



Pengantar Komputasi Statistika

#10 Meeting

Alur Kontrol & Metode Iterasi

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





Alur Kontrol: If, else if, else

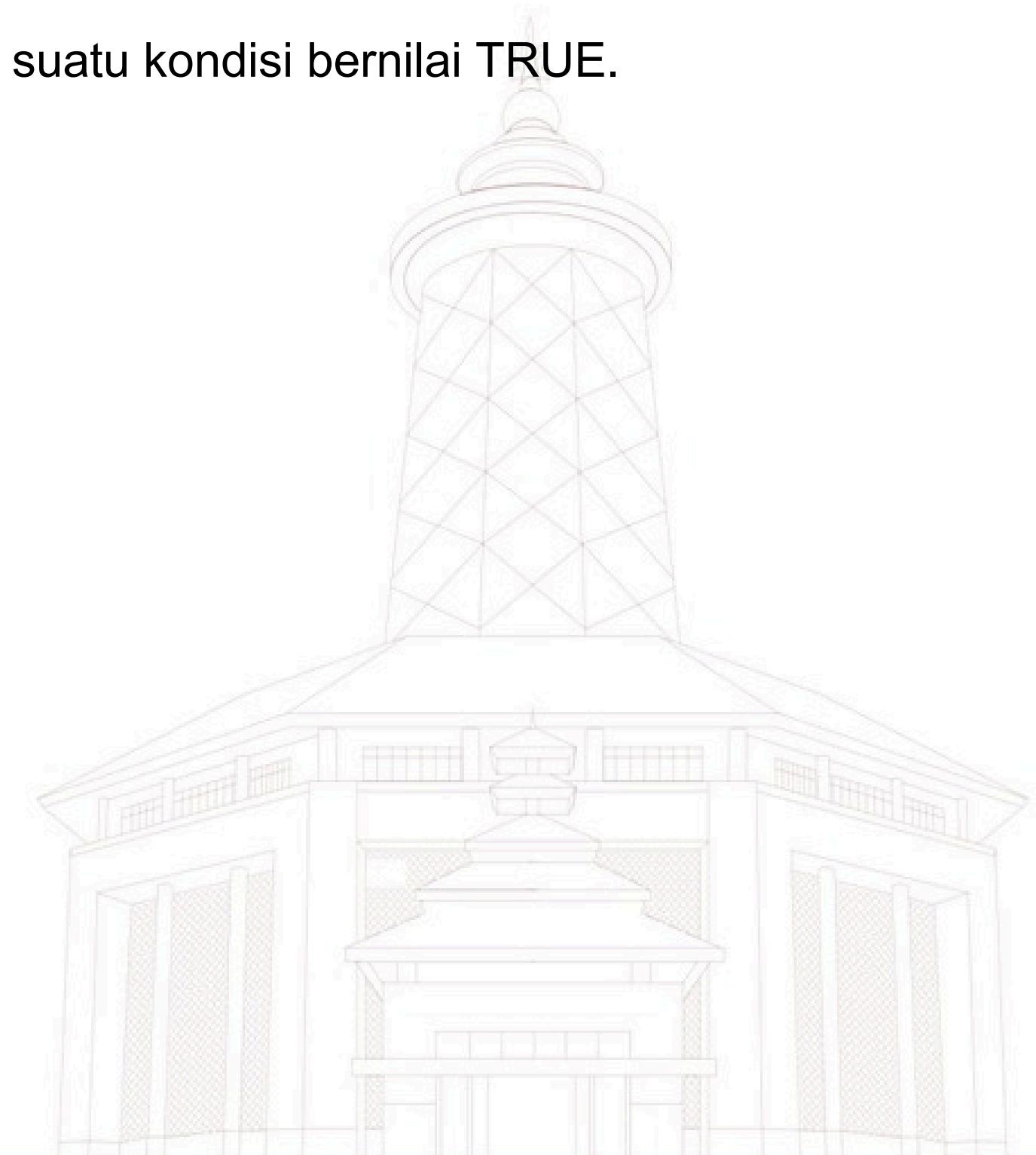
Struktur if adalah "pintu gerbang" logika. Ia hanya menjalankan kode jika suatu kondisi bernilai TRUE.

- if (kondisi): "Jika ini benar, lakukan A."
- else: "Jika tidak (jika if salah), lakukan B."
- else if (kondisi_lain): "Jika if salah, coba cek kondisi lain ini."

```
# Cek nilai mahasiswa  
nilai <- 75
```

```
if (nilai >= 80) {  
  print("Nilai Anda A, sangat baik!")  
} else if (nilai >= 60) {  
  print("Nilai Anda B/C, lulus.")  
} else {  
  print("Nilai Anda D/E, harap mengulang.")  
}
```

```
# Output: [1] "Nilai Anda B/C, lulus."
```





Alur Kontrol : ifelse()

Ini adalah versi if-else yang tervektorisasi (bekerja pada seluruh vektor sekaligus). Ini sangat berguna untuk membuat kategori atau kolom baru di data frame.

Struktur: ifelse(test_kondisi, nilai_jika_TRUE, nilai_jika_FALSE)

```
# Vektor nilai ujian
```

```
vektor_nilai <- c(90, 55, 78, 40, 81)
```

```
# Membuat vektor status kelulusan (Kriteria lulus >= 60)
```

```
status_lulus <- ifelse(vektor_nilai >= 60, "Lulus", "Gagal")
```

```
print(status_lulus)
```

```
# Output: [1] "Lulus" "Gagal" "Lulus" "Gagal" "Lulus"
```





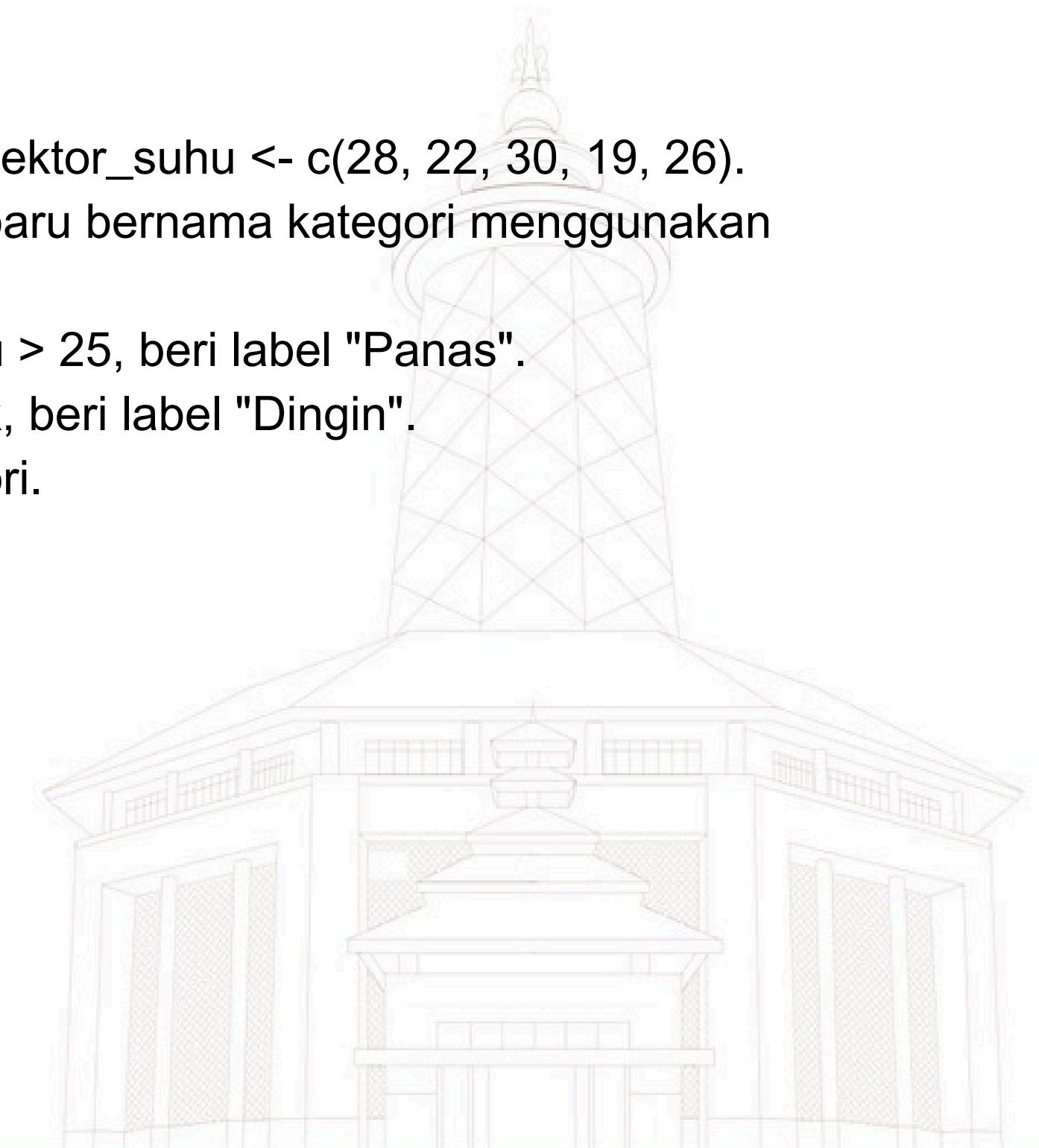
Latihan

Latihan 1:

- Buat variabel angka <- -5.
- Tulis kode if-else untuk mengecek:
 - Jika angka > 0, cetak "Positif".
 - Jika angka < 0, cetak "Negatif".
 - Jika tidak keduanya (berarti 0), cetak "Nol".
- (Coba ganti nilai angka menjadi 10 atau 0 untuk menguji kode Anda).

Latihan 2:

- Buat vektor vektor_suhu <- c(28, 22, 30, 19, 26).
- Buat vektor baru bernama kategori menggunakan ifelse():
 - Jika suhu > 25, beri label "Panas".
 - Jika tidak, beri label "Dingin".
- Cetak kategori.





Jawaban Latihan

Latihan 1:

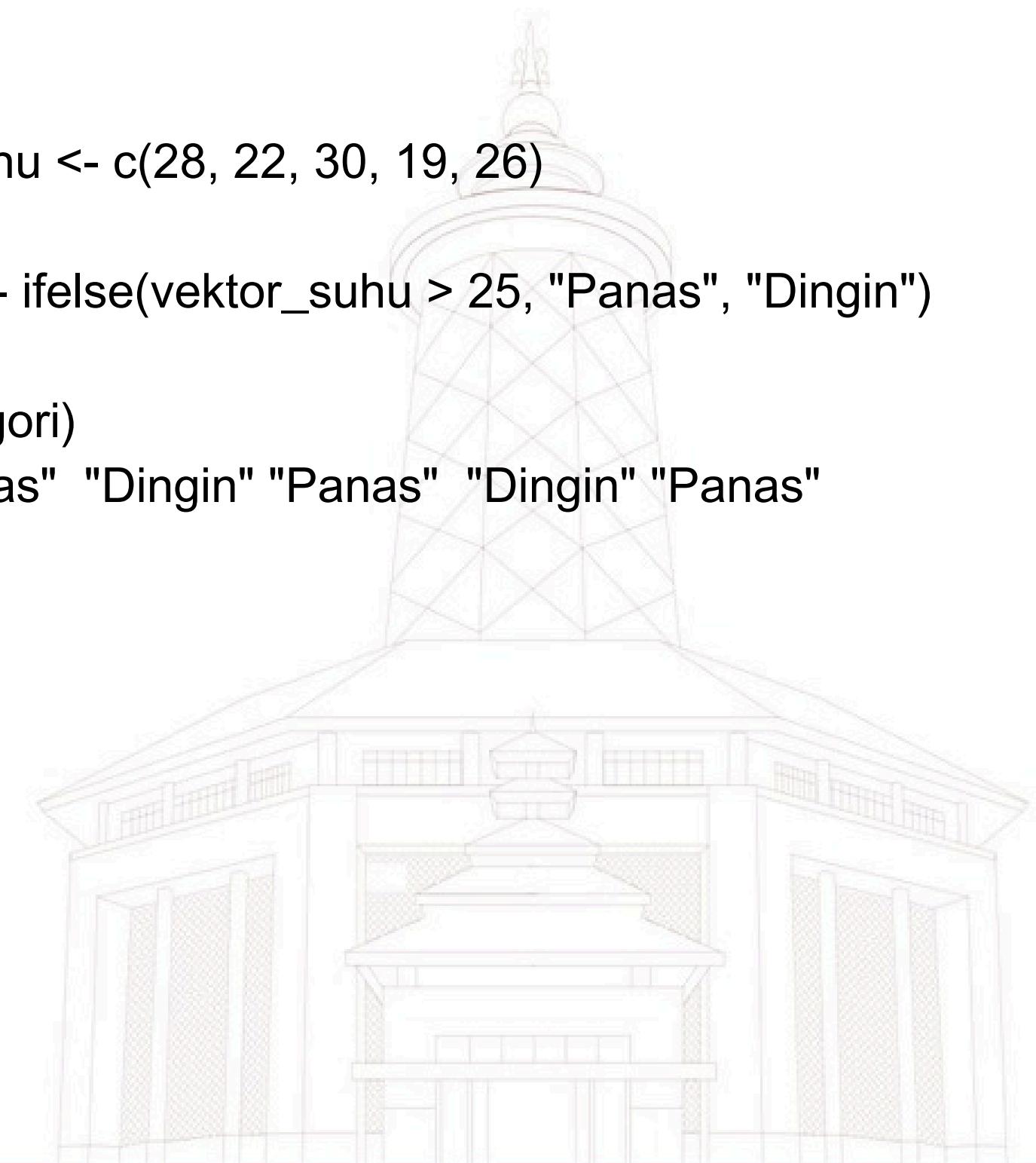
angka < -5

```
if (angka > 0) {  
  print("Positif")  
} else if (angka < 0) {  
  print("Negatif")  
} else {  
  print("Nol")  
}
```

Latihan 2:

vektor_suhu <- c(28, 22, 30, 19, 26)

```
kategori <- ifelse(vektor_suhu > 25, "Panas", "Dingin")  
print(kategori)  
# [1] "Panas" "Dingin" "Panas" "Dingin" "Panas"
```



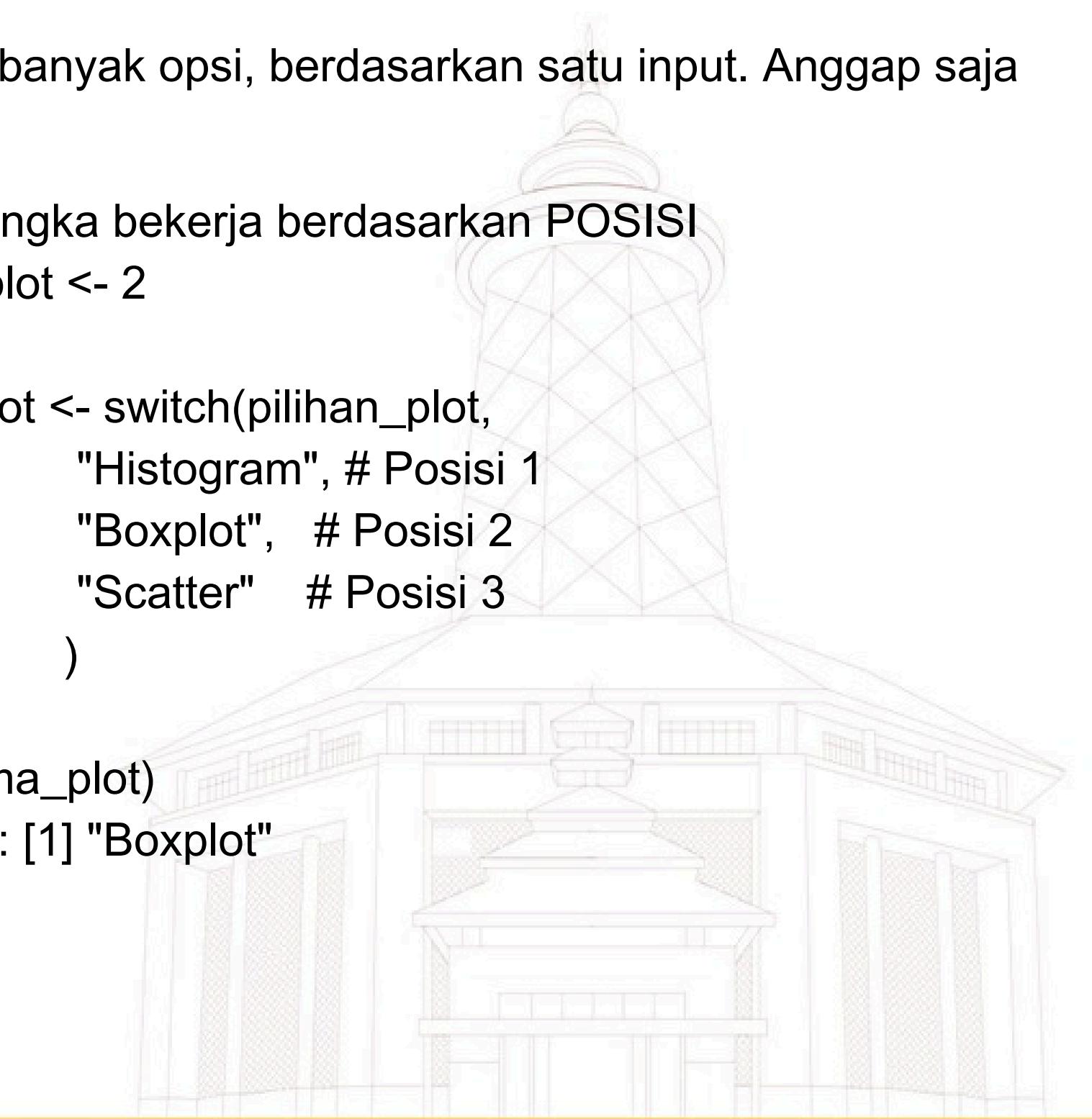


Alur Kontrol : switch()

Fungsi switch() adalah cara efisien untuk memilih satu tindakan dari banyak opsi, berdasarkan satu input. Anggap saja seperti menu di restoran.

```
# Data kita  
data_angka <- c(10, 20, 30, 40, 50)  
  
# Pilihan 'menu' kita  
pilihan_stat <- "median"  
  
# switch() akan MENJALANKAN fungsi yang sesuai  
hasil <- switch(pilihan_stat,  
    "rata-rata" = mean(data_angka),  
    "median" = median(data_angka),  
    "total" = sum(data_angka),  
    "Pilihan tidak valid" # Default  
)  
print(paste("Hasil kalkulasi:", hasil))  
# Output: [1] "Hasil kalkulasi: 30"
```

```
# Input angka bekerja berdasarkan POSISI  
pilihan_plot <- 2  
  
nama_plot <- switch(pilihan_plot,  
    "Histogram", # Posisi 1  
    "Boxplot", # Posisi 2  
    "Scatter" # Posisi 3  
)  
  
print(nama_plot)  
# Output: [1] "Boxplot"
```





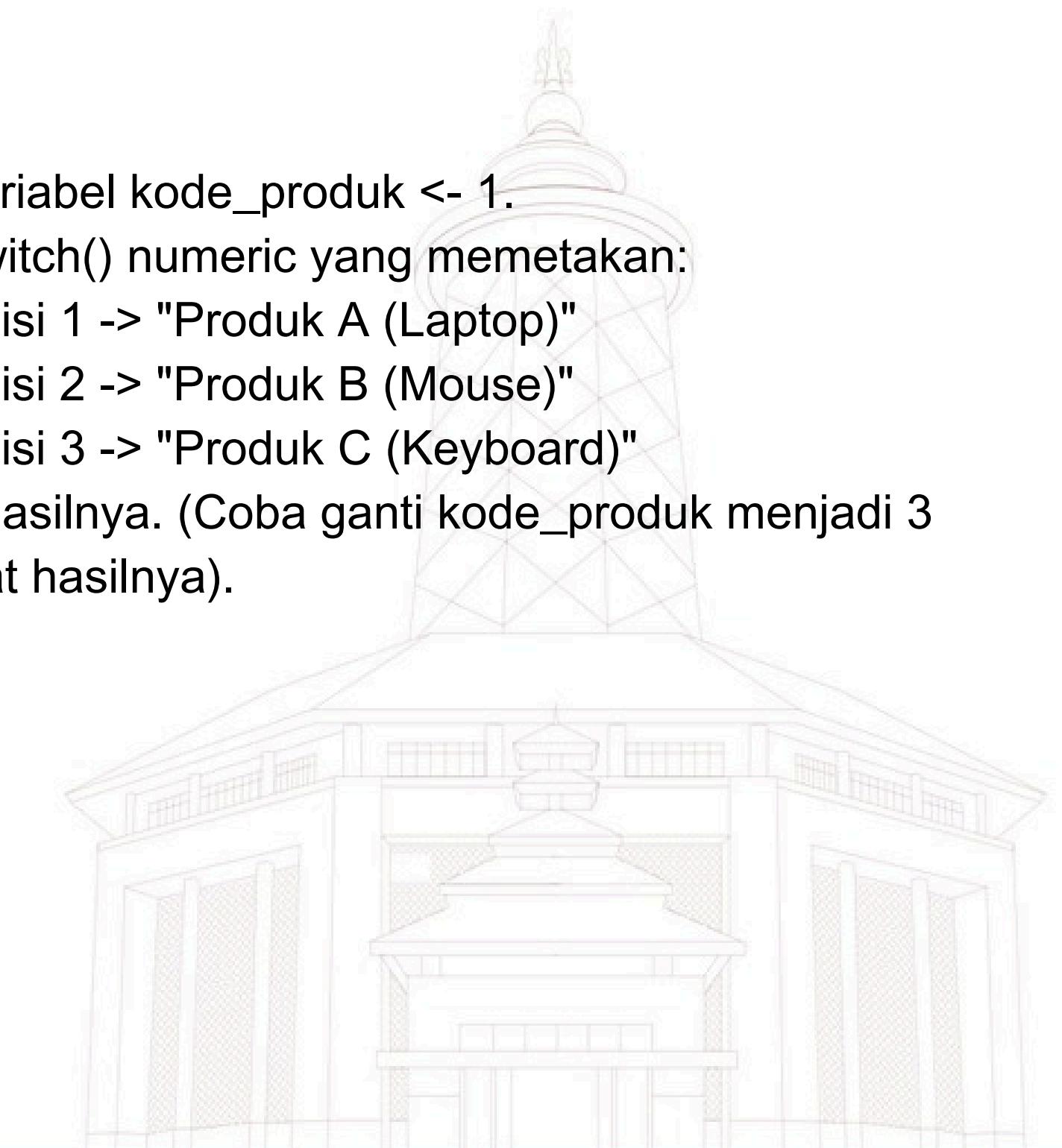
Latihan

Latihan 1:

- Buat variabel hari_ini <- "Rabu".
- Buat switch() yang memetakan:
 - "Senin" -> "Kuliah Statistika"
 - "Rabu" -> "Praktikum R"
 - "Jumat" -> "Presentasi Proyek"
 - Default -> "Hari bebas!"
- Cetak hasilnya.

Latihan 2:

- Buat variabel kode_produkt <- 1.
- Buat switch() numeric yang memetakan:
 - Posisi 1 -> "Produk A (Laptop)"
 - Posisi 2 -> "Produk B (Mouse)"
 - Posisi 3 -> "Produk C (Keyboard)"
- Cetak hasilnya. (Coba ganti kode_produkt menjadi 3 dan lihat hasilnya).





Jawaban Latihan

Latihan 1:

```
hari_ini <- "Rabu" # Coba ganti ini ke "Senin" atau  
"Selasa"  
  
jadwal <- switch(hari_ini,  
    "Senin" = "Kuliah Statistika",  
    "Rabu" = "Praktikum R",  
    "Jumat" = "Presentasi Proyek",  
    "Hari bebas!" # Ini adalah nilai default jika tidak  
ada yg cocok  
    )
```

```
print(jadwal)  
# [1] "Praktikum R"
```

Latihan 2:

```
kode_produkt <- 1 # Coba ganti ke 3  
  
nama_produkt <- switch(kode_produkt,  
    "Produk A (Laptop)", # Posisi 1  
    "Produk B (Mouse)", # Posisi 2  
    "Produk C (Keyboard)" # Posisi 3  
    )  
print(nama_produkt)  
# Output: [1] "Produk A (Laptop)"
```



Metode Iterasi (Perulangan)

Iterasi (atau looping) adalah proses melakukan tugas yang sama berulang kali.

Kita akan fokus pada dua jenis utama:

1. for loop (Perulangan Terbatas):

- "Lakukan ini untuk setiap item di dalam koleksi."
- (Contoh: Sapa setiap nama mahasiswa di daftar absensi).

2. while loop (Perulangan Kondisional):

- "Selama kondisi ini benar, terus lakukan ini."
- (Contoh: Selama tabungan belum 1 juta, terus menabung 10 ribu).

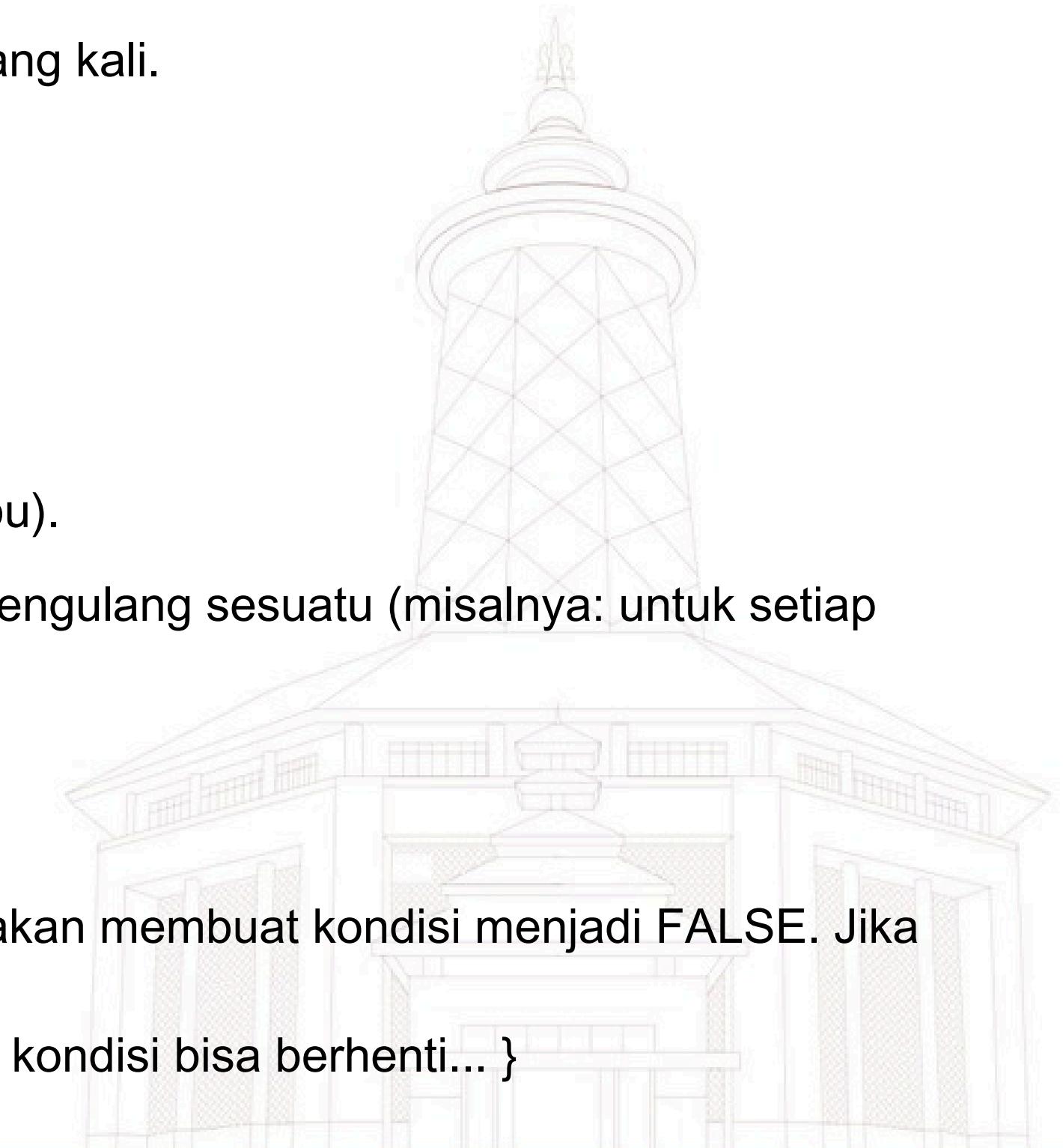
for loop sangat cocok ketika Anda tahu persis berapa kali Anda ingin mengulang sesuatu (misalnya: untuk setiap elemen dalam vektor, atau 10 kali).

Struktur: `for (variabel_sementara in koleksi) { ...lakukan sesuatu... }`

while loop akan terus berjalan selama kondisi logikanya bernilai TRUE.

PENTING: PASTIKAN ada sesuatu di dalam loop yang pada akhirnya akan membuat kondisi menjadi FALSE. Jika tidak, akan menciptakan infinite loop (loop tak terbatas)!

Struktur: `while (kondisi_masih_TRUE) { ...lakukan sesuatu... ...pastikan kondisi bisa berhenti... }`





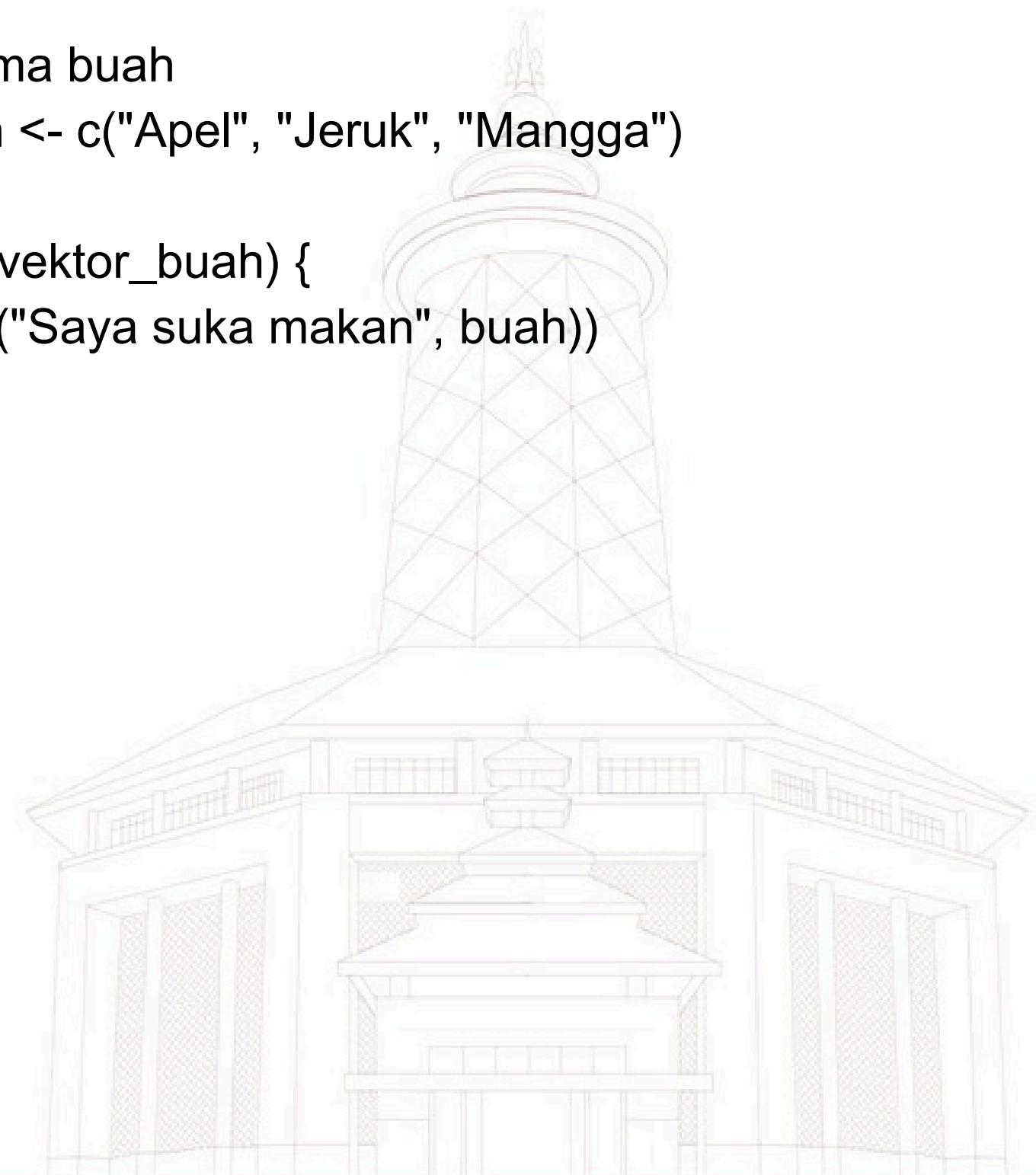
Metode Iterasi - For Loop

```
# 'i' adalah variabel sementara yang berubah di setiap  
iterasi
```

```
for (i in 1:5) {  
  print(paste("Ini adalah iterasi ke-", i))  
}
```

```
# Vektor nama buah  
vektor_buah <- c("Apel", "Jeruk", "Mangga")
```

```
for (buah in vektor_buah) {  
  print(paste("Saya suka makan", buah))  
}
```





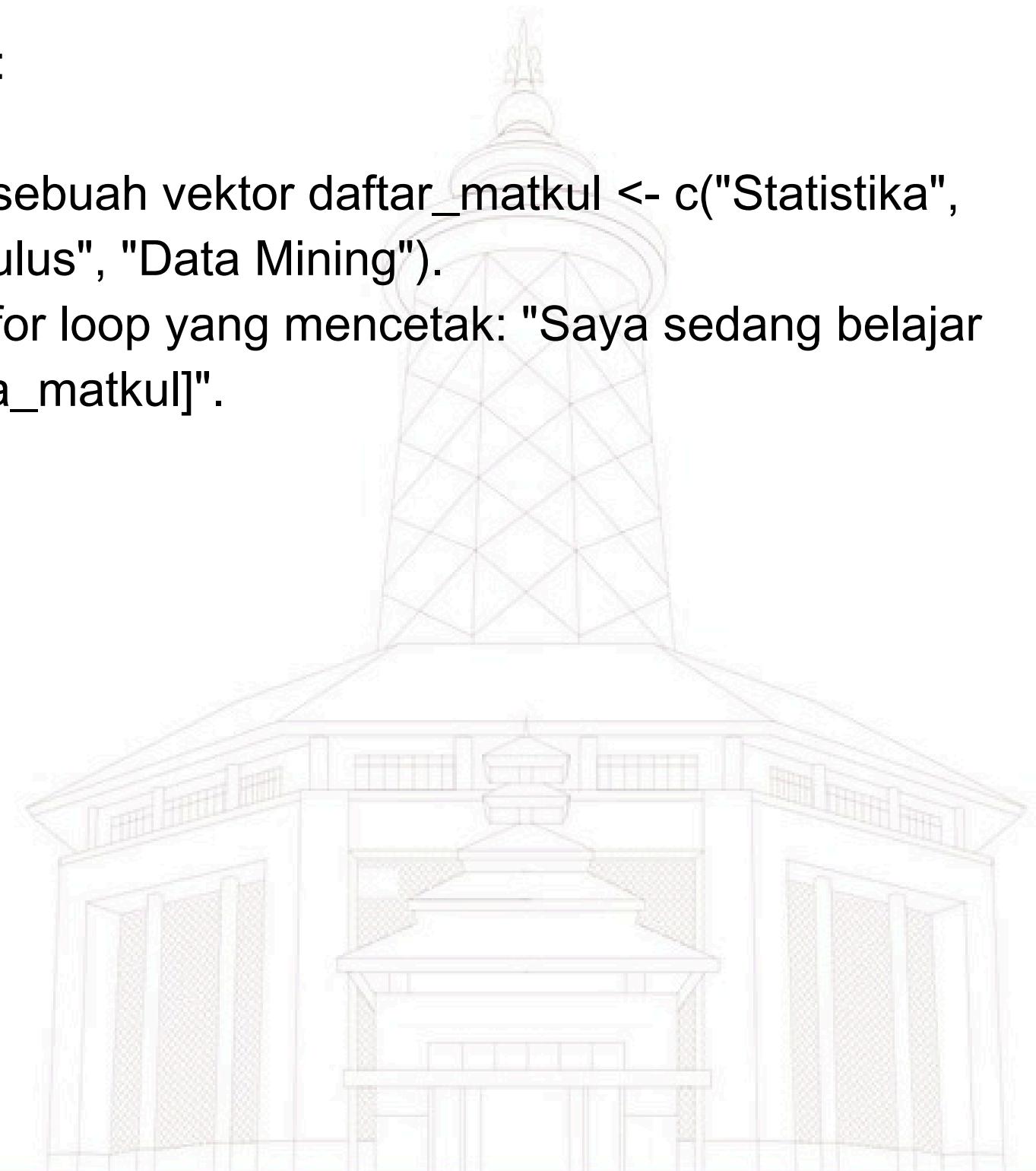
Latihan

Latihan 1:

- Buatlah for loop yang mencetak angka genap dari 2 sampai 10.
- (Tips: seq(2, 10, by = 2))

Latihan 2:

- Buat sebuah vektor daftar_matkul <- c("Statistika", "Kalkulus", "Data Mining").
- Buat for loop yang mencetak: "Saya sedang belajar [nama_matkul]".





Jawaban Latihan

Latihan 1:

```
# seq(2, 10, by = 2) menghasilkan vektor: c(2, 4, 6, 8, 10)
for (i in seq(2, 10, by = 2)) {
  print(i)
}
```

Latihan 2:

```
daftar_matkul <- c("Statistika", "Kalkulus", "Data Mining")
for (matkul in daftar_matkul) {
  print(paste("Saya sedang belajar", matkul))
}
```





Metode Iterasi - While Loop

```
# 'n' adalah variabel kontrol kita
```

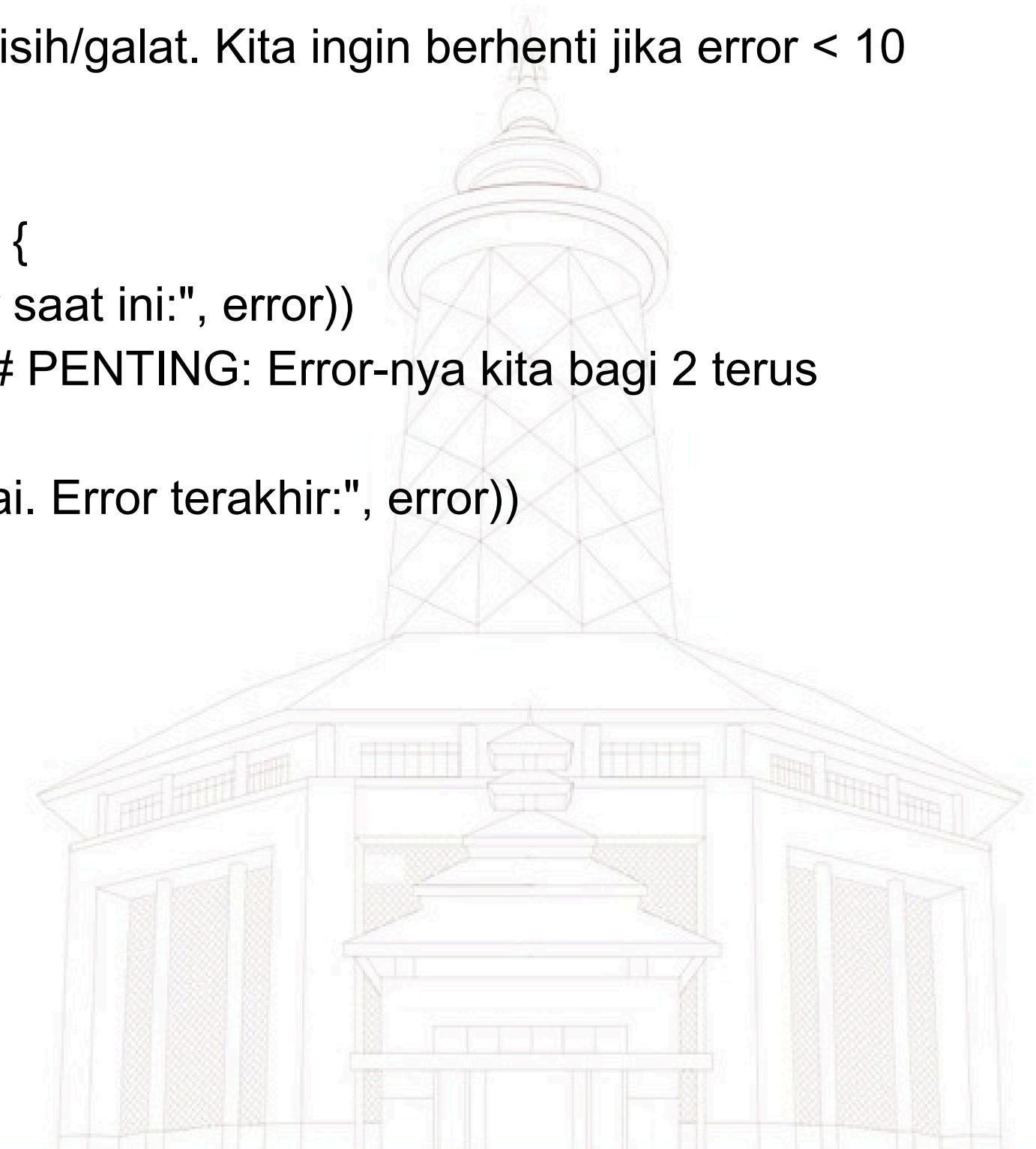
```
n <- 3
```

```
while (n > 0) {  
  print(paste(n, "..."))  
  n <- n - 1 # PENTING: Langkah untuk berhenti  
  (membuat 'n' mengecil)  
}  
print("Mulai!")
```

```
# 'error' adalah selisih/galat. Kita ingin berhenti jika error < 10
```

```
error <- 100
```

```
while (error >= 10) {  
  print(paste("Error saat ini:", error))  
  error <- error / 2 # PENTING: Error-nya kita bagi 2 terus  
}  
print(paste("Selesai. Error terakhir:", error))
```





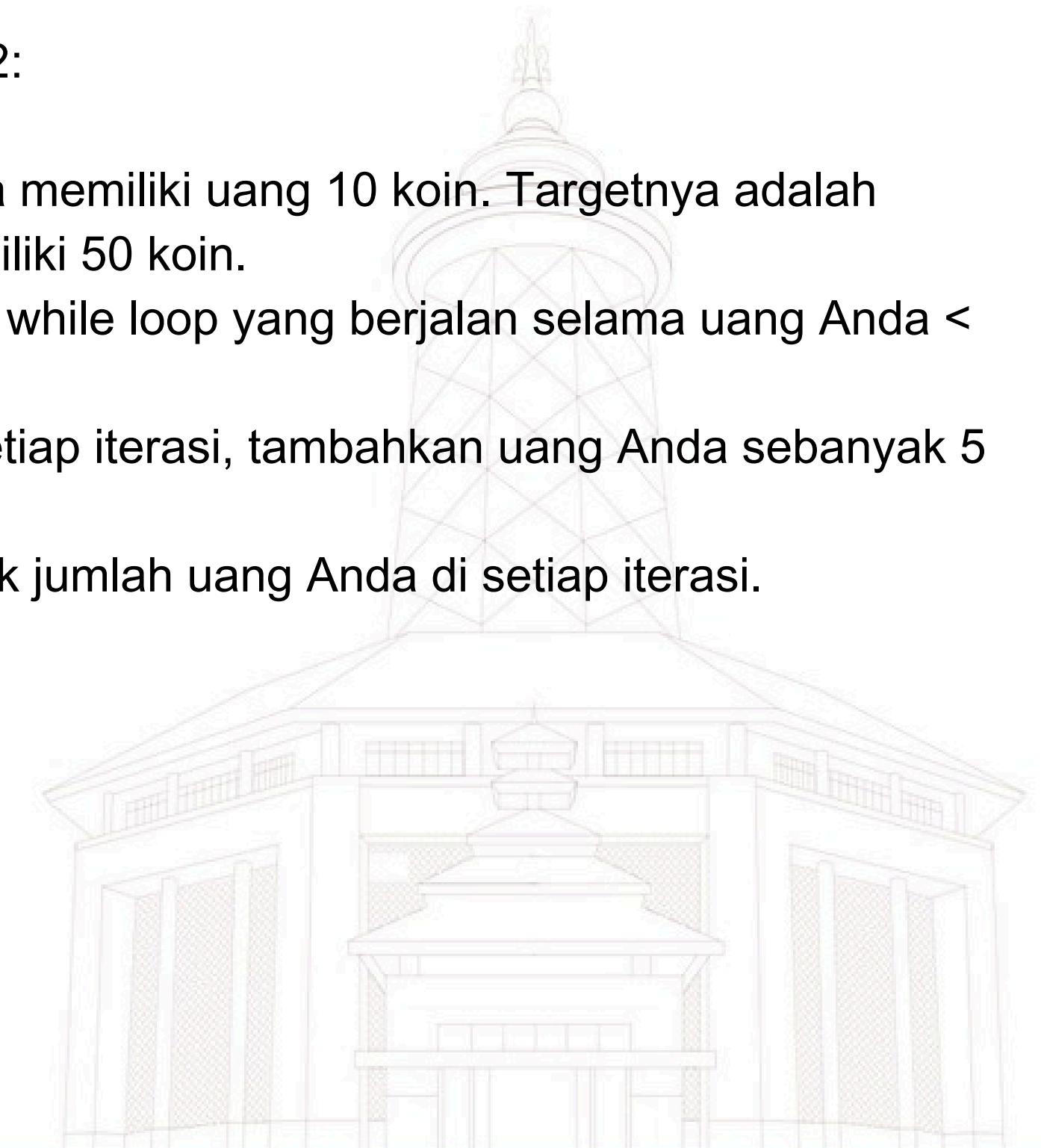
Latihan

Latihan 1:

- Buat while loop yang melakukan hitung maju dari 1 sampai 5.
- Anda perlu variabel n yang dimulai dari 1.
- Cetak: "Hitungan ke: [n]".

Latihan 2:

- Anda memiliki uang 10 koin. Targetnya adalah memiliki 50 koin.
- Buat while loop yang berjalan selama uang Anda < 50.
- Di setiap iterasi, tambahkan uang Anda sebanyak 5 koin.
- Cetak jumlah uang Anda di setiap iterasi.





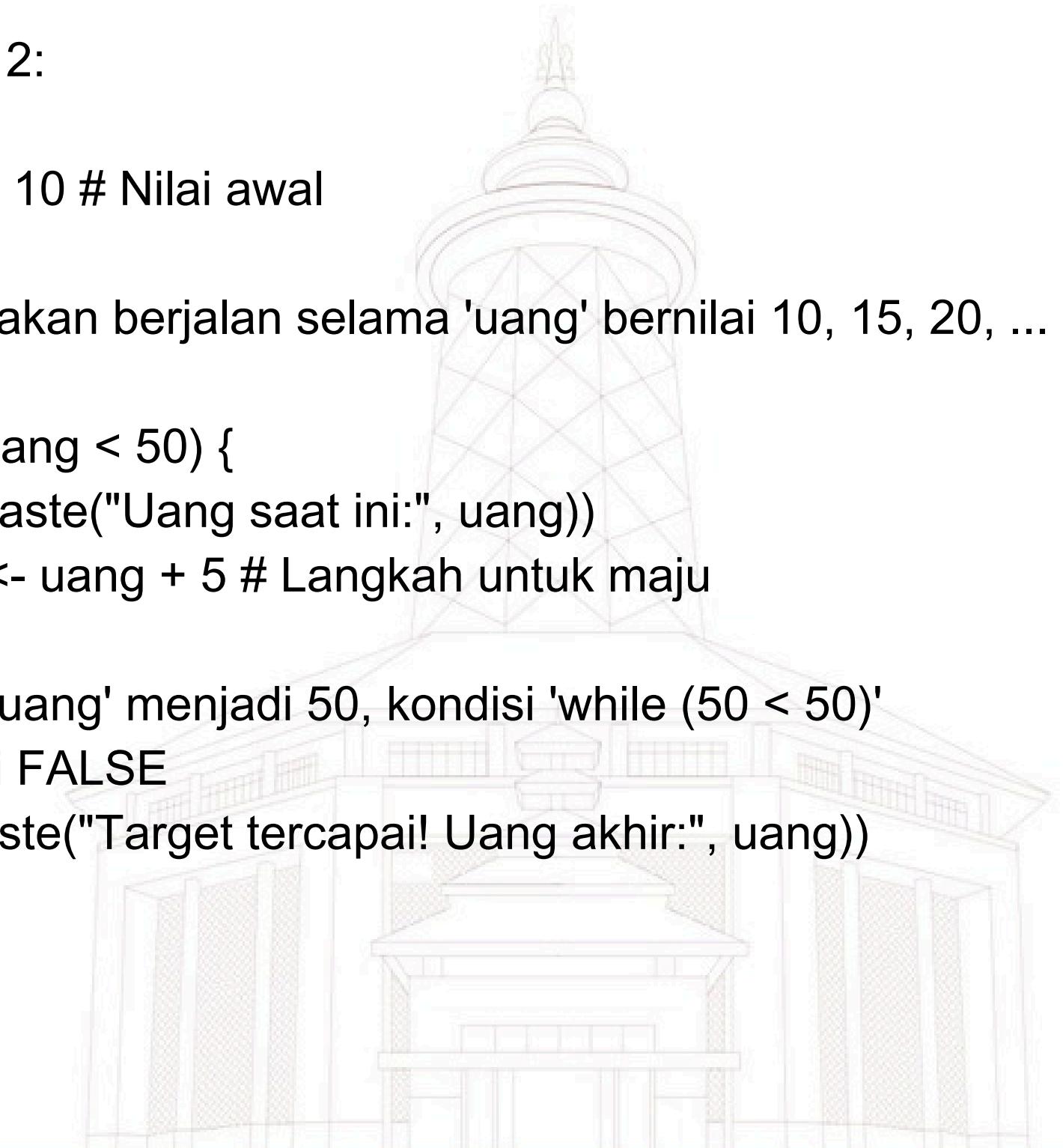
Jawaban Latihan

Latihan 1:

```
n <- 1 # Titik mulai  
  
while (n <= 5) { # Kondisi: Berhenti JIKA SUDAH LEBIH  
DARI 5  
  print(paste("Hitungan ke:", n))  
  n <- n + 1 # PENTING: Tambah 1 di setiap iterasi  
}
```

Latihan 2:

```
uang <- 10 # Nilai awal  
  
# Loop akan berjalan selama 'uang' bernilai 10, 15, 20, ...  
45  
while (uang < 50) {  
  print(paste("Uang saat ini:", uang))  
  uang <- uang + 5 # Langkah untuk maju  
}  
# Saat 'uang' menjadi 50, kondisi 'while (50 < 50)'  
menjadi FALSE  
print(paste("Target tercapai! Uang akhir:", uang))
```





SEE YOU NEXT WEEK !

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

NIP. 199005202024061001

ferdian.bangkit@untirta.ac.id

