

**LAPORAN PRAKTIKUM**

**Algoritma Pemrograman**

**MODUL 4**

**MODUL 4. I/O, TIPE DATA & VARIABEL**



**Disusun oleh:**

**ALMA BONITA MIA WARDHANA**

**109082500015**

**S1IF-13-04**

**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 x, y, z int

    fmt.Print("Masukkan detik: ")

    fmt.Scan(&x)

    y = x / 3600           // Menghitung jam

    z = (x % 3600) / 60   // Menghitung menit

    x = x % 60             // Menghitung detik sisa

    fmt.Printf("%d jam, %d menit, %d detik\n", y, z, x)

}
```

#### Screenshot program

The screenshot shows a code editor interface with a Go file named 'main.go' open. The code calculates the number of hours, minutes, and remaining seconds from a given number of seconds. The code is as follows:

```
package main

import "fmt"

func main() {
    var x, y, z int
    fmt.Print("masukan detik: ")
    fmt.Scan(&x)
    y = x / 3600
    z = (x % 3600) / 60
    x = x % 60
    fmt.Printf("%d jam, %d menit, %d detik\n", y, z, x)
}
```

Below the code editor, a terminal window shows the execution of the program. It runs the command 'go run "d:\Modul4\_AlproAlma\contohsoal1\main.go"' twice. The first run asks for input ('masukan detik: ') and prints the output ('1 jam, 1 menit, 1 detik'). The second run also asks for input ('masukan detik: ') and prints the output ('1 jam, 0 menit, 0 detik'). The terminal window also displays system status information at the bottom.

### **Deskripsi program**

Program di atas adalah program sederhana dalam bahasa Go yang digunakan untuk mengonversi waktu dari satuan detik menjadi jam, menit, dan detik. Cara kerjanya dimulai dengan mendeklarasikan tiga variabel bernama x, y, dan z yang bertipe int atau bilangan bulat. Program kemudian menampilkan teks “Masukkan detik:” di layar untuk meminta pengguna mengetikkan jumlah detik yang ingin dihitung. Nilai yang dimasukkan oleh pengguna akan disimpan ke dalam variabel x menggunakan perintah fmt.Scan(&x).

Setelah itu, program mulai melakukan perhitungan waktu. Variabel y digunakan untuk menyimpan hasil pembagian total detik dengan angka 3600, karena satu jam terdiri dari 3600 detik. Dengan begitu, y mewakili jumlah jam dari total detik yang dimasukkan. Kemudian variabel z dihitung dari sisa pembagian detik terhadap 3600 (menggunakan operator %), lalu dibagi 60, karena satu menit sama dengan 60 detik. Hasilnya, z akan menyimpan jumlah menit setelah jam dikurangi. Selanjutnya, variabel x diperbarui lagi dengan sisa pembagian detik terhadap 60, yang berarti x sekarang menyimpan jumlah detik yang tersisa setelah jam dan menit diambil.

Terakhir, program menampilkan hasil perhitungan tersebut menggunakan fmt.Printf, dengan format yang rapi: jumlah jam, jumlah menit, dan jumlah detik. Misalnya, jika pengguna memasukkan nilai 3665, maka hasil yang ditampilkan adalah “1 jam, 1 menit, 5 detik”. Program ini membantu mengubah waktu dalam detik menjadi bentuk yang lebih mudah dipahami oleh manusia.

## **2. Guided 2**

### **Source Code**

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3 int

    fmt.Print("Masukkan bilangan tiga angka: ")

    fmt.Scan(&bilangan)

    d1 = bilangan / 100          // Mengambil angka
    ratusan
```

```

d2 = (bilangan % 100) / 10 // Mengambil angka
puluhan

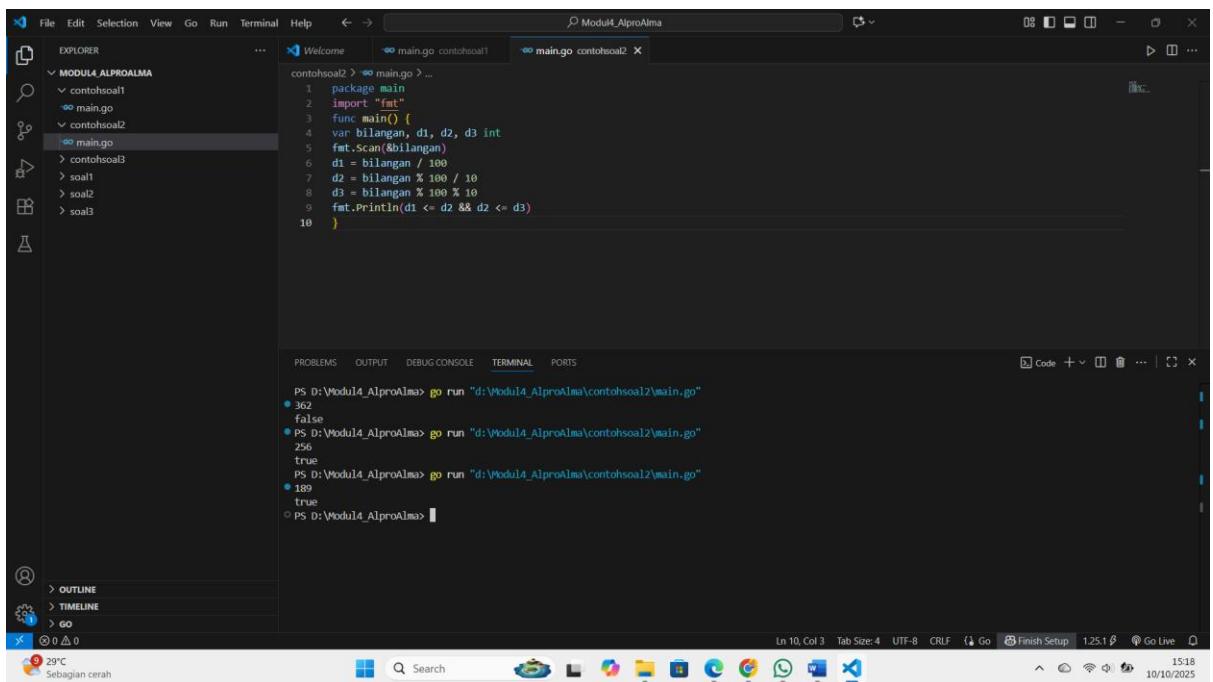
d3 = bilangan % 10 // Mengambil angka
satuan

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

}

```

## Screenshot program



## Deskripsi program

Program di atas merupakan program sederhana dalam bahasa Go yang berfungsi untuk memeriksa apakah tiga digit dari sebuah bilangan berurutan dari kecil ke besar atau tidak. Di awal program, terdapat deklarasi tiga variabel yaitu d1, d2, dan d3, yang masing-masing digunakan untuk menyimpan digit ratusan, puluhan, dan satuan dari bilangan yang dimasukkan oleh pengguna. Program kemudian meminta pengguna untuk mengetikkan sebuah bilangan, lalu membaca nilai tersebut menggunakan perintah `fmt.Scan(&bilangan)`.

Setelah nilai dimasukkan, program memecah bilangan tersebut menjadi tiga bagian. Variabel d1 diisi dengan hasil pembagian bilangan dengan 100, sehingga nilainya mewakili digit ratusan. Variabel d2 dihitung dari sisa pembagian bilangan dengan 100, kemudian hasilnya dibagi 10, yang berarti mengambil digit puluhan. Sedangkan d3 dihitung dari sisa pembagian bilangan dengan 100, lalu diambil sisa 10-nya, sehingga nilainya mewakili digit satuan.

Setelah ketiga digit diperoleh, program melakukan perbandingan dengan ekspresi `d1 <= d2 && d2 <= d3`. Artinya, program akan mengecek apakah digit pertama lebih kecil

atau sama dengan digit kedua, dan digit kedua lebih kecil atau sama dengan digit ketiga. Jika kedua kondisi ini benar, maka hasilnya true; sebaliknya, jika salah satu tidak terpenuhi, hasilnya false. Hasil ini kemudian ditampilkan di layar menggunakan fmt.Println. Misalnya, jika pengguna memasukkan angka 345, program akan menampilkan true, sedangkan jika memasukkan 531, hasilnya akan false.

### 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)

}
```

#### Screenshot program

The screenshot shows a Go development environment with the following details:

- Explorer:** Shows a project structure under "MODUL4\_ALPROALMA" with files: contohsoal1, main.go, contohsoal2, main.go, contohsoal3, main.go, soal1, soal2, and soal3.
- Code Editor:** Displays the source code for "main.go" in the "contohsoal3" package.
- Terminal:** Shows command-line history:
  - PS D:\Modul4\_AlproAlma> go run "d:\Modul4\_AlproAlma\contohsoal3\main.go"
  - 70 1.75
  - 22.86
  - PS D:\Modul4\_AlproAlma> go run "d:\Modul4\_AlproAlma\contohsoal3\main.go"
  - 60 1.6
  - 23.44
  - PS D:\Modul4\_AlproAlma> go run "d:\Modul4\_AlproAlma\contohsoal3\main.go"
  - 88 1.8
  - 24.69
  - PS D:\Modul4\_AlproAlma>
- Status Bar:** Includes system icons, language settings (1.25.1), and a date/time stamp (10/10/2025).

## **Deskripsi program**

Program di atas adalah program sederhana dalam bahasa Go yang digunakan untuk menghitung BMI (Body Mass Index) atau Indeks Massa Tubuh seseorang berdasarkan berat badan dan tinggi badan yang dimasukkan. Di dalam program, terdapat tiga variabel bertipe float64, yaitu beratBadan, tinggiBadan, dan bmi. Tipe data ini dipilih karena nilai berat dan tinggi badan biasanya mengandung angka desimal, misalnya 60.5 kilogram atau 1.70 meter.

Ketika program dijalankan, pengguna diminta untuk memasukkan dua nilai, yaitu berat badan dan tinggi badan, yang dibaca menggunakan perintah fmt.Scan(&beratBadan, &tinggiBadan). Setelah kedua nilai tersebut dimasukkan, program akan menghitung nilai BMI menggunakan rumus standar, yaitu berat badan dibagi dengan kuadrat dari tinggi badan. Rumus ini ditulis dalam kode sebagai bmi = beratBadan / (tinggiBadan \* tinggiBadan).

Hasil perhitungan kemudian ditampilkan di layar menggunakan perintah fmt.Printf("%.2f", bmi), yang berarti hasil BMI akan dicetak dengan dua angka di belakang koma agar tampak lebih rapi dan mudah dibaca. Misalnya, jika pengguna memasukkan berat badan 60 dan tinggi badan 1.65, maka program akan menghitung BMI-nya menjadi sekitar 22.04. Program ini membantu pengguna mengetahui nilai BMI mereka, yang bisa digunakan untuk menilai apakah berat badan tergolong ideal, kurang, atau berlebih.

## **TUGAS**

### **1. Tugas 1**

#### **Source code**

```
package main

import "fmt"

func main() {

    var beli, diskon int

    var hasil int


    fmt.Println("Masukkan harga beli:")

    fmt.Scan(&beli)


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

    fmt.Scan(&diskon)
```

```

    hasil = beli * diskon / 100

    hasil2 := beli - hasil

    fmt.Println("Harga setelah diskon adalah:", hasil2)

}

```

## Screenshot program

```

package main

import "fmt"

func main() {
    var beli, diskon int
    var hasil int

    fmt.Println("Masukkan harga beli:")
    fmt.Scan(&beli)

    fmt.Println("Masukkan diskon (%): ")
    fmt.Scan(&diskon)

    hasil = beli * diskon / 100
    hasil2 := beli - hasil

    fmt.Println("Harga setelah diskon adalah:", hasil2)
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Modul4\_AlproAlma> go run "d:\Modul4\_AlproAlma\soal1\main.go"
● Masukkan harga beli:
100000
Masukkan diskon (%): 10
Harga setelah diskon adalah: 90000
PS D:\Modul4\_AlproAlma> go run "d:\Modul4\_AlproAlma\soal1\main.go"
● Masukkan harga beli:
200000
Masukkan diskon (%): 20
Harga setelah diskon adalah: 160000
PS D:\Modul4\_AlproAlma> go run "d:\Modul4\_AlproAlma\soal1\main.go"
● Masukkan harga beli:
150000
Masukkan diskon (%): 15
Harga setelah diskon adalah: 127500
○ PS D:\Modul4\_AlproAlma>

## Deskripsi program

Program di atas adalah program sederhana dalam bahasa Go yang berfungsi untuk menghitung harga akhir suatu barang setelah diberi diskon. Di awal program, terdapat dua variabel bertipe int bernama beli dan diskon, yang masing-masing digunakan untuk menyimpan harga awal barang dan besar persentase diskon yang diberikan. Selain itu, ada juga variabel hasil yang digunakan untuk menyimpan nilai potongan harga dari hasil perhitungan diskon tersebut.

Saat program dijalankan, pengguna akan diminta untuk memasukkan harga beli barang melalui perintah `fmt.Println("Masukkan harga beli:")` dan nilainya disimpan ke dalam variabel beli menggunakan `fmt.Scan(&beli)`. Setelah itu, program meminta pengguna memasukkan besar diskon dalam persen, misalnya 10 untuk 10%, yang kemudian disimpan dalam variabel diskon.

Setelah kedua nilai tersebut diperoleh, program menghitung besarnya potongan harga dengan rumus  $hasil = beli * diskon / 100$ . Artinya, harga beli dikalikan dengan persentase diskon, lalu dibagi 100 untuk mendapatkan jumlah potongan dalam satuan rupiah. Nilai potongan ini kemudian digunakan untuk menghitung harga akhir setelah

diskon, yaitu hasil2 := beli - hasil. Jadi, harga awal dikurangi dengan jumlah potongan diskon menghasilkan harga yang harus dibayar.

Terakhir, program menampilkan hasilnya ke layar dengan perintah `fmt.Println("Harga setelah diskon adalah:", hasil2)`. Misalnya, jika pengguna memasukkan harga beli 200000 dan diskon 10, maka program akan menampilkan hasil 180000 sebagai harga setelah diskon. Dengan kata lain, program ini membantu pengguna mengetahui berapa harga yang perlu dibayar setelah potongan harga diterapkan.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {

    var bmi, tinggi float64

    var berat float64


    fmt.Print("Masukkan BMI: ")

    fmt.Scan(&bmi)

    fmt.Print("Masukkan tinggi (meter): ")

    fmt.Scan(&tinggi)

    berat = bmi * (tinggi * tinggi)

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

}
```

### Screenshoot program

```

File Edit Selection View Go Run Terminal Help < > Welcome main.go contohsoal1 main.go contohsoal2 main.go contohsoal3 main.go soal1 main.go soal2
EXPLORER MODUL4_ALPROALMA
  contohsoal1
    main.go
  contohsoal2
    main.go
  main.go
  contohsoal3
    main.go
    soal1
    main.go
    soal2
    main.go
    > soal3
  > main.go
  > soal3

soal2 > `main.go` ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bmi, tinggi float64
7     var berat float64
8
9     fmt.Println("Masukkan BMI: ")
10    fmt.Scan(&bmi)
11
12    fmt.Println("Masukkan tinggi (meter): ")
13    fmt.Scan(&tinggi)
14
15    berat = bmi * (tinggi * tinggi)
16    fmt.Printf("Perkiraan berat badan: %.0f kg\n", berat)
17 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\soal2\main.go"
● Masukkan BMI: 22.85
Masukkan tinggi (meter): 1.75
Perkiraan berat badan: 70 kg
● PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\soal2\main.go"
Masukkan BMI: 23.43
Masukkan tinggi (meter): 1.6
Perkiraan berat badan: 60 kg
● PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\soal2\main.go"
Masukkan BMI: 24.69
Masukkan tinggi (meter): 1.8
Perkiraan berat badan: 88 kg
○ PS D:\Modul4_AlproAlma>

```

The screenshot shows the Visual Studio Code interface with the Go extension installed. The Explorer sidebar shows a project structure with files like `main.go`, `contohsoal1`, `contohsoal2`, `contohsoal3`, `soal1`, `soal2`, and `soal3`. The main editor window displays a Go program that calculates BMI. The terminal below shows three runs of the program with different input values (BMI: 22.85, 23.43, 24.69) and their corresponding calculated weights (70 kg, 60 kg, 88 kg). The status bar at the bottom right indicates the date and time as 10/10/2025 and 15:28.

## Deskripsi program

Program di atas adalah program sederhana dalam bahasa Go yang digunakan untuk menghitung perkiraan berat badan seseorang berdasarkan nilai BMI (Body Mass Index) dan tinggi badannya. Di awal program, terdapat tiga variabel bertipe float64 yaitu bmi, tinggi, dan berat. Tipe data float64 dipilih karena nilai-nilai seperti tinggi badan dan BMI biasanya mengandung angka desimal, misalnya 1.75 meter atau 22.5. Ketika program dijalankan, pengguna akan diminta untuk memasukkan nilai BMI terlebih dahulu melalui perintah `fmt.Println("Masukkan BMI: ")`, kemudian nilainya disimpan dalam variabel bmi menggunakan `fmt.Scan(&bmi)`. Setelah itu, program meminta pengguna untuk memasukkan tinggi badan dalam satuan meter melalui `fmt.Println("Masukkan tinggi (meter): ")`, dan nilainya disimpan ke dalam variabel tinggi. Setelah kedua nilai tersebut diperoleh, program menghitung perkiraan berat badan dengan menggunakan rumus  $\text{berat} = \text{BMI} \times (\text{tinggi} \times \text{tinggi})$ . Rumus ini berasal dari rumus dasar perhitungan BMI, yaitu  $\text{BMI} = \text{berat} / (\text{tinggi} \times \text{tinggi})$ . Dalam program ini, rumus tersebut dibalik untuk mencari berat badan berdasarkan nilai BMI yang sudah diketahui.

Hasil perhitungan kemudian ditampilkan ke layar menggunakan perintah `fmt.Printf("Perkiraan berat badan: %.0f kg\n", berat)`. Format `%.0f` digunakan agar hasil ditampilkan tanpa angka di belakang koma, sehingga hasilnya lebih sederhana dibaca. Misalnya, jika pengguna memasukkan BMI 22 dan tinggi 1.7, maka program akan menghitung berat badan sebagai  $22 \times (1.7 \times 1.7)$ , yaitu sekitar 64 kg. Dengan demikian, program ini membantu memperkirakan berat badan ideal seseorang berdasarkan nilai BMI dan tinggi badan yang dimasukkan.

### 3. Tugas 3

#### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var x1, y1, x2, y2, x3, y3 float64

    fmt.Print("Masukkan nilai x1 dan y1: ")
    fmt.Scan(&x1, &y1)

    fmt.Print("Masukkan nilai x2 dan y2: ")
    fmt.Scan(&x2, &y2)

    fmt.Print("Masukkan nilai x3 dan y3: ")
    fmt.Scan(&x3, &y3)

    // Menghitung panjang sisi-sisi segitiga
    AB := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
    BC := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
    CA := math.Sqrt(math.Pow(x1-x3, 2) + math.Pow(y1-y3, 2))

    // Menentukan sisi terpanjang
    max := AB
    nama := "AB"

    if BC > max {
        max = BC
        nama = "BC"
    }
    if CA > max {
        max = CA
        nama = "CA"
    }

    // Menampilkan hasil
    fmt.Printf("Sisi terpanjang adalah %s dengan panjang\n%.2f\n", nama, max)
}
```

### Screenshoot program

File Edit Selection View Go Run Terminal Help ← → 🔍 Modul4\_AlproAlma

EXPLORER

MODUL4\_ALPROALMA

- ↳ contoh1.go
- ↳ contoh2.go
- ↳ contoh3.go
- ↳ main.go
- ↳ main3.go
- ↳ main.go
- ↳ main.go
- ↳ main.go
- ↳ main.go

... Welcome main.go main.go main.go main.go main.go main.go main.go

```
soil3 > main.go ⚡ main
1
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var x1, y1, x2, y2, x3, y3 float64
10
11    fmt.Print("Masukkan nilai x1 dan y1: ")
12    fmt.Scan(&x1, &y1)
13
14    fmt.Print("Masukkan nilai x2 dan y2: ")
15    fmt.Scan(&x2, &y2)
16
17    fmt.Print("Masukkan nilai x3 dan y3: ")
18    fmt.Scan(&x3, &y3)
19
20
21    // Menghitung jarak sisi-sisi segitiga
22    AB := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
23    BC := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
24    CA := math.Sqrt(math.Pow(x1-x3, 2) + math.Pow(y1-y3, 2))
25
26    // Menentukan sisi terpanjang
27    max := AB
28    name := "AB"
29
30    if BC > max {
31        max = BC
32        name = "BC"
33    }
34    if CA > max {
35        max = CA
36        name = "CA"
37    }
38
39    // Menampilkan hasil
40    fmt.Println("Sisi terpanjang adalah", name, "dengan panjang", max)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Modul4\_AlproAlma\go run "D:\Modul4\_AlproAlma\soil3\main.go"

Masukkan nilai x1 dan y1: 1.0 1.0  
Masukkan nilai x2 dan y2: 4.0 4.0  
Masukkan nilai x3 dan y3: 3.0 4.0  
Sisi terpanjang adalah BC dengan panjang 5.00

PS D:\Modul4\_AlproAlma\go run "D:\Modul4\_AlproAlma\soil3\main.go"

Masukkan nilai x1 dan y1: 1.0 1.0  
Masukkan nilai x2 dan y2: 3.0 0.0  
Masukkan nilai x3 dan y3: 3.0 4.0  
Sisi terpanjang adalah BC dengan panjang 5.00

PS D:\Modul4\_AlproAlma\

OUTLINE TIMELINE GO PACKAGE OUTLINE

9 Hari panas yang... 29°C

Search

Ln 34 Col 6 Tab Size 4 UFT-8 CPU: 4 Go Finish Setup 1.25 L 15:33 10/10/2025

## Deskripsi program

Program di atas adalah program dalam bahasa Go yang digunakan untuk menentukan sisi terpanjang dari sebuah segitiga berdasarkan tiga titik koordinat yang dimasukkan pengguna. Di awal program, terdapat enam variabel bertipe float64, yaitu x1, y1, x2, y2, x3, dan y3, yang masing-masing digunakan untuk menyimpan koordinat titik A, B, dan C. Tipe float64 digunakan karena nilai koordinat bisa berupa angka desimal.

Ketika program dijalankan, pengguna diminta untuk memasukkan nilai koordinat titik pertama ( $x_1, y_1$ ), kemudian titik kedua ( $x_2, y_2$ ), dan terakhir titik ketiga ( $x_3, y_3$ ). Nilai-nilai ini dibaca menggunakan perintah `fmt.Scan` dan disimpan ke dalam variabel yang sesuai. Setelah semua nilai koordinat dimasukkan, program mulai menghitung panjang ketiga sisi segitiga, yaitu  $AB$ ,  $BC$ , dan  $CA$ .

Perhitungan panjang sisi dilakukan dengan menggunakan rumus jarak antar dua titik pada bidang koordinat, yaitu  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ . Dalam program, rumus ini ditulis menggunakan fungsi `math.Sqrt` untuk menghitung akar kuadrat dan `math.Pow` untuk menghitung pangkat dua. Dengan cara ini, program memperoleh nilai panjang dari setiap sisi segitiga.

Setelah mendapatkan panjang ketiga sisi, program kemudian membandingkan ketiganya untuk menentukan sisi mana yang paling panjang. Awalnya, variabel max diisi dengan nilai panjang sisi AB, dan variabel nama diberi nilai "AB". Program kemudian memeriksa apakah panjang sisi BC atau CA lebih besar dari nilai max. Jika iya, maka nilai max dan nama diperbarui sesuai sisi yang lebih panjang.

Terakhir, program menampilkan hasil ke layar dengan perintah `fmt.Printf`, yang menuliskan sisi terpanjang beserta panjangnya dengan dua angka di belakang koma. Misalnya, jika pengguna memasukkan tiga titik yang membentuk segitiga dengan sisi

AB sepanjang 5.00, BC sepanjang 6.32, dan CA sepanjang 4.47, maka program akan menampilkan pesan “Sisi terpanjang adalah BC dengan panjang 6.32”. Dengan demikian, program ini membantu pengguna mengetahui sisi mana yang paling panjang dari segitiga yang dibentuk oleh tiga titik koordinat yang diberikan.

## TUGAS PENDAHULUAN

### Soal 1

Pemberi Soal: Johanson Leeroy - 109082500017

#### Deskripsi Soal:

Buatlah sebuah program dalam bahasa Go yang dapat menghitung rata-rata dari 5 bilangan bulat. Program harus menerima 5 masukan berupa bilangan bulat positif.

#### Contoh Input/Output:

Input	Output
10 20 30 40 50	30
79 89 67 90 97	84.4
34 34 34 34 34	34

#### Source code

```
package main

import "fmt"

func main() {
    var a, b, c, d, e int
    var rata float64

    fmt.Println("Masukkan 5 bilangan bulat: ")
    fmt.Scan(&a, &b, &c, &d, &e)

    rata = float64(a+b+c+d+e) / 5

    fmt.Printf("Rata-rata = %.1f\n", rata)
}
```

#### Screenshot program

```

File Edit Selection View Go Run Terminal Help < > Modul4_AlproAlma
EXPLORER ... main.go contohsoal1 main.go contohsoal2 main.go contohsoal3 main.go soal1 main.go soal2 main.go soal3 main.go TP1_AJ ...
MODUL4_ALPROALMA
contohsoal1
main.go
contohsoal2
main.go
contohsoal3
main.go
soal1
main.go
soal2
main.go
soal3
main.go
TP1_AJ
main.go
TP2_AJ

TP1_AJ > main.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b, c, d, e int
7     var rata float64
8
9     fmt.Println("Masukkan 5 bilangan bulat: ")
10    fmt.Scanf("%d %d %d %d %d", &a, &b, &c, &d, &e)
11
12    rata = float64(a+b+c+d+e) / 5
13
14    fmt.Printf("Rata-rata = %.1f\n", rata)
15 }
16

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\TP1_AJ\main.go"
● Masukkan 5 bilangan bulat: 10 20 30 40 50
Rata-rata = 30.0
PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\TP1_AJ\main.go"
Masukkan 5 bilangan bulat: 79 89 67 99 97
Rata-rata = 84.4
PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\TP1_AJ\main.go"
Masukkan 5 bilangan bulat: 34 34 34 34 34
Rata-rata = 34.0
PS D:\Modul4_AlproAlma>

```

Ln 16, Col 1 Tab Size: 4 UTF-8 CR/LF ⌂ Go ⌂ Finish Setup 1.25.1 ⌂ Go Live ⌂ 15:43 10/10/2025

## Soal 2

**Pemberi Soal:** Johanson Leeroy - 109082500017

### Deskripsi Soal:

Buatlah sebuah program dalam bahasa Go yang dapat mengkonversi satuan jarak: meter menjadi kilometer. Program harus menerima 1 masukan berupa bilangan bulat positif.

### Contoh Input/Output:

Input	Output
6000	6 km
8600	8.6 km
450	0.45 km

### Source code

```

package main

import "fmt"

func main() {
    var meter float64
    var kilometer float64
}

```

```

fmt.Println("Masukkan jarak (meter): ")

fmt.Scan(&meter)

kilometer = meter / 1000

fmt.Printf("%.2f km\n", kilometer)

}

```

## Screenshoot program

The screenshot shows a code editor interface with a terminal window at the bottom. The terminal window displays the execution of a Go program named 'main.go' from the directory 'D:\Modul4\_AlproAlma\TP2\_AJ'. The program prompts for a distance in meters, reads it, converts it to kilometers, and prints the result. The terminal output is as follows:

```

PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\TP2_AJ\main.go"
Masukkan jarak (meter): 6600
6.60 km
PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\TP2_AJ\main.go"
Masukkan jarak (meter): 8600
8.60 km
PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\TP2_AJ\main.go"
Masukkan jarak (meter): 450
0.45 km
PS D:\Modul4_AlproAlma>

```