

# MODUL LOGIKA DAN PENGULANGAN

Zahra Mahendra Putri, Regina Dwirahma Alisya

2024-12-06

## Contents

<b>BAB 1 : DASAR PEMOGRAMAN R</b>	<b>2</b>
1.1 Pengertian . . . . .	2
1.2 Pentingnya untuk analisis data . . . . .	3
<b>BAB 2 : LOGIKA DAN PENGULANGAN DALAM R</b>	<b>4</b>
2.1 Function if . . . . .	4
2.2 Function ifelse . . . . .	4
2.3 Function for loop . . . . .	4
2.4 Funciton While Loop . . . . .	4
2.5 Repeat Loop . . . . .	5
2.6 Function Nested ifelse . . . . .	5
2.7 Function “and (&)”, “or(())” . . . . .	5
<b>BAB 3 : Implementasi study kasus nyata</b>	<b>6</b>
3.1 STUDY KASUS IF . . . . .	6
3.2 STUDY KASUS IFELSE . . . . .	7
3.3 For loop . . . . .	7
3.4 While loop . . . . .	8
3.5 Repeat Loop . . . . .	9
3.6 Nested if-else . . . . .	9
3.7 Logika Dasar and (&), or (()) . . . . .	10
3.8 Logika Dasar or (()) . . . . .	11
<b>BAB 4 : Kesimpulan</b>	<b>12</b>
4.1 Komparansi antar fungsi . . . . .	12
<b>DAFTAR PUSTAKA</b>	<b>13</b>

# BAB 1 : DASAR PEMOGRAMAN R

## 1.1 Pengertian

### 1. Statement If

Fungsi if adalah statement yang umum digunakan pada percabangan. If adalah fungsi untuk membuat perbandingan logis antara nilai dan apa yang diharapkan dengan menguji kondisi dan mengembalikan hasil jika True atau False.

### 2. If Else

Fungsi ini di gunakan jika ada perintah yang akan dijalankan ketika masing-masing kondisi bernilai True atau False. Fungsi ini juga dikenal sebagai percabangan bersyarat, Ifelse berfungsi untuk memastikan bahwa instruksi tertentu hanya akan dijalankan jika suatu kondisi terpenuhi.

### 3. For loop

For loop merupakan jenis loop yang paling sederhana dan sering digunakan. For-loop digunakan untuk mengulang sekumpulan objek, seperti vektor, daftar, matriks, atau kerangka data, dan menerapkan serangkaian operasi yang sama pada setiap item dari struktur data tertentu. Selain for loop, ada juga jenis loop lainnya, yaitu while loop dan repeat loop.

### 4. While loop

While loop merupakan loop yang digunakan ketika kita telah menetapkan stop condition sebelumnya. Blok statement/kode yang sama akan terus dijalankan sampai stop condition ini tercapai. Stop condition akan di cek sebelum melakukan proses loop.

### 5. Repeat loop

Fungsi repeat loop adalah untuk menjalankan kode atau statement yang sama secara berulang-ulang hingga kondisi yang ditentukan tercapai.

### 6. Nested Ifelse

Fungsi nested if else adalah untuk memeriksa kondisi baru setelah kondisi sebelumnya telah ditemukan benar atau salah. Nested if else juga dikenal sebagai if bersarang, yaitu kondisi yang di dalamnya terdapat kondisi lagi.

### 7. Fungsi logika “and”, “or”

Fungsi logika “and” dan “or” di RStudio berfungsi untuk membandingkan dua kondisi logika, yaitu logika benar (TRUE) dan logika salah (FALSE). Operator “and” ini akan mengembalikan TRUE jika kedua nilai logika yang dibandingkan bernilai TRUE sedangkan operator “or” ini akan mengembalikan TRUE jika kedua nilai logika yang dibandingkan bernilai TRUE.

## 1.2 Pentingnya untuk analisis data

### 1. Statement If

Fungsi if adalah statement yang umum digunakan pada percabangan. If adalah fungsi untuk membuat perbandingan logis antara nilai dan apa yang diharapkan dengan menguji kondisi dan mengembalikan hasil jika True atau False.

### 2. If Else

Fungsi ini di gunakan jika ada perintah yang akan dijalankan ketika masing-masing kondisi bernilai True atau False. Fungsi ini juga dikenal sebagai percabangan bersyarat, Ifelse berfungsi untuk memastikan bahwa instruksi tertentu hanya akan dijalankan jika suatu kondisi terpenuhi.

### 3. For loop

For loop merupakan jenis loop yang paling sederhana dan sering digunakan. For-loop digunakan untuk mengulang sekumpulan objek, seperti vektor, daftar, matriks, atau kerangka data, dan menerapkan serangkaian operasi yang sama pada setiap item dari struktur data tertentu. Selain for loop, ada juga jenis loop lainnya, yaitu while loop dan repeat loop.

### 4. While loop

While loop digunakan untuk mengulang proses selama kondisi tertentu terpenuhi. Singkatnya, misalkan kita ingin menabung 10 juta, tetapi banyaknya waktu yang dibutuhkan untuk mencapai target Tabungan tersebut tidak diketahui, namun di sisi lain kita mengetahui kapan harus berhenti menabung disaat target sudah tercapai, bagaimana kita mengetahuinya? Maka kita dapat mengetahuinya ketika tabungan itu mencapai target.

### 5. Repeat Loop

Repeat loop dalam analisis data digunakan untuk menjalankan perintah tertentu secara berulang hingga kondisi syarat tersebut telah terpenuhi.

### 6. Nested Ifelse

Fungsi nested if else adalah untuk memeriksa kondisi baru setelah kondisi sebelumnya telah ditemukan benar atau salah. Nested if else juga dikenal sebagai if bersarang, yaitu kondisi yang di dalamnya terdapat kondisi lagi.

### 7. Fungsi logika “and”, “or”

Fungsi logika “and” dan “or” di RStudio berfungsi untuk membandingkan dua kondisi logika, yaitu logika benar (TRUE) dan logika salah (FALSE). Operator “and” ini akan mengembalikan TRUE jika kedua nilai logika yang dibandingkan bernilai TRUE sedangkan operator “or” ini akan mengembalikan TRUE jika kedua nilai logika yang dibandingkan bernilai TRUE.

## BAB 2 : LOGIKA DAN PENGULANGAN DALAM R

### 2.1 Function if

Bentuk umum dari perintah if() adalah if(kondisi) ekspresi dan argumen yang dibutuhkan pada fungsi if() adalah sebuah vector logical tunggal bernilai TRUE atau FALSE. Berikut ini adalah sintaks dari fungsi If.

```
kondisi <- 1
if(kondisi) {
}
```

```
## NULL
```

### 2.2 Function ifelse

Bentuk umum dari perintah if() adalah if(kondisi) ekspresi dan argumen yang dibutuhkan pada fungsi if() adalah sebuah vector logical tunggal bernilai TRUE atau FALSE. Berikut ini adalah sintaks dari fungsi If.

```
kondisi <- 1
if(kondisi) {
}
```

```
## NULL
```

### 2.3 Function for loop

Pengulangan menggunakan statement for, Berikut ini sintaks dari for loop.

```
for(i in 1) {
  print(i)
}
```

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

### 2.4 Function While Loop

Cara lain untuk melakukan pengulangan adalah dengan menggunakan statement while. Sintaks dari statement ini adalah sebagai berikut.

```
i = 1
n <- 2>1
while(i < n){
  print(i);
  i = i+1;
}
```

## 2.5 Repeat Loop

Repeat digunakan untuk menjalankan blok pernyataan berulang kali hingga pernyataan break jump ditemukan dan untuk mengakhiri atau keluar dari loop harus menggunakan pernyataan break. Jika tidak menggunakan ini, pernyataan repeat akan berakhir dalam loop tak terbatas. Berikut ini adalah sintaks dari repeat loop.

```
i = 1
n <- 10
repeat{
  print(i)
  if(i==n) {break()}
  i = i + 1
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

## 2.6 Function Nested ifelse

Nested Ifelse adalah fungsi If yang berada di dalam If lainnya. Semakin menjorok ke dalam, maka pernyataan tersebut akan semakin ada di Tingkat bawah. Berikut ini sintaks dari nested ifelse.

```
kondisi1 <- c(2, 6, 9, 3)
kondisi2 <- c(2, 3, 4, 5)
ifelse(kondisi1, kondisi2)
```

```
## [1] 2 3 4 5
```

```
cat("kondisi", kondisi1, kondisi2)
```

```
## kondisi 2 6 9 3 2 3 4 5
```

## 2.7 Function “and (&)”, “or(|)”

### 1. Fungsi “and”

Operator “and” diwakili dengan tanda ampersand ‘&’ dan ‘&&’. Berikut ini adalah sintaks dari logika “and”.

```

x <- c("x", "y", "z")
y <- c(85, 65, 78, 90, 70, 73)
z <- c(85, 70, 85, 80, 60, 80)
kondisi <- ifelse(x >= 75 & y >= 80, "BENAR", "SALAH")
#status kondisi
status <- c()
for (i in 1:length(x)) {
  if (y[i] >= 75 && z[i] >= 80) {
    status[i] <- "BENAR"
  } else {
    status[i] <- "SALAH"
  }
}

```

## 2. Fungsi “or”

Operator “or” diwakili dengan symbol pike ( | ) dan ( || ). Berikut ini adalah sintaks dari logika “or”.

```

x <- c("x", "y", "z")
y <- c(85, 65, 78, 90, 70, 73)
z <- c(85, 70, 85, 80, 60, 80)
#status kondisi
status <- c()
for (i in 1:length(x)) {
  if (y[i] >= 75 && z[i] >= 80) {
    status[i] <- "BENAR"
  } else {
    status[i] <- "SALAH"
  }
}

```

# BAB 3 : Implementasi study kasus nyata

## 3.1 STUDY KASUS IF

Sebuah perusahaan ingin memberikan bonus 10% dari gaji berdasarkan performa kerjanya. karyawan layak mendapat bonus jika performanya diatas 80. Peforma yang dimiliki oleh karyawan yaitu 85 dengan gaji 6..000.000. Apakah seorang karyawan berhak mendapatkan bonus ?

```

gaji <- 6000000
peforma <- 85
#fungsi if
if (peforma > 80) {
  bonus <- gaji * 0.1
  print(paste("Karyawan mendapat bonus sebesar :", bonus))
} else {
  print(paste("Karyawan tidak mendapat bonus sebesar."))
}

```

```
## [1] "Karyawan mendapat bonus sebesar : 6e+05"
```

## 3.2 STUDY KASUS IFELSE

Sebuah perusahaan memiliki data gaji dan performa kerja beberapa karyawan. karyawan mendapatkan bonus 10% dari gaji jika performa mereka diatas 85. tampilkan data karyawan, performa, besaran bonus dan status apakah beberapa karyawan di perusahaan tersebut layak mendapatkan bonus.

```
gaji <- c(6000000, 5000000, 4500000, 7000000, 8000000, 3000000)
peforma <- c(86, 70, 75, 80, 90, 75)
#menghitung bonus
bonus <- ifelse(peforma > 85, gaji *0.1, 0)
keterangan <- ifelse(peforma > 85, "Karyawan mendapatkan bonus:", "Karyawan tidak mendapatkan bonus")
data_karyawan <- data.frame(Gaji = gaji, performa = peforma, Bonus = bonus, Status = keterangan)
print(data_karyawan)
```

```
##      Gaji performa Bonus                               Status
## 1 6000000      86 6e+05      Karyawan mendapatkan bonus:
## 2 5000000      70 0e+00 Karyawan tidak mendapatkan bonus
## 3 4500000      75 0e+00 Karyawan tidak mendapatkan bonus
## 4 7000000      80 0e+00 Karyawan tidak mendapatkan bonus
## 5 8000000      90 8e+05      Karyawan mendapatkan bonus:
## 6 3000000      75 0e+00 Karyawan tidak mendapatkan bonus
```

## 3.3 For loop

Pengecekan angka 1-30 menggunakan for loop untuk menentukan ganjil dan genap dari sebuah bilangan menggunakan for loop

```
for (i in 1:30) {      #for loop untuk mengecek angka ganjil atau genap
  if (i %% 2 == 0) {    #jika angka dibagi 2 bersisa 0 maka itu genap
    cat(i, "adalah angka genap\n")
  } else {
    cat(i, "adalah angka ganjil\n")
  }
}
```

```
## 1 adalah angka ganjil
## 2 adalah angka genap
## 3 adalah angka ganjil
## 4 adalah angka genap
## 5 adalah angka ganjil
## 6 adalah angka genap
## 7 adalah angka ganjil
## 8 adalah angka genap
## 9 adalah angka ganjil
## 10 adalah angka genap
## 11 adalah angka ganjil
## 12 adalah angka genap
## 13 adalah angka ganjil
## 14 adalah angka genap
## 15 adalah angka ganjil
## 16 adalah angka genap
## 17 adalah angka ganjil
```

```
## 18 adalah angka genap
## 19 adalah angka ganjil
## 20 adalah angka genap
## 21 adalah angka ganjil
## 22 adalah angka genap
## 23 adalah angka ganjil
## 24 adalah angka genap
## 25 adalah angka ganjil
## 26 adalah angka genap
## 27 adalah angka ganjil
## 28 adalah angka genap
## 29 adalah angka ganjil
## 30 adalah angka genap
```

### 3.4 While loop

Pengecekan angka 1-30 menggunakan for loop untuk menentukan ganjil dan genap dari sebuah bilangan menggunakan for loop

```
i <- 1 #mulai dari angka 1
while (i <= 30) {
  if (i %% 2 == 0) { #jika angka dibagi 2 bersisa 0 maka itu genap
    cat(i, "adalah angka genap\n")
  } else { #jika tidak berarti ganjil
    cat(i, "adalah angka ganjil\n")
  }
  i <- i + 1 #setiap iterasi, i+1 untuk melanjutkan ke angka berikutnya
}
```

```
## 1 adalah angka ganjil
## 2 adalah angka genap
## 3 adalah angka ganjil
## 4 adalah angka genap
## 5 adalah angka ganjil
## 6 adalah angka genap
## 7 adalah angka ganjil
## 8 adalah angka genap
## 9 adalah angka ganjil
## 10 adalah angka genap
## 11 adalah angka ganjil
## 12 adalah angka genap
## 13 adalah angka ganjil
## 14 adalah angka genap
## 15 adalah angka ganjil
## 16 adalah angka genap
## 17 adalah angka ganjil
## 18 adalah angka genap
## 19 adalah angka ganjil
## 20 adalah angka genap
## 21 adalah angka ganjil
## 22 adalah angka genap
## 23 adalah angka ganjil
## 24 adalah angka genap
```



```
## 25 adalah angka ganjil
## 26 adalah angka genap
## 27 adalah angka ganjil
## 28 adalah angka genap
## 29 adalah angka ganjil
## 30 adalah angka genap
```

### 3.5 Repeat Loop

Membuat program menggunakan repeat loop yang akan menambahkan angka dari 1,2,3 dan seterusnya hingga jumlah atau totalnya mencapai 50. ketika total pencapaian melebihi 50 maka program akan berhenti.

```
total <- 0
awal <- 1
repeat{
  total <- total + awal
  cat("Menambahkan :", awal, "Total saat ini :", total, "\n")
  awal <- awal+1
  if (total >= 50) {
    cat("Penjumlahan selesai. Total akhir :", total, "\n")
    break
  }
}
```

```
## Menambahkan : 1 Total saat ini : 1
## Menambahkan : 2 Total saat ini : 3
## Menambahkan : 3 Total saat ini : 6
## Menambahkan : 4 Total saat ini : 10
## Menambahkan : 5 Total saat ini : 15
## Menambahkan : 6 Total saat ini : 21
## Menambahkan : 7 Total saat ini : 28
## Menambahkan : 8 Total saat ini : 36
## Menambahkan : 9 Total saat ini : 45
## Menambahkan : 10 Total saat ini : 55
## Penjumlahan selesai. Total akhir : 55
```

### 3.6 Nested if-else

Penentuan nilai grade siswa berdasarkan nilai ujian :

Terdapat data nilai 6 siswa berikut :

1. Patricia : 90
2. Faiz : 85
3. Kinan : 70
4. Andika : 60
5. Rafi : 75
6. Stella : 88
7. Fabian : 45

- “A : Sangat Baik” jika nilai  $\geq 85$

- “B : Baik” jika nilai 70-84
- “C : Cukup” jika nilai 50-69
- “D : Kurang” jika nilai < 50

```
nama <- c ("Patricia", "Faiz", "Kinan", "Andika", "Rafi", "Stella", "Fabian")
nilai <- c(90, 85, 70, 60, 75, 88, 45)
keterangan <- character(length(nilai))

#nested if-else
for (i in 1:length(nilai)) {
  if (nilai[i] >= 85){
    keterangan[i] <- "A : Sangat Baik"
  } else if (nilai[i] >= 70) {
    keterangan[i] <- "B : Baik"
  } else if (nilai[i] >= 50) {
    keterangan[i] <- "C : Cukup Baik"
  } else {
    keterangan[i] <- "D : Kurang Baik"
  }
}

#membuat menjadi data frame
hasil <- data.frame(
  Nama = nama,
  Nilai = nilai,
  Keterangan = keterangan
)
print(hasil)
```

```
##      Nama Nilai      Keterangan
## 1 Patricia   90 A : Sangat Baik
## 2   Faiz    85 A : Sangat Baik
## 3   Kinan   70      B : Baik
## 4  Andika   60 C : Cukup Baik
## 5    Rafi   75      B : Baik
## 6  Stella   88 A : Sangat Baik
## 7  Fabian   45 D : Kurang Baik
```

### 3.7 Logika Dasar and (&), or (|)

Universitas Sultan Ageng Tirtayasa menentukan status kelulusan mahasiswa berdasarkan dua kriteria berikut :

1. Nilai rata-rata mahasiswa harus  $\geq 75$
2. Kehadiran mahasiswa harus  $\geq 80\%$

Jika kedua kriteria terpenuhi, mahasiswa dinyatakan lulus. jika kedua kriteria tidak terpenuhi, mahasiswa dinyatakan tidak lulus.

Tentukan status kelulusan untuk daftar mahasiswa berikut :

- Stella : 85, Kehadiran : 85%
- Rafly : 65, Kehadiran : 70%
- Faiz : 78, Kehadiran : 85%
- Pingkan : 90, Kehadiran : 80%
- Zilda : 70, Kehadiran : 60%
- Kinan : 73, Kehadiran : 80%

Tampilkan tabel yang menunjukkan Nama, Nilai rata-rata, Kehadiran, dan Status Kelulusan masing-masing mahasiswa.

```
nama <- c("Stella", "Rafly", "Faiz", "Pingkan", "Zilda", "Kinan")
nilai <- c(85, 65, 78, 90, 70, 73)
kehadiran <- c(85, 70, 85, 80, 60, 80)
kelulusan <- ifelse(nilai >= 75 & kehadiran >= 80, "Lulus", "Tidak Lulus")
#status kelulusan untuk setiap mahasiswa
status <- c()
for (i in 1:length(nama)) {
  if (nilai[i] >= 75 && kehadiran[i] >= 80) {
    status[i] <- "Lulus"
  } else {
    status[i] <- "Tidak Lulus"
  }
}

#menampilkan hasil
hasil <- data.frame>Nama=nama, Nilai=nilai, Kehadiran=kehadiran, Status=status)
print(hasil)
```

##	Nama	Nilai	Kehadiran	Status
## 1	Stella	85	85	Lulus
## 2	Rafly	65	70	Tidak Lulus
## 3	Faiz	78	85	Lulus
## 4	Pingkan	90	80	Lulus
## 5	Zilda	70	60	Tidak Lulus
## 6	Kinan	73	80	Tidak Lulus

### 3.8 Logika Dasar or (|)

Universitas Sultan Ageng Tirtayasa menentukan status kelulusan mahasiswa berdasarkan dua kriteria berikut :

1. Nilai rata-rata mahasiswa harus  $\geq 75$
2. Kehadiran mahasiswa harus  $\geq 80\%$

Jika salah satu kriteria terpenuhi, mahasiswa dinyatakan lulus. jika salah satu kriteria tidak terpenuhi, mahasiswa dinyatakan tidak lulus.

Tentukan status kelulusan untuk daftar mahasiswa berikut :

- Stella : 85, Kehadiran : 85%
- Rafly : 65, Kehadiran : 70%
- Faiz : 78, Kehadiran : 85%
- Pingkan : 90, Kehadiran : 80%
- Zilda : 70, Kehadiran : 60%
- Kinan : 80, Kehadiran : 90%

Tampilkan tabel yang menunjukkan Nama, Nilai rata-rata, Kehadiran, dan Status Kelulusan masing-masing mahasiswa.

```
nama <- c("Stella", "Rafly", "Faiz", "Pingkan", "Zilda", "Kinan")
nilai <- c(85, 65, 78, 90, 70, 73)
kehadiran <- c(85, 70, 85, 80, 60, 80)

for (i in 1:length(nama)) {
  if (nilai[i] >= 75 || kehadiran[i] >= 80) {
    status[i] <- "Lulus"
  } else {
    status[i] <- "Tidak Lulus"
  }
}

#menampilkan hasil
hasil <- data.frame>Nama=nama, Nilai=nilai, Kehadiran=kehadiran, Status=status)
print(hasil)
```

##	Nama	Nilai	Kehadiran	Status
## 1	Stella	85	85	Lulus
## 2	Rafly	65	70	Tidak Lulus
## 3	Faiz	78	85	Lulus
## 4	Pingkan	90	80	Lulus
## 5	Zilda	70	60	Tidak Lulus
## 6	Kinan	73	80	Lulus

## BAB 4 : Kesimpulan

### 4.1 Komparansi antar fungsi

- If vs Ifelse
  - a. If digunakan untuk kondisi sederhana yang hanya terdapat data tunggal
  - b. Ifelse digunakan untuk kondisi yang lebih kompleks dengan memiliki data yang lebih dari 1 data atau pada vektor yang lebih besar.
- For vs While
  - a. For loop digunakan ketika sudah mengetahui batas awal dan akhir pengulangan

- b. While loop digunakan ketika hanya tau syarat batas angka yang diberikan (misalnya ketika didalam soal batas angka harus dibawah atau sama dengan 30), tetapi pengulangannya diatur sendiri.

- Nested ifelse

Nested if-else digunakan untuk menangani suatu kondisi yang bergantung pada kondisi sebelumnya. contoh pada study kasus diatas ingin memeriksa kategori grade nilai sebuah mahasiswa bergantung dengan kriteria yang sudah disebutkan disoal.

- Repeat loop

Repeat loop adalah pengulangan yang akan terus berjalan hingga diberikan perintah untuk berhenti. perbedaannya dengan loop lain adalah untuk menentukan pengulangan itu akan berhenti menggunakan perintah break. seperti pada study kasus diatas bahwa repeat loop diminta untuk melakukan pengulangan angka hingga jumlah atau totalnya mencapai 50. ketika total pencapaian melebihi 50 maka program akan berhenti dan itu menggunakan perintah break.

- Logika dasar and (&), or(|)

Fungsi and (&) digunakan untuk memastikan bahwa kedua kondisi yaitu (nilai ujian dan absensi) harus memenuhi kriteria agar mahasiswa lulus. jika salah satu kondisi tidak terpenuhi, siswa tidak akan lulus

Fungsi or (|) digunakan untuk memastikan bahwa salah satu kondisi baik antara nilai ujian dan absensi telah memenuhi kriteria agar mahasiswa lulus. Jika salah satu kondisi tidak terpenuhi, siswa tidak akan lulus.

## DAFTAR PUSTAKA<sup>1</sup>

---

<sup>1</sup>Muhammad Rizka Fadhlil. November 2021. Intro to R Volume 3 : Dasar-Dasar Bahasa Pemograman R