

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 4**  
**“TIPE DATA & VARIABEL”**



**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.

## CONTOH SOAL

1. Sebuah program digunakan untuk mengkonversi detik ke jam, menit dan detik.

**Masukan** terdiri dari suatu bilangan bulat positif yang menyatakan waktu dalam satuan detik.

**Keluaran** terdiri dari suatu teks yang menyatakan jam, menit dan detik.

### Source Code:

```
package main

import "fmt"

func main() {
    var detik, jam, menit int
    fmt.Scan(&detik)
    jam = detik / 3600
    menit = (detik % 3600) / 60
    detik = detik % 60
    fmt.Println(jam, "jam", menit, "menit", detik, "detik")
}
```

### Output:

```
D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\coso1\coso1.go"
3661
1 jam 1 menit 1 detik

D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\coso1\coso1.go"
7322
2 jam 2 menit 2 detik

D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\coso1\coso1.go"
3600
1 jam 0 menit 0 detik
```

2. Sebuah program digunakan untuk menentukan apakah setiap digit pada suatu bilangan terurut membesar atau tidak.

**Masukan** terdiri dari suatu bilangan bulat positif dengan jumlah digit adalah 3 (atau antara 100 sampai dengan 999)

**Keluaran** adalah boolean yang menyatakan bilangan terurut membesar atau tidak.

### Source Code:

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3 int
    fmt.Scan(&bilangan)
    d1 = bilangan / 100
    d2 = bilangan % 100 / 10
```

```

    d3 = bilangan % 10
    fmt.Println(d1 <= d2 && d2 <= d3)
}

```

Output:

```

D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\coso2\coso2.go"
362
false

D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\coso2\coso2.go"
256
true

D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\coso2\coso2.go"
189
true

```

- Sebuah program digunakan untuk menghitung BMI atau Body Mass Indeks, yang mana BMI merupakan hasil bagi dari berat badan dengan kuadrat dari tinggi badan. **Masukan** terdiri dari dua bilangan riil yang menyatakan berat (kg) dan tinggi badan (m).

**Keluaran** berupa nilai BMI berdasarkan berat dan tinggi badan yang diberikan.

**Source Code:**

```

package main

import "fmt"

func main() {
    var beratBadan, tinggiBadan, bmi float64
    fmt.Print("Masukkan Berat Badan (kg): ")
    fmt.Scan(&beratBadan)
    fmt.Print("Masukkan Tinggi Badan (m): ")
    fmt.Scan(&tinggiBadan)
    bmi = beratBadan / (tinggiBadan * tinggiBadan)
    fmt.Printf("BMI anda: %.2f", bmi)
}

```

Output:

```

D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\coso3\tempCodeRunnerFile.go"
Masukkan Berat Badan (kg): 70
Masukkan Tinggi Badan (m): 1.75
BMI anda: 22.86
D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\coso3\tempCodeRunnerFile.go"
Masukkan Berat Badan (kg): 60
Masukkan Tinggi Badan (m): 1.6
BMI anda: 23.44
D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\coso3\tempCodeRunnerFile.go"
Masukkan Berat Badan (kg): 80
Masukkan Tinggi Badan (m): 1.8
BMI anda: 24.69

```

## LATIHAN SOAL

### 1. Latihan1

Source Code:

```
package main

import "fmt"

func main() {
    var (
        totalbelanja, diskon      int
        hargasetelahdiskon, hargadiskon float64
    )
    fmt.Print("Masukkan total belanja: ")
    fmt.Scan(&totalbelanja)
    fmt.Print("Masukkan diskon: ")
    fmt.Scan(&diskon)
    hargadiskon = float64(totalbelanja) * (float64(diskon) / 100)
    hargasetelahdiskon = float64(totalbelanja) - hargadiskon
    fmt.Printf("%.0f", hargasetelahdiskon)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\latso\latso1\latso1.go"
Masukkan total belanja: 100000
Masukkan diskon: 10
90000
D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\latso\latso1\latso1.go"
Masukkan total belanja: 200000
Masukkan diskon: 20
160000
D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\latso\latso1\latso1.go"
Masukkan total belanja: 150000
Masukkan diskon: 15
127500
```

Deskripsi Program:

Program kasir untuk mendapatkan harga diskon, dari total belanja kemudian dikalikan dengan diskon. Kemudian total belanja akan dikurangi oleh harga diskon. Melakukan casting float64 pada rumus hargadiskon dikarenakan input awal merupakan integer, untuk melakukan perkalian bilangan riil maka perlu dilakukan konversi. Kemudian terdapat format “%.0f” pada fmt.Printf() yaitu untuk membulatkan hasil.

### 2. Latihan2

Source Code:

```
package main

import "fmt"
```

```
func main() {
    var beratBadan, tinggiBadan, bmi float64
    fmt.Print("Masukkan skor BMI: ")
    fmt.Scan(&bmi)
    fmt.Print("Masukkan Tinggi Badan (m): ")
    fmt.Scan(&tinggiBadan)
    beratBadan = bmi * (tinggiBadan * tinggiBadan)
    fmt.Printf("Berat Badan anda adalah: %.0f", beratBadan)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\latso\latso3\latso3.go"
Masukkan skor BMI: 22.85
Masukkan Tinggi Badan (m): 1.75
Berat Badan anda adalah: 70
D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\latso\latso3\latso3.go"
Masukkan skor BMI: 23.43
Masukkan Tinggi Badan (m): 1.6
Berat Badan anda adalah: 60
D:\Tugas\Alpro\Praktikum\Laprak3>go run "d:\Tugas\Alpro\Praktikum\Laprak3\modul4\latso\latso3\latso3.go"
Masukkan skor BMI: 24.69
Masukkan Tinggi Badan (m): 1.8
Berat Badan anda adalah: 80
```

Deskripsi Program:

Program untuk mencari berat badan dari bmi dan tinggi badan. Jika rumus mencari BMI adalah Berat badan / (tinggi\*tinggi), maka rumus mencari berat badan adalah Berat Badan = BMI \* (tinggi\*tinggi). Kemudian terdapat format “%.0f” pada `fmt.Printf()` yaitu untuk membulatkan hasil.

### 3. Latihan3

Source Code:

```
package main

import (
    "fmt"
    "math"
)

type koorA struct {
    xA, yA float64
}
type koorB struct {
    xB, yB float64
}
type koorC struct {
    xC, yC float64
}

func main() {
```



```

var (
    AB, BC, CA, panjangsisisegitiga float64
    A          koorA
    B          koorB
    C          koorC
)

fmt.Print("Masukkan koordinat A: ")
fmt.Scanln(&A.xA, &A.yA)
fmt.Print("Masukkan koordinat B: ")
fmt.Scanln(&B.xB, &B.yB)
fmt.Print("Masukkan koordinat C: ")
fmt.Scanln(&C.xC, &C.yC)
AB = math.Sqrt(math.Pow(B.xB-A.xA, 2) + math.Pow(B.yB-A.yA, 2))
BC = math.Sqrt(math.Pow(C.xC-B.xB, 2) + math.Pow(C.yC-B.yB, 2))
CA = math.Sqrt(math.Pow(A.xA-C.xC, 2) + math.Pow(A.yA-C.yC, 2))

if AB >= BC && AB >= CA {
    panjangsisisegitiga = AB
} else if BC >= AB && BC >= CA {
    panjangsisisegitiga = BC
} else {
    panjangsisisegitiga = CA
}
fmt.Printf("%.2f", panjangsisisegitiga)
// println(AB, BC, CA, panjangsisisegitiga)
}

```

Output:

```

D:\Tugas\Alpro\Praktikum\Laprak3>go run
Masukkan koordinat A: 1.0 1.0
Masukkan koordinat B: 4.0 1.0
Masukkan koordinat C: 1.0 5.0
5.00
D:\Tugas\Alpro\Praktikum\Laprak3>go run
D:\Tugas\Alpro\Praktikum\Laprak3>go run
Masukkan koordinat A: 0.0 0.0
Masukkan koordinat B: 3.0 0.0
Masukkan koordinat C: 3.0 4.0
5.00
D:\Tugas\Alpro\Praktikum\Laprak3>

```

Deskripsi Program:

Program menentukan sisi terpanjang segitiga pada koordinat kartesius. Dapat menggunakan rumus teorema Pythagoras yaitu contoh:

$$AB = \sqrt{(Bx - Ax)^2 + (By - Ay)^2}.$$

Setelah menggunakan rumus diatas, akan dilakukan pengecekan untuk memasukkan sisi terpanjang ke dalam variabel panjangsisisegitiga. Kemudian terdapat format “%.2f” pada fmt.Printf() yaitu untuk membulatkan 2 angka dibelakang koma.

## 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>