

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2**

**MATERI**

**REVIEW STRUKTUR CONTROL**



**Telkom  
University**

Oleh:

Muhammad Faris Rachmadi

103112400079

IF12-01

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

## I. DASAR TEORI

### Dasar Teori

#### 1. For-Loop

##### 1.1 Paradigma Perulangan

Perulangan merupakan salah satu struktur kontrol yang memungkinkan suatu instruksi yang sama dilakukan berulang kali dalam waktu atau jumlah yang lama. Tanpa instruksi perulangan, maka suatu instruksi akan ditulis dalam jumlah yang sangat banyak. Sebagai contoh adalah menuliskan suatu teks "CAK1BAB3 Algoritma Pemrograman 1" sebanyak 1000 baris. Maka tidak mungkin kita menuliskan instruksi `fmt.Println("CAK1BAB3 Algoritma Pemrograman 1")` sebanyak 1000x (walaupun memungkinkan menggunakan copy-paste).

##### 1.2 Karakteristik For-Loop

Salah satu instruksi perulangan yang paling mudah adalah for-loop, yang mana dengan instruksi ini dapat digunakan untuk mengulangi instruksi sebanyak  $n$  kali (iterasi). Batasan besar nilai dari  $n$  menyesuaikan dengan batasan dari tipe data integer yang digunakan.

##### 1.2.1 Instruksi for-loop memiliki beberapa komponen, yaitu:

- 1) inisialisasi merupakan assignment variabel iterasi yang bertipe integer. Pada contoh di atas biasanya variabel iterasi = 0 atau 1, artinya iterasi dimulai dari 0 atau 1.
- 2) kondisi merupakan suatu operasi bernilai boolean yang menyatakan kapan perulangan harus dilakukan. Pada contoh di atas kondisi adalah variabel iterasi  $\leq n$  (kurang dari atau sama dengan)
- 3) update merupakan ekspresi yang menyatakan perubahan nilai dari variabel iterasi. Pada contoh di atas update adalah variabel iterasi = variabel iterasi + 1.

#### 2. If-Then

##### 2.1 Paradigma Percabangan

Struktur kontrol if-then, atau sering disebut juga dengan percabangan if, adalah salah satu konsep fundamental dalam pemrograman yang memungkinkan alur eksekusi program untuk bercabang berdasarkan kondisi tertentu.

Tujuan utama dari if-then adalah untuk memungkinkan program mengambil keputusan. Artinya, program dapat memilih untuk menjalankan blok kode tertentu hanya jika kondisi yang ditentukan terpenuhi (benar).

## 2.2 Karakteristik If-Then

Penulisan struktur kontrol percabangan dengan menggunakan if-then pada dasarnya terdiri dari dua bagian, yaitu:

- 1) Kondisi, yaitu sesuatu syarat atau ketentuan dari suatu percabangan. Kondisi ini harus bernilai boolean, baik itu variabel ataupun operasi tipe data.
- 2) Aksi, yaitu kumpulan instruksi yang akan dilakukan apabila kondisi terpenuhi atau bernilai true. Artinya baris aksi tidak secara default dieksekusi, tetapi ada syarat yang harus terpenuhi terlebih dahulu sehingga aksi dapat dieksekusi.

## 3.If-Else

Struktur kontrol if-else adalah salah satu konsep fundamental dalam pemrograman yang memungkinkan program untuk membuat keputusan berdasarkan kondisi tertentu. Secara sederhana, if-else memungkinkan program untuk menjalankan blok kode tertentu jika suatu kondisi terpenuhi (benar), dan menjalankan blok kode lain jika kondisi tersebut tidak terpenuhi (salah).

Penulisan struktur kontrol percabangan dengan menggunakan else-if terdiri dari dua bagian, yaitu:

- 1) Kondisi, yaitu sesuatu syarat atau ketentuan dari suatu percabangan. Kondisi ini harus bernilai boolean, baik itu variabel ataupun operasi tipe data.
- 2) Aksi, yaitu kumpulan instruksi yang akan dilakukan apabila kondisi terpenuhi atau bernilai true.
- 3) Aksi lain, yaitu kumpulan instruksi yang akan dilakukan apabila kondisi terpenuhi atau bernilai false. Artinya aksi dan aksi lain ini merupakan pilihan, yang mana hanya salah satu aksi saja yang akan dieksekusi sesuai dengan nilai dari kondisi.

## II. GUIDED

### Guided 1

Code:

```
1  package main
2
3  import "fmt"
4
5  func main() {
6      var greetings = "selamat datang di dunia DAP"
7      var a, b int
8      fmt.Println(greetings)
9      fmt.Scan(&a, &b)
10     fmt.Printf("%v + %v = %v\n", a, b, a+b)
11 }
12
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2> go run "c:\Users\Faris\Documents\ALGORITMA
selamat datang di dunia DAP
7 5
7 + 5 = 12
```

Deskripsi:

Program Go ini adalah program sederhana yang menggunakan paket "fmt" untuk mencetak pesan sambutan dan melakukan operasi penjumlahan dua angka yang dimasukkan oleh pengguna.

## Guided 2

Code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     for i := 1; i <= 5; i++ {
7         fmt.Println("Iterasi ke-", i)
8     }
9 }
10
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2> go run "c:\Users\Faris\
Iterasi ke- 1
Iterasi ke- 2
Iterasi ke- 3
Iterasi ke- 4
Iterasi ke- 5
```

Deskripsi:

Program ini adalah sebuah contoh program sederhana yang menggunakan pengulangan (loop) untuk mencetak angka dari 1 hingga 5.

## Guided 3

Code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     nilai := 80
8     pctHadir := 0.75
9     adaTubes := true
10
11     var indeks string
12
13     if nilai > 75 && adaTubes {
14         indeks = "A"
15     } else if nilai > 65 {
16         indeks = "B"
17     } else if nilai > 50 && pctHadir > 0.7 {
18         indeks = "C"
19     } else {
20         indeks = "F"
21     }
22     fmt.Printf("Nilai %d dengan kehadiran %.2f%% dan buat tubes= %t, mendapat indeks %s\n",
23     nilai, pctHadir*100, adaTubes, indeks)
24 }
25
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2> go run "c:\Users\Faris\
Nilai 80 dengan kehadiran 75.00% dan buat tubes= true, mendapat indeks A
```

Deskripsi:

Program Golang ini menentukan indeks nilai (A, B, C, atau F) berdasarkan nilai numerik, persentase kehadiran, dan status penyelesaian tugas besar (tubes) menggunakan serangkaian kondisi if-else, kemudian mencetak nilai, persentase kehadiran, status tubes, dan indeks nilai yang dihasilkan.

### III. UNGUIDED

#### Unguided 1

Code:

```
1 package main
2
3 //Muhammad Faris Rachmadi
4 //103112400079
5 import "fmt"
6
7 func main() {
8     var tahun int
9     fmt.Print("Tahun: ")
10    fmt.Scan(&tahun)
11    if (tahun%400 == 0) || (tahun%4 == 0 && tahun%100 != 0) {
12        fmt.Println("kabisat: true")
13    } else {
14        fmt.Println("kabisat: false")
15    }
16 }
17
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Unguided 1.go"
Tahun: 2016
kabisat: true
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Unguided 1.go"
Tahun: 2018
kabisat: false
```

Deskripsi:

Program Go ini berfungsi untuk menentukan apakah suatu tahun adalah tahun kabisat atau bukan. Program meminta input tahun dari pengguna, kemudian memeriksa apakah tahun tersebut memenuhi kriteria tahun kabisat, yaitu jika tahun tersebut dapat dibagi oleh 400, atau jika dapat dibagi oleh 4 tetapi tidak dibagi oleh 100. Jika salah satu dari kondisi tersebut terpenuhi, program akan menampilkan kabisat: true, yang menandakan bahwa tahun tersebut adalah tahun kabisat, sedangkan jika tidak, program akan menampilkan kabisat: false. Program ini menggunakan pernyataan kondisi if untuk mengevaluasi tahun yang dimasukkan dan memberikan hasil yang sesuai.

## Unguided 2

Code:

```
1 package main
2
3 //Muhammad Faris Rachmadi
4 //10311240079
5
6 import "fmt"
7
8 func main() {
9     var k float64
10
11     fmt.Print("Nilai K: ")
12     fmt.Scan(&k)
13
14     pembilang := (4*k + 2) * (4*k + 2)
15     penyebut := (4*k + 1) * (4*k + 3)
16     hasil := pembilang / penyebut
17
18     fmt.Printf("Nilai f(K) = %.10f\n", hasil)
19 }
20
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2> go run "c:\Users\Faris\
Nilai K: 100
Nilai f(K) = 1.0000061880
```

Deskripsi:

Program Go ini bertujuan untuk menghitung nilai dari suatu fungsi matematika berbasis input variabel K. Program akan meminta pengguna untuk memasukkan nilai K, lalu menghitung hasil dari rumus berikut:

$$f(K) = \frac{(4K+2)^2}{(4K+1) \times (4K+3)}$$

Hasil perhitungan akan ditampilkan dengan presisi hingga 10 angka desimal.



## Unguided 2.1

Code:

```
1  package main
2
3  //Muhammad Faris Rachmadi
4  //10311240079
5
6  import "fmt"
7
8  func main() {
9      var k int
10
11      fmt.Print("Nilai K: ")
12      fmt.Scan(&k)
13
14      hasil := 1.0
15      for i := 0; i <= k; i++ {
16          pembilang := float64((4*i + 2) * (4*i + 2))
17          penyebut := float64((4*i + 1) * (4*i + 3))
18          hasil *= pembilang / penyebut
19      }
20
21      fmt.Printf("Nilai akar 2 : %.10f\n", hasil)
22  }
23
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2> go run "c:\Users\Faris\
Nilai K: 10
Nilai akar 2 : 1.4062058441
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2> go run "c:\Users\Faris\
Nilai K: 100
Nilai akar 2 : 1.4133387072
```

Deskripsi:

Program ini merupakan modifikasi dari program sebelumnya yang menghitung fungsi berbasis input variabel K. Namun, kali ini program bertujuan untuk menghitung nilai akar 2 menggunakan pendekatan deret berulang berdasarkan input nilai K. Deret ini didasarkan pada rumus:

$$\sqrt{2} \approx \prod_{i=0}^K \frac{(4i+2)^2}{(4i+1)(4i+3)}$$

Program ini meminta pengguna untuk memasukkan nilai K, kemudian menghitung hasil perkalian dari deret hingga K iterasi. Hasil akhir akan ditampilkan dengan presisi hingga 10 angka desimal.

## Unguided 3

Code:

```
1 package main
2
3 //Muhammad Faris Rachmadi
4 //10311240079
5
6 import "fmt"
7
8 func main() {
9     var beratgram int
10
11     fmt.Print("Berat parcel (gram): ")
12     fmt.Scan(&beratgram)
13
14     beratKg := beratgram / 1000
15     sisagram := beratgram % 1000
16
17     biayaKg := beratKg * 10000
18     biayasisa := 0
19
20     if beratKg > 10 && sisagram < 1000 {
21         biayasisa = 0
22     } else if sisagram >= 500 {
23         biayasisa = sisagram * 5
24     } else {
25         biayasisa = sisagram * 15
26     }
27
28     totalbiaya := biayaKg + biayasisa
29
30     fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisagram)
31     fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg, biayasisa)
32     fmt.Printf("Total biaya: Rp. %d\n", totalbiaya)
33 }
34
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2> go run "c:\Users
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
```

Deskripsi:

Program Go ini berfungsi untuk menghitung biaya pengiriman parcel berdasarkan beratnya. Pengguna diminta untuk memasukkan berat parcel dalam gram, yang kemudian dikonversi menjadi kilogram dan sisa gram. Biaya pengiriman dihitung berdasarkan tarif Rp. 10.000 per kilogram, dengan biaya tambahan untuk sisa gram yang dihitung menggunakan tarif berbeda: Rp. 5 per gram jika berat sisa lebih dari atau sama dengan 500 gram, atau Rp. 15 per gram jika kurang dari 500 gram. Program ini akhirnya menampilkan rincian berat dalam kilogram dan gram, biaya untuk berat kilogram dan sisa gram, serta total biaya pengiriman yang harus dibayar.

#### **IV. KESIMPULAN**

Kesimpulan dari laporan praktikum ini adalah bahwa pemahaman terhadap struktur kontrol dalam pemrograman, seperti perulangan dengan for-loop dan percabangan dengan if-then serta if-else, sangat penting untuk membangun program yang efisien dan mudah dipahami. Praktikum ini menekankan penerapan struktur kontrol dalam berbagai situasi, seperti pengulangan untuk menghasilkan output berulang dan percabangan untuk menentukan jalur eksekusi program berdasarkan kondisi tertentu. Beberapa contoh program yang diberikan, seperti penentuan tahun kabisat, perhitungan fungsi matematika, dan perhitungan biaya pengiriman, memperlihatkan bagaimana struktur kontrol ini digunakan untuk memecahkan masalah yang spesifik. Secara keseluruhan, praktikum ini memberikan pemahaman yang lebih baik tentang bagaimana struktur kontrol dapat membantu dalam menyusun solusi yang lebih terstruktur dan terorganisir dalam pemrograman.

## **V. REFERENSI**

Prasti Eko Yunanto, S.T., M.Kom. MODUL PRAKTIKUM 5 & 6 FOR & LOOP ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA

Prasti Eko Yunanto, S.T., M.Kom. MODUL PRAKTIKUM 9 - IF-THEN ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA

Prasti Eko Yunanto, S.T., M.Kom. MODUL PRAKTIKUM 10 - ELSE-IF ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA

MODUL 2 PRAKTIKUM ALGORITMA PEMROGRAMAN 2 - REVIEW STRUKTUR CONTROL