketiga (Q3), yaitu 75% nilai terendah. IQR <- Q3 - Q1: Menghitung IQR atau selisih antara Q3 dan Q1. lower_bound <- Q1 - 1.5 * IQR dan upper_bound <- Q3 + 1.5 * IQR: Ini adalah rumus standar untuk menentukan batas bawah dan atas. Setiap nilai di luar batas ini dianggap sebagai outlier. outliers <- ...: Baris ini menyaring data untuk menemukan nilai-nilai yang berada di luar batas yang sudah dihitung. cat("Nilai outlier pada pendapatan:", outliers, "\n"): Menampilkan nilai-nilai yang teridentifikasi sebagai outlier. Berdasarkan data ini, outputnya akan menunjukkan 15000, yang secara signifikan lebih tinggi dari pendapatan lainnya.

c. Praktik menyiapkan Data

```
# 1. Membuat dataset
data <- data.frame(
  id = c(1, 1, 2, 3, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
         19, 20, 21, 22, 23, 24, 25, 26, 27, 28),
  umur = c(25, 25, 30, 22, 22, 35, 28, 40, 33, 27, 29, NA, 31, 26, 38, 45, 23,
           29, 34, 36, 41, 27, 30, 32, NA, 28, 39, 24, 31, 50),
  pendapatan = c(5000, 5000, 6000, 4500, 4500, 7000, 5500, 8000, 6200, NA,
                 5100, 4900, 7500, 5200, 6800, 9000, 4700, 5300, 6100, 7200,
                 NA, 5400, 6500, 4800, 5100, 5900, 7800, 4600, 6700, 15000),
  kategori = c("PNS", "PNS", "Swasta", "Wirausaha", "Wirausaha", "PNS",
               "Swasta", "PNS", "Wirausaha", "Swasta", "PNS", "Swasta",
               "PNS", "Wirausaha", "Swasta", "PNS", "Wirausaha", "Swasta",
               "PNS", "Swasta", "Wirausaha", "PNS", "Swasta", "Wirausaha",
               "PNS", "Swasta", "PNS", "Wirausaha", "Swasta", "PNS")
)
print(data)

# Muat paket yang diperlukan
library(dplyr)
library(ggplot2)

# 2. Persiapan Data

# a. Menangani nilai hilang
cat("=== a. Menangani Nilai Hilang ===\n")
# Cek jumlah NA
cat("Jumlah NA per kolom sebelum penanganan:\n")
print(colSums(is.na(data)))

# Opsi 1: Menghapus baris yang ada NA
data_no_na <- na.omit(data)
cat("\nOpsi 1 menghapus baris yang ada NA\n")
cat("Jumlah baris setelah menghapus NA:", nrow(data_no_na), "\n")

# Opsi 2: Mengisi NA dengan median untuk umur dan pendapatan
data_imputed <- data
data_imputed$umur[is.na(data_imputed$umur)] <- median(data_imputed$umur, na.rm =
TRUE)
data_imputed$pendapatan[is.na(data_imputed$pendapatan)] <-
median(data_imputed$pendapatan, na.rm = TRUE)
cat("\nOpsi 2 Mengisi NA dengan median untuk umur dan pendapatan\n")
cat("Jumlah NA per kolom setelah imputasi:\n")
print(colSums(is.na(data_imputed)))
cat("Jumlah baris setelah mengisi NA dengan median:", nrow(data_imputed), "\n")


# b. Menghapus data duplikat
cat("\n=== b. Menghapus Data Duplikat ===\n")
```

```
cat("Jumlah baris duplikat:", sum(duplicated(data_imputed)), "\n")
data_no_duplicates <- distinct(data_imputed)
cat("Jumlah baris setelah menghapus duplikat:", nrow(data_no_duplicates), "\n")

# c. Memastikan tipe data sesuai
cat("\n=== c. Memastikan Tipe Data Sesuai ===\n")
# Cek tipe data
cat("Tipe data sebelum konversi:\n")
print(str(data_no_duplicates))

# Konversi tipe data
data_no_duplicates$kategori <- as.factor(data_no_duplicates$kategori)
data_no_duplicates$umur <- as.numeric(data_no_duplicates$umur)
data_no_duplicates$pendapatan <- as.numeric(data_no_duplicates$pendapatan)
cat("Tipe data setelah konversi:\n")
print(str(data_no_duplicates))

# d. Menangani outlier (Metode IQR)
cat("\n=== d. Menangani Outlier ===\n")
Q1 <- quantile(data_no_duplicates$pendapatan, 0.25, na.rm = TRUE)
Q3 <- quantile(data_no_duplicates$pendapatan, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
cat("Batas bawah:", lower_bound, "\n")
cat("Batas atas:", upper_bound, "\n")

# Identifikasi outlier
outliers <- data_no_duplicates$pendapatan[data_no_duplicates$pendapatan <
lower_bound |
                                          data_no_duplicates$pendapatan >
upper_bound]
cat("Nilai outlier pada pendapatan:", outliers, "\n")

# Hapus outlier
data_no_outliers <- filter(data_no_duplicates,
                           pendapatan >= lower_bound & pendapatan <= upper_bound)
cat("Jumlah baris setelah menghapus outlier:", nrow(data_no_outliers), "\n")


# e. Membuat variabel baru
cat("\n=== e. Membuat Variabel Baru ===\n")
data_with_new_var <- mutate(data_no_outliers,
                            log_pendapatan = log(pendapatan),  # Transformasi log
                            umur_kategori = case_when(
                              umur < 30 ~ "Muda",
                              umur >= 30 & umur < 40 ~ "Dewasa",
                              TRUE ~ "Lansia"
                            ))
cat("Beberapa baris dengan variabel baru:\n")
print(head(data_with_new_var))

# f. Memfilter data
cat("\n=== f. Memfilter Data ===\n")
data_filtered <- filter(data_with_new_var,
                        umur >= 18,
                        kategori %in% c("PNS", "Swasta"))
cat("Jumlah baris setelah memfilter (umur >= 18, kategori PNS/Swasta):",
    nrow(data_filtered), "\n")
print(head(data_filtered))

# Simpan data yang sudah disiapkan
write.csv(data_filtered, "data_prepared.csv", row.names = FALSE)
cat("Data disimpan sebagai 'data_prepared.csv'\n")
```

Pembahasan: pada praktik ini kode dimulai dengan membuat data.frame bernama data yang sama seperti sebelumnya, lengkap dengan nilai NA dan baris duplikat. print(data): Perintah ini akan mencetak seluruh data frame ke konsol, memberikan gambaran lengkap tentang data awal. Simpan Data write.csv(data_filtered, "data_prepared.csv", row.names = FALSE): Ini adalah langkah terakhir yang sangat praktis. Fungsi ini menyimpan data frame data_filtered ke dalam file CSV di Google Colab. row.names = FALSE memastikan R tidak menambahkan kolom nomor baris di dalam file

Output

```
   id umur pendapatan  kategori
1   1   25       5000       PNS
2   1   25       5000       PNS
3   2   30       6000    Swasta
4   3   22       4500 Wirausaha
5   3   22       4500 Wirausaha
6   4   35       7000       PNS
7   5   28       5500    Swasta
8   6   40       8000       PNS
9   7   33       6200 Wirausaha
10  8   27         NA    Swasta
11  9   29       5100       PNS
12 10   NA       4900    Swasta
13 11   31       7500       PNS
14 12   26       5200 Wirausaha
15 13   38       6800    Swasta
16 14   45       9000       PNS
17 15   23       4700 Wirausaha
18 16   29       5300    Swasta
19 17   34       6100       PNS
20 18   36       7200    Swasta
21 19   41         NA Wirausaha
22 20   27       5400       PNS
23 21   30       6500    Swasta
24 22   32       4800 Wirausaha
25 23   NA       5100       PNS
26 24   28       5900    Swasta
27 25   39       7800       PNS
28 26   24       4600 Wirausaha
29 27   31       6700    Swasta
30 28   50      15000       PNS
=== a. Menangani Nilai Hilang ===
Jumlah NA per kolom sebelum penanganan:
        id       umur pendapatan   kategori
         0          2          2          0

Opsi 1 menghapus baris yang ada NA
Jumlah baris setelah menghapus NA: 26

Opsi 2 Mengisi NA dengan median untuk umur dan pendapatan
Jumlah NA per kolom setelah imputasi:
        id       umur pendapatan   kategori
         0          0          0          0
Jumlah baris setelah mengisi NA dengan median: 30

=== b. Menghapus Data Duplikat ===
Jumlah baris duplikat: 2
Jumlah baris setelah menghapus duplikat: 28

=== c. Memastikan Tipe Data Sesuai ===
Tipe data sebelum konversi:
'data.frame':   28 obs. of  4 variables:
 $ id        : num  1 2 3 4 5 6 7 8 9 10 ...
 $ umur      : num  25 30 22 35 28 40 33 27 29 30 ...
 $ pendapatan: num  5000 6000 4500 7000 5500 8000 6200 5700 5100 4900 ...
 $ kategori  : chr  "PNS" "Swasta" "Wirausaha" "PNS" ...
NULL
Tipe data setelah konversi:
'data.frame':   28 obs. of  4 variables:
 $ id        : num  1 2 3 4 5 6 7 8 9 10 ...
 $ umur      : num  25 30 22 35 28 40 33 27 29 30 ...
 $ pendapatan: num  5000 6000 4500 7000 5500 8000 6200 5700 5100 4900 ...
 $ kategori  : Factor w/ 3 levels "PNS","Swasta",..: 1 2 3 1 2 1 3 2 1 2 ...
NULL

=== d. Menangani Outlier ===
Batas bawah: 2475
Batas atas: 9475
Nilai outlier pada pendapatan: 15000
Jumlah baris setelah menghapus outlier: 27


 === e. Membuat Variabel Baru ===
 Beberapa baris dengan variabel baru:
   id umur pendapatan  kategori log_pendapatan umur_kategori
 1  1   25       5000       PNS       8.517193          Muda
 2  2   30       6000    Swasta       8.699515        Dewasa
 3  3   22       4500 Wirausaha       8.411833          Muda
 4  4   35       7000       PNS       8.853665        Dewasa
 5  5   28       5500    Swasta       8.612503          Muda
 6  6   40       8000       PNS       8.987197        Lansia

 === f. Memfilter Data ===
 Jumlah baris setelah memfilter (umur >= 18, kategori PNS/Swasta): 20
   id umur pendapatan kategori log_pendapatan umur_kategori
 1  1   25       5000      PNS       8.517193          Muda
 2  2   30       6000   Swasta       8.699515        Dewasa
 3  4   35       7000      PNS       8.853665        Dewasa
 4  5   28       5500   Swasta       8.612503          Muda
 5  6   40       8000      PNS       8.987197        Lansia
 6  8   27       5700   Swasta       8.648221          Muda
 Data disimpan sebagai 'data_prepared.csv'
```

Pembahasan:

a. Menangani Nilai Hilang (NA)

Kode ini menunjukkan dua opsi umum untuk menangani NA:

Opsi 1: Menghapus baris yang ada NA

data_no_na <- na.omit(data): Fungsi na.omit() adalah cara cepat dan sederhana untuk menghapus setiap baris yang memiliki setidaknya satu nilai NA. Outputnya: Jumlah baris setelah menghapus NA: 26. Ini menunjukkan 4 baris yang mengandung NA (dua di umur dan dua di pendapatan) telah dihapus.

Opsi 2: Mengisi NA dengan median (Imputasi)

data_imputed <- data: Pertama, buat salinan data agar data asli tidak berubah. data_imputed$umur[is.na(data_imputed$umur)] <- median(...): Baris ini menemukan nilai NA di kolom umur (is.na(...)) lalu menggantinya dengan nilai median dari kolom umur. na.rm = TRUE sangat penting, karena ia memberitahu R untuk mengabaikan nilai NA saat menghitung median. Jika tidak, median akan bernilai NA. Langkah serupa dilakukan untuk kolom pendapatan. Outputnya: Jumlah NA per kolom setelah imputasi: id = 0, umur = 0, pendapatan = 0, kategori = 0. Ini mengonfirmasi bahwa semua nilai hilang sudah diisi.

b. Menghapus Data Duplikat

sum(duplicated(data_imputed)): Menghitung jumlah baris duplikat. Outputnya adalah 2. data_no_duplicates <- distinct(data_imputed): Fungsi distinct() dari paket dplyr adalah cara terbaik untuk menghapus baris yang sepenuhnya duplikat. Ia akan menyimpan hanya baris unik. Outputnya: Jumlah baris setelah menghapus duplikat: 28. Ini menunjukkan dua baris duplikat (yang memiliki id 1 dan 3) telah dihapus, dan data sekarang memiliki 28 baris unik.

c. Memastikan Tipe Data Sesuai

str(data_no_duplicates): Dicetak dua kali, sebelum dan sesudah konversi. Ini memungkinkan melihat perubahannya. data_no_duplicates$kategori <- as.factor(...): Mengonversi kolom kategori menjadi tipe data factor. data_no_duplicates$umur <- as.numeric(...) dan data_no_duplicates$pendapatan <- as.numeric(...): Meskipun kemungkinan besar tipe data ini sudah numerik, langkah ini memastikan bahwa tidak ada masalah yang terlewat, seperti angka yang dibaca sebagai karakter.

d. Menangani Outlier (Metode IQR)

Ini adalah proses yang sama seperti di penjelasan sebelumnya, di mana kita menggunakan metode Interquartile Range (IQR) untuk menemukan batas bawah dan atas. outliers <- ...: Baris ini akan mengidentifikasi nilai 15000 sebagai outlier di kolom pendapatan. data_no_outliers <- filter(...): Ini adalah langkah penting. Fungsi filter() dari dplyr digunakan untuk memilih baris yang memenuhi kondisi tertentu. Di sini hanya menyimpan baris yang nilai pendapatannya berada di antara lower_bound dan upper_bound. Outputnya: Jumlah baris setelah menghapus outlier: 27. Ini menunjukkan satu baris (dengan pendapatan 15000) telah dihapus.

e. Membuat Variabel Baru

data_with_new_var <- mutate(...): Fungsi mutate() dari dplyr adalah cara efisien untuk membuat kolom baru tanpa mengubah kolom yang sudah ada. log_pendapatan = log(pendapatan): Membuat kolom baru bernama log_pendapatan yang berisi nilai logaritma dari kolom pendapatan. Transformasi log sering digunakan untuk menormalisasi data yang memiliki sebaran tidak merata, seperti pendapatan. umur_kategori = case_when(...): Ini adalah cara yang fleksibel dan kuat untuk membuat variabel kategorikal baru berdasarkan kondisi. umur < 30 ~ "Muda": Jika umur kurang dari 30, beri label "Muda". umur >= 30 & umur < 40 ~ "Dewasa": Jika umur di antara 30 dan 40, beri label "Dewasa". TRUE

~ "Lansia": Semua nilai lainnya (yang tidak memenuhi kondisi di atas) diberi label "Lansia".

f. Memfilter Data

data_filtered <- filter(...): Menggunakan filter() lagi, tetapi kali ini untuk memilih baris yang diinginkan untuk analisis lebih lanjut. umur >= 18: Menyimpan hanya baris di mana umur setidaknya 18. kategori %in% c("PNS", "Swasta"): Menjaga hanya baris di mana kolom kategori bernilai "PNS" atau "Swasta". Outputnya akan menunjukkan jumlah baris yang tersisa dan beberapa baris pertama dari data yang sudah disaring.

B. PEMBAHASAN LATIHAN

Lihat folder sampel_data yang ada di Google Colab

1. Impor dataset california_housing_tes.csv
2. Tampilkan 6 data di awal.
3. Lakukan pemeriksaan data untuk melihat struktur data, dimensi data, ringkasan data, data hilang, data duplikat, outlier.
4. Lakukan penyiapan data dengan menghapus data hilang dan data yang duplikat.

```
# 1. Impor dataset california_housing_test.csv
cat("=== Impor Dataset ===\n")
data <- read.csv("/content/sample_data/california_housing_test.csv")
cat("Dataset 'california_housing_test.csv' berhasil diimpor.\n")

# 2. Tampilkan 6 data di awal.
cat("\n=== 6 Data Pertama ===\n")
print(head(data))

# 3. Lakukan pemeriksaan data
cat("\n=== Pemeriksaan Data ===\n")

# a. Memeriksa struktur data
cat("\n=== a. Memeriksa Struktur Data ===\n")
str(data)

# b. Memeriksa dimensi data
cat("\n=== b. Memeriksa Dimensi Data ===\n")
cat("Dimensi (baris, kolom):", dim(data), "\n")
cat("Jumlah baris:", nrow(data), "\n")
cat("Jumlah kolom:", ncol(data), "\n")
cat("Nama kolom:", names(data), "\n")

# c. Ringkasan statistik
cat("\n=== c. Ringkasan Statistik ===\n")
summary(data)

# d. Mengecek nilai hilang
cat("\n=== d. Mengecek Nilai Hilang ===\n")
```

```
cat("Jumlah NA per kolom:\n")
print(colSums(is.na(data)))
cat("Jumlah baris lengkap (tanpa NA):", sum(complete.cases(data)), "\n")

# e. Mengecek duplikat
cat("\n=== e. Mengecek Duplikat ===\n")
cat("Jumlah baris duplikat:", sum(duplicated(data)), "\n")
cat("Baris duplikat:\n")
print(data[duplicated(data), ])

# f. Mengecek outlier (contoh pada kolom median_income)
cat("\n=== f. Mengecek Outlier (median_income) ===\n")
Q1 <- quantile(data$median_income, 0.25, na.rm = TRUE)
Q3 <- quantile(data$median_income, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
cat("Batas bawah (Q1 - 1.5 * IQR):", lower_bound, "\n")
cat("Batas atas (Q3 + 1.5 * IQR):", upper_bound, "\n")
outliers <- data$median_income[data$median_income < lower_bound |
                               data$median_income > upper_bound &
!is.na(data$median_income)]
cat("Nilai outlier pada median_income:", outliers, "\n")

# 4. Lakukan penyiapan data

# a. Menghapus data hilang
cat("\n=== a. Menghapus Data Hilang ===\n")
data_no_na <- na.omit(data)
cat("Jumlah baris setelah menghapus NA:", nrow(data_no_na), "\n")

# b. Menghapus data duplikat
cat("\n=== b. Menghapus Data Duplikat ===\n")
data_prepared <- unique(data_no_na) # unique() works for rows in data frames
cat("Jumlah baris setelah menghapus duplikat:", nrow(data_prepared), "\n")

cat("\n=== Data Siap Digunakan (Setelah NA dan Duplikat Dihapus) ===\n")
print(head(data_prepared))
```

Pembahasan: pada latihan pertama kode dimulai dengan memuat dataset yang sudah tersedia di google colab. data <- read.csv(...): Fungsi ini membaca file CSV yang merupakan standar di R. Jalur /content/sample_data/california_housing_test.csv adalah lokasi tetap file sampel yang disediakan oleh Google Colab. Seluruh data dari file ini dimuat ke dalam variabel data. Output: Mengonfirmasi bahwa file telah berhasil diimpor. print(head(data)) Fungsi head() menampilkan 6 baris pertama dari data frame. Ini adalah cara cepat untuk memverifikasi bahwa data telah dibaca dengan benar dan kolom-kolomnya (seperti longitude, latitude, median_income) terlihat seperti yang diharapkan. Output Menampilkan 6 baris pertama dengan 9 kolom yang sesuai, ini menunjukkan data koordinat geografis dan statistik perumahan di California.

Output

```
=== Impor Dataset ===
Dataset 'california_housing_test.csv' berhasil diimpor.

=== 6 Data Pertama ===
  longitude latitude housing_median_age total_rooms total_bedrooms population
1   -122.05    37.37                27        3885            661       1537
2   -118.30    34.26                43        1510            310        809
3   -117.81    33.78                27        3589            507       1484
4   -118.36    33.82                28          67             15         49
5   -119.67    36.33                19        1241            244        850
6   -119.56    36.51                37        1018            213        663
  households median_income median_house_value
1        606        6.6085             344700
2        277        3.5990             176500
3        495        5.7934             270500
4         11        6.1359             330000
5        237        2.9375              81700
6        204        1.6635              67000

=== Pemeriksaan Data ===

=== a. Memeriksa Struktur Data ===
'data.frame':   3000 obs. of  9 variables:
 $ longitude         : num  -122 -118 -118 -118 -120 ...
 $ latitude          : num  37.4 34.3 33.8 33.8 36.3 ...
 $ housing_median_age: num  27 43 27 28 19 37 43 19 15 31 ...
 $ total_rooms       : num  3885 1510 3589 67 1241 ...
 $ total_bedrooms    : num  661 310 507 15 244 213 225 471 617 632 ...
 $ population        : num  1537 809 1484 49 850 ...
 $ households        : num  606 277 495 11 237 204 218 441 599 603 ...
 $ median_income     : num  6.61 3.6 5.79 6.14 2.94 ...
 $ median_house_value: num  344700 176500 270500 330000 81700 ...

=== b. Memeriksa Dimensi Data ===
Dimensi (baris, kolom): 3000 9
Jumlah baris: 3000
Jumlah kolom: 9
Nama kolom: longitude latitude housing_median_age total_rooms total_bedrooms population households median_income median_house_value

=== c. Ringkasan Statistik ===
   longitude         latitude     housing_median_age  total_rooms
 Min.   :-124.2   Min.   :32.56   Min.   : 1.00      Min.   :    6
 1st Qu.:-121.8   1st Qu.:33.93   1st Qu.:18.00      1st Qu.: 1401
 Median :-118.5   Median :34.27   Median :29.00      Median : 2106
 Mean   :-119.6   Mean   :35.64   Mean   :28.85      Mean   : 2600
 3rd Qu.:-118.0   3rd Qu.:37.69   3rd Qu.:37.00      3rd Qu.: 3129
 Max.   :-114.5   Max.   :41.92   Max.   :52.00      Max.   :30450
 total_bedrooms   population      households     median_income
 Min.   :   2   Min.   :    5   Min.   :   2.0   Min.   : 0.4999
 1st Qu.: 291   1st Qu.:  780   1st Qu.: 273.0   1st Qu.: 2.5440
 Median : 437   Median : 1155   Median : 409.5   Median : 3.4872
 Mean   : 530   Mean   : 1403   Mean   : 489.9   Mean   : 3.8073
 3rd Qu.: 636   3rd Qu.: 1743   3rd Qu.: 597.2   3rd Qu.: 4.6565
 Max.   :5419   Max.   :11935   Max.   :4930.0   Max.   :15.0001
 median_house_value
 Min.   : 22500
 1st Qu.:121200
 Median :177650
 Mean   :205846
 3rd Qu.:263975
 Max.   :500001

=== d. Mengecek Nilai Hilang ===
Jumlah NA per kolom:
         longitude           latitude housing_median_age        total_rooms
                 0                  0                  0                  0
    total_bedrooms         population         households      median_income
                 0                  0                  0                  0
median_house_value
                 0
Jumlah baris lengkap (tanpa NA): 3000

=== e. Mengecek Duplikat ===
Jumlah baris duplikat: 0
Baris duplikat:
[1] longitude          latitude           housing_median_age total_rooms
[5] total_bedrooms     population         households         median_income
[9] median_house_value
<0 rows> (or 0-length row.names)

=== f. Mengecek Outlier (median_income) ===
Batas bawah (Q1 - 1.5 * IQR): -0.6247125
Batas atas (Q3 + 1.5 * IQR): 7.825187
Nilai outlier pada median_income: 15.0001 10.1007 13.6623 10.5981 12.6417 8.1926 15.0001 8.5938 9.8214 10.9722 10.476 12.3767 10.9529 11.806 10.8111

=== a. Menghapus Data Hilang ===
Jumlah baris setelah menghapus NA: 3000

=== b. Menghapus Data Duplikat ===
Jumlah baris setelah menghapus duplikat: 3000

=== Data Siap Digunakan (Setelah NA dan Duplikat Dihapus) ===
  longitude latitude housing_median_age total_rooms total_bedrooms population
1   -122.05    37.37                27        3885            661       1537
2   -118.30    34.26                43        1510            310        809
3   -117.81    33.78                27        3589            507       1484
4   -118.36    33.82                28          67             15         49
5   -119.67    36.33                19        1241            244        850
6   -119.56    36.51                37        1018            213        663
  households median_income median_house_value
1        606        6.6085             344700
2        277        3.5990             176500
3        495        5.7934             270500
4         11        6.1359             330000
5        237        2.9375              81700
6        204        1.6635              67000
```

Pembahasan :

a. Memeriksa Struktur Data

str(data) Fungsi str() memberikan ringkasan struktur internal data frame. Output 'data.frame' 3000 obs. of 9 variables Mengonfirmasi bahwa dataset adalah data frame dengan 3000 observasi (baris) dan 9 variabel (kolom). Semua variabel teridentifikasi sebagai tipe num (numerik), yang berarti semua data berupa angka desimal.

b. Memeriksa Dimensi Data

dim(), nrow(), ncol() Fungsi-fungsi ini memberikan informasi dimensi yang lebih spesifik. names() Menampilkan semua nama kolom. Output Semua perintah mengonfirmasi data berukuran 3000 baris dan 9 kolom

c. Ringkasan Statistik

summary(data) Fungsi ini menghitung statistik deskriptif untuk setiap kolom numerik. Output Menunjukkan nilai Min., Max., Median, Mean, Q1, dan Q3 untuk setiap variabel. Misalnya, pada kolom median_income, nilai minimum adalah $0.4999 (ribu USD) dan maksimumnya adalah $15.0001 (ribu USD).

d. Mengecek Nilai Hilang (NA)

colSums(is.na(data)) Menghitung total nilai NA (hilang) per kolom. sum(complete.cases(data)) Menghitung jumlah baris yang lengkap (tidak ada NA). Output Jumlah NA per kolom ... 0 ...: akan menunjukkan bahwa tidak ada nilai hilang sama sekali di dataset ini (semua kolom bernilai 0). Jumlah baris lengkap (tanpa NA) 3000: Karena tidak ada NA, semua baris lengkap.

e. Mengecek Duplikat

sum(duplicated(data)) ini untuk menghitung total baris yang merupakan duplikat penuh. data[duplicated(data), untuk mencetak baris-baris yang duplikat tersebut. Output Jumlah baris duplikat: 0 ini mengonfirmasi bahwa tidak ada baris yang merupakan duplikat di dataset ini.

f. Mengecek Outlier (Median Income)

Perhitungan ini menemukan Q1 (25% kuartil) dan Q3 (75% kuartil), kemudian menggunakan rumus Batas=Q±1.5×IQR. Nilai di luar batas ini dianggap outlier. Output Batas bawah (Q1 - 1.5 * IQR): -0.6247125 Batas atas (Q3 + 1.5 * IQR): 7.825187 Nilai outlier pada median_income: 15.0001 10.1007 ...: Karena Batas Atas hanya $7.825, semua nilai median_income yang lebih besar dari itu dianggap outlier. Output menunjukkan sejumlah nilai pendapatan yang sangat tinggi (di atas $7.825, atau $78.250 USD) yang terdeteksi sebagai outlier statistik.

Penyiapan data sederhana meskipun pemeriksaan data menunjukkan tidak ada NA atau duplikat, bagian ini akan mendemonstrasikan bagaimana melakukan pembersihan jika seandainya ada.

a. Menghapus Data Hilang

data_no_na <- na.omit(data): Menghapus semua baris yang mengandung NA. Outputnya Jumlah baris setelah menghapus NA: 3000. Karena data awal bersih dari NA, jumlah baris tidak berubah.

b. Menghapus Data Duplikat

data_prepared <- unique(data_no_na): Fungsi unique() akan menghapus baris duplikat. Outputnya Jumlah baris setelah menghapus duplikat: 3000. Karena data awal bersih dari duplikat, jumlah baris tidak berubah.

C. PEMBAHASAN TUGAS

Download dan simpan data pada Google Colab dataset pada link https://www.kaggle.com/datasets/mohankrishnathalla/diabetes-health-indicators-dataset.

1. Impor dataset tersebut
2. Tampilkan 6 data di awal.
3. Lakukan pemeriksaan data untuk melihat struktur data, dimensi data, ringkasan data, data hilang, data duplikat, outlier.n
4. Lakukan penyiapan data dengan menghapus data hilang dan data yang duplikat.

```
# 1. Impor dataset
cat("=== 1. Impor Dataset ===\n")
# Sesuaikan nama file jika berbeda
data <- read_csv("/content//diabetes_dataset.csv") # Updated file path
cat("Dataset '/content//diabetes_dataset.csv' berhasil diimpor.\n") # Updated
success message

# 2. Tampilkan 6 data di awal
cat("\n=== 2. Tampilkan 6 Data Pertama (Head) ===\n")
print(head(data))

# 3. Lakukan pemeriksaan data (EDA - Exploratory Data Analysis)
cat("\n=== 3. Pemeriksaan Data (EDA) ===\n")

# a. Memeriksa struktur data
cat("\n=== a. Struktur Data (str) ===\n")
str(data)

# b. Memeriksa dimensi data
cat("\n=== b. Dimensi Data (rows, columns) ===\n")
cat("Dimensi (baris, kolom):", dim(data), "\n")
cat("Jumlah baris:", nrow(data), "\n")
cat("Jumlah kolom:", ncol(data), "\n")
cat("Nama kolom:", names(data), "\n")

# c. Ringkasan statistik
cat("\n=== c. Ringkasan Statistik (summary) ===\n")
print(summary(data))

# d. Mengecek nilai hilang (NA)
cat("\n=== d. Mengecek Nilai Hilang (NA) ===\n")
cat("Jumlah NA per kolom:\n")
print(colSums(is.na(data)))
cat("Jumlah baris lengkap (tanpa NA):", sum(complete.cases(data)), "\n")

# e. Mengecek duplikat
cat("\n=== e. Mengecek Duplikat ===\n")
total_duplikat <- sum(duplicated(data))
cat("Jumlah baris duplikat:", total_duplikat, "\n")
if (total_duplikat > 0) {
    cat("Beberapa contoh baris duplikat:\n")
    print(data[duplicated(data), ] |> head())
} else {
    cat("Tidak ditemukan baris duplikat.\n")
```

```
}

# f. Mengecek outlier (Contoh: BMI)
cat("\n=== f. Mengecek Outlier (BMI) ===\n")
# Perhitungan IQR
Q1_bmi <- quantile(data$BMI, 0.25, na.rm = TRUE)
Q3_bmi <- quantile(data$BMI, 0.75, na.rm = TRUE)
IQR_bmi <- Q3_bmi - Q1_bmi
lower_bound_bmi <- Q1_bmi - 1.5 * IQR_bmi
upper_bound_bmi <- Q3_bmi + 1.5 * IQR_bmi

cat("Batas bawah BMI (Q1 - 1.5 * IQR):", lower_bound_bmi, "\n")
cat("Batas atas BMI (Q3 + 1.5 * IQR):", upper_bound_bmi, "\n")

# Identifikasi outlier
outliers_bmi <- data$BMI[data$BMI < lower_bound_bmi | data$BMI > upper_bound_bmi]
cat("Jumlah outlier BMI yang terdeteksi:", length(outliers_bmi), "\n")
cat("Contoh nilai outlier BMI:", unique(outliers_bmi) |> head(10), "\n")

# 4. Lakukan penyiapan data (Data Cleaning)
cat("\n=== 4. Penyiapan Data (Pembersihan) ===\n")

# a. Menghapus data hilang (NA)
data_no_na <- na.omit(data)
cat("Jumlah baris setelah menghapus NA:", nrow(data_no_na), " (Awal: ",
nrow(data), ")\n")

# b. Menghapus data duplikat
# Gunakan fungsi distinct() dari dplyr untuk menghapus baris duplikat
data_prepared <- data_no_na %>% distinct()

cat("Jumlah baris setelah menghapus duplikat:", nrow(data_prepared), "\n")

cat("\n=== Data Siap Digunakan ===\n")
cat("Dataset akhir yang bersih memiliki", nrow(data_prepared), "baris dan",
ncol(data_prepared), "kolom.\n")
print(head(data_prepared))
```

Output

```
=== 1. Impor Dataset ===
Rows: 100000 Columns: 31
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (7): gender, ethnicity, education_level, income_level, employment_statu...
dbl (24): age, alcohol_consumption_per_week, physical_activity_minutes_per_w...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
Dataset '/content//diabetes_dataset.csv' berhasil diimpor.

=== 2. Tampilkan 6 Data Pertama (Head) ===
# A tibble: 6 × 31
    age gender ethnicity education_level income_level employment_status
  <dbl> <chr>  <chr>     <chr>           <chr>        <chr>
1    58 Male   Asian     Highschool      Lower-Middle Employed
2    48 Female White     Highschool      Middle       Employed
3    60 Male   Hispanic  Highschool      Middle       Unemployed
4    74 Female Black     Highschool      Low          Retired
5    46 Male   White     Graduate        Middle       Retired
6    46 Female White     Highschool      Upper-Middle Employed
# ℹ 25 more variables: smoking_status <chr>,
#   alcohol_consumption_per_week <dbl>,
#   physical_activity_minutes_per_week <dbl>, diet_score <dbl>,
#   sleep_hours_per_day <dbl>, screen_time_hours_per_day <dbl>,
#   family_history_diabetes <dbl>, hypertension_history <dbl>,
#   cardiovascular_history <dbl>, bmi <dbl>, waist_to_hip_ratio <dbl>,
#   systolic_bp <dbl>, diastolic_bp <dbl>, heart_rate <dbl>, …
```

Pembahasan : Output menunjukkan bahwa sudah berhasil mengimpor dataset dengan sukses. Baris Rows: 100000 Columns: 31 dan Dataset '/content//diabetes_dataset.csv' berhasil diimpor ini adalah konfirmasi bahwa data telah dimuat ke dalam R. Output ini juga memberikan tipe data kolom-kolom. R secara otomatis mendeteksi bahwa ada 7 kolom bertipe chr (karakter) seperti gender dan ethnicity, serta 24 kolom bertipe dbl (double/numerik) seperti age dan bmi.

Tampilan awal menunjukkan dataset yang mencakup demografi (age, gender, ethnicity), faktor gaya hidup (smoking_status, alcohol_consumption_per_week, sleep_hours_per_day), dan indikator klinis rinci (bmi, systolic_bp, hba1c, insulin_level). Variabel target utama kemungkinan adalah diagnosed_diabetes (0 atau 1) atau diabetes_stage (karakter).879kjl

```
=== 3. Pemeriksaan Data (EDA) ===

=== a. Struktur Data (str) ===
spc_tbl_ [100,000 × 31] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ age                              : num [1:100000] 58 48 60 74 46 46 75 62 42 59 ...
 $ gender                           : chr [1:100000] "Male" "Female" "Male" "Female" ...
 $ ethnicity                        : chr [1:100000] "Asian" "White" "Hispanic" "Black" ...
 $ education_level                  : chr [1:100000] "Highschool" "Highschool" "Highschool" "Highschool" ...
 $ income_level                     : chr [1:100000] "Lower-Middle" "Middle" "Middle" "Low" ...
 $ employment_status                : chr [1:100000] "Employed" "Employed" "Unemployed" "Retired" ...
 $ smoking_status                   : chr [1:100000] "Never" "Former" "Never" "Never" ...
 $ alcohol_consumption_per_week     : num [1:100000] 0 1 1 0 1 2 0 1 1 3 ...
 $ physical_activity_minutes_per_week: num [1:100000] 215 143 57 49 109 124 53 75 114 86 ...
 $ diet_score                       : num [1:100000] 5.7 6.7 6.4 3.4 7.2 9 9.2 4.1 6.7 8.2 ...
 $ sleep_hours_per_day              : num [1:100000] 7.9 6.5 10 6.6 7.4 6.2 7.8 9 8.5 5.3 ...
 $ screen_time_hours_per_day        : num [1:100000] 7.9 8.7 8.1 5.2 5 5.4 8 12.9 8.5 7.4 ...
 $ family_history_diabetes          : num [1:100000] 0 0 1 0 0 0 0 0 0 0 ...
 $ hypertension_history             : num [1:100000] 0 0 0 0 0 0 1 1 0 0 ...
 $ cardiovascular_history           : num [1:100000] 0 0 0 0 0 0 0 1 1 0 ...
 $ bmi                              : num [1:100000] 30.5 23.1 22.2 26.8 21.2 26.1 25.1 23.9 24.7 26.7 ...
 $ waist_to_hip_ratio               : num [1:100000] 0.89 0.8 0.81 0.88 0.78 0.85 0.88 0.86 0.84 0.81 ...
 $ systolic_bp                      : num [1:100000] 134 129 115 120 92 95 129 128 103 124 ...
 $ diastolic_bp                     : num [1:100000] 78 76 73 93 67 81 77 83 71 81 ...
 $ heart_rate                       : num [1:100000] 68 67 74 68 67 57 81 76 72 70 ...
 $ cholesterol_total                : num [1:100000] 239 116 213 171 210 218 238 241 187 188 ...
 $ hdl_cholesterol                  : num [1:100000] 41 55 66 50 52 61 46 49 33 52 ...
 $ ldl_cholesterol                  : num [1:100000] 160 50 99 79 125 119 161 159 132 103 ...
 $ triglycerides                    : num [1:100000] 145 30 36 140 160 179 155 120 98 104 ...
 $ glucose_fasting                  : num [1:100000] 136 93 118 139 137 100 101 110 116 76 ...
 $ glucose_postprandial             : num [1:100000] 236 150 195 253 184 133 100 189 172 109 ...
 $ insulin_level                    : num [1:100000] 6.36 2 5.07 5.28 12.74 ...
 $ hba1c                            : num [1:100000] 8.18 5.63 7.51 9.03 7.2 6.03 5.24 7.04 6.9 4.99 ...
 $ diabetes_risk_score              : num [1:100000] 29.6 23 44.7 38.2 23.5 23.5 36.1 34.2 26.7 30 ...
 $ diabetes_stage                   : chr [1:100000] "Type 2" "No Diabetes" "Type 2" "Type 2" ...
 $ diagnosed_diabetes               : num [1:100000] 1 0 1 1 1 0 0 1 1 0 ...
```

Pembahasan : A. Struktur Data (str) Tujuan: Memeriksa tipe data dari 31 kolom yang ada.. Variabel seperti age, bmi, hba1c, dan tekanan darah (systolic_bp, diastolic_bp) dikategorikan sebagai numerik (num atau dbl). Variabel seperti gender dan education_level dikategorikan sebagai karakter (chr).

```
=== b. Dimensi Data (rows, columns) ===
Dimensi (baris, kolom): 100000 31
Jumlah baris: 100000
Jumlah kolom: 31
Nama kolom: age gender ethnicity education_level income_level employment_status smoking_status alcohol_consumpt:

=== c. Ringkasan Statistik (summary) ===
      age           gender           ethnicity         education_level
 Min.   :18.00   Length:100000      Length:100000      Length:100000
 1st Qu.:39.00   Class :character   Class :character   Class :character
 Median :50.00   Mode  :character   Mode  :character   Mode  :character
 Mean   :50.12
 3rd Qu.:61.00
 Max.   :90.00
 income_level       employment_status  smoking_status
 Length:100000      Length:100000      Length:100000
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character
```

Pembahasan :

b. Dimensi Data. Dimensi (baris, kolom): 100000 31: Ini menunjukkan bahwa dataset yang dianalisis memiliki 100.000 baris (observasi) dan 31 kolom (variabel). Jumlah baris: 100000: Menegaskan jumlah total data yang ada, yang merupakan dataset berukuran sangat besar dan ideal untuk pemodelan machine learning. Jumlah kolom: 31: Mengindikasikan kekayaan data, di mana setiap observasi memiliki 31 atribut atau fitur yang berbeda. Nama kolom: Daftar nama kolom yang sangat beragam, mencakup informasi demografis, gaya hidup, hingga data klinis yang spesifik.

c. Ringkasan Statistik

Bagian ini memberikan gambaran umum tentang distribusi dan tipe data setiap kolom. age: Merupakan variabel numerik (kuantitatif) dengan rentang usia dari 18 hingga 90 tahun, dan rata-rata usia sekitar 50.12 tahun. Distribusi ini menunjukkan bahwa dataset mencakup berbagai kelompok usia dewasa. gender, ethnicity, education_level, income_level, employment_status, smoking_status: Semua kolom ini adalah variabel karakter (kategorikal). Output Length:100000, Class:character, dan Mode:character menegaskan bahwa variabel-variabel ini menyimpan data tekstual atau kategori, bukan angka.

```
 alcohol_consumption_per_week physical_activity_minutes_per_week
 Min.   : 0.000               Min.   :  0.0
 1st Qu.: 1.000               1st Qu.: 57.0
 Median : 2.000               Median :100.0
 Mean   : 2.004               Mean   :118.9
 3rd Qu.: 3.000               3rd Qu.:160.0
 Max.   :10.000               Max.   :833.0
   diet_score     sleep_hours_per_day screen_time_hours_per_day
 Min.   : 0.000   Min.   : 3.000      Min.   : 0.500
 1st Qu.: 4.800   1st Qu.: 6.300      1st Qu.: 4.300
 Median : 6.000   Median : 7.000      Median : 6.000
 Mean   : 5.995   Mean   : 6.998      Mean   : 5.996
 3rd Qu.: 7.200   3rd Qu.: 7.700      3rd Qu.: 7.700
 Max.   :10.000   Max.   :10.000      Max.   :16.800
 family_history_diabetes hypertension_history cardiovascular_history
 Min.   :0.0000          Min.   :0.0000       Min.   :0.0000
 1st Qu.:0.0000          1st Qu.:0.0000       1st Qu.:0.0000
 Median :0.0000          Median :0.0000       Median :0.0000
 Mean   :0.2194          Mean   :0.2508       Mean   :0.0792
 3rd Qu.:0.0000          3rd Qu.:1.0000       3rd Qu.:0.0000
 Max.   :1.0000          Max.   :1.0000       Max.   :1.0000
      bmi        waist_to_hip_ratio  systolic_bp     diastolic_bp
 Min.   :15.00   Min.   :0.6700     Min.   : 90.0   Min.   : 50.00
 1st Qu.:23.20   1st Qu.:0.8200     1st Qu.:106.0   1st Qu.: 70.00
 Median :25.60   Median :0.8600     Median :116.0   Median : 75.00
 Mean   :25.61   Mean   :0.8561     Mean   :115.8   Mean   : 75.23
 3rd Qu.:28.00   3rd Qu.:0.8900     3rd Qu.:125.0   3rd Qu.: 81.00
 Max.   :39.20   Max.   :1.0600     Max.   :179.0   Max.   :110.00
   heart_rate     cholesterol_total hdl_cholesterol ldl_cholesterol
```

```
      Min.   : 40.00   Min.   :100       Min.   :20.00   Min.   : 50
      1st Qu.: 64.00   1st Qu.:164       1st Qu.:47.00   1st Qu.: 78
      Median : 70.00   Median :186       Median :54.00   Median :102
      Mean   : 69.63   Mean   :186       Mean   :54.04   Mean   :103
      3rd Qu.: 75.00   3rd Qu.:208       3rd Qu.:61.00   3rd Qu.:126
      Max.   :105.00   Max.   :318       Max.   :98.00   Max.   :263
      triglycerides   glucose_fasting glucose_postprandial insulin_level
      Min.   : 30.0   Min.   : 60.0   Min.   : 70         Min.   : 2.000
      1st Qu.: 91.0   1st Qu.:102.0   1st Qu.:139         1st Qu.: 5.090
      Median :121.0   Median :111.0   Median :160         Median : 8.790
      Mean   :121.5   Mean   :111.1   Mean   :160         Mean   : 9.061
      3rd Qu.:151.0   3rd Qu.:120.0   3rd Qu.:181         3rd Qu.:12.450
      Max.   :344.0   Max.   :172.0   Max.   :287         Max.   :32.220
          hba1c       diabetes_risk_score diabetes_stage     diagnosed_diabetes
      Min.   :4.000   Min.   : 2.70       Length:100000      Min.   :0.0
      1st Qu.:5.970   1st Qu.:23.80       Class :character   1st Qu.:0.0
      Median :6.520   Median :29.00       Mode  :character   Median :1.0
      Mean   :6.521   Mean   :30.22                          Mean   :0.6
      3rd Qu.:7.070   3rd Qu.:35.60                          3rd Qu.:1.0
      Max.   :9.800   Max.   :67.20                          Max.   :1.0
```

Pembahasan : dilihat dari factor gaya hidup, Riwayat Kesehatan, biometrik dan klinis, indicator diabetes langsung, varibel hasil . Variabel riwayat kesehatan ini bersifat biner (0 atau 1). Nilai rata-rata secara langsung menunjukkan proporsi responden yang memiliki riwayat tersebut. Proporsi hipertensi (25%) dan riwayat keluarga diabetes (22%) cukup signifikan dan akan menjadi prediktor kuat dalam model diabetes. Rata-rata HbA1c 6.521% berada tepat di ambang diagnosis diabetes (≥6.5%). Kuartil 1 (5.97%) menunjukkan 25% populasi berada di zona Pradiabetes (5.7−6.4%).

```
=== d. Mengecek Nilai Hilang (NA) ===
Jumlah NA per kolom:
                              age                            gender
                                0                                 0
                        ethnicity                   education_level
                                0                                 0
                     income_level                  employment_status
                                0                                 0
                   smoking_status       alcohol_consumption_per_week
                                0                                 0
physical_activity_minutes_per_week                        diet_score
                                0                                 0
              sleep_hours_per_day          screen_time_hours_per_day
                                0                                 0
          family_history_diabetes               hypertension_history
                                0                                 0
           cardiovascular_history                               bmi
                                0                                 0
               waist_to_hip_ratio                        systolic_bp
                                0                                 0
                     diastolic_bp                         heart_rate
                                0                                 0
              cholesterol_total                    hdl_cholesterol
                                0                                 0
                ldl_cholesterol                       triglycerides
                                0                                 0
                glucose_fasting              glucose_postprandial
                                0                                 0
                  insulin_level                             hba1c
                                0                                 0
             diabetes_risk_score                    diabetes_stage
                                0                                 0
              diagnosed_diabetes
                                0
```

Pembahasan : dapat dilihat tidak ada satupun nilai hilang yang secara eksplisit ditandai sebagai NA di seluruh baris. Hal ini menghilangkan kebutuhan untuk imputasi, menjadikan proses pemodelan lebih cepat dan akurat.

```
Jumlah baris lengkap (tanpa NA): 100000

=== e. Mengecek Duplikat ===
Jumlah baris duplikat: 0
Tidak ditemukan baris duplikat.

=== f. Mengecek Outlier (BMI) ===
Warning message:
“Unknown or uninitialised column: `BMI`.”
Warning message:
“Unknown or uninitialised column: `BMI`.”
Batas bawah BMI (Q1 - 1.5 * IQR): NA
Batas atas BMI (Q3 + 1.5 * IQR): NA
Warning message:
“Unknown or uninitialised column: `BMI`.”
Warning message:
“Unknown or uninitialised column: `BMI`.”
Warning message:
“Unknown or uninitialised column: `BMI`.”
Jumlah outlier BMI yang terdeteksi: 0
Contoh nilai outlier BMI:
```

Pembahasan : Dataset ini bersih dari duplikasi baris sempurna. Setiap observasi (pasien) unik, menjamin bahwa model yang dilatih tidak akan bias oleh data yang berulang.

```
=== 4. Penyiapan Data (Pembersihan) ===
Jumlah baris setelah menghapus NA: 100000  (Awal:  100000 )
Jumlah baris setelah menghapus duplikat: 100000

=== Data Siap Digunakan ===
Dataset akhir yang bersih memiliki 100000 baris dan 31 kolom.
# A tibble: 6 × 31
    age gender ethnicity education_level income_level employment_status
  <dbl> <chr>  <chr>     <chr>           <chr>        <chr>
1    58 Male   Asian     Highschool      Lower-Middle Employed
2    48 Female White     Highschool      Middle       Employed
3    60 Male   Hispanic  Highschool      Middle       Unemployed
4    74 Female Black     Highschool      Low          Retired
5    46 Male   White     Graduate        Middle       Retired
6    46 Female White     Highschool      Upper-Middle Employed
# i 25 more variables: smoking_status <chr>,
#   alcohol_consumption_per_week <dbl>,
#   physical_activity_minutes_per_week <dbl>, diet_score <dbl>,
#   sleep_hours_per_day <dbl>, screen_time_hours_per_day <dbl>,
#   family_history_diabetes <dbl>, hypertension_history <dbl>,
#   cardiovascular_history <dbl>, bmi <dbl>, waist_to_hip_ratio <dbl>,
#   systolic_bp <dbl>, diastolic_bp <dbl>, heart_rate <dbl>, …
```

Pembahasan :

a. Menghapus Nilai Hilang (NA) Langkah ini memastikan bahwa tidak ada observasi yang terhapus. Hal ini mengonfirmasi temuan di langkah EDA (3.d) bahwa dataset sangat bersih dan tidak mengandung nilai hilang (NA) eksplisit.

b. Menghapus Duplikat Langkah ini mengonfirmasi temuan EDA (3.e) bahwa tidak ada baris yang merupakan duplikat sempurna. Setiap observasi dalam dataset adalah unik, menjamin bahwa model tidak akan mengalami bias yang disebabkan oleh entri data yang ganda.

Data Siap Digunakan Integritas Data Terjaga Dataset akhir mempertahankan ukuran penuhnya: 100.000 baris dan 31 kolom. Ini berarti proses pembersihan tidak mengakibatkan kehilangan data. Kualitas Data ini telah terverifikasi tidak memiliki masalah nilai hilang eksplisit maupun duplikat. Selain itu, berdasarkan ringkasan statistik (3.c), data ini juga bersih dari masalah umum "nilai 0 sebagai nilai hilang" pada metrik klinis.

D. KESIMPULAN

Kesimpulan dari semua praktikum yang sudah dikerjakan adalah saya berhasil mempraktikkan alur kerja standar dalam pengelolaan data menggunakan R di Google Colab. Awalnya, saya menggunakan data sederhana untuk mempelajari langkah-langkah dasar seperti membuat data frame, memeriksa struktur, dimensi, serta mendeteksi nilai hilang, duplikat, dan outlier. selanjutnya adalah menerapkan pengetahuan dasar ini pada dataset yang lebih besar dan realistis, yaitu dataset diabetes. Proses ini mengonfirmasi bahwa dataset tersebut memiliki kualitas yang sangat tinggi, tidak mengandung nilai hilang atau duplikat, yang merupakan hal langka dan sangat menguntungkan.