

Modul 1 Algoritam Iteratif

06 September 2026

Runtunan

Program dengan permasalahan yang sederhana pada umumnya berisi instruksi tanpa adanya proses dalam perulangan atau percabangan. Contoh :

```
#Luas Segitiga
alas <- 10
tinggi <- 6

luas <- 0.5 * alas * tinggi
luas

## [1] 30
```

Percabangan

Algoritma yang di dalam prosesnya terdapat pilihan langkah (cabang) yang akan diambil, tergantung pada kondisi tertentu. Dengan kata lain, alur eksekusi tidak selalu linier, melainkan bisa bercabang sesuai syarat yang diberikan.

Ada kondisi/keputusan (biasanya berupa perbandingan atau logika). Jika kondisi benar, program menjalankan satu perintah. Jika kondisi salah, program bisa berhenti, menjalankan perintah lain, atau melewati bagian tertentu.

Satu Kasus

```
#Luas Segitiga
nilai <- 75

if (nilai >= 60) {
  print("Lulus")
}

## [1] "Lulus"
```

Dua Kasus

```
nilai <- 75

if (nilai >= 60) {
  cat("Lulus\n")
} else {
  cat("Tidak Lulus\n")
}
```

```
## Lulus
```

Tiga Kasus

```
x <- -3

if (x > 0) {
  cat("Positif\n")
} else if (x < 0) {
  cat("Negatif\n")
} else {
  cat("Nol\n")
}
```

```
## Negatif
```

Berikut diberikan contoh menggunakan percabangan dengan dataset “mtcars”.

```
data(mtcars)

mtcars$Kategori <- ifelse(mtcars$mpg > 25,
                          "Irit",
                          ifelse(mtcars$mpg > 15,
                                  "Sedang",
                                  "Boros"))
mtcars$Tenaga <- ifelse(mtcars$hp > 150,
                       "Kuat",
                       "Standar")
head(mtcars[, c("mpg", "hp", "Kategori", "Tenaga")])

##           mpg  hp Kategori  Tenaga
## Mazda RX4    21.0 110   Sedang Standar
## Mazda RX4 Wag 21.0 110   Sedang Standar
## Datsun 710    22.8  93   Sedang Standar
## Hornet 4 Drive 21.4 110   Sedang Standar
## Hornet Sportabout 18.7 175   Sedang Kuat
## Valiant      18.1 105   Sedang Standar
```

Algoritma diatas digunakan untuk mengklasifikasikan mobil berdasarkan nilai mpg (mile per gallon) dan hp (horsepower).

Perulangan

Perulangan yaitu suatu instruksi dengan pola yang sama secara berulang-ulang.

Perulangan For digunakan jika jumlah pengulangan sudah diketahui pasti.

```
for (i in 1:5) {
  cat("Iterasi ke-", i, "\n")
}
```

```
## Iterasi ke- 1
## Iterasi ke- 2
## Iterasi ke- 3
## Iterasi ke- 4
## Iterasi ke- 5
```

Perulangan While digunakan jika jumlah pengulangan tidak pasti, tetapi bergantung pada kondisi logika.

```
x <- 1
while (x <= 5) {
  cat("x =", x, "\n")
  x <- x + 1
}

## x = 1
## x = 2
## x = 3
## x = 4
## x = 5
```

Perulangan Repeat mirip dengan while, tetapi selalu dijalankan minimal sekali, dan harus dihentikan dengan break.

```
x <- 1
repeat {
  cat("x =", x, "\n")
  x <- x + 1
  if (x > 5) {
    break
  }
}

## x = 1
## x = 2
## x = 3
## x = 4
## x = 5
```

Selanjutnya, akan diberikan algoritma iteratif dalam kasus tertentu.

Kasus I

```
x=c(15,18,10,8,20)
n=length(x)
for(i in 2:n){
  if (x[i]>x[1]){
    a=x[1]
    x[1]=x[i]
    x[i]=a
  }
}
```

```

    }
a=x[1]
a
## [1] 20

```

Algoritma diatas digunakan untuk mencari nilai maksimum dari sekumpulan dataset.

Kasus II

```

X = 1
E = 2
while (E>10**(-5)){
  C = X
  X = X-(5*X**2-1.25)/(10*X)
  E = abs(X-C)
}
X
## [1] 0.5

```

Algoritma di atas menggunakan tebakan awal $X=1$, algoritma akan konvergen ke akar positif mendekati 0.5

Kasus III

```

x <- c(15, 18, 10, 8, 20)
n <- length(x)

for (i in 1:(n-1)) {
  for (j in (i+1):n) {
    if (x[i] > x[j]) {
      # tukar nilai
      temp <- x[i]
      x[i] <- x[j]
      x[j] <- temp
    }
  }
}
print(x)
## [1] 8 10 15 18 20

```

Algoritma diatas berfungsi mengurutkan data (sorting) dari kecil ke besar (ascending order).

Kasus IV

```

i <- 1
repeat {
  if (mtcars$hp[i] > 200) {
    cat("Mobil pertama dengan hp > 200 adalah:", rownames(mtcars)[i], "\n")
  }
}

```

```
    break  
  }  
  i <- i + 1  
}
```

Mobil pertama dengan hp > 200 adalah: Duster 360

Algoritma diatas menggunakan repeat loop untuk mencari mobil pertama di dataset mtcars yang punya horsepower (hp) lebih dari 200.