

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



Oleh:

DWI OKTA SURYANINGRUM

103112400066

12-IF-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## I. DASAR TEORI

Struktur kontrol adalah elemen dasar dalam bahasa pemrograman apa pun. Fungsinya untuk menjalankan blok kode tertentu berdasarkan kondisi tertentu atau mengulangi proses dengan serangkaian nilai. Dalam bahasa Go, terdapat tiga struktur kontrol utama, yaitu:

1. **if** – Menjalankan blok kode jika kondisi terpenuhi.
2. **for** – Melakukan perulangan pada sekumpulan nilai atau sejumlah iterasi tertentu.
3. **switch** – Memilih dan menjalankan salah satu blok kode berdasarkan beberapa kemungkinan kondisi.

### 1. Pernyataan if

Pernyataan **if** digunakan untuk mengeksekusi blok kode hanya jika kondisi yang diberikan bernilai benar. Jika kondisi salah, Anda bisa menggunakan blok **else** untuk mengeksekusi kode lainnya. Selain itu, **else if** memungkinkan Anda mengecek beberapa kondisi secara berurutan.

### 2. Perulangan for

Struktur **for** digunakan untuk melakukan iterasi pada sekumpulan nilai atau mengeksekusi blok kode berulang kali hingga kondisi tertentu terpenuhi. Tidak seperti bahasa pemrograman lain yang memiliki perulangan **while** atau **do-while**, bahasa Go hanya menggunakan **for** untuk semua jenis perulangan.

Ada tiga bentuk utama **for** dalam Go:

- **for init; condition; increment** – Bentuk yang paling umum, mirip dengan perulangan **for** pada C, C++, dan Java.
- **for condition** – Bertindak seperti perulangan **while**, yang akan terus berjalan selama kondisi benar.
- **for** tanpa parameter – Perulangan tak terbatas yang hanya berhenti saat ada pernyataan **break** atau **return**.

### 3. Pernyataan switch

Pernyataan **switch** digunakan untuk memilih salah satu dari beberapa blok kode berdasarkan kondisi tertentu. Pernyataan ini adalah alternatif yang lebih sederhana daripada menggunakan banyak pernyataan **if-else**.

### Kesimpulan

Pemahaman tentang struktur kontrol **if**, **for**, dan **switch** sangat penting dalam membangun alur program pada bahasa Go. Menguasai konsep ini akan membantu Anda menulis kode yang lebih bersih, efisien, dan mudah dipelihara, sehingga meningkatkan kualitas Anda sebagai pengembang Go.

## II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

### a. Coso 1

```
1 // DWI OKTA SURYANINGRUM
2
3 package main
4
5 import "fmt"
6
7 func main() {
8     var greetings = "Selamat datang di dunia DAB" //membuat variabel greetings dengan tipe data string yang menyimpan teks "Selamat datang di dunia DAB"
9     var a, b int //membuat variabel a,b dengan tipe data integer
10
11     fmt.Println(greetings) //menampilkan teks yang disimpan di variabel greetings
12     fmt.Scan(&a, &b) //membaca input dari pengguna kemudian disimpan ke variabel a dan b
13
14     fmt.Printf("%v + %v = %v\n", a, b, a+b) //menampilkan hasil penjumlahan dalam format. %v digunakan sebagai placeholder untuk menampilkan nilai dari variabel.
15     // a + b akan dihitung, lalu hasilnya ditampilkan.
16 }
```

Output :

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE EXPLORER PORTS TERMINAL
mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Coso/Coso-1.go"
Selamat datang di dunia DAB
2 4
2 + 4 = 6
```

### b. Coso 2

```
1 // DWI OKTA SURYANINGRUM
2
3 package main
4
5 import "fmt"
6
7 func main() {
8     for i := 1; i <= 5; i++ {
9         fmt.Println("Iterasi ke-", i)
10     }
11 }
12
13 // kode tersebut menggunakan perulangan untuk mencetak teks sebanyak 5 kali
14 // Perulangan dimulai dari nilai i = 1 dan akan terus berjalan selama i <= 5
15 // Setiap putaran, angka tersebut dicetak bersama tulisan "Iterasi ke-" diikuti dengan nilai i. Setelah mencapai angka 5, perulangan berhenti secara otomatis.
```

Output :

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE EXPLORER PORTS TERMINAL
mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Coso/Coso-2.go"
Iterasi ke- 1
Iterasi ke- 2
Iterasi ke- 3
Iterasi ke- 4
Iterasi ke- 5
```

### c. Coso 3

```
1 // DWI OKTA SURYANINGRUM
2 package main
3
4 import "fmt"
5
6 func main() {
7     nilai := 80 // deklarasi variabel dengan tipe data int dan menyimpan nilai 80
8     pctHadir := 0.75 // deklarasi variabel dengan tipe data float64 dan menyimpan nilai 0.75 / 75% sebagai (presentase kehadiran)
9     adaTubes := true // deklarasi variabel dengan tipe data boolean bernilai "true" yang menunjukkan mahasiswa mengerjakan tubes
10
11     var indeks string // deklarasi variabel indeks dengan tipe data string
12
13     if nilai > 75 && adaTubes { // kondisi jika nilai lebih dari 75 dan mengerjakan tubes maka indeks nya adalah "A"
14         indeks = "A"
15     } else if nilai > 65 { // jika nilai lebih dari 65 maka indeks nya "B"
16         indeks = "B"
17     } else if nilai > 50 && pctHadir > 0.7 { // jika nilai lebih dari 50 dan presentasi hadir lebih dari 70% maka indeks nya "C"
18         indeks = "C"
19     } else { // tidak memenuhi semua syarat maka indeks nya "F"
20         indeks = "F"
21     }
22
23     // output akan menampilkan hasil berupa nilai, kehadiran, tugas besar, dan indeks yang didapat.
24     fmt.Printf("Nilai %d dengan kehadiran %.2f%% dan buat tubes = %t, mendapatkan indeks %s\n", nilai, pctHadir*100, adaTubes, indeks)
25 }
```

Output :

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE EXPLORER PORTS TERMINAL
mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Coso/Coso-3.go"
Nilai 80 dengan kehadiran 75.00% dan buat tubes = true, mendapatkan indeks A
```

### III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

#### a. Latsol 1

```
1 // DWI OKTA SURYANINGRUM
2 package main
3
4 import "fmt"
5
6 func main() {
7     var tahun int // deklarasi variabel tahun dengan tipe data integer
8     var kabisat bool // deklarasi variabel kabisat dengan tipe data boolean
9
10    fmt.Print("Tahun : ") // menampilkan string "Tahun : "
11    fmt.Scan(<math>tahun</math>) // membaca input dari pengguna dan menyimpannya dalam variabel tahun
12
13    //logika dari var kabisat dimana jika tahun habis dibagi 400 atau habis dibagi 4 tetapi tidak habis ketika dibagi 100 maka hasilnya bernilai "true"
14    kabisat = tahun % 400 == 0 || tahun % 4 == 0 && 4 % 100 != 0
15
16    fmt.Println("Kabisat :", kabisat) // menampilkan hasil pengecekan apakah tahun kabisat atau tidak
17 }
```

Output :

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE EXPLORER PORTS TERMINAL
● mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-1.go"
Tahun : 2016
Kabisat : true
● mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-1.go"
Tahun : 2000
Kabisat : true
● mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-1.go"
Tahun : 2018
Kabisat : false
```

#### b. Latsol 2

```
1 // DWI OKTA SURYANINGRUM
2 package main
3
4 import "fmt"
5 import "math"
6
7 func main() {
8     var k, fk float64 // deklarasi variabel k, fk dengan tipe data float64
9
10    fmt.Scan(<math>k</math>) // membaca input dari pengguna dan menyimpan di variabel k
11
12    fk = math.Pow((4*k+2),2) / ((4*k+1) * (4*k+3)) // menghitung (4k + 2) pangkat 2. kemudian dibagi dengan (4k+1) dikali (4k+3)
13
14    fmt.Println("Nilai K =", k) // menampilkan nilai k yang diinputkan user
15    fmt.Printf("Nilai f(K) = %.10f\n", fk) // menampilkan hasil perhitungan dari variabel fk dengan presisi 10 angka di belakang koma.
16 }
```

Output :

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE EXPLORER PORTS TERMINAL
● mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-2.go"
100
Nilai K = 100
Nilai f(K) = 1.0000061880
```

Modifikasi

```

1 // DWI OKTA SURYANINGRUM
2 package main
3
4 import (
5     "fmt"
6     "math"
7 )
8
9 func main() {
10     var k int // deklarasi variabel dengan tipe data integer
11
12     fmt.Print("Nilai K: ") // menampilkan string "Nilai K: "
13     fmt.Scan(&k) // membaca input dari user dan menyimpan di variabel k
14
15     result := 1.0 // deklarasi variabel result dengan tipe data float64 dan menyimpan nilai 1.0
16
17     for i := 0; i < k; i++ { // (perulangan) melakukan perhitungan sebanyak K kali:
18         numerator := math.Pow(float64(4*i+2), 2) // Menghitung pangkat dua dari (4*i + 2)
19         denominator := float64((4*i + 1) * (4*i + 3)) // Mengalikan (4*i + 1) dengan (4*i + 3)
20         result *= numerator / denominator // Hasil bagi pembilang dan penyebut akan dikalikan dengan variabel result secara terus-menerus.
21     }
22
23     fmt.Printf("Hasil dari operasi fungsi = %.10f\n", result) // hasil akhir ditampilkan dengan ketelitian 10 angka di belakang koma:
24 }

```

Output :

```

PROBLEMS 8 OUTPUT DEBUG CONSOLE EXPLORER PORTS TERMINAL
• mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-2.go"
Nilai K: 10
Hasil dari operasi fungsi = 1.4054086752
• mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-2.go"
Nilai K: 100
Hasil dari operasi fungsi = 1.4133299615
• mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-2.go"
Nilai K: 1000
Hasil dari operasi fungsi = 1.4141251768

```

### c. Latsol 3

```

1 // DWI OKTA SURYANINGRUM
2 package main
3
4 import "fmt"
5
6 func main() {
7     var berat_parsel, tambahan_biaya int // deklarasi variabel dengan tipe data integer
8
9     fmt.Print("Berat parsel (gram) : ") // menampilkan teks "Berat parsel (gram) : "
10    fmt.Scan(&berat_parsel) // membaca input dari pengguna kemudian menyimpannya dalam variabel berat_parsel
11
12    total_berat := berat_parsel/1000 // perhitungan total berat dalam satuan kilogram (dibagi 1000)
13    sisa_berat := berat_parsel - (total_berat*1000) // perhitungan sisa berat dalam satuan gram (berat yang tidak mencapai 1kg)
14    jasa_kirim := total_berat*10000 // biaya pengiriman per kg yaitu 10.000
15
16    if sisa_berat >= 500 { // Jika sisa berat lebih dari atau sama dengan 500 gram, biaya tambahan = Rp. 5 per gram.
17        tambahan_biaya = (5*sisa_berat)
18    } else { // Jika sisa berat kurang dari 500 gram, biaya tambahan = Rp. 15 per gram.
19        tambahan_biaya = (15 * sisa_berat)
20    }
21
22    total_biaya := jasa_kirim + tambahan_biaya // untuk total biaya dihitung dari jasa kirim + tambahan biaya
23
24    fmt.Println("Detail berat :", total_berat, "kg + ", sisa_berat, "gr") // menampilkan Detail berat dalam kg dan gram.
25    fmt.Println("Detail Biaya : Rp.", jasa_kirim, "+ Rp.", tambahan_biaya) // menampilkan Rincian biaya jasa kirim dan tambahan biaya.
26    fmt.Println("Total biaya : Rp.", total_biaya) // menampilkan Total biaya pengiriman.
27 }

```

Output :

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE EXPLORER PORTS TERMINAL
● mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-3.go"
Berat parcel (gram) : 8500
Detail berat : 8 kg + 500 gr
Detail Biaya : Rp. 80000 + Rp. 2500
Total biaya : Rp. 82500
● mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-3.go"
Berat parcel (gram) : 9250
Detail berat : 9 kg + 250 gr
Detail Biaya : Rp. 90000 + Rp. 3750
Total biaya : Rp. 93750
● mymac@mymacs-MacBook-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/Week-1/Latsol/latsol-3.go"
Berat parcel (gram) : 11750
Detail berat : 11 kg + 750 gr
Detail Biaya : Rp. 110000 + Rp. 3750
Total biaya : Rp. 113750
```



#### **IV. KESIMPULAN**

Struktur kontrol di Go memberikan fleksibilitas dalam mengontrol alur eksekusi program. Meskipun Go memiliki if-else, switch, dan for sebagai struktur utama, dengan pendekatan yang sederhana dan jelas, Go mendorong penggunaan kode yang mudah dipahami dan dibaca. Struktur kontrol ini memungkinkan penulisan program yang efisien, baik dalam menentukan kondisi tertentu untuk dieksekusi maupun dalam pengulangan tugas yang berulang.

## **V.     REFERENSI**

Codex, A. C. (2023). Understanding Go's control structures: if, for, and switch.

Retrieved from <https://reintech.io/blog/understanding-gos-control-structures-if-for-switch>