

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



**Oleh:**

**NAMA: ICHYA ULUMIDDIIN**

**NIM: 103112400076**

**KELAS: 12IF-01-A**

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

## I. DASAR TEORI

Dalam bahasa pemrograman Go, struktur kontrol digunakan untuk mengatur alur eksekusi program berdasarkan kondisi tertentu. Struktur kontrol utama dalam Go meliputi pernyataan if, perulangan for, dan pernyataan switch.

- **Pernyataan if**

Pernyataan if digunakan untuk mengeksekusi blok kode jika kondisi yang diberikan bernilai benar. Jika kondisi tersebut tidak terpenuhi, blok kode dalam else (jika ada) akan dieksekusi. Go tidak memerlukan tanda kurung { } untuk blok kode dalam if atau else jika hanya terdapat satu pernyataan.

- **Pernyataan for**

Go memiliki satu jenis perulangan, yaitu for. Perulangan ini dapat digunakan dalam berbagai bentuk:

- Perulangan dengan penghitung: Mirip dengan perulangan for tradisional, di mana inisialisasi, kondisi, dan increment diletakkan dalam satu baris.
- Perulangan tanpa kondisi: Perulangan ini akan berjalan terus menerus hingga kondisi break atau return terpenuhi.
- Perulangan dengan kondisi di awal: Kondisi perulangan diletakkan sebelum blok kode, mirip dengan perulangan while di bahasa lain.

- **Pernyataan Switch**

Pernyataan switch digunakan untuk memilih salah satu blok kode yang akan dieksekusi berdasarkan nilai dari ekspresi yang diberikan. Berbeda dengan bahasa lain, Go tidak memerlukan pernyataan break di akhir setiap blok case, karena Go secara otomatis keluar dari pernyataan switch setelah menjalankan blok yang sesuai.

## II. GUIDED

### Source Code Guided 1

```
//ICHYA ULUMIDDIIN
package main

import "fmt"

func main() {
    for i := 1; i <= 5; i++ {
        fmt.Println("iterasi ke-", i)
    }
}
```

### Output

```
PS C:\Users\ICHYA ULUMIDDIIN\OneDrive\Documents\guided\guided.go"
iterasi ke- 1
iterasi ke- 2
iterasi ke- 3
iterasi ke- 4
iterasi ke- 5
```

### Penjelasan

Tujuan dari program ini adalah untuk menunjukkan cara menggunakan perulangan `for` dalam bahasa Go untuk mencetak urutan teks yang berisi nomor iterasi, dimulai dari 1 hingga 5.

### Source Code Guided 2

```
//ICHYA ULUMIDDIIN
package main

import "fmt"

func main() {
    var greetings = "Selamat Datang di Dunia DAP"
    var a, b int
    fmt.Println(greetings)
    fmt.Scanln(&a, &b)
    fmt.Printf("%v + %v = %v\n", a, b, a+b)
}
```

## Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\guidedM2\guidedM2.go"
Selamat Datang di Dunia DAP
5 5
5 + 5 = 10
```

## Penjelasan

Tujuan dari program ini adalah untuk menerima input dua angka dari pengguna, menghitung jumlahnya, dan menampilkan hasil perhitungan tersebut. Program ini juga mencetak pesan sambutan dan menggunakan fungsi `fmt.Scanln()` untuk mengambil input, serta `fmt.Printf()` untuk menampilkan hasilnya.

## Source Code Guided 3

```
// ICHYA ULUMIDDIIN
package main

import "fmt"

func main() {
    nilai := 80
    pctHadir := 0.75
    adaTubes := true

    var indeks string

    if nilai > 75 && adaTubes {
        indeks = "A"
    } else if nilai > 65 {
        indeks = "B"
    } else if nilai > 50 && pctHadir > 0.7 {
        indeks = "C"
    } else {
        indeks = "F"
    }

    fmt.Printf("Nilai %d dengan kehadiran %.2f%% dan
    buat tubes =%t, mendapat indeks %s\n", nilai,
    pctHadir*100, adaTubes, indeks)
}
```

## Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\Alpro 2> go run "c:\Users\Ic
dom\GuidedM21\GuidedM21.go"
Nilai 80 dengan kehadiran 75.00% dan buat tubes =true, mendapat indeks A
```

## Penjelasan

Tujuan dari program ini adalah untuk menentukan indeks nilai seorang mahasiswa berdasarkan beberapa kondisi, yaitu nilai yang diperoleh, persentase kehadiran, dan apakah mahasiswa tersebut mengikuti ujian ulang (tubes). Program ini menggunakan struktur pengkondisian `if-else` untuk mengevaluasi kondisi-kondisi tersebut. Jika nilai lebih dari 75 dan mahasiswa mengikuti ujian ulang, maka indeks yang diberikan adalah "A". Jika nilai lebih dari 65, indeks yang diberikan adalah "B". Jika nilai lebih dari 50 dan persentase kehadiran lebih dari 70%, maka indeks yang diberikan adalah "C". Jika tidak memenuhi salah satu kondisi tersebut, maka indeks yang diberikan adalah "F". Program ini kemudian menampilkan hasil perhitungan indeks bersama dengan informasi nilai, persentase kehadiran, dan status ujian ulang menggunakan fungsi `fmt.Printf()`.

### III. UNGUIDED

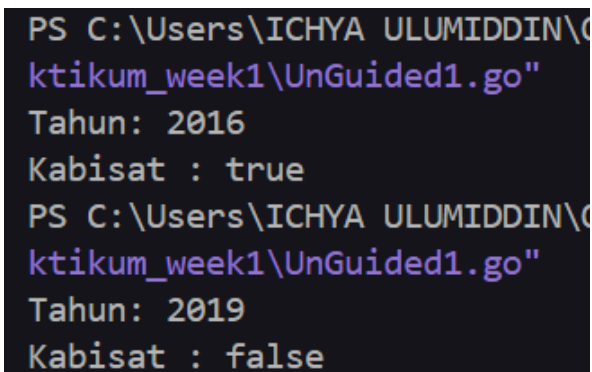
#### Source Code Unguided 1

```
//ICHYA ULUMIDDIIN
package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Tahun: ")
    fmt.Scan(&tahun)
    fmt.Println("Kabisat :", (tahun % 400 == 0)
|| (tahun % 4 == 0 && tahun % 100 != 0))
}
```

#### Output



```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Desktop>
ktikum_week1\UnGuided1.go"
Tahun: 2016
Kabisat : true
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Desktop>
ktikum_week1\UnGuided1.go"
Tahun: 2019
Kabisat : false
```

#### Penjelasan

Tujuan dari program ini adalah untuk menentukan apakah suatu tahun adalah tahun kabisat atau bukan. Program ini meminta input tahun dari pengguna, kemudian memeriksa apakah tahun tersebut memenuhi kondisi sebagai tahun kabisat. Sebuah tahun dianggap kabisat jika dapat dibagi habis oleh 400, atau dapat dibagi habis oleh 4 tetapi tidak dibagi habis oleh 100. Program menggunakan operator modulus (`%`) untuk memeriksa kondisi ini, dan hasilnya dicetak sebagai `true` (jika tahun kabisat) atau `false` (jika bukan tahun kabisat).

### Source Code Unguided 3

```
//ICHYA ULUMIDDIIN
package main

import (
    "fmt"
)

func main() {
    var beratParsel int
    fmt.Print("Berat Parsel (gram): ")
    fmt.Scan(&beratParsel)
    beratKg := beratParsel / 1000
    sisaGr := beratParsel % 1000
    biaya := beratKg * 10000
    if beratKg > 10 {
        sisaGr = 0
    }
    if sisaGr >= 500 {
        biaya += sisaGr * 5
    } else {
        biaya += sisaGr * 15
    }
    fmt.Printf("\nDetail berat : %d kg + %d gr\n",
    beratKg, sisaGr)
    fmt.Printf("Detail Biaya : Rp. %d + Rp. %d\n",
    beratKg*10000, biaya-(beratKg*10000))
    fmt.Printf("Total Biaya : Rp. %d\n", biaya)
}
```

### **Output**

```
PS C:\Users\ICHYA ULUMIDDIIN\OneDrive\Documents\Alp
ktikum_week1\unguided3\UnGuided3.go"
Berat Parsel (gram): 8500

Detail berat : 8 kg + 500 gr
Detail Biaya : Rp. 80000 + Rp. 2500
Total Biaya : Rp. 82500
```

## Penjelasan

Program di atas bertujuan untuk menghitung biaya pengiriman berdasarkan berat parcel dalam gram. Pertama, program meminta input berat parcel dari pengguna. Berat parcel ini kemudian dikonversi menjadi kilogram dan sisa gram dihitung. Biaya pengiriman dihitung dengan tarif Rp 10.000 per kilogram. Jika berat total lebih dari 10 kg, sisa gram akan digratiskan. Untuk sisa gram, biaya tambahan dikenakan sebesar Rp 5 per gram jika lebih dari atau sama dengan 500 gram, dan Rp 15 per gram jika kurang dari 500 gram. Program kemudian mencetak detail berat, biaya per kilogram, biaya tambahan, dan total biaya pengiriman.

## Source Code Unguided 2

```
package main

import "fmt"

func main() {
    var k int
    fmt.Print("Masukkan Nilai K: ")
    fmt.Scan(&k)
    hasil := 1.0
    for i := 0; i <= k; i++ {
        penyebutAtas := (4*float64(i) + 2) *
(4*float64(i) + 2)
        penyebutBawah := (4*float64(i) + 1) *
(4*float64(i) + 3)
        hasil *= penyebutAtas / penyebutBawah
    }
    fmt.Printf("%.10f\n", hasil)
}
```

## Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\Alpro
2\Praktikum_week1\unguided2\UnGuided2.go"
Masukkan Nilai K: 10
1.4062058441
```



### Penjelasan

Kode ini menghitung hasil dari rumus matematika dengan menggunakan perulangan. Pengguna memasukkan nilai `K`, dan program menghitung hasil perkalian dari pecahan yang melibatkan ekspresi  $(4*i + 2)$  di bagian atas dan  $(4*i + 1)$  serta  $(4*i + 3)$  di bagian bawah, untuk setiap nilai `i` dari 0 hingga `K`. Hasil akhirnya ditampilkan dengan format 10 angka desimal.

#### IV. KESIMPULAN

Kesimpulan dari pembahasan di atas adalah bahwa struktur kontrol dalam bahasa pemrograman Go memungkinkan pengaturan alur eksekusi program berdasarkan kondisi tertentu. Tiga struktur kontrol utama yang digunakan adalah:

**1. Pernyataan ``if``:** Digunakan untuk mengeksekusi blok kode berdasarkan kondisi yang diberikan, dengan dukungan opsi ``else`` untuk menangani kondisi yang tidak terpenuhi.

**2. Pernyataan ``for``:** Merupakan satu-satunya jenis perulangan dalam Go yang dapat digunakan dalam berbagai bentuk, baik dengan penghitung, tanpa kondisi, maupun dengan kondisi di awal.

Struktur kontrol ini memungkinkan pengembangan program yang efisien dengan mengatur alur eksekusi berdasarkan kondisi yang ada dalam kode.

## **V. REFERENSI**

Ihsan. (2022, August 22). *Mengenal Struktur Control, Perulangan, Kondisi dan Map Pada Golang*. Retrieved from Santekno:  
<https://www.santekno.com/mengenal-struktur-control-perulangan-kondisi-dan-map-pada-golang/>

Putri, F. (2024, August 1). *Panduan Lengkap Belajar Golang: Dari Pemula Sampai Mahir*. Retrieved from dibimbing:  
<https://dibimbing.id/blog/detail/panduan-lengkap-belajar-golang-dari-pemula-sampai-mahir>