

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 4
“I/O, TIPE DATA & VARIABEL”



DISUSUN OLEH:
RAIHAN ADI ARBA
103112400071
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

I/O (Input/Output) merupakan konsep penting dalam pemrograman yang mengatur bagaimana data masuk dan keluar dari program. Dalam Golang, I/O dilakukan melalui paket seperti `fmt` untuk operasi dasar seperti membaca input dari pengguna dan menampilkan output ke layar.

Tipe data dalam Golang menentukan jenis data yang dapat disimpan dan diolah oleh program. Seperti kebanyakan bahasa pemrograman, Golang memiliki dua kategori utama tipe data:

1. **Tipe Data Primitif:** Ini adalah tipe dasar yang langsung disediakan oleh bahasa tanpa perlu definisi tambahan. Golang menyediakan tipe data primitif seperti:
 - **Numerik:** Termasuk integer (`int`, `int8`, `int16`, `int32`, `int64`), unsigned integer (`uint`, `uint8`, `uint16`, `uint32`, `uint64`), dan floating point (`float32`, `float64`).
 - **Boolean:** Merepresentasikan nilai kebenaran (`true` atau `false`).
 - **String:** Untuk menyimpan teks, yang diwakili oleh kumpulan karakter.
2. **Tipe Data Non-Primitif atau Tipe Data Referensi:** Tipe data ini lebih kompleks dan dibuat sesuai dengan kebutuhan pengembang. Beberapa tipe data non-primitif di Golang adalah:
 - **Array:** Kumpulan data tipe sama yang disimpan dalam urutan tertentu.
 - **Slice:** Versi lebih fleksibel dari array yang ukurannya bisa berubah.
 - **Map:** Koleksi pasangan kunci-nilai dinamis.
 - **Struct:** Kumpulan berbagai tipe data yang didefinisikan sebagai satu unit.

Variabel adalah entitas yang digunakan untuk menyimpan data dalam program. Di Golang, variabel dideklarasikan dengan menyebutkan tipe data, diikuti dengan nama variabel. Contoh deklarasi variabel di Golang adalah `var angka int`, yang menunjukkan bahwa variabel `angka` akan menyimpan data bertipe integer.

Konstanta di Golang adalah variabel yang nilainya tetap yang tidak berubah selama waktu eksekusi program. Konstanta dideklarasikan menggunakan kata kunci `const`, dan nilai dari konstanta ini harus ditetapkan pada saat deklarasi. Penggunaan konstanta di Golang mirip dengan penggunaannya di bahasa pemrograman lain, membantu dalam menjaga integritas dan keamanan nilai-nilai yang tidak boleh berubah selama eksekusi program. Contoh deklarasi konstanta adalah `const Pi = 3.14159`.

Dengan memahami I/O, tipe data, variabel, dan konstanta, pemrogram dapat mengorganisir dan mengoperasikan data secara efektif dalam pengembangan perangkat lunak menggunakan Golang, memungkinkan penanganan data yang lebih efisien dan program yang lebih rapi dan mudah dipelihara.

A. GUIDED

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

Source code :

```
package main

import "fmt"

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

Output :

```
● raihan@Raihans-MacBook-Pro minggu 3 % go run c1.go
3661
1 jam 1 menit 1 detik
● raihan@Raihans-MacBook-Pro minggu 3 % go run c1.go
3600
1 jam 0 menit 0 detik
```

2. Sebuah program digunakan untuk menentukan apakah setiap digit pada suatu 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 :

```
● raihan@Raihans-MacBook-Pro minggu 3 % go run c2.go
189
true
● raihan@Raihans-MacBook-Pro minggu 3 % go run c2.go
133
true
● raihan@Raihans-MacBook-Pro minggu 3 % go run c2.go
321
false
```

3. Sebuah program digunakan untuk menghitung BMI atau Body Mass Indeks, yang mana BMI merupakan hasil bagi dari berat badan dengan kuadrat dari tinggi badan.

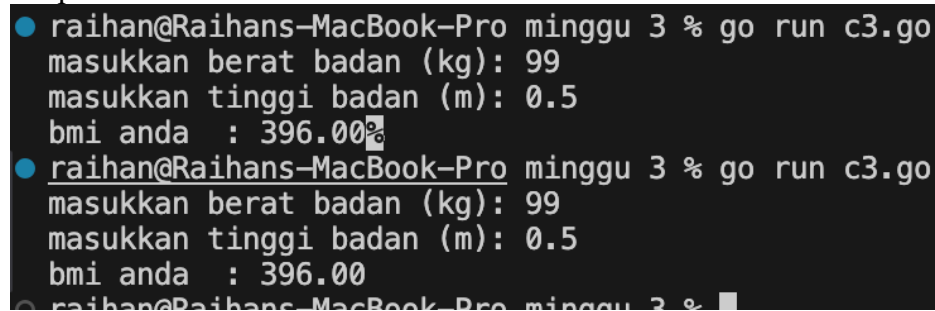
Source code:

```
package main

import "fmt"

func main() {
    var bb, tb, bmi float64
    fmt.Print("masukkan berat badan (kg): ")
    fmt.Scan(&bb)
    fmt.Print("masukkan tinggi badan (m): ")
    fmt.Scan(&tb)
    bmi = bb / (tb * tb)
    fmt.Printf("bmi anda : %.2f", bmi)
}
```

Output :



The screenshot shows a terminal window with two instances of the BMI program being run. The first instance shows the input '99' for weight and '0.5' for height, resulting in a BMI of '396.00%'. The second instance shows the same inputs, resulting in a BMI of '396.00'.

```
● raihan@Raihans-MacBook-Pro minggu 3 % go run c3.go
masukkan berat badan (kg): 99
masukkan tinggi badan (m): 0.5
bmi anda : 396.00%
● raihan@Raihans-MacBook-Pro minggu 3 % go run c3.go
masukkan berat badan (kg): 99
masukkan tinggi badan (m): 0.5
bmi anda : 396.00
● raihan@Raihans-MacBook-Pro minggu 3 %
```

B. UNGUIDED

1. Latihan 1

Sebuah program digunakan untuk menghitung total harga setelah memperoleh diskon dengan besaran tertentu.

Source Code:

```
package main

import "fmt"

func main() {
    var hargawal, diskon, total int
    fmt.Scan(&hargawal)
    fmt.Scan(&diskon)

    total = hargawal - (hargawal*diskon)/100
    fmt.Println(total)
}
```

Output :

```

● raihan@Raihans-MacBook-Pro minggu 3 % go run p1.go
20000
30
14000
● raihan@Raihans-MacBook-Pro minggu 3 % go run p1.go
10000
10
9000

```

Penjelasan Program:

Program ini meminta dua input teks dari pengguna untuk nilai harga awal dan persentase diskon. Setelah menerima input, program menghitung harga setelah diskon dengan mengurangi sejumlah persentase dari harga awal. Formula yang digunakan adalah harga awal dikurangi hasil dari harga awal dikali diskon, dibagi seratus. Program kemudian menampilkan harga akhir setelah diskon.

2. Latihan 2

Buatlah program yang digunakan untuk menentukan berat badan seseorang apabila diketahui nilai BMI dan tinggi badannya.

Source Code:

```

package main

import "fmt"

func main() {
    var bmi, tb, bb float64
    fmt.Scan(&bmi, &tb)

    bb = bmi * tb * tb

    fmt.Printf("%.f\n", bb)
}

```

Output:

```

● raihan@Raihans-MacBook-Pro minggu 3 % go run p2.go
22.85 1.75
70
● raihan@Raihans-MacBook-Pro minggu 3 % go run p2.go
23.43 1.6
60
● raihan@Raihans-MacBook-Pro minggu 3 % go run p2.go
24.69 1.8
80

```

Deskripsi Program:

Program ini meminta dua input numerik dari pengguna: nilai BMI (Body Mass Index) dan tinggi badan (dalam meter). Setelah menerima input, program menghitung berat badan ideal dengan menggunakan rumus yang mengalikan nilai BMI dengan kuadrat dari tinggi badan. Kemudian, program menampilkan berat badan ideal yang telah dihitung dalam format bilangan bulat tanpa desimal.

3. Latihan 3

Dalam sebuah peta, terdapat tiga titik yang membentuk segitiga. Titik A, B, dan C masing-masing memiliki koordinat dalam sistem kartesius 2 dimensi. Tugas Anda adalah menghitung panjang sisi-sisi segitiga yang dibentuk oleh titik-titik tersebut dan menentukan sisi terpanjang dari segitiga tersebut menggunakan teorema Pythagoras. Masukan terdiri dari tiga baris, yang mana masing-masing berisi dua bilangan riil yang menyatakan koordinat titik A, B, dan C dalam format x y.

Source Code:

```
package main

import (
    "fmt"
    "math"
)

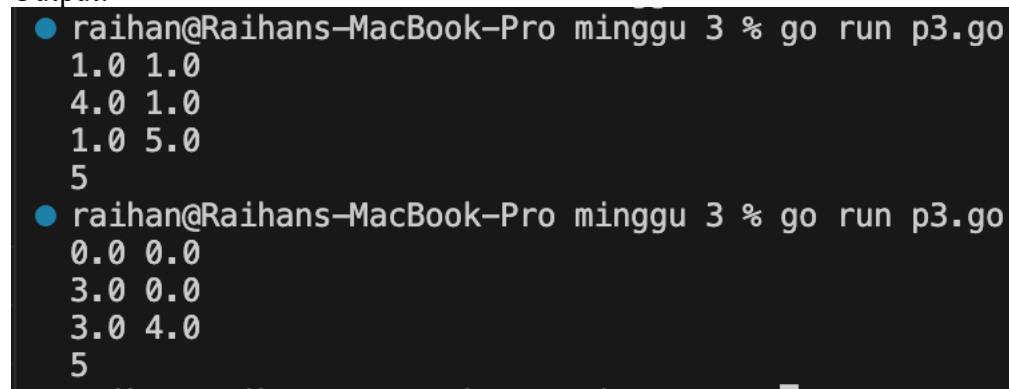
func main() {
    var x1, x2, x3, y1, y2, y3 float64
    fmt.Scan(&x1, &y1)
    fmt.Scan(&x2, &y2)
    fmt.Scan(&x3, &y3)

    pjgAB := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
    pjgBC := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
    pjgCA := math.Sqrt(math.Pow(x1-x3, 2) + math.Pow(y1-y3, 2))

    plgpjg := math.Max(pjgAB, math.Max(pjgBC, pjgCA))

    fmt.Println(plgpjg)
}
```

Output:



```
● raihan@Raihans-MacBook-Pro minggu 3 % go run p3.go
1.0 1.0
4.0 1.0
1.0 5.0
5
● raihan@Raihans-MacBook-Pro minggu 3 % go run p3.go
0.0 0.0
3.0 0.0
3.0 4.0
5
```

Deskripsi Program:

Program ini meminta input koordinat tiga titik pada bidang dua dimensi dari pengguna. Setiap titik diwakili oleh sepasang nilai (x) dan (y) , yang dimasukkan secara bertahap. Setelah menerima input, program menghitung panjang dari tiga garis yang terbentuk antara setiap pasangan titik menggunakan formula jarak antara dua

titik di geometri Euclidean. Program selanjutnya mengidentifikasi garis dengan panjang terbesar dari ketiga garis tersebut menggunakan fungsi matematika untuk menemukan nilai maksimum. Akhirnya, panjang garis terpanjang tersebut ditampilkan sebagai output.

Untuk mendukung perhitungan ini, program memanfaatkan paket ``math`` yang merupakan bagian dari pustaka standar Go. Paket ini menyediakan konstanta dan fungsi matematika dasar, seperti ``math.Sqrt`` yang menghitung akar kuadrat, ``math.Pow`` untuk pemangkatan, dan ``math.Max`` yang menentukan nilai maksimal dari dua nilai. Penggunaan ``import "math"`` dalam program memungkinkan implementasi operasi matematika yang kompleks dengan presisi tinggi, esensial dalam menghitung jarak antar titik dan menentukan panjang garis terpanjang.

DAFTAR PUSTAKA

Prayogo, N. A. (2021). *Dasar Pemrograman Go. Ebook*