

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 14
“KOMPOSISI”



DISUSUN OLEH:
Keishin Naufa Alfaridzhi
103112400061
S1 IF-12-01
DOSEN:
Yohani Setiya Rafika Nur, M. Kom.

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

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.

CONTOH SOAL

1. Buatlah program yang digunakan untuk menampilkan sejumlah bilangan ganjil.

Masukan terdiri dari sebuah bilangan bulat.

Keluaran berupa barisan bilangan ganjil dari 1 hingga bilangan bulat yang diberikan pada masukan.

Source Code:

```
package main
import "fmt"

func main() {
    var bil, j int
    fmt.Scan(&bil)
    for j = 1; j <= bil; j++ {
        if j%2 != 0 {
            fmt.Print(j, " ")
        }
    }
}
```

Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\satu\tempCodeRunnerFile.go"
3
1 3
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\satu\tempCodeRunnerFile.go"
2
1
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\satu\tempCodeRunnerFile.go"
7
1 3 5 7
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\satu\tempCodeRunnerFile.go"
10
1 3 5 7 9
```

2. Buatlah program Go yang digunakan untuk mencari nilai terbesar dan terkecil antara tiga bilangan yang diberikan.

Masukan terdiri dari 3 bilangan bulat.

Keluaran terdiri dari dua bilangan yang menyatakan bilangan terbesar dan terkecil dari tiga bilangan yang diberikan.

Source Code:

```
package main
import "fmt"

func main() {
    var b1, b2, b3, max, min int
    fmt.Scan(&b1, &b2, &b3)
    if b1 > b2 {
        max = b1
    }
```

```

        min = b2
    } else {
        max = b2
        min = b1
    }
    if max < b3 {
        max = b3
    }
    if min > b3 {
        min = b3
    }
    fmt.Println("Terbesar", max)
    fmt.Println("Terkecil", min)
}

```

Output:

```

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\dua\dua.go"
1 2 3
Terbesar 3
Terkecil 1

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\dua\dua.go"
5 5 5
Terbesar 5
Terkecil 5

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\dua\dua.go"
12 31 -43
Terbesar 31
Terkecil -43

```

3. Buatlah program yang digunakan untuk menampilkan faktor bilangan dari suatu nilai. Faktor bilangan dari x adalah himpunan bilangan yang habis membagi bilangan x. **Masukan** terdiri dari suatu bilangan bulat positif. **Keluaran** berupa baris bilangan yang menyatakan semua faktor bilangan dari bilangan yang diberikan pada masukan. **Source Code:**

```

package main
import "fmt"

func main() {
    var bil, j int
    fmt.Scan(&bil)
    for j = 1; j <= bil; j++ {
        if bil%j == 0 {
            fmt.Print(j, " ")
        }
    }
}

```

Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\tiga\tiga.go"
5
1 5
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\tiga\tiga.go"
12
1 2 3 4 6 12
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\tiga\tiga.go"
20
1 2 4 5 10 20
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\coso\tiga\tiga.go"
27
1 3 9 27
D:\Tugas\Alpro\Praktikum>
```

LATIHAN SOAL

1. Latihan no. 3

Source Code:

```
package main
import "fmt"

func main() {
    const correctWarna1 = "merah"
    const correctWarna2 = "kuning"
    const correctWarna3 = "hijau"
    const correctWarna4 = "ungu"

    var (
        warna1, warna2, warna3, warna4 string
        total          int
        isTrue          bool
    )

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)
        fmt.Scan(&warna1)
        fmt.Scan(&warna2)
        fmt.Scan(&warna3)
        fmt.Scan(&warna4)
        if warna1 == correctWarna1 && warna2 == correctWarna2 && warna3 ==
correctWarna3 && warna4 == correctWarna4 {
            total = total + 1
        }
    }
    isTrue = total == 5

    fmt.Println("BERHASIL: ", isTrue)
}
```


Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\latso\tiga\tiga.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\latso\tiga\tiga.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
```

Deskripsi Program:

Program yang menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. True bila urutan warna benar. Menyimpan value dari warna ke dalam konstanta / const dengan nama correctWarna1 hingga correctWarna4. Kemudian menggunakan variable dengan nama warna1 hingga warna4, variabel total untuk mencatat berapa kali urutan warna benar, dan variabel isTrue untuk cek boolean. Dalam perulangan input warna1 hingga warna4 secara berurutan, jika warna1 hingga warna4 sama dengan correctWarna1 hingga correctWarna4 secara berurutan maka total +1, Setelah itu cek dengan isTrue apakah total = 5, jika iya maka true.

2. Latihan no. 4

Source Code:

```
package main
import "fmt"

func main() {
    var input, allInput string
    var n int
    for {
        fmt.Printf("Bunga %d: ", n+1)
        fmt.Scan(&input)
        if input == "SELESAI" {
            break
        }
        allInput += input + " - "
        n++
    }
    fmt.Println("Pita:", allInput)
    fmt.Println("Bunga:", n)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul14\latso\empat\empat.go"
Bunga 1: Kertas
Bunga 2: Mawar
Bunga 3: Tulip
Bunga 4: SELESAI
Pita: Kertas - Mawar - Tulip -
Bunga: 3
```

Deskripsi Program:

Program akan meminta input berupa nama bunga secara berulang, proses input akan berhenti apabila user mengetikkan 'SELESAI'. Menggunakan 2 variabel string yaitu input untuk input, dan allInput untuk menyimpan nilai yang telah di-input. Lalu 1 variabel integer yaitu n untuk mencatat berapa bunga. Pertama dalam perulangan menginputkan bunga, setelah itu cek apakah input adalah "SELESAI", jika iya maka break (perulangan berhenti) dan jika bukan maka perulangan akan terus berlanjut. Lalu memasukkan nilai input ke dalam allInput dengan menggunakan operator "+".

3. Latihan no.6

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

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

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>