

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2  
MODUL 2  
REVIEW STRUKTUR KONTROL**



Oleh:

**HISYAM NURDIATMOKO**

103112400049

IF-12-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## **I. DASAR TEORI**

### **For-Loop (Perulangan)**

Perulangan adalah struktur kontrol yang memungkinkan eksekusi instruksi yang sama berulang kali sesuai dengan jumlah atau kondisi tertentu. For-loop merupakan jenis perulangan yang dilakukan berdasarkan iterasi. Terdiri dari tiga bagian utama: inisialisasi (penentuan nilai awal variabel iterasi), kondisi (syarat untuk melanjutkan perulangan), dan update (perubahan nilai variabel iterasi). Struktur ini sangat efisien digunakan untuk tugas yang melibatkan pengulangan instruksi, seperti menampilkan deret angka atau pengolahan data.

### **If-Then (Percabangan)**

Percabangan digunakan untuk mengeksekusi instruksi berdasarkan kondisi tertentu. Dalam struktur `if-then`, kondisi yang bernilai benar akan memicu eksekusi aksi yang ditentukan. Struktur ini hanya menjalankan aksi jika kondisi yang diuji bernilai `true`, dan mengabaikan aksi jika kondisi tersebut `false`. Percabangan ini berguna untuk membuat keputusan dalam program, seperti memeriksa apakah suatu bilangan positif atau tidak.

### **Else-If (Percabangan Bertingkat)**

Else-if adalah pengembangan dari percabangan `if-then` yang memungkinkan pemeriksaan beberapa kondisi secara berurutan. Jika kondisi pertama tidak terpenuhi, program akan memeriksa kondisi berikutnya, dan seterusnya. Ini memungkinkan untuk menangani beberapa kemungkinan kondisi dalam satu struktur percabangan yang lebih kompleks, seperti menentukan kategori nilai berdasarkan beberapa kriteria.

## II. GUIDED

Source code Guided 1:

```
coso1 > -go 1.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var greethings = "Selamat datang di dunia DAP"
7      var a, b int
8      fmt.Println(greethings)
9      fmt.Scanln(&a, &b)
10     fmt.Printf("%v + %v = %v\n", a, b, a+b)
11 }
```

Output :

```
● PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\coso1\1.go"
Selamat datang di dunia DAP
20 7
20 + 7 = 27
○ PS D:\1_Matkul\Alpro2\week2> █
```

Deskripsi :

Program Go Guided 1 ini menampilkan pesan "Selamat datang di dunia DAP", kemudian meminta input dua angka dari pengguna dengan fungsi `fmt.Scanln`. Kemudian program menjumlahkan kedua angka dan menampilkan hasilnya menggunakan `fmt.Printf`.

Source code Guided 2 :

```
coso2 > go 2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     for i := 1; i <= 5; i++ {
7         fmt.Println("Literasi ke-", i)
8     }
9 }
```

Output :

```
PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\coso2\2.go"
Literasi ke- 1
Literasi ke- 2
Literasi ke- 3
Literasi ke- 4
Literasi ke- 5
```

Deskripsi :

Program Go Guided 2 ini menggunakan perulangan for untuk mencetak teks yang menunjukkan nomor iterasi dari 1 hingga 5. Pada setiap iterasi, nilai i akan bertambah satu, dimulai dari 1 hingga 5, dan setiap nilai tersebut dicetak dengan pesan "Literasi ke-" diikuti oleh nilai iterasi saat ini kemudian mengeluarkan output yang dicetak dengan fungsi `fmt.Println`.

### Source code Guided 3:

```
coso3 > go 3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     nilai := 80
7     pctHadir := 0.75
8     adaTubes := true
9     var indeks string
10    if nilai > 75 && adaTubes {
11        indeks = "A"
12    } else if nilai > 65 {
13        indeks = "B"
14    } else if nilai > 50 && pctHadir >= 0.70 {
15        indeks = "C"
16    } else {
17        indeks = "F"
18    }
19    fmt.Printf("Nilai %d dengan kehadiran %.2f%% dan buat tubes=%t, mendapat indeks %s\n", nilai, pctHadir*100, adaTubes, indeks)
20 }
```

### Output:

```
PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\coso3\3.go"
Nilai 80 dengan kehadiran 75.00% dan buat tubes=true, mendapat indeks A
PS D:\1_Matkul\Alpro2\week2> 
```

### Deskripsi:

Program Go Guided 3 ini digunakan untuk menentukan indeks nilai seorang siswa berdasarkan beberapa kondisi. Variabel `nilai`, `pctHadir`, dan `adaTubes` digunakan untuk menyimpan nilai ujian, persentase kehadiran, dan apakah siswa mengerjakan tugas besar (TUBES). Program menggunakan `if-else` untuk mengecek kondisi yang berbeda dan menentukan indeks yang sesuai. Jika nilai lebih dari 75 dan siswa mengerjakan tugas besar, indeks yang diberikan adalah "A". Jika nilai lebih dari 65, indeksnya "B". Jika nilai lebih dari 50 dan persentase kehadiran lebih dari atau sama dengan 70%, indeksnya "C". Jika tidak ada kondisi yang terpenuhi, indeksnya "F". Hasil akhirnya dicetak menggunakan `fmt.Printf`, menampilkan nilai, persentase kehadiran, status ujian ulang, dan indeks yang diperoleh.

### III. UNGUIDED

Source Code Unguided 1 (2A no 2):

```
2a_2 > -go 1.go > ...
1  package main
2  //HISYAM NURDIATMOKO 103112400049
3  import "fmt"
4
5  func main() {
6      var tahun int
7      fmt.Print("Tahun: ")
8      fmt.Scan(&tahun)
9      if (tahun%400 == 0) || (tahun%4 == 0 && tahun%100 != 0) {
10         fmt.Println("kabisat: true")
11     } else {
12         fmt.Println("kabisat: false")
13     }
14 }
```

Output:

```
PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\2a_2\1.go"
Tahun: 2016
kabisat: true
● PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\2a_2\1.go"
Tahun: 2000
kabisat: true
● PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\2a_2\1.go"
Tahun: 2018
kabisat: false
```

Deskripsi:

Program Go Unguided 1 ini digunakan untuk menentukan apakah tahun yang diinputkan merupakan tahun kabisat atau bukan. Kemudian program memeriksa apakah tahun tersebut memenuhi kriteria tahun kabisat atau tidak. Jika tahun tersebut bisa dibagi habis oleh 400, atau bisa dibagi habis oleh 4 dan tidak bisa dibagi habis oleh 100, maka tahun tersebut adalah tahun kabisat dan program mencetak "kabisat: true". Jika tidak, program mencetak "kabisat: false". Program Unguided 1 ini menggunakan modulus untuk memeriksa pembagian tahun tersebut.

Source code Unguided 2 (2B no 4):

```
2b_4 > go 2.go > ...
1  package main
2  //HISYAM NURDIATMOKO 103112400049
3  import "fmt"
4
5  func main() {
6      var k int
7      var hasil float64
8      fmt.Print("Nilai K = ")
9      fmt.Scan(&k)
10     hasil = float64((4*k+2)*(4*k+2)) / float64((4*k+1)*(4*k+3))
11     fmt.Printf("Nilai f(K) = %.10f\n", hasil)
12 }
```

Output:

```
PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\2b_4\2.go"
Nilai K = 100
Nilai f(K) = 1.0000061880
```

Deskripsi:

Program Go Unguided 2 ini digunakan untuk menghitung nilai fungsi matematika  $f(K)$  berdasarkan nilai input  $k$  yang diinputkan. Program dimulai dengan deklarasi variabel  $k$  sebagai integer dan hasil sebagai float64. Pengguna diminta untuk memasukkan nilai  $k$ , yang kemudian digunakan dalam rumus matematika berikut:

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

Setelah perhitungan dilakukan, hasilnya dicetak dengan format angka desimal hingga 10 angka di belakang koma menggunakan `fmt.Printf`.

Source code Unguied 2 akar 2 (2B no 4):

```
2b_4 > go 22.go > ...
1 package main
2 //HISYAM NURDIATMOKO 103112400049
3 import "fmt"
4
5 func main() {
6     var k int
7     var hasil float64
8     fmt.Print("Nilai K = ")
9     fmt.Scan(&k)
10    hasil = 1.0
11    for i := 0; i <= k; i++ {
12        pembilang := (4*i + 2) * (4*i + 2)
13        penyebut := (4*i + 1) * (4*i + 3)
14        hasil = hasil * float64(pembilang) / float64(penyebut)
15    }
16    fmt.Printf("nilai akar 2 = %.10f\n", hasil)
17 }
```

Output:

```
PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\2b_4\22.go"
Nilai K = 10
nilai akar 2 = 1.4062058441
● PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\2b_4\22.go"
Nilai K = 100
nilai akar 2 = 1.4133387072
● PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\2b_4\22.go"
Nilai K = 1000
nilai akar 2 = 1.4141252651
```

Deskripsi:

Program Go Unguied 2 akar 2 ini menghitung perkiraan nilai akar 2 dari program Unguied 2.



Source code Unguided 3 (2C no 1):

```
2c_1 > -go 3.go > ...
1  package main
2  //HISYAM NURDIATMOKO 103112400049
3  import "fmt"
4
5  func main() {
6      var berat int
7      fmt.Print("Berat parsel (gram): ")
8      fmt.Scan(&berat)
9      beratKg := berat / 1000
10     sisaGram := berat % 1000
11     biaya := beratKg * 10000
12     biayaTambahan := 0
13     if beratKg > 10 {
14         biayaTambahan = 0
15     } else if sisaGram >= 500 {
16         biayaTambahan = sisaGram * 5
17     } else {
18         biayaTambahan = sisaGram * 15
19     }
20     fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisaGram)
21     fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biaya, biayaTambahan)
22     totalBiaya := biaya + biayaTambahan
23     fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
24 }
```

Output:

```
PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\2c_1\3.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
• PS D:\1_Matkul\Alpro2\week2> go run "d:\1_Matkul\Alpro2\week2\2c_1\3.go"
• Berat parsel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
```

Deskripsi:

Program Go Unguided 3 ini menghitung biaya pengiriman parsel berdasarkan berat dalam gram. Program meminta input berat parsel dalam gram. Berat ini kemudian diubah menjadi kilogram. Biaya dasar dihitung berdasarkan kilogram, dengan biaya Rp 10.000 per kilogram. Kemudian, program menghitung biaya tambahan tergantung pada sisa gram: jika lebih dari atau sama dengan 500 gram, biaya tambahan dihitung dengan tarif Rp 5 per gram; jika kurang dari 500 gram, tarif biaya tambahan menjadi Rp 15 per gram. Program ini kemudian menampilkan rincian berat, biaya, dan total biaya pengiriman menggunakan `fmt.Printf`.

#### **IV. KESIMPULAN**

Berdasarkan laporan praktikum ini, dapat disimpulkan bahwa pemahaman mengenai struktur kontrol dalam pemrograman, seperti for-loop, if-then, dan else-if, sangat penting dalam mengembangkan aplikasi yang efisien dan terstruktur. Praktikum ini menunjukkan bagaimana setiap struktur kontrol dapat diaplikasikan pada berbagai kasus nyata, mulai dari perhitungan matematika, pemeriksaan tahun kabisat, hingga penentuan biaya pengiriman berdasarkan berat.

Pada program guided, mahasiswa diajarkan untuk mengimplementasikan dasar-dasar pemrograman seperti input/output dan pengulangan menggunakan for-loop. Sementara itu, dalam unguided practice, mahasiswa diberikan tantangan untuk mengembangkan program dengan logika yang lebih kompleks seperti menghitung nilai fungsi matematika dan biaya pengiriman.

Secara keseluruhan, laporan ini memperlihatkan bahwa penguasaan struktur kontrol dasar sangat krusial untuk menyelesaikan berbagai permasalahan dalam pemrograman komputer dan memperdalam pemahaman terkait algoritma.

## **V. REFERENSI**

MODUL PRAKTIKUM 5 & 6 - FOR-LOOP ALGORITMA  
DAN PEMROGRAMAN 1 S1 INFORMATIKA

MODUL PRAKTIKUM 9 - IF-THEN ALGORITMA DAN  
PEMROGRAMAN 1 S1 INFORMATIKA

MODUL PRAKTIKUM 10 - ELSE-IF ALGORITMA DAN  
PEMROGRAMAN 1 S1 INFORMATIKA