



Pengantar Komputasi Statistika

#11 Meeting

Data Frame & Apply

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 199005202024061001

Data Frame adalah struktur data 2 dimensi (baris dan kolom) yang paling penting dan paling umum di R.

- Ini adalah representasi "tabel" atau "spreadsheet" (seperti Excel).
- Perbedaan Kunci vs Matriks:
 - Matriks: Semua elemen harus bertipe data SAMA (misal: semua numeric).
 - Data Frame: Setiap kolom bisa memiliki tipe data BERBEDA. (Kolom 1 character, Kolom 2 numeric, Kolom 3 logical).
- Semua kolom dalam Data Frame harus memiliki panjang (jumlah baris) yang sama.
- "Beda Panjang" (Error): R akan GAGAL jika Anda mencoba menggabungkan vektor yang panjangnya berbeda secara struktural.
- "Data Hilang" (NA / Sel Kosong): Ini BUKAN beda panjang. NA tetap dihitung sebagai satu elemen. Vektornya tetap memiliki panjang yang sama.

Membuat Data Frame

1. Buat vektor-vektornya dulu (harus sama panjang!)

```
vektor_npm <- c("1001", "1002", "1003")
```

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

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

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

2. Gabungkan menjadi data frame

```
df_mahasiswa <- data.frame(
```

```
  NPM = vektor_npm,
```

```
  Nama = vektor_nama,
```

```
  Nilai = vektor_nilai,
```

```
  Lulus = vektor_lulus
```

```
)
```

3. Cetak untuk melihat hasilnya

```
print(df_mahasiswa)
```

4. Cek struktur (PENTING!)

```
str(df_mahasiswa)
```

```
> print(df_mahasiswa)
```

	NPM	Nama	Nilai	Lulus
1	1001	Budi	85	TRUE
2	1002	Ani	78	TRUE
3	1003	Citra	92	TRUE

```
>
```

```
> # 4. Cek struktur (PENTING!)
```

```
> str(df_mahasiswa)
```

```
'data.frame':  3 obs. of  4 variables:  
 $ NPM   : chr  "1001" "1002" "1003"  
 $ Nama  : chr  "Budi" "Ani" "Citra"  
 $ Nilai : num  85 78 92  
 $ Lulus : logi  TRUE TRUE TRUE
```

Membuat Data Frame

AKAN GAGAL: arguments imply differing number of rows: 3, 2
`data.frame>Nama = c("A", "B", "C"), Nilai = c(90, 80))`

AKAN BERHASIL: Panjang kedua vektor sama-sama 3
`data.frame>Nama = c("A", "B", "C"), Nilai = c(90, 80, NA))`



Impor Data Frame dari Excel

```
# Muat 'library' setiap kali memulai R
library(readxl)

# Tentukan path (lokasi) file
# (Perhatikan, kita ganti \ menjadi / agar R bisa
membacanya)
file_path <- "C:/Users/user/OneDrive -
untirta.ac.id/UNTIRTA/Bahan Ajar/Pengantar Komputasi
Statistika/Github PKS/DataFrame_Perkuliahan.xlsx"

# Membaca SHEET 1 ("Nilai_UTS")
df_uts <- read_excel(file_path, sheet = "Nilai_UTS")

# Cek hasil impor
print(df_uts)
str(df_uts)
```

```
> print(df_uts)
# A tibble: 4 × 4
  NIM Nama   Nilai_UTS Kehadiran_Persen
  <dbl> <chr>   <dbl>         <dbl>
1  1001 Budi     85           100
2  1002 Ani     78           90
3  1003 Citra   92           95
4  1004 Doni    65           80

> str(df_uts)
tibble [4 × 4] (S3: tbl_df/tbl/data.frame)
 $ NIM          : num [1:4] 1001 1002 1003 1004
 $ Nama         : chr [1:4] "Budi" "Ani" "Citra" "Doni"
 $ Nilai_UTS    : num [1:4] 85 78 92 65
 $ Kehadiran_Persen: num [1:4] 100 90 95 80
```



Buat Data Frame dari Matrix

ingat: matriks hanya punya 1 tipe data.

```
# 1. Buat sebuah matriks (semua isinya 'numeric')
mat_nilai <- matrix(c(80, 85, 90, 95), nrow = 2, ncol = 2)
colnames(mat_nilai) <- c("UTS", "UAS")
#   UTS UAS
# [1,] 80 90
# [2,] 85 95
```

```
# 2. Konversi ke data frame
df_dari_matriks <- as.data.frame(mat_nilai)
```

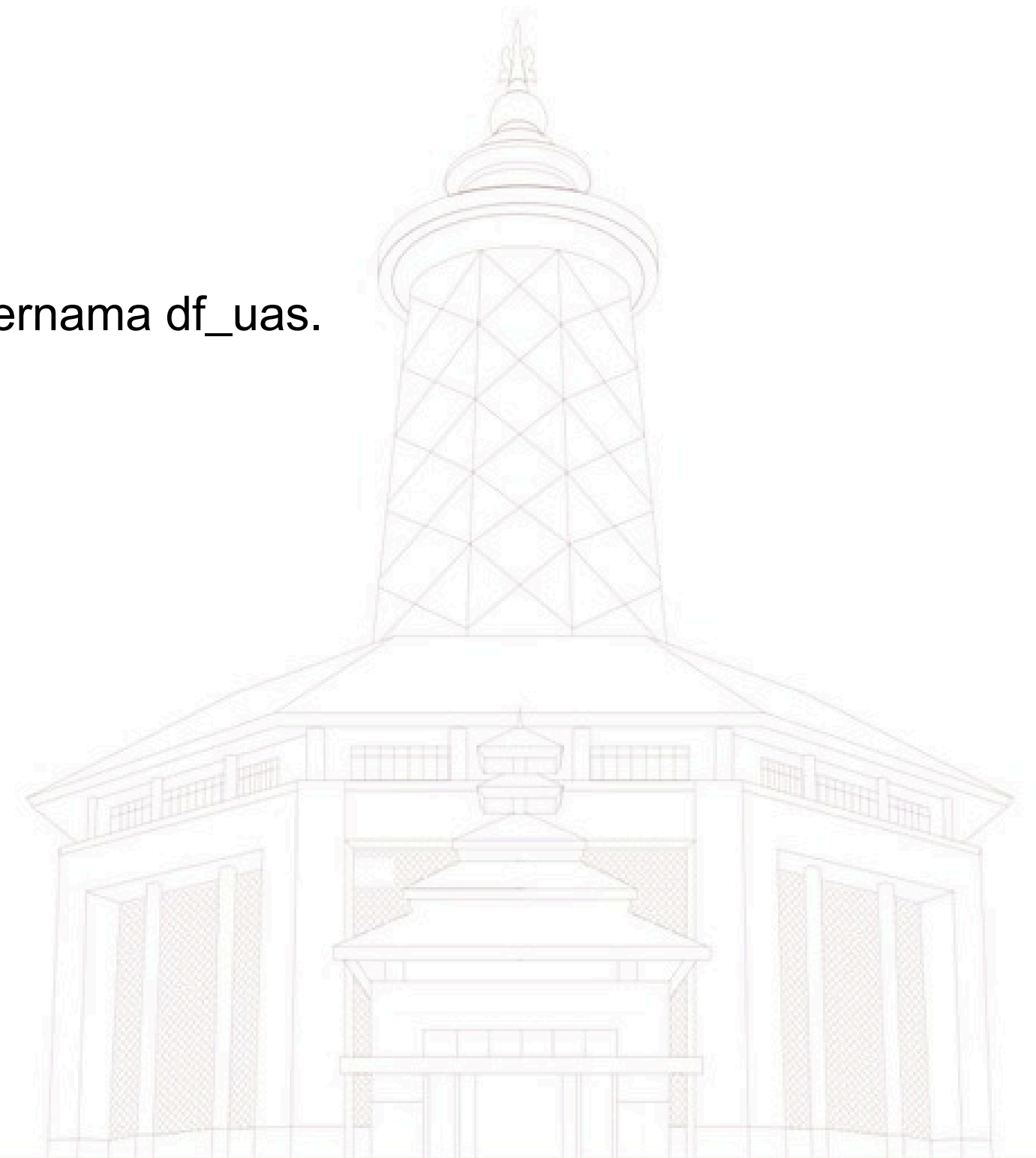
```
# 3. Cek struktur
str(df_dari_matriks)
# 'data.frame': 2 obs. of 2 variables:
# $ UTS: num 80 85
# $ UAS: num 90 95
# (Semua kolom adalah numeric, sesuai asalnya)
```


Impor Data Frame dari Excel

Latihan

Tugas:

1. Gunakan variabel `file_path` yang sama dari slide sebelumnya.
2. Impor data dari sheet bernama "Nilai_UAS" ke dalam variabel baru bernama `df_uas`.
3. Cetak `df_uas` ke console.
4. Periksa strukturnya menggunakan `str(df_uas)`.



Impor Data Frame dari Excel

Jawaban Latihan

```
library(readxl)
# Path lengkap ke file Anda
file_path <- "C:/Users/user/OneDrive - untirta.ac.id/UNTIRTA/Bahan Ajar/Pengantar Komputasi Statistika/Github
PKS/DataFrame_Perkuliahan.xlsx"

# 1. Impor sheet "Nilai_UAS"
df_uas <- read_excel(file_path, sheet = "Nilai_UAS")

# 2. Cetak
print(df_uas)

# 3. Cek struktur
str(df_uas)
```


Indeks Data Frame

Cara 1: Simbol \$ (Paling Umum) Digunakan untuk mengakses satu kolom berdasarkan NAMA-nya. Hasilnya adalah vektor.

```
# Mengambil kolom 'Nama'
```

```
df_uts$Nama
```

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

```
# Menghitung rata-rata kolom 'Nilai_UTS'
```

```
mean(df_uts$Nilai_UTS)
```

```
# [1] 80
```

Cara 2: Kurung Siku Ganda [[...]] Fungsinya identik dengan \$.

```
# Mengambil kolom 'Nama' (hasilnya vektor)
```

```
df_uts[["Nama"]]
```

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

Indeks Data Frame

Cara 3: Kurung Siku Tunggal [...] (Paling Fleksibel) Berfungsi seperti matriks: df[baris, kolom]

(Kita pakai data df_uts)

NIM Nama Nilai_UTS Kehadiran_Persen

[1,] 1001 Budi 85 100

[2,] 1002 Ani 78 90

[3,] 1003 Citra 92 95

[4,] 1004 Doni 65 80

Mengambil Baris 1, SEMUA kolom

df_uts[1,]

NPM Nama Nilai_UTS Kehadiran_Persen

1 1001 Budi 85 100

Mengambil SEMUA baris, Kolom 2 ('Nama')

df_uts[, 2]

[1] "Budi" "Ani" "Citra" "Doni"

Mengambil Baris 3, Kolom 3 (Satu sel)

df_uts[3, 3]

[1] 92

Mengambil Baris 1 & 3, Kolom 2 & 3 (Sub-tabel)

df_uts[c(1, 3), c(2, 3)]

Nama Nilai_UTS

1 Budi 85

3 Citra 92

Menambah dan Menghapus Kolom

Menambah Kolom Baru Gunakan \$ dan operator <-

(Kita pakai data df_uts)

1. Tambah kolom 'Status' berdasarkan 'Nilai_UTS'

```
df_uts$Status <- ifelse(df_uts$Nilai_UTS >= 70, "Lulus",  
"Gagal")
```

```
print(df_uts)
```

```
# NPM Nama Nilai_UTS Kehadiran_Persen Status
```

```
# 1 1001 Budi      85      100 Lulus
```

```
# 2 1002 Ani       78       90 Lulus
```

```
# 3 1003 Citra     92       95 Lulus
```

```
# 4 1004 Doni      65       80 Gagal
```

```
> print(df_uts)
```

```
# A tibble: 4 × 5
```

	NIM	Nama	Nilai_UTS	Kehadiran_Persen	Status
	<dbl>	<chr>	<dbl>	<dbl>	<chr>
1	1001	Budi	85	100	Lulus
2	1002	Ani	78	90	Lulus
3	1003	Citra	92	95	Lulus
4	1004	Doni	65	80	Gagal

Menambah dan Menghapus Kolom

Menghapus Kolom Tetapkan kolom tersebut menjadi NULL.

2. Kita hapus kolom 'Kehadiran_Persen'

```
df_uts$Kehadiran_Persen <- NULL
```

```
print(df_uts)
```

```
# NPM Nama Nilai_UTS Status
```

```
# 1 1001 Budi      85 Lulus
```

```
# 2 1002 Ani       78 Lulus
```

```
# 3 1003 Citra     92 Lulus
```

```
# 4 1004 Doni      65 Gagal
```

```
> print(df_uts)
```

```
# A tibble: 4 × 4
```

	NIM	Nama	Nilai_UTS	Status
	<dbl>	<chr>	<dbl>	<chr>
1	1001	Budi	85	Lulus
2	1002	Ani	78	Lulus
3	1003	Citra	92	Lulus
4	1004	Doni	65	Gagal

Fungsi apply ()

Fungsi apply() adalah cara cepat untuk menjalankan fungsi (seperti mean, sum, max) pada setiap baris atau setiap kolom dari sebuah MATRIKS (atau data frame numerik).

Struktur: apply(X, MARGIN, FUN)

- X: Matriks atau Data Frame (yang akan diproses).
- MARGIN:
 - 1 = Operasi dilakukan per BARIS (horizontal).
 - 2 = Operasi dilakukan per KOLOM (vertikal).
- FUN: Fungsi yang ingin diterapkan (misal: sum, mean, min, max).



apply () pada Matrix

Buat matriks 3x2

```
mat <- matrix(1:6, nrow = 3, ncol = 2)
```

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

```
# [1,]  1  4
```

```
# [2,]  2  5
```

```
# [3,]  3  6
```

Contoh 1: Hitung JUMLAH per BARIS (MARGIN = 1)

```
apply(mat, 1, sum)
```

```
# [1] 5 7 9 (Hasil dari: 1+4, 2+5, 3+6)
```

Contoh 2: Hitung RATA-RATA per KOLOM (MARGIN = 2)

```
apply(mat, 2, mean)
```

```
# [1] 2 5 (Hasil dari: (1+2+3)/3 , (4+5+6)/3)
```

```
> apply(mat, 1, sum)
```

```
[1] 5 7 9
```

```
> # [1] 5 7 9 (Hasil dari: 1+4, 2+5, 3+6)
```

```
>
```

```
> # Contoh 2: Hitung RATA-RATA per KOLOM (MARGIN = 2)
```

```
> apply(mat, 2, mean)
```

```
[1] 2 5
```

```
> # [1] 2 5 (Hasil dari: (1+2+3)/3 , (4+5+6)/3)
```

apply () pada Data Frame

apply() bisa digunakan pada Data Frame, TAPI ia akan memaksa (coerce) data frame menjadi matriks terlebih dahulu.

- Jika ada kolom character (seperti 'Nama'), semua kolom numeric akan diubah menjadi character juga!
- Ini akan menyebabkan fungsi mean atau sum GAGAL.

```
# (Kita pakai data df_uts)
# Ini akan GAGAL karena ada kolom 'Nama' (character)
apply(df_uts, 2, mean)
# Error: 'trim' must be numeric...
```

```
# Sekarang jalankan apply pada data frame numerik tersebut
apply(df_uts_numerik, 2, mean)
#      Nilai_UTS Kehadiran_Persen
#      80.0      91.25
```

```
# CARA BENAR: Pilih HANYA kolom numerik terlebih dahulu
df_uts_numerik <- df_uts[, c("Nilai_UTS",
"Kehadiran_Persen")]
#      Nilai_UTS Kehadiran_Persen
# 1      85      100
# 2      78      90
# 3      92      95
# 4      65      80
```


na.rm = TRUE adalah argumen (perintah tambahan) yang sangat penting dalam R. Singkatnya, itu berarti: "Tolong Hapus (Remove) semua nilai NA (data hilang) sebelum melakukan perhitungan."

1. Apa itu NA?

- NA adalah singkatan dari "Not Available" (Tidak Tersedia).
- Ini adalah cara R menandai data yang hilang atau missing value (seperti sel kosong di Excel).

2. Apa yang Terjadi Tanpa na.rm? (Default)

- Secara default, R sangat berhati-hati. Jika mencoba menghitung rata-rata atau jumlah dari sebuah vektor yang mengandung setidaknya satu nilai NA, R akan mengembalikan NA.
- Logikanya: "Bagaimana saya bisa memberi Anda rata-rata yang pasti jika ada data yang hilang? Jawabannya adalah 'Tidak Diketahui' (NA)."

3. Apa yang Dilakukan na.rm = TRUE?

- na.rm adalah singkatan dari NA Remove (Hapus NA).
- Dengan menambahkan na.rm = TRUE ke dalam fungsi (seperti mean(), sum(), max(), min(), rowMeans()), Anda memberi instruksi: "Saya tahu ada data yang hilang. Abaikan saja data yang hilang itu, dan hitung sisanya."

Data Blank

```
vektor_tugas2 <- c(95, 88, 90, NA)
```

```
# 1. Perhitungan GAGAL (Default)
```

```
mean(vektor_tugas2)
```

```
# Output: [1] NA
```

```
# (Karena  $95 + 88 + 90 + NA = NA$ )
```

```
# 2. Perhitungan BERHASIL (dengan na.rm = TRUE)
```

```
mean(vektor_tugas2, na.rm = TRUE)
```

```
# Output: [1] 91
```

```
# (R menghitung:  $(95 + 88 + 90) / 3$ )
```

```
> # 1. Perhitungan GAGAL (Default)
```

```
> mean(vektor_tugas2)
```

```
[1] NA
```

```
> # Output: [1] NA
```

```
> # (Karena  $95 + 88 + 90 + NA = NA$ )
```

```
>
```

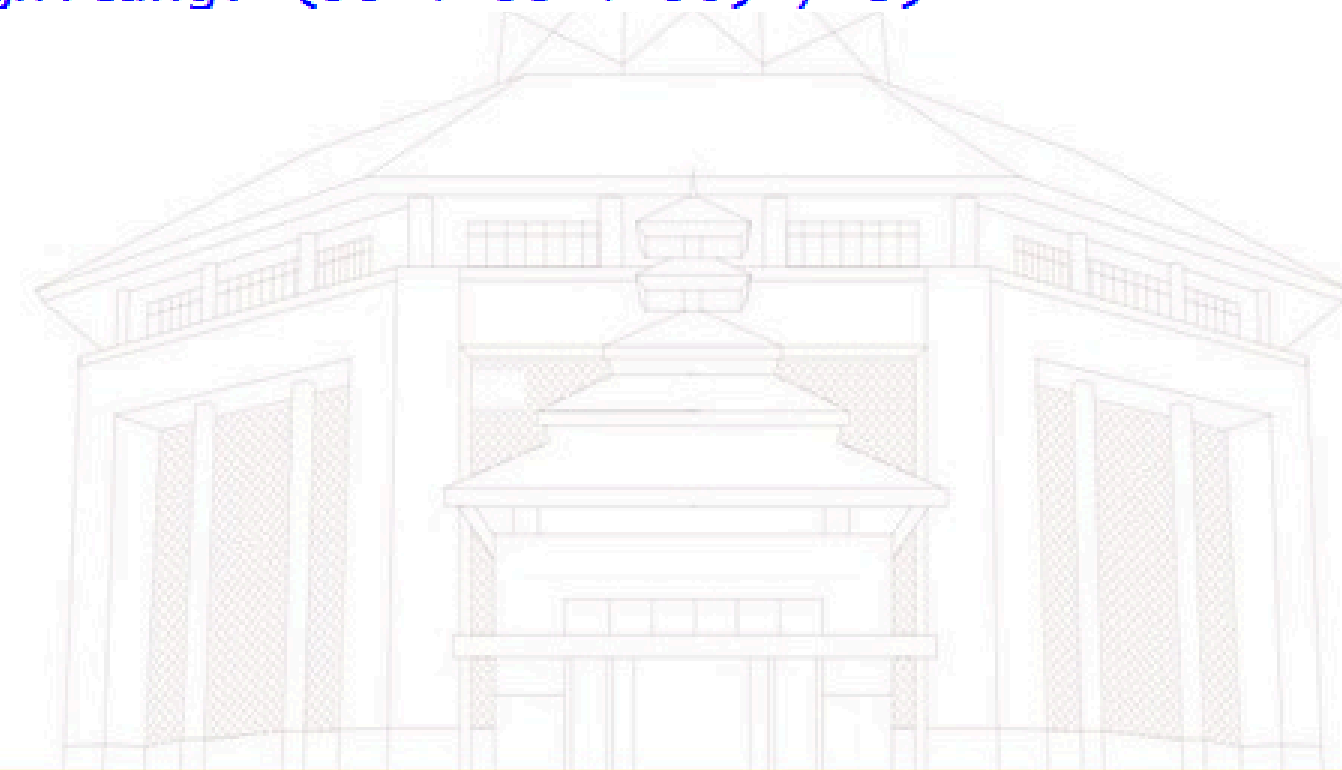
```
> # 2. Perhitungan BERHASIL (dengan na.rm = TRUE)
```

```
> mean(vektor_tugas2, na.rm = TRUE)
```

```
[1] 91
```

```
> # Output: [1] 91
```

```
> # (R menghitung:  $(95 + 88 + 90) / 3$ )
```



Tugas

Tugas 1: (Memodifikasi Data Frame)

1. Data df_uas Anda memiliki NA (data hilang) di Tugas_2.
2. Buat kolom baru Tugas_RataRata yang berisi rata-rata dari Tugas_1 dan Tugas_2 untuk setiap mahasiswa.
3. TANTANGAN: Gunakan rowMeans() dengan argumen na.rm = TRUE.

```
> print(df_uas)
# A tibble: 4 x 5
  NIM `Nilai UAS` Tugas_1 Tugas_2 Tugas_RataRata
  <dbl>      <dbl>   <dbl>   <dbl>      <dbl>
1  1001        88     90     95      92.5
2  1002        80     85     88      86.5
3  1003        90     92     90       91
4  1004        70     75     NA       75
```

Tugas 2: (Fungsi apply)

1. Hitung nilai maksimum untuk setiap kolom (NIM, Nilai_UAS, Tugas_1, Tugas_2) menggunakan apply().
2. (Tips: Gunakan MARGIN = 2 dan FUN = max. Hati-hati dengan NA! Gunakan na.rm = TRUE di dalam fungsi max).

Nilai UAS	Tugas_1	Tugas_2
90	92	95

Jawaban Tugas

(Pastikan df_uas sudah ada dari latihan impor)

Solusi Tugas 1

```
df_uas$Tugas_RataRata <- rowMeans(  
  df_uas[, c("Tugas_1", "Tugas_2")],  
  na.rm = TRUE  
)  
print(df_uas)
```

Solusi Tugas 2

(NPM tidak ikut dihitung karena itu ID, jadi kita pilih kolom 2-4)

```
data_nilai_uas <- df_uas[, 2:4]
```

Kita harus membuat fungsi kustom kecil untuk 'max' agar bisa 'na.rm'

```
apply(data_nilai_uas, 2, max, na.rm = TRUE)
```

Output:

# Nilai_UAS	Tugas_1	Tugas_2
# 90	92	95





SEE YOU NEXT WEEK !

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 199005202024061001
ferdian.bangkit@untirta.ac.id