

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 1



DISUSUN OLEH:

RIZKINA AZIZAH

103112400082

S1 IF-12-01

DOSEN:

Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025/2026

DASAR TEORI

Struktur Kontrol

Dalam Go, struktur kontrol hanyalah konstruksi yang mengendalikan aliran eksekusi dalam sebuah program. Struktur kontrol memungkinkan Anda untuk melakukan berbagai tindakan berdasarkan kondisi atau mengeksekusi blok kode secara berulang.

Beberapa struktur kontrol utama dalam Go meliputi:

1. Pernyataan If/Else

Pernyataan [if/else](#) dalam Go mengeksekusi blok kode berdasarkan suatu kondisi. Jika kondisi mengembalikan true, kode di dalam if blok tersebut dieksekusi. Jika kondisi mengembalikan false, else blok (jika ada) dieksekusi.

2. Pernyataan Switch

Pernyataan [switch](#) adalah cabang multi arah yang memungkinkan Anda untuk mengeksekusi blok kode yang berbeda tergantung pada nilai suatu ekspresi. Pernyataan ini lebih mudah dibaca daripada beberapa pernyataan if/else.

3. Untuk Perulangan

Go hanya memiliki satu konstruksi perulangan, yaitu [perulangan for](#). Perulangan ini dapat digunakan dalam berbagai bentuk: perulangan tradisional, perulangan berbasis rentang (untuk mengulang irisan, peta, dan sebagainya), dan perulangan tak terbatas.

4. Fungsi

Fungsi dalam Go adalah blok kode yang menjalankan tugas tertentu. Fungsi membantu Anda mengatur kode dengan memungkinkan Anda menyusun logika kode yang dapat digunakan kembali, yang lebih mudah dikelola dan dipahami.

5. Pointer

Dalam Go, [pointer](#) adalah variabel yang menyimpan alamat memori dari variabel lain. Pointer "menunjuk ke" wilayah dalam memori tempat nilai sebenarnya disimpan, bukan menyimpan nilai itu sendiri.

Pointer berguna saat Anda perlu meneruskan referensi ke struktur besar atau saat Anda ingin mengubah nilai variabel dari dalam suatu fungsi. Pointer juga penting untuk manajemen memori.

6. Goroutine dan Konkurensi

[Goroutine](#) dan konkurensi adalah konsep yang memungkinkan kode Anda secara efisien menjalankan beberapa tugas secara paralel.

Goroutine adalah fungsi yang berjalan bersamaan dengan fungsi lainnya. Goroutine sangat ringan, dengan jejak memori yang kecil, sehingga memungkinkan Anda menjalankan ribuan (atau bahkan jutaan) goroutine secara bersamaan tanpa membebani sumber daya sistem.

Di sisi lain, konkurensi mengacu pada kapasitas program untuk menangani sejumlah tugas pada saat yang bersamaan. Hal ini tidak selalu berarti bahwa tugas-tugas tersebut dijalankan secara bersamaan (yang merupakan paralelisme), tetapi tugas-tugas tersebut berjalan secara independen.

7. Struct

Dalam Go, a struct adalah tipe data komposit yang mengatur variabel (bidang) menjadi satu tipe. Bidang-bidang ini dapat mencakup berbagai tipe data, sehingga struct cocok untuk menggambarkan struktur data yang kompleks. Struct dalam Go berfungsi sama seperti kelas dalam bahasa pemrograman lain tetapi tanpa metode pewarisan.

CONTOH SOAL

1. Latihan1

Source Code:

```
//Rizkina Azizah_10311240082
package main

import "fmt"

func main(){
    var greetings = "Selamat datang di dunia DAB"
    var a, b int
    fmt.Println(greetings)
    fmt.Scan(&a, &b)
    fmt.Printf("%v + %v = %v\n", a, b, a+b)
}
```

Output:

```
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run "c:\Users\HP\Doc
Selamat datang di dunia DAB
4 5
4 + 5 = 9
```

Deskripsi Program:

- Program ini digunakan untuk menjumlahkan dua angka yang dimasukkan oleh pengguna.
 - Mendeklarasikan dua variabel yaitu **a** dan **b** dengan tipe data integer
 - Instruksi Scan digunakan untuk menerima input nilai a dan b
- Menggunakan instruksi Printf untuk menampilkan hasil penjumlahan dari a dan b dengan format: a + b = hasil

2. Latihan2

Source Code:

```
//Rizkina Azizah_10311240082
package main

import "fmt"

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

Output:

```
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run "c
iterasi ke- 1
iterasi ke- 2
iterasi ke- 3
iterasi ke- 4
iterasi ke- 5
```

Deskripsi Program:

- Program ini digunakan untuk menampilkan iterasi ke-n sebanyak 5 kali.
- Menggunakan paradigma perulangan dengan variabel i dengan tipe data integer yang digunakan sebagai penghitung dalam loop
- Intruksi `fmt.Println` akan mencetak output setiap iterasi yang diikuti dengan spasi

3. Latihan3

Source Code:

```
//Rizkina Azizah _ 10311240082
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\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run "c:\Users\HP\D
Nilai 80 dengan kehadiran 75.00 dan buat tubes=true, mendapat indeks A
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> █
```

Deskripsi Program:

- Program ini digunakan untuk menentukan indeks nilai
- Menggunakan paradigma percabangan yaitu if-else :
 - Jika nilai lebih dari 75 dan (adaTubes = true), indeks adalah "A".
 - Jika nilai lebih dari 65, indeks adalah "B".
 - Jika nilai lebih dari 50 dan persentase kehadiran lebih dari 70%, indeks adalah "C".
 - Jika tidak memenuhi syarat-syarat di atas, indeks adalah "F".
- Program akan mencetak nilai, persentase kehadiran, status ujian ulang, dan indeks yang didapatkan

SOAL LATIHAN

1. Latihan 1

Tahun kabisat adalah tahun yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100. Buatlah sebuah program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut merupakan tahun kabisat (true) atau bukan (false). (Contoh input/output, Teks bergaris bawah adalah input dari user):

NO	Contoh Masukan dan Keluaran
1	Tahun: 2016 Kabisat: true
2	Tahun: 2000 Kabisat: true
3	Tahun: 2018 Kabisat: false

Source Code:

```
//Rizkina Azizah_103112400082
package main

import "fmt"

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

Output:

```
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run "c
Tahun :2016
Kabisat :true
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run "c
Tahun :2000
Kabisat :true
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run "c
Tahun :2018
Kabisat :false
```

Deskripsi Program:

- Program ini digunakan untuk menentukan tahun kabisat
- Mendeklarasikan dua variabel yaitu n dengan tipe data integer dan tahun dengan tipe data boolean
- Intruksi Scan digunakan untuk mendeklarasikan input n
- Tahun kabisat jika habis dibagi 4 atau 400
- Menggunakan intruksi Print untuk mencetak output tahun merupakan tahun kabisat atau bukan

2. Latihan 2

Source Code:

```
1 //Rizkina Aziazah_103112400082
2 package main
3
4 import "fmt"
5
6 func main(){
7     var k int
8     var akar2 float64
9     fmt.Print("k: ")
10    fmt.Scan(&k)
11    akar2 = 1.0
12    for i := 0; i <= k; i++ {
13        akar2 *= (float64(4*i + 2)) * (float64(4*i + 2)) / (float64(4*i + 1) * float64(4*i + 3))
14    }
15    fmt.Printf("akar2: %.10f\n", akar2)
16 }
17 }
```

Output:

```
k: 10
akar2: 1.4062058441
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run "c:\Use
k: 100
akar2: 1.4133387072
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run "c:\Use
k: 1000
akar2: 1.4141252651
```

Deskripsi Program:

- Program ini digunakan untuk menghitung nilai akar 2
- Menggunakan 2 variabel yaitu k dengan tipe data integer dan akar2 dengan tipe data real
- Intruksi fmt.Scan digunakan untuk mendeklarasikan input
- Menggunakan paradigma perulangan
- Intruksi fmt.Printf digunakan untuk menghasilkan output dari akar2
- %.10f berguna agar angka dibelakang koma ada 10 angka

3. Latihan 3

PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program BiayaPos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut! Dari berat parcel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

1	Contoh #1 Berat parcel (gram): 8500 Detail berat: 8 kg + 500 gr Detail biaya: Rp. 80000 + Rp. 2500 Total biaya: Rp. 82500
2	Contoh #2 Berat parcel (gram): 9250 Detail berat: 9 kg + 250 gr Detail biaya: Rp. 90000 + Rp. 3750 Total biaya: Rp. 93750
3	Contoh #3 Berat parcel (gram): 11750 Detail berat: 11 kg + 750 gr Detail biaya: Rp. 110000 + Rp. 3750 Total biaya: Rp. 110000

Source Code:

```
//Rizkina Azizah_10312400082
package main
import "fmt"

func main(){
    var bp, tambah int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&bp)
    kg := bp/1000
    sisa_gr := bp%1000
    kirim := kg * 10000
    if kg > 10 {
        tambah = 0
    }else if sisa_gr >= 500 {
        tambah = sisa_gr * 5
    }else if sisa_gr < 500 {
        tambah = sisa_gr * 15
    }
    total := kirim + tambah
    fmt.Printf("Detail Berat: %d kg + %d gr\n ", kg, sisa_gr)
    fmt.Printf("Detail Biaya: Rp. %d kg + Rp. %d\n", kirim, tambah)
    fmt.Printf("Total biaya: Rp. %d\n ", total)
}
```

Output:

```
PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
Berat parsel (gram): 8500
Detail Berat: 8 kg + 500 gr
Detail Biaya: Rp. 80000 kg + Rp. 2500
Total biaya: Rp. 82500

PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
Berat parsel (gram): 9250
Detail Berat: 9 kg + 250 gr
Detail Biaya: Rp. 90000 kg + Rp. 3750
Total biaya: Rp. 93750

PS C:\Users\HP\Documents\ittp\TUP smst 2\alpro smt 2> go run
Berat parsel (gram): 11750
Detail Berat: 11 kg + 750 gr
Detail Biaya: Rp. 110000 kg + Rp. 0
Total biaya: Rp. 110000
```

Deskripsi Program:

- Program ini digunakan untuk menghitung biaya pengiriman
- Mendeklarasikan dua variabel yaitu bp dan tambah dengan tipe data integer

- Intruksi Scan digunakan untuk mendeklarasikan input bp
- Menggunakan paradigma percabangan yaitu if-else :
 - If kondisi atau syarat = jika berat kilo lebih dari 10 then ({}) sebagai aksi = maka tambah = 0
 - apabila syarat pertama tidak terpenuhi akan menjalankan syarat dan aksi lain atau disebut dengan else if = jika sisa gram lebih dari 500 aksi lain = maka tambah = sisa gram * 5
 - apabila syarat pertama dan kedua tidak terpenuhi maka akan menjalankan aksi lain = maka tambah = sisa gram * 15 •
- Menggunakan intruksi Printf sebagai deklarasi output agar otomatis tertulis kg ataupun gr pada hasil keluaran

DAFTAR PUSTAKA

TomitopOyedele(2024), Key Golang Concepts You Should Learn as a Beginner Go Developer
(<https://www.freecodecamp.org/news/key-golang-concepts-for-beginner-go-devs/> ,2024)