

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 4

INPUT / OUTPUT, TIPE DATA & VARIABEL



Disusun oleh:

ALIN KARISA HIZANNAH

109082500010

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {

    var detik, jam, menit int

    fmt.Print("Masukkan detik waktu : ")

    fmt.Scan(&detik)

    jam = detik / 3600

    menit = (detik % 3600) / 60

    detik = detik % 60

    fmt.Println(jam, "jam", menit, "menit dan", detik,
"detik")

}
```

Screenshoot program

The image shows a Go program in a code editor and its execution output in a terminal window. The code is as follows:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var detik, jam, menit int
7     fmt.Print("Masukkan detik waktu : ")
8     fmt.Scan(&detik)
9     jam = detik / 3600
10    menit = (detik % 3600) / 60
11    detik = detik % 60
12    fmt.Println(jam, "jam", menit, "menit dan", detik, "detik")
13 }
14
```

The terminal window shows the following output:

```
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run guide1.go
Masukkan detik waktu : 3661
1 jam 1 menit dan 1 detik
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run guide1.go
Masukkan detik waktu : 7322
2 jam 2 menit dan 2 detik
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run guide1.go
Masukkan detik waktu : 3600
1 jam 0 menit dan 0 detik
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4>
```

The output window shows the following text:

```
109082500010
IF-13-07
ALIN KARISA HIZANNAH
```

Deskripsi program

Program di atas merupakan program konversi waktu yang ditulis menggunakan bahasa pemrograman Go (Golang). Tujuan utama dari program ini adalah untuk mengubah nilai waktu dalam satuan detik menjadi bentuk yang lebih mudah dipahami, yaitu terdiri dari jam, menit, dan detik. Pada bagian awal program, terdapat deklarasi tiga variabel bertipe data int yaitu detik, jam, dan menit. Variabel detik berfungsi sebagai input utama yang akan dimasukkan oleh pengguna, sementara jam dan menit digunakan untuk menyimpan hasil perhitungan konversi waktu.

Selanjutnya, program menampilkan pesan "Masukkan detik waktu : " yang bertujuan sebagai panduan bagi pengguna agar mengetahui data apa yang harus dimasukkan. Setelah pengguna mengetikkan nilai dalam satuan detik, program membaca input tersebut menggunakan fungsi `fmt.Scan(&detik)` dan menyimpannya ke dalam variabel detik. Proses perhitungan dimulai dengan menentukan jumlah jam melalui operasi `detik / 3600`. Angka 3600 digunakan karena dalam satu jam terdapat 3600 detik (hasil dari 60 menit dikali 60 detik). Kemudian, untuk menghitung jumlah menit, program menggunakan sisa dari pembagian sebelumnya dengan operasi `(detik % 3600) / 60`. Operator modulus `%` di sini berfungsi untuk mendapatkan sisa detik yang belum

terkonversi menjadi jam. Setelah itu, sisa detik yang tersisa dihitung dengan $\text{detik} = \text{detik} \% 60$ untuk memperoleh nilai detik akhir setelah dikurangi jumlah menit yang telah dihitung.

Langkah terakhir adalah menampilkan hasil konversi tersebut ke layar menggunakan perintah `fmt.Println()`. Program mencetak hasil dalam format yang mudah dibaca, misalnya: "1 jam 25 menit dan 45 detik". Dengan demikian, pengguna dapat dengan cepat mengetahui representasi waktu yang lebih informatif dari input detik yang dimasukkan. Secara keseluruhan, program ini menggambarkan konsep dasar pembagian integer, penggunaan operator modulus, serta interaksi input-output sederhana dalam bahasa Go. Program ini juga bisa dikembangkan lebih lanjut, misalnya dengan menambahkan validasi input atau menampilkan hasil dengan format waktu digital seperti HH:MM:SS agar lebih menarik dan fungsional.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var bilangan, d1, d2, d3 int

    fmt.Print("Masukkan sebuah bilangan: ")

    fmt.Scan(&bilangan)

    d1 = bilangan / 100

    d2 = bilangan % 100 / 10

    d3 = bilangan % 100 % 10

    fmt.Println(d1 <= d2 && d2 <= d3)

}
```

Screenshoot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan, d1, d2, d3 int
7     fmt.Print("Masukkan sebuah bilangan: ")
8     fmt.Scan(&bilangan)
9     d1 = bilangan / 100
10    d2 = bilangan % 100 / 10
11    d3 = bilangan % 100 % 10
12    fmt.Println(d1 <= d2 && d2 <= d3)
13 }
14
```

PROBLEMS TERMINAL PORTS OUTPUT DEBUG CONSOLE

- PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run guide2.go
Masukkan sebuah bilangan: 362
false
- PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run guide2.go
Masukkan sebuah bilangan: 256
true
- PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run guide2.go
Masukkan sebuah bilangan: 189
true
- PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4>

File Edit View 109 100% Wind UTF-8

109082500010
IF-13-07
ALIN KARISA HIZANNAH

Ln 3, Col 11 42 character xA Fo 100% Wind UTF-8

Deskripsi program

Program di atas merupakan program sederhana dalam bahasa Go (Golang) yang berfungsi untuk memeriksa apakah tiga digit dalam suatu bilangan tersusun secara berurutan dari kecil ke besar. Program ini menggunakan tipe data int dan operator logika untuk menentukan hasilnya berupa nilai true (benar) atau false (salah).

Pertama, program mendeklarasikan empat variabel bertipe int, yaitu bilangan, d1, d2, dan d3. Variabel bilangan digunakan untuk menampung input dari pengguna, sedangkan d1, d2, dan d3 masing-masing digunakan untuk menyimpan digit pertama, kedua, dan ketiga dari bilangan tersebut.

Program kemudian meminta pengguna memasukkan sebuah bilangan dengan perintah `fmt.Print("Masukkan sebuah bilangan: ")`, lalu membaca input tersebut menggunakan `fmt.Scan(&bilangan)`. Setelah pengguna memasukkan bilangan, program memproses setiap digitnya. Digit pertama (d1) diperoleh dengan operasi pembagian `bilangan / 100`, karena pada bilangan tiga digit, pembagian dengan 100

akan menghasilkan digit paling kiri. Digit kedua (d2) dihitung menggunakan bilangan $\% 100 / 10$, yaitu dengan mengambil sisa dua digit terakhir dari bilangan menggunakan modulus 100, lalu membaginya dengan 10 untuk mendapatkan digit di tengah. Sedangkan digit ketiga (d3) diperoleh melalui bilangan $\% 100 \% 10$, atau bisa disederhanakan menjadi bilangan $\% 10$, yang mengambil digit paling kanan dari bilangan.

Setelah ketiga digit diperoleh, program memeriksa apakah digit-digit tersebut tersusun secara menaik (non-decreasing order) menggunakan ekspresi logika $d1 \leq d2 \ \&\& \ d2 \leq d3$. Operator \leq digunakan untuk membandingkan apakah setiap digit lebih kecil atau sama dengan digit setelahnya, dan operator logika $\&\&$ memastikan bahwa kedua kondisi harus benar agar hasil akhir bernilai true. Hasil perbandingan tersebut kemudian ditampilkan menggunakan `fmt.Println()`.

Sebagai contoh, jika pengguna memasukkan angka 123, maka $d1 = 1$, $d2 = 2$, dan $d3 = 3$, sehingga kondisi $1 \leq 2 \ \&\& \ 2 \leq 3$ bernilai true. Namun, jika pengguna memasukkan angka 321, maka $d1 = 3$, $d2 = 2$, dan $d3 = 1$, sehingga kondisi $3 \leq 2 \ \&\& \ 2 \leq 1$ bernilai false. Dengan demikian, program ini dapat digunakan untuk mengecek pola urutan angka dalam bilangan tiga digit dan memperkenalkan konsep dasar pembagian, modulus, serta logika perbandingan dalam bahasa Go.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var beratBadan, tinggiBadan, bmi float64

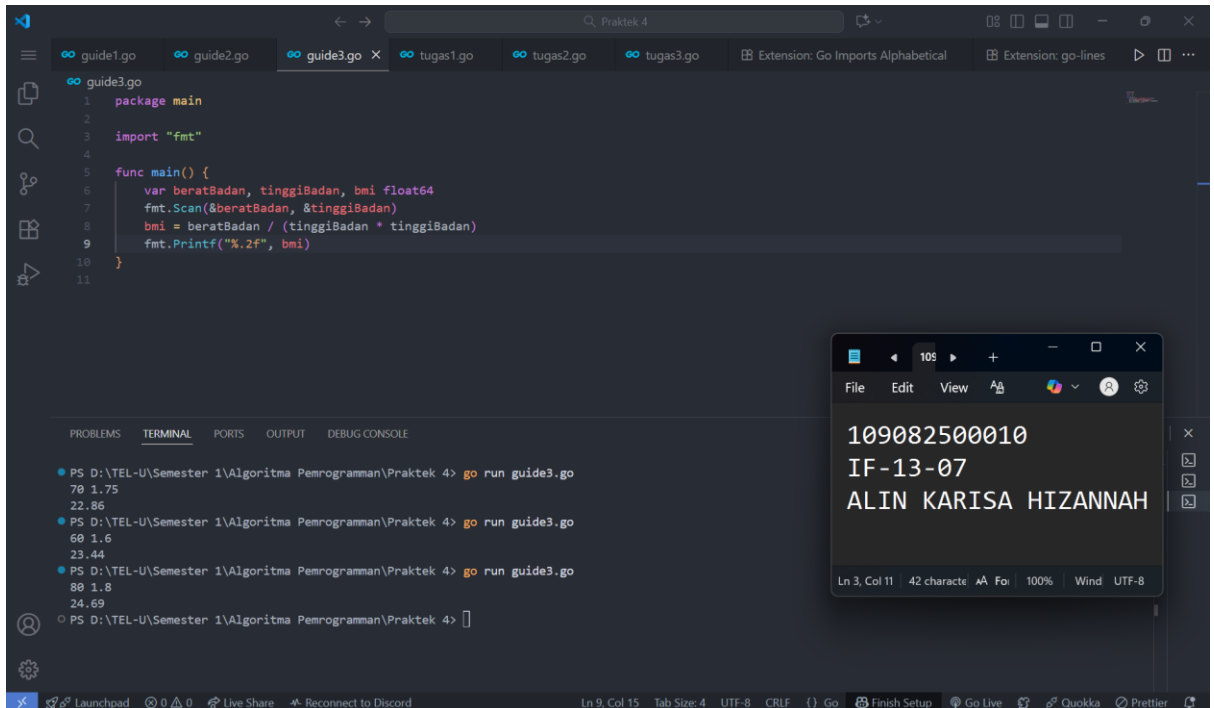
    fmt.Scan(&beratBadan, &tinggiBadan)
```

```
bmi = beratBadan / (tinggiBadan * tinggiBadan)

fmt.Printf("%.2f", bmi)

}
```

Screenshoot program



Deskripsi program

Program ini merupakan program sederhana yang digunakan untuk menghitung Body Mass Index (BMI) atau Indeks Massa Tubuh (IMT) seseorang dengan menggunakan bahasa pemrograman Go (Golang). Tujuan utama dari program ini adalah untuk membantu pengguna mengetahui apakah berat badan mereka tergolong ideal, kurang, atau berlebih berdasarkan perbandingan antara berat badan dan tinggi badan. Program bekerja dengan cara meminta pengguna memasukkan dua data, yaitu berat badan dalam satuan kilogram dan tinggi badan dalam satuan meter. Setelah data dimasukkan, program secara otomatis akan melakukan perhitungan menggunakan rumus BMI, yaitu berat badan dibagi dengan kuadrat tinggi badan.

Hasil dari perhitungan tersebut akan menghasilkan angka desimal yang menunjukkan

nilai BMI seseorang. Nilai ini kemudian ditampilkan ke layar dengan format dua angka di belakang koma agar lebih presisi dan mudah dibaca. Sebagai contoh, jika seseorang memiliki berat badan 70 kilogram dan tinggi badan 1,75 meter, maka hasil BMI-nya sekitar 22,86. Nilai ini menunjukkan bahwa berat badan orang tersebut berada pada kategori normal atau ideal.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var totalBelanja, diskonPersen float64

    var totalAkhir float64

    fmt.Print("Masukkan total belanja: ")

    fmt.Scan(&totalBelanja)

    fmt.Print("Masukkan besarnya diskon (%): ")

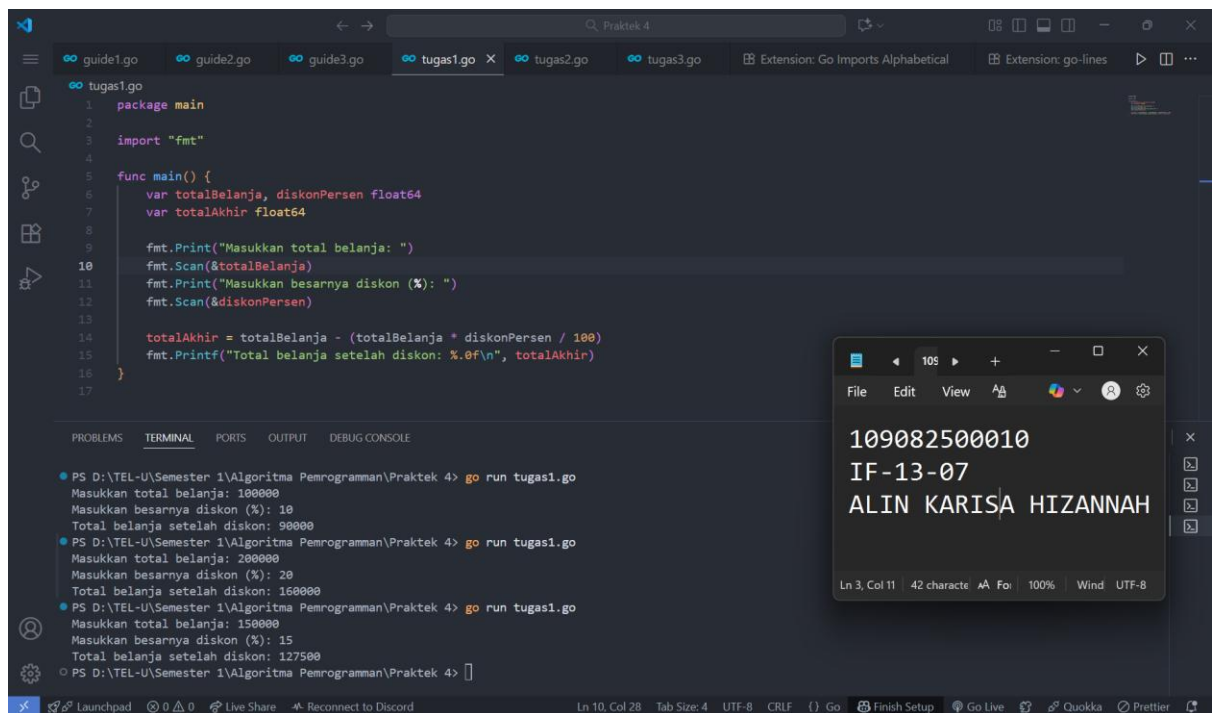
    fmt.Scan(&diskonPersen)

    totalAkhir = totalBelanja - (totalBelanja *
diskonPersen / 100)

    fmt.Printf("Total belanja setelah diskon: %.0f\n",
totalAkhir)

}
```

Screenshoot program



The screenshot shows a Go program in a text editor and its execution output in a terminal window. The program, named `tugas1.go`, is a simple application that calculates the total amount after a discount. It declares three variables: `totalBelanja`, `diskonPersen`, and `totalAkhir`, all of type `float64`. It prompts the user to enter the total amount and the discount percentage, then calculates the final total using the formula: $\text{totalAkhir} = \text{totalBelanja} - (\text{totalBelanja} \times \text{diskonPersen} / 100)$. The terminal window shows the program being run three times with different inputs, resulting in the final total being displayed for each case.

```
package main
import "fmt"

func main() {
    var totalBelanja, diskonPersen float64
    var totalAkhir float64

    fmt.Print("Masukkan total belanja: ")
    fmt.Scan(&totalBelanja)
    fmt.Print("Masukkan besarnya diskon (%): ")
    fmt.Scan(&diskonPersen)

    totalAkhir = totalBelanja - (totalBelanja * diskonPersen / 100)
    fmt.Printf("Total belanja setelah diskon: %.0f\n", totalAkhir)
}
```

```
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run tugas1.go
Masukkan total belanja: 100000
Masukkan besarnya diskon (%): 10
Total belanja setelah diskon: 90000
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run tugas1.go
Masukkan total belanja: 200000
Masukkan besarnya diskon (%): 20
Total belanja setelah diskon: 160000
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run tugas1.go
Masukkan total belanja: 150000
Masukkan besarnya diskon (%): 15
Total belanja setelah diskon: 127500
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4>
```

Deskripsi program

Program di atas merupakan program sederhana dalam bahasa Go (Golang) yang berfungsi untuk menghitung total belanja setelah mendapatkan potongan diskon. Program ini dirancang agar pengguna dapat mengetahui berapa jumlah uang yang harus dibayarkan setelah diskon diterapkan pada total belanja.

Pertama, program mendeklarasikan tiga variabel bertipe `float64`, yaitu `totalBelanja`, `diskonPersen`, dan `totalAkhir`. Variabel `totalBelanja` digunakan untuk menampung jumlah belanja sebelum diskon, `diskonPersen` menyimpan besarnya diskon dalam bentuk persentase, dan `totalAkhir` digunakan untuk menyimpan hasil akhir setelah potongan diskon diterapkan. Program kemudian meminta pengguna untuk memasukkan nilai total belanja dan besarnya diskon menggunakan perintah `fmt.Scan`.

Selanjutnya, program melakukan perhitungan dengan rumus $\text{total akhir} = \text{total belanja} - (\text{total belanja} \times \text{diskon} / 100)$. Rumus ini digunakan untuk mengurangi total belanja dengan besarnya potongan harga sesuai persentase diskon yang dimasukkan oleh pengguna. Hasil perhitungan kemudian disimpan ke dalam variabel `totalAkhir`.

Terakhir, program menampilkan hasil perhitungan ke layar menggunakan `fmt.Printf` dengan format `%.0f`, yang berarti hasil ditampilkan tanpa angka di belakang koma agar terlihat seperti nilai uang yang bulat. Sebagai contoh, jika pengguna memasukkan total belanja sebesar 500.000 dan diskon sebesar 10%, maka program akan menampilkan hasil 450.000 sebagai total belanja setelah diskon.

Secara keseluruhan, program ini menggambarkan penerapan operasi aritmetika, input-output, dan format tampilan angka dalam bahasa Go, serta bermanfaat dalam konteks kehidupan sehari-hari, khususnya dalam menghitung potongan harga saat berbelanja.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var bmi, tinggi, berat float64

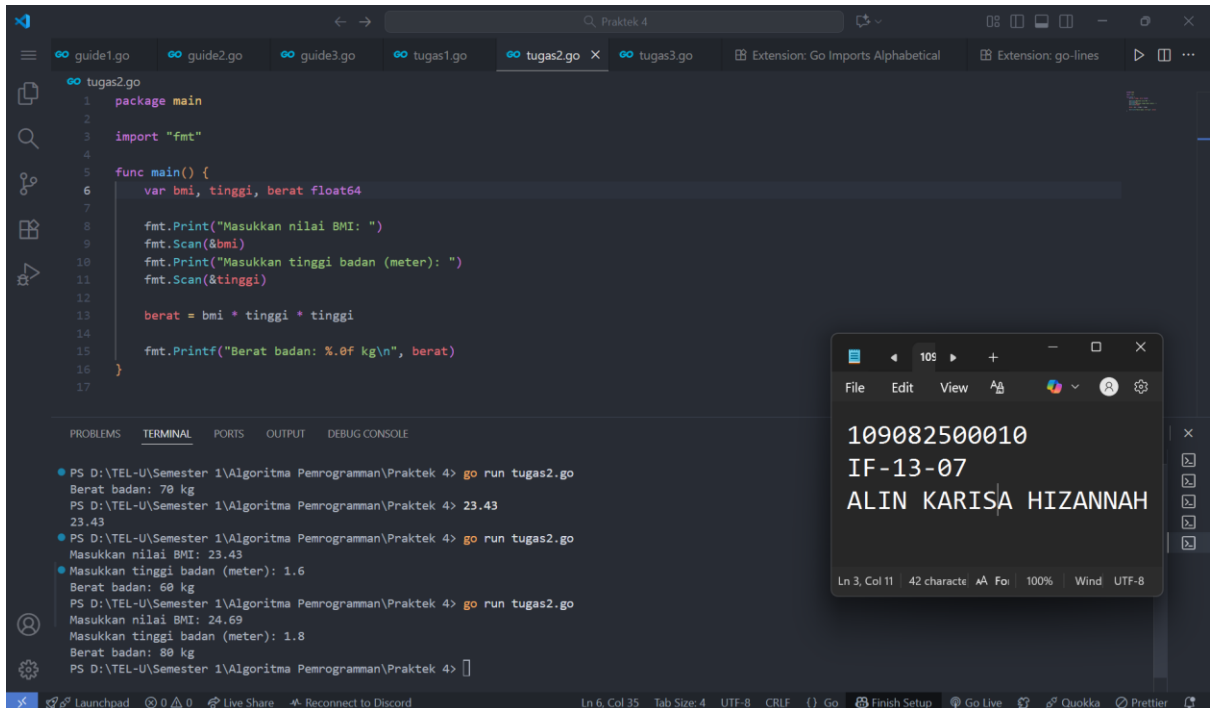
    fmt.Print("Masukkan nilai BMI: ")
    fmt.Scan(&bmi)

    fmt.Print("Masukkan tinggi badan (meter): ")
    fmt.Scan(&tinggi)

    berat = bmi * tinggi * tinggi
```

```
        fmt.Printf("Berat badan: %.0f kg\n", berat)
    }
}
```

Screenshoot program



Deskripsi program

Program di atas merupakan program sederhana dalam bahasa Go (Golang) yang digunakan untuk menghitung berat badan berdasarkan nilai Body Mass Index (BMI) dan tinggi badan seseorang. Program ini bekerja dengan membalik rumus perhitungan BMI, di mana biasanya BMI dihitung dari berat badan dibagi dengan kuadrat tinggi badan. Dalam program ini, rumus tersebut dibalik untuk mencari berat badan, yaitu

$$\text{berat} = \text{BMI} \times (\text{tinggi} \times \text{tinggi}).$$

Pada bagian awal, program mendeklarasikan tiga variabel bertipe float64, yaitu bmi, tinggi, dan berat. Variabel bmi digunakan untuk menampung nilai BMI yang dimasukkan pengguna, tinggi menyimpan nilai tinggi badan dalam satuan meter, dan berat akan digunakan untuk menyimpan hasil perhitungan berat badan. Program kemudian meminta pengguna untuk memasukkan nilai BMI serta tinggi badan melalui perintah `fmt.Print` dan membaca inputnya menggunakan `fmt.Scan`.

Setelah data dimasukkan, program menghitung berat badan dengan mengalikan nilai BMI dengan kuadrat tinggi badan. Hasilnya kemudian disimpan ke dalam variabel berat. Selanjutnya, hasil perhitungan tersebut ditampilkan ke layar menggunakan `fmt.Printf` dengan format `%.0f`, yang berarti hasil ditampilkan tanpa angka desimal agar terlihat seperti satuan kilogram yang utuh.

Sebagai contoh, jika pengguna memasukkan nilai BMI sebesar 23 dan tinggi badan 1.70 meter, maka berat badan yang dihasilkan adalah $23 \times (1.70 \times 1.70) = 66.49$, sehingga hasil yang ditampilkan adalah 66 kg. Program ini sederhana namun bermanfaat untuk memperkirakan berat badan ideal seseorang berdasarkan nilai BMI dan tinggi badan, serta memperkenalkan konsep dasar aritmetika, input-output, dan format angka dalam bahasa Go.

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"

)

func main() {
    var xA, yA, xB, yB, xC, yC float64
    var AB, BC, CA, maxSide float64

    // Input koordinat titik A
    fmt.Print("Masukkan koordinat titik A (x y): ")
```

```

    fmt.Scan(&xA, &yA)

    // Input koordinat titik B
    fmt.Print("Masukkan koordinat titik B (x y): ")
    fmt.Scan(&xB, &yB)

    // Input koordinat titik C
    fmt.Print("Masukkan koordinat titik C (x y): ")
    fmt.Scan(&xC, &yC)

    // Hitung panjang sisi-sisi segitiga
    AB = math.Sqrt(math.Pow(xB-xA, 2) + math.Pow(yB-yA,
2))
    BC = math.Sqrt(math.Pow(xC-xB, 2) + math.Pow(yC-yB,
2))
    CA = math.Sqrt(math.Pow(xA-xC, 2) + math.Pow(yA-yC,
2))

    // Tentukan sisi terpanjang
    maxSide = AB
    if BC > maxSide {
        maxSide = BC
    }
    if CA > maxSide {
        maxSide = CA
    }

    // Tampilkan hasil
    fmt.Printf("Panjang sisi terpanjang: %.2f\n",
maxSide)
}

```

Screenshoot program

```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8
9 func main() {
10     var xA, yA, xB, yB, xC, yC float64
11     var AB, BC, CA, maxSide float64
12
13     // Input koordinat titik A
14     fmt.Print("Masukkan koordinat titik A (x y): ")
15     fmt.Scan(&xA, &yA)
16
17     // Input koordinat titik B
18     fmt.Print("Masukkan koordinat titik B (x y): ")
19     fmt.Scan(&xB, &yB)
```

```
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run tugas3.go
Masukkan koordinat titik A (x y): 1.0 1.0
Masukkan koordinat titik B (x y): 4.0 1.0
Masukkan koordinat titik C (x y): 1.0 5.0
Panjang sisi terpanjang: 5.00
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4> go run tugas3.go
Masukkan koordinat titik A (x y): 0.0 0.0
Masukkan koordinat titik B (x y): 3.0 0.0
Masukkan koordinat titik C (x y): 3.0 4.0
Panjang sisi terpanjang: 5.00
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 4>
```

```
109082500010
IF-13-07
ALIN KARISA HIZANNAH
```

Deskripsi program

Program di atas merupakan program dalam bahasa Go (Golang) yang berfungsi untuk menentukan panjang sisi terpanjang dari sebuah segitiga berdasarkan koordinat tiga titik, yaitu titik A, B, dan C. Program ini menggunakan konsep geometri bidang datar dan rumus jarak antara dua titik dalam sistem koordinat Kartesius.

Pertama, program mendeklarasikan enam variabel bertipe float64, yaitu xA, yA, xB, yB, xC, dan yC, yang masing-masing menyimpan nilai koordinat dari tiga titik pembentuk segitiga. Selain itu, terdapat tiga variabel lain yaitu AB, BC, dan CA yang digunakan untuk menyimpan panjang sisi-sisi segitiga, serta satu variabel maxSide yang digunakan untuk menyimpan nilai sisi terpanjang.

Program kemudian meminta pengguna memasukkan nilai koordinat untuk ketiga titik, dimulai dari titik A, kemudian titik B, dan terakhir titik C. Setelah semua data dimasukkan, program menghitung panjang masing-masing sisi menggunakan rumus jarak dua titik, yaitu:

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

$$BC = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2}$$

$$CA = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2}$$

Perhitungan dilakukan dengan bantuan fungsi `math.Sqrt()` untuk akar kuadrat dan `math.Pow()` untuk perpangkatan. Setelah ketiga panjang sisi diketahui, program menentukan sisi mana yang memiliki nilai terbesar dengan membandingkan AB, BC, dan CA. Hasil perbandingan disimpan dalam variabel `maxSide`.

Terakhir, program menampilkan hasil ke layar menggunakan `fmt.Printf` dengan format `%.2f`, sehingga panjang sisi ditampilkan dengan dua angka di belakang koma. Misalnya, jika pengguna memasukkan titik A(0,0), B(3,0), dan C(0,4), maka sisi terpanjangnya adalah 5.00, yang merupakan hasil dari sisi miring segitiga siku-siku.

Secara keseluruhan, program ini menunjukkan penerapan operasi matematika, input-output, serta logika perbandingan dalam Go, dan dapat digunakan untuk memahami konsep jarak antar titik serta pengenalan bentuk geometri segitiga secara komputasional.