

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



**DISUSUN OLEH:
Keishin Naufa Alfaridzhi
103112400061
S1 IF-12-01**

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. DASAR TEORI

A. Bahasa Yang Digunakan

Pada praktikum ini bahasa pemrograman yang digunakan adalah bahasa pemrograman Go, sesuai dengan modul yang menjadi acuan praktikum. Golang (atau) Go adalah bahasa pemrograman baru, yang mulai dilirik oleh para developer karena kelebihan-kelebihan yang dimilikinya. Sudah banyak Perusahaan besar yang menggunakan bahasa ini untuk produk-produk mereka hingga di level production.

B. Komentar

Komentar biasa dimanfaatkan untuk menyisipkan catatan pada kode program, menulis penjelasan atau deskripsi mengenai suatu blok kode, atau bisa juga digunakan untuk me-remark kode (men-non-aktifkan kode yang tidak digunakan). Komentar akan diabaikan Ketika kompilasi maupun eksekusi program.

Ada 2 jenis komentar di Golang, yaitu inline dan multiline.

1. Komentar Inline

Penulisan komentar jenis ini diawali dengan tanda *double slash* (*//*) lalu diikuti pesan komentarnya. Komentar inline hanya berlaku untuk satu baris pesan saja. Jika pesan komentar lebih dari satu baris, maka tanda *double slash* harus ditulis lagi di baris selanjutnya.

2. Komentar Multiline

Komentar yang cukup panjang akan lebih rapi jika ditulis menggunakan teknik komentar multiline. Ciri dari komentar jenis ini adalah penulisannya diawali dengan tanda *(/**) dan diakhiri *(*/)*.

C. Variabel

Golang mengadopsi 2 jenis penulisan variabel, yang dituliskan tipe data-nya dan yang tidak. Kedua cara tersebut intinya adalah sama, pembedanya hanyalah cara penulisannya saja. Untuk penulisan variabel dengan tipe data, keyword *var* digunakan untuk deklarasi variabel kemudian diakhiri dengan tipe data misalnya *string*. Kemudian untuk penulisan variabel tanpa tipe data, variabel dideklarasikan dengan menggunakan metode type inference. Penandanya tipe data tidak dituliskan pada saat deklarasi. Pada penggunaan metode ini, operand (*=*) harus diganti dengan (*:=*) dan keyword *var* dihilangkan.

Golang memiliki aturan unik yang tidak dimiliki bahasa lain, yaitu tidak boleh ada satupun variabel yang menganggur. Artinya, semua variabel yang dideklarasikan harus digunakan. Jika terdapat variabel yang tidak digunakan tapi dideklarasikan, program akan gagal dikompilasi. Untuk mengatasi itu, Golang memiliki variabel yaitu underscore. Underscore (`_`) adalah predefined variabel yang bisa dimanfaatkan untuk menampung nilai yang tidak dipakai.

D. Tipe Data

Golang mengenal beberapa jenis tipe data, diantaranya adalah tipe data numerik (decimal dan non-desimal), string, dan boolean.

1. Tipe Data Numerik Non-Desimal (uint, int)
2. Tipe Data Numerik Desimal (float64, float32)
3. Tipe Data Bool (true, false)
4. Tipe Data String (string, “ ”)

E. Operator Aritmatika

Operator aritmatika merupakan operator yang digunakan untuk operasi yang sifatnya perhitungan. Golang mendukung beberapa operator aritmatika standar, yaitu:

1. Penjumlahan (+)
2. Pengurangan (-)
3. Perkalian (*)
4. Pembagian (/)
5. Modulus atau sisa hasil pembagian (%)

F. Seleksi Kondisi

Seleksi kondisi pada program berguna untuk mengontrol sebuah blok kode yang akan dieksekusi. Yang dijadikan acuan oleh seleksi kondisi adalah nilai bertipe bool, bisa berasal dari variabel, ataupun hasil operasi perbandingan. Nilai tersebut menentukan blok kode mana yang akan dieksekusi. Go memiliki 2 macam keyword untuk seleksi kondisi, yaitu if else dan switch.

1. If Expression

If adalah salah satu kata kunci yang digunakan dalam percabangan. Percabangan artinya kita bisa mengeksekusi kode program tertentu ketika suatu kondisi terpenuhi. Hampir semua bahasa pemrograman mendukung if expression.

2. Else if expression

Terkadang kita butuh membuat beberapa kondisi. Kasus seperti ini dapat menggunakan else if expression. If mendukung short statement sebelum kondisi.

Hal ini sangat cocok untuk membuat statement yang sederhana sebelum melakukan pengecekan terhadap kondisi.

3. Switch-Case

Switch merupakan seleksi kondisi yang sifatnya fokus pada satu variabel, lalu kemudian di-cek nilainya. Contoh sederhananya seperti penentuan apakah nilai variabel x adalah: 1, 2, 3, atau lainnya. Perlu diketahui, switch pada pemrograman Go memiliki perbedaan dibanding bahasa lain. Di Go, ketika sebuah case terpenuhi, tidak akan dilanjutkan ke pengecekan case selanjutnya, meskipun tidak ada keyword “break” di situ. Konsep ini berkebalikan dengan switch pada umumnya pemrograman lain (yang ketika sebuah case terpenuhi, maka akan tetap dilanjutkan mengecek case selanjutnya kecuali ada keyword “break”).

G. Perulangan

Perulangan merupakan proses mengulang dan mengeksekusi blok kode tanpa henti sesuai dengan kondisi yang dijadikan acuan. Biasanya disiapkan variabel untuk iterasi atau penanda kapan perulangan akan dihentikan.

a. For Loop

For loop merupakan statement perulangan dasar dan cukup sering ditemui. Format for loop yaitu sebagai berikut.

- *Init Statement*: bagian ini akan dieksekusi sebelum perulangan dimulai. Biasanya diisi dengan mendeklarasi variabel iterasi.
- *Condition Expression*: bagian ini akan dicek dan dieksekusi setiap perulangan yang dilakukan, jika true maka perulangan akan terus berjalan hingga kondisi bernilai false.
- *Post Statement*: statement ini akan dieksekusi pada akhir iterasi. Jika terdapat range, maka perulangan akan dieksekusi untuk setiap item pada range.

b. While Loop

While loop merupakan perulangan yang akan terus berjalan hingga suatu kondisi terpenuhi. Penulisan while loop adalah dengan menuliskan kondisi setelah keyword for (hanya kondisi). Deklarasi dan iterasi variabel counter tidak dituliskan setelah keyword, hanya kondisi perulangan saja. Konsepnya mirip seperti while milik bahasa pemrograman lain.

c. Repeat Until

Untuk Repeat Until ini mirip seperti for loop biasa namun hanya menggunakan inisiasi dan kondisi saja.

II. GUIDED

1. Source Code:

```
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:

```
D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul2\coso\guided.go"
Selamat datang di dunia DAP
1 2
1 + 2 = 3
```

Penjelasan:

Program sederhana untuk mencetak output “Selamat datang di dunia DAP” dan melakukan operasi aritmatika sederhana yaitu pertambahan.

2. Source Code:

```
package main
import "fmt"

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

Output:

```
D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul2\coso\guided2.go"
iterasi ke- 1
iterasi ke- 2
iterasi ke- 3
iterasi ke- 4
iterasi ke- 5
```

Penjelasan:

Program sederhana untuk mencatat berapa kali perulangan diulang.

3. Source Code:

```
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:

```

D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul2\coso\guided3.go"
Nilai 80 dengan kehadiran 75.00% dan buat tubes = true, mendapat indeks A

```

Penjelasan:

Program indeks nilai dan kehadiran dengan kriteria sebagai berikut: Jika nilai lebih dari 75 & adaTubes = true maka indeks “A”; Jika nilai lebih dari 65 maka indeks “B”; Jika nilai lebih dari 50 dan persentase kehadiran lebih dari 70% maka indeks “C”; Selain kriteria sebelumnya maka indeks “F”.

III. UNGUIDED

1. Latihan no. 2A2

Source Code:

```

package main
import "fmt"

func main() {
    var tahun int
    var kabisat bool
    fmt.Print("Masukkan tahun : ")
    fmt.Scanln(&tahun)
    if tahun%4 == 0 && (tahun%100 != 0 || tahun%400 == 0) {
        kabisat = true
    } else {
        kabisat = false
    }
    fmt.Println("Tahun: ", tahun)
}

```

```
    fmt.Println("Kabisat: ", kabisat)
}
```

Output:

```
D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul2\latso\2A2\main.go"
Masukkan tahun : 2016
Tahun: 2016
Kabisat: true

D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul2\latso\2A2\main.go"
Masukkan tahun : 2000
Tahun: 2000
Kabisat: true

D:\Tugas\SEM2\Alpro\Praktikum>go run "d:\Tugas\SEM2\Alpro\Praktikum\modul2\latso\2A2\main.go"
Masukkan tahun : 2018
Tahun: 2018
Kabisat: false
```

Deskripsi Program:

Menentukan nilai Boolean dari tahun yang diinput. Dengan melakukan pengecekan dengan menggunakan if, else statement. Menggunakan 2 cek, Pengecekan pertama dengan tahun dimodulo dengan 4 kemudian melakukan cek nilai, jika sama dengan 0 maka true DAN lanjut cek tahun dimodulo dengan 100, jika tidak sama dengan 0 maka true atau cek tahun dimodulo dengan 400, jika sama dengan 0 maka true. Jika 2 cek tersebut tidak terpenuhi, maka kabisat bernilai false.

2. Latihan no. 2B4

Source Code:

```
package main
import "fmt"

func main() {
    var (
        K, toK int
        sqrt2 float64 = 1.0
    )
    fmt.Print("Nilai K = ")
    fmt.Scan(&toK)

    for K = 0; K <= toK; K++ {
        pembilang := float64((4*K + 2) * (4*K + 2))
        penyebut := float64((4*K + 1) * (4*K + 3))
        sqrt2 *= pembilang / penyebut
    }
    fmt.Printf("Nilai akar 2 = %.10f\n", sqrt2)
}
```


Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\latso\enam\enam.go"
Nilai K = 10
Nilai akar 2 = 1.4062058441

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\latso\enam\enam.go"
Nilai K = 100
Nilai akar 2 = 1.4133387072

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\latso\enam\enam.go"
Nilai K = 1000
Nilai akar 2 = 1.4141252651
```

Deskripsi Program:

Program mencari nilai $\sqrt{2}$ dari variabel K. Menggunakan 2 variabel integer K untuk perkalian berulang dan toK untuk batas perkalian, lalu juga menggunakan variabel bertipe data float64 yaitu sqrt2 dan didefinisikan value nya 1. Nilai K akan terus bertambah 1 dan program akan terus melakukan perkalian dari variabel sqrt2 hingga K mencapai nilai dari variabel toK. Input nilai untuk melakukan perkalian berulang ke dalam toK. Dalam perulangan dilakukan perkalian sqrt2 dengan rumus

$$\prod_{k=0}^{\infty} \frac{(4K+2)^2}{(4K+1)(4K+3)}.$$

3. Latihan no. 2C1

Source Code:

```
package main
import "fmt"

func main() {
    var beratParsel, kg, gr, biayaKg, biayaGr, totalBiaya int

    fmt.Scan(&beratParsel)
    kg = beratParsel / 1000
    gr = beratParsel % 1000

    if gr >= 500 {
        biayaGr = gr * 5
    } else {
        biayaGr = gr * 15
    }

    biayaKg = kg * 10000
    if kg > 10 {
        totalBiaya = biayaKg
    } else {
        totalBiaya = biayaKg + biayaGr
    }
}
```

```

    }

    fmt.Printf("Berat parcel (gram): %v \nDetail berat: %v kg + %v gr \nDetail biaya:
Rp. %v + Rp. %v \nTotal biaya: Rp. %v", beratParsel, kg, gr, biayaKg, biayaGr,
totalBiaya)
}

```

Output:

```

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\satu\satu.go"
8500
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\satu\satu.go"
9250
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\satu\satu.go"
11750
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 3750
Total biaya: Rp. 110000

```

Deskripsi Program:

Program untuk menghitung biaya jasa kirim parcel dengan satuan berat gram. Alurnya adalah pertama input berat parcel dalam gram. Masuk logic pertama kita perlu mendefinisikan variable kg (karena akan digunakan untuk biaya per-kg) dan variable gr (untuk mengambil sisa gram) kemudian melakukan cek jika kondisi gr lebih dari 500 maka $biayaGr = gr * 5$, jika kondisi gr kurang dari 500 maka $biayaGr = gr * 15$. Kedua, akan mendefinisi biayaKg untuk mencari biaya kirim per-kg. Kemudian cek jika paket lebih dari 10kg maka tidak ada tambahan biaya sisa gram parcel. Lalu output detail berat, detail biaya, dan total biaya.

IV. KESIMPULAN

Pada praktikum ini telah dibahas perihal beberapa struktur kontrol yang ada pada bahasa pemrograman Go diantaranya adalah percabangan If-Else dan juga perulangan For loop.

V. DAFTAR PUSTAKA

Noval Agung Prayogo. *Dasar Pemrograman Golang*. Diakses pada 01 Oktober 2024.
<https://dasarpemrogramangolang.novalagung.com>

Annisa Nur Isnaeni. *Golang — Seleksi Kondisi*. Diakses pada 01 Oktober 2024.
<https://medium.com/@annisaisna/golang-seleksi-kondisi-f988ead004b4>

Parvez Alam, *Golang for loop example | Golang Loops Tutorial – Phpflow.com*
<https://medium.com/@parvez1487/golang-for-loop-example-golang-loops-tutorial-phpflow-com-f4b2b0e57944>