

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 2
“REVIEW STRUKTUR KONTROL”



DISUSUN OLEH:
Muhammad Shabrian Fadly
103112400087
S1 IF-12-01

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

I. DASAR TEORI

Struktur kontrol adalah konsep fundamental dalam pemrograman yang memungkinkan pengembang untuk mengontrol aliran eksekusi program berdasarkan kondisi atau perulangan. Dalam Go (Golang), struktur kontrol dirancang dengan kesederhanaan dan kejelasan sebagai prioritas utama. Berikut penjelasan dasar tentang struktur kontrol dalam Go:

1. Percabangan (Branching)

Percabangan memungkinkan program untuk mengeksekusi blok kode tertentu berdasarkan kondisi yang diberikan.

- if dan else

Struktur if digunakan untuk mengevaluasi suatu kondisi. Jika kondisi bernilai true, blok kode di dalam if akan dieksekusi. Jika tidak, blok kode di dalam else (jika ada) akan dieksekusi.

- if dengan Inisialisasi

Go memungkinkan inisialisasi variabel sebelum kondisi if. Variabel ini hanya berlaku di dalam blok if.

- switch

Struktur switch digunakan untuk mengevaluasi nilai dari sebuah ekspresi dan mengeksekusi blok kode yang sesuai dengan case yang cocok.

2. Perulangan (Looping)

Go hanya memiliki satu struktur perulangan, yaitu for tetapi dapat digunakan dalam berbagai bentuk.

- for Loop

Struktur for digunakan untuk mengulang eksekusi blok kode selama kondisi tertentu terpenuhi.

- for sebagai while

Go tidak memiliki kata kunci while, tetapi Anda dapat menggunakan for tanpa inisialisasi dan post-statement untuk meniru perilaku while.

- Range

range digunakan untuk mengiterasi elemen dalam array, slice, string, map, atau channel.

II. GUIDED

Source Code Guided 1:

```
Go contoh1.go 2 X  Go contoh2.go 1  Go contoh3.go 1
Week1 > guided > Go contoh1.go > ...
1 //10311240087
2
3 package main
4
5 import "fmt"
6
7 func main() { main redeclared in this block (see details)
8     var greetings = "selamat datang di dunia DAP"
9     var a, b int
10    fmt.Println(greetings)
11    fmt.Scanln(&a, &b)
12    fmt.Printf("%v + %v = %v\n", a, b, a+b)
13 }
14
```

Output :

```
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\Week1\guided\contoh1.go"
selamat datang di dunia DAP
8 9
8 + 9 = 17
PS D:\Coding manja\semester2> |
```

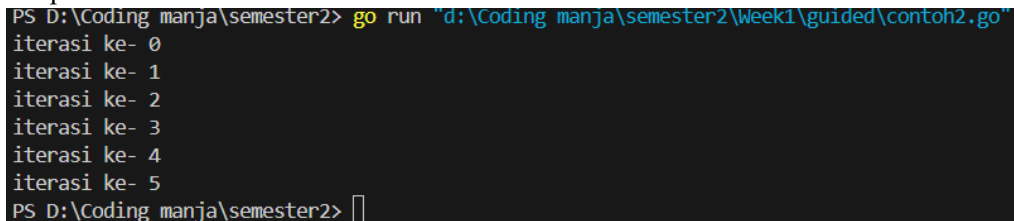
Deskripsi Program: program ini menampilkan pesan selamat datang, kemudian meminta input dua angka dari pengguna, dan output menampilkan hasil penjumlahan dari kedua angka tersebut.

Source Code Guided 2:



```
Week1 > guided > -GO contoh2.go > ...
1 //10311240087
2
3 package main
4
5 import "fmt"
6
7 func main() { main redeclared in this block
8     for i := 0; i <= 5; i++ {
9         fmt.Println("iterasi ke-", i)
10    }
11 }
12
```

Output:



```
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\week1\guided\contoh2.go"
iterasi ke- 0
iterasi ke- 1
iterasi ke- 2
iterasi ke- 3
iterasi ke- 4
iterasi ke- 5
PS D:\Coding manja\semester2> 
```

Deskripsi Program: Program ini menggunakan perulangan for untuk mencetak urutan angka dari 0 hingga 5. Setiap iterasi, program menampilkan teks "iterasi ke-" diikuti angka yang menunjukkan urutan iterasi saat itu. Jadi, program ini menampilkan 6 iterasi yang dimulai dari 0 hingga 5.

Source Code Guided 3:

```
Week1 > guided > -o contoh3.go > ...
1 //10311240087
2
3 package main
4
5 import "fmt"
6
7 func main() { main redeclared in this block
8     nilai := 80
9     pcthadir := 0.75
10    adatubes := true
11
12    var indeks string
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
19        indeks = "C"
20    } else {
21        indeks = "F"
22    }
23    fmt.Printf("nilai %d dengan kehadiran %.2f%% dan buat tubes=%t, mendapatkan indeks %s\n", nilai, pcthadir*100, adatubes, indeks)
24 }
25
```

Output:

```
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\Week1\guided\contoh3.go"
nilai 80 dengan kehadiran 75.00% dan buat tubes=true, mendapatkan indeks A
```

Deskripsi Program: Program ini menentukan indeks kelulusan berdasarkan nilai, kehadiran, dan status ujian ulang. Jika nilai lebih dari 75 dan ada ujian ulang, indeksnya "A". Jika nilai lebih dari 65, indeksnya "B". Jika nilai lebih dari 50 dan kehadirannya lebih dari 70%, indeksnya "C". Selain itu, indeksnya "F".

III. UNGUIDED

Source Code Unguided 1:

```
Week1 > unguided > -o unguided1.go > ...
1 // 103112400087
2 package main
3
4 import "fmt"
5
6 func isLeapYear(tahun int) bool {
7     if tahun%400 == 0 {
8         return true
9     } else if tahun%4 == 0 && tahun%100 != 0 {
10        return true
11    }
12    return false
13 }
14 func main() {
15     var tahun int
16     fmt.Print("masukkan tahun: ")
17     fmt.Scan(&tahun)
18
19     if isLeapYear(tahun) {
20         fmt.Println("kabisat: true")
21     } else {
22         fmt.Println("kabisat: false")
23     }
24 }
25
```

Output:

```
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\Week1\unguided\unguided1.go"
masukkan tahun: 2000
kabisat: true
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\Week1\unguided\unguided1.go"
masukkan tahun: 2018
kabisat: false
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\Week1\unguided\unguided1.go"
masukkan tahun: 2016
kabisat: true
```

Deskripsi Program: Program ini bertujuan untuk menentukan apakah suatu tahun merupakan tahun kabisat atau tidak. Tahun kabisat adalah tahun yang dapat dibagi 4, tetapi tidak boleh dibagi 100, kecuali jika bisa dibagi 400.

Source Code Unguided 2 (2B no 4):

```
// 103112400087

package main

import "fmt"

func main() {    main redeclared in this block
    var k float64
    fmt.Print("Nilai k = ")
    fmt.Scan(&k)

    fk := ((4*k + 2) * (4*k + 2)) / ((4*k + 1) * (4*k + 3))
    fmt.Printf("nilai f(k) = %.10f\n", fk)
}
```

Output:

```
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\Week1\unguided\unguided2.go"
Nilai k = 100
nilai f(k) = 1.0000061880
```

Deskripsi Program: Program ini digunakan untuk menghitung nilai fungsi $f(k)$ berdasarkan rumus matematika yang diberikan. Pengguna diminta untuk memasukkan nilai k , lalu program menghitung dan menampilkan hasilnya dengan akurasi hingga 10 angka desimal.

Source code Unguided 2 (2B no 4B)

```
package main

import "fmt"

func main() {    main redeclared in this block
    var k int
    fmt.Print("Nilai K = ")
    fmt.Scan(&k)

    hasil := 1.0
    for i := 0; i < k; i++ {
        hasil *= float64((4*i+2)*(4*i+2)) / float64((4*i+1)*(4*i+3))
    }


    fmt.Printf("Nilai akar 2 = %.10f\n", hasil)
```

Output:

```
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\Week1\unguided\unguided22.go"
Nilai K = 10
Nilai akar 2 = 1.4054086752
```

Deskripsi Program: Program ini digunakan untuk menghitung perkiraan nilai akar 2 dengan menggunakan deret perkalian tertentu. Pengguna memasukkan nilai k, yang menentukan jumlah iterasi dalam perhitungan. Semakin besar nilai k, semakin akurat hasil perkiraannya. Program ini menampilkan hasil dengan 10 angka desimal, dan hasil akhirnya akan semakin mendekati akar 2 sesuai dengan banyaknya iterasi yang dilakukan.

Source Code unguided 3:

```
Week1 > unguided >  unguided3.go > ...
1  //103112400087 Muhammad Shabrian Fadly
2  package main
3
4  import "fmt"
5
6  func main() {  main redeclared in this block
7      var berat int
8      fmt.Print("berat parsel (gram): ")
9      fmt.Scan(&berat)
10     kg := berat / 1000
11     gram := berat % 1000
12     biaya := kg * 10000
13     if berat > 10000 {
14         gram = 0
15     }
16     if gram >= 500 {
17         biaya += gram * 5
18     } else {
19         biaya += gram * 15
20     }
21
22     fmt.Printf("detail berat: %d kg + %d gr\n", kg, gram)
23     fmt.Printf("total biaya: Rp. %d\n", biaya)
24 }
25
```

Output:

```
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\Week1\unguided\unguided3.go"
berat parsel (gram): 8500
detail berat: 8 kg + 500 gr
total biaya: Rp. 82500
PS D:\Coding manja\semester2> go run "d:\Coding manja\semester2\Week1\unguided\unguided3.go"
berat parsel (gram): 9250
detail berat: 9 kg + 250 gr
total biaya: Rp. 93750
```

Deskripsi Program: Program ini digunakan untuk menghitung biaya pengiriman berdasarkan berat parsel. Pengguna memasukkan berat dalam gram, yang kemudian diubah menjadi kilogram dan gram. Biaya dihitung berdasarkan berat kilogram dan tarif berbeda untuk gram di atas atau di bawah 500 gram. Setelah itu, program menampilkan detail berat dan total biaya pengiriman yang harus dibayar.

IV. KESIMPULAN

Praktikum Modul 2 Algoritma dan Pemrograman 2 memberikan pemahaman mendalam tentang struktur kontrol dalam bahasa pemrograman Go. Melalui berbagai studi kasus, mempelajari cara menggunakan percabangan dan perulangan untuk mengendalikan alur program. Struktur kontrol seperti if, else, switch, dan for membantu membuat kode program menjadi lebih terstruktur, efisien, dan mudah dibaca.

V. REFERENSI

https://go.dev/doc/effective_go

<https://gobyexample.com/>

<https://go.dev/tour/flowcontrol/>