

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL IV  
I/O, TIPE DATA & VARIABEL**



**Disusun oleh:**  
**BENING PUTRI NARESWARI SUKARNO**  
**109082500211**  
**S1IF-13-02**

**Asisten Praktikum**  
Adithana dharma putra  
Alfin Ilham Berlianto

**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 time, jam, menit, detik int

    fmt.Println("waktu")

    fmt.Scan(&time)

    jam = time / 3600

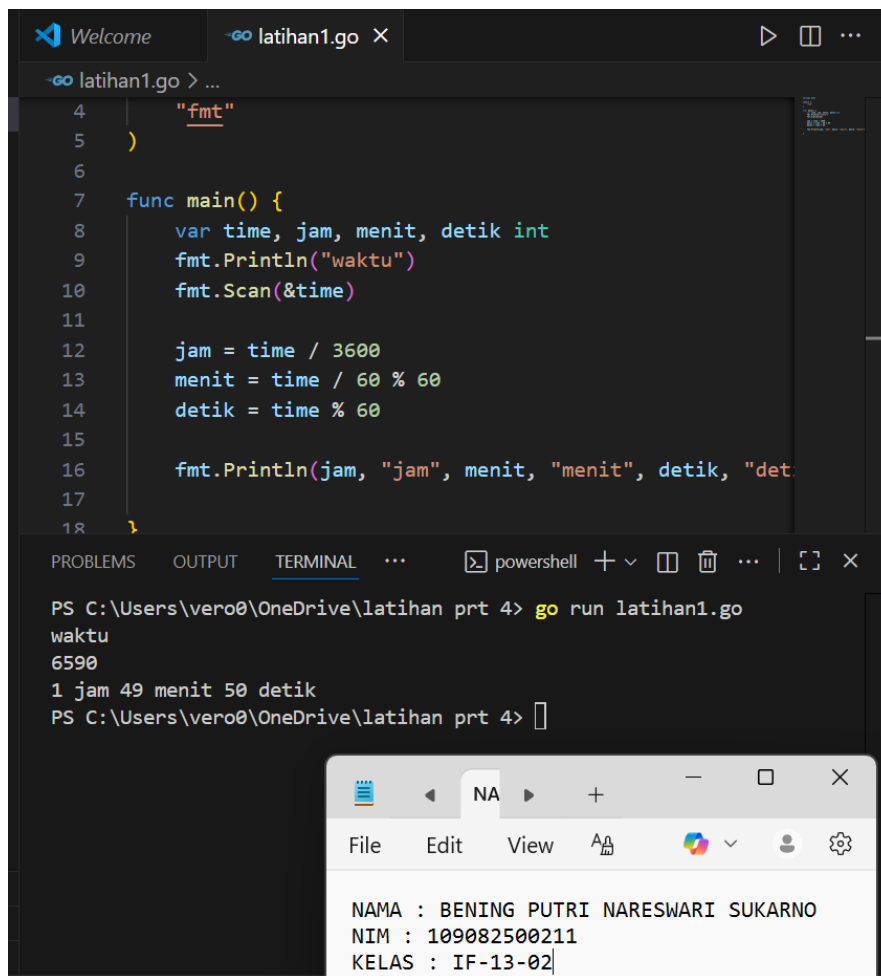
    menit = time / 60 % 60

    detik = time % 60

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

}
```

## Screenshoot program



The screenshot shows a Go program in a VS Code editor. The program is named `latihan1.go` and contains the following code:

```
4   "fmt"
5   )
6
7   func main() {
8       var time, jam, menit, detik int
9       fmt.Println("waktu")
10      fmt.Scan(&time)
11
12      jam = time / 3600
13      menit = time / 60 % 60
14      detik = time % 60
15
16      fmt.Println(jam, "jam", menit, "menit", detik, "detik")
17  }
```

The terminal output shows the program being run with the command `go run latihan1.go`. The output is:

```
PS C:\Users\vero0\OneDrive\latihan prt 4> go run latihan1.go
waktu
6590
1 jam 49 menit 50 detik
PS C:\Users\vero0\OneDrive\latihan prt 4>
```

A small window titled "NA" is also visible, displaying the following text:

```
NAMA : BENING PUTRI NARESWARI SUKARNO
NIM : 109082500211
KELAS : IF-13-02
```

## Deskripsi program

Program tersebut adalah konverter waktu yang berfungsi untuk mengubah total waktu dalam satuan detik yang dimasukkan pengguna menjadi waktu dalam jam, menit, dan detik.

Cara Kerja:

1. Program menampilkan "waktu" dan kemudian menunggu pengguna memasukkan nilai total detik. Nilai ini disimpan dalam variabel `time`.
2. Perhitungan dilakukan berdasarkan fakta bahwa 1 jam adalah 3600 detik dan 1 menit adalah 60 detik.
3. Program menampilkan hasil perhitungan (jam, menit, dan detik) ke konsol dengan format yang jelas (misalnya: 1 jam 49 menit 50 detik).

Contoh input:

1. Ketika program meminta waktu, pengguna memasukkan 6590
2. Kemudian program akan mengkonversi jam:  $6590/3600 = 1$
3. Konversi menit:  $(6590/60) \pmod{60} = 49$
4. Detik:  $6590 \pmod{60} = 50$
5. Hasil konversi 1 jam 49 menit 50 detik.

## 2. Guided 2

### Source Code

```
package main

import (
    "fmt"
)

func main() {
    var nilai, nilai1, nilai2, nilai3 int
    var hasil bool

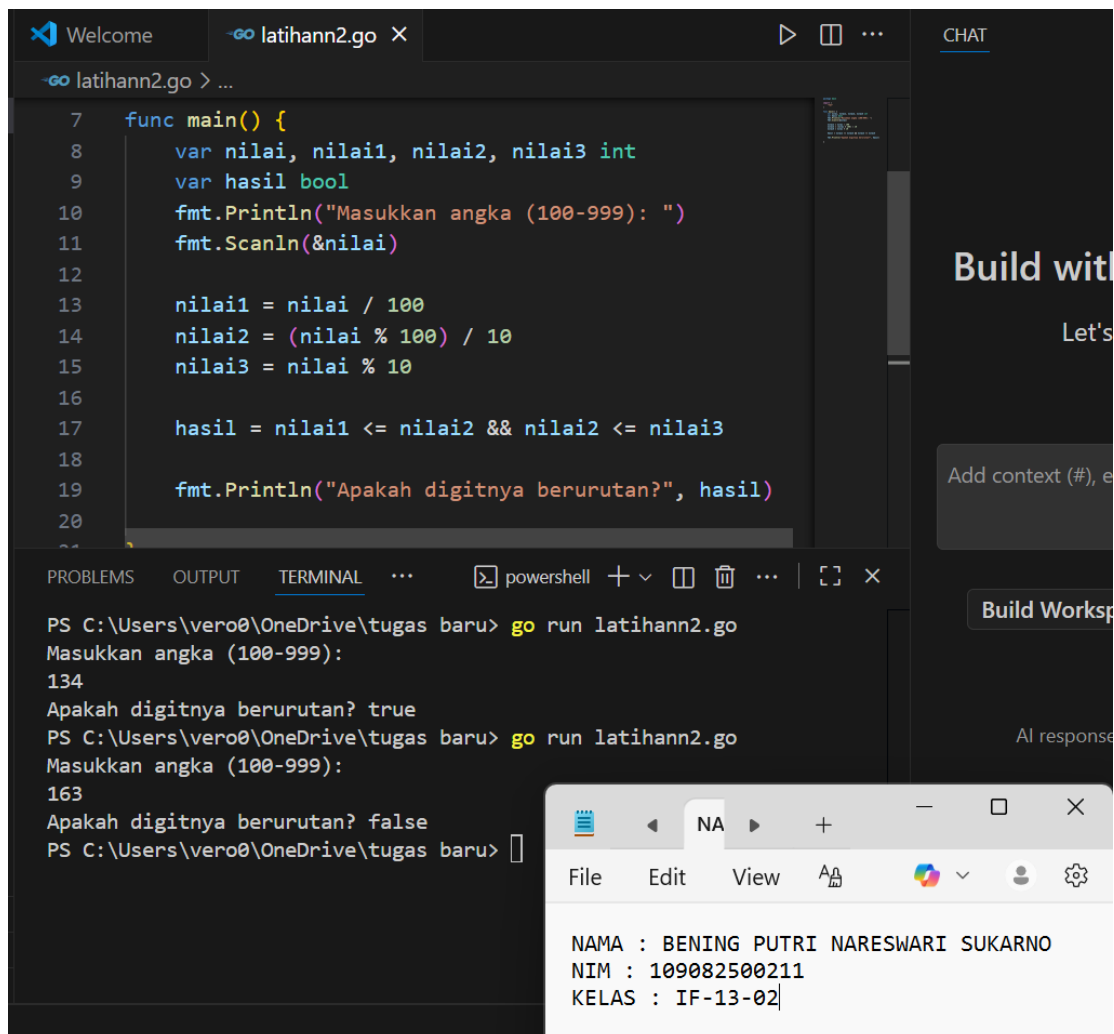
    fmt.Println("Masukkan angka (100-999): ")
    fmt.Scanln(&nilai)

    nilai1 = nilai / 100
    nilai2 = (nilai % 100) / 10
    nilai3 = nilai % 10

    hasil = nilai1 <= nilai2 && nilai2 <= nilai3

    fmt.Println("Apakah digitnya berurutan?", hasil)
}
```

## Screenshoot program



The screenshot shows a Go program in a VS Code editor and its execution in a terminal. The program is named `latihann2.go` and is located in the directory `C:\Users\vero0\OneDrive\tugas baru`.

```
7 func main() {
8     var nilai, nilai1, nilai2, nilai3 int
9     var hasil bool
10    fmt.Println("Masukkan angka (100-999): ")
11    fmt.Scanln(&nilai)
12
13    nilai1 = nilai / 100
14    nilai2 = (nilai % 100) / 10
15    nilai3 = nilai % 10
16
17    hasil = nilai1 <= nilai2 && nilai2 <= nilai3
18
19    fmt.Println("Apakah digitnya berurutan?", hasil)
20 }
```

The terminal shows the program being run twice. In the first run, the user enters `134`, and the program outputs `Apakah digitnya berurutan? true`. In the second run, the user enters `163`, and the program outputs `Apakah digitnya berurutan? false`.

Below the terminal, a small window displays the following information:

```
NAMA : BENING PUTRI NARESWARI SUKARNO
NIM  : 109082500211
KELAS : IF-13-02
```

## Deskripsi program

Program tersebut akan memecah angka tersebut menjadi tiga digit terpisah, kemudian memeriksa apakah digit-digitnya berurutan naik atau sama.

Cara Kerja:

1. Program memisahkan angka yang Anda masukkan menjadi tiga digit, yaitu digit ratusan, puluhan, dan satuan.
2. Kemudian memeriksa apakah setiap digit berikutnya lebih besar dari atau sama dengan digit sebelumnya (misalnya, digit ratusan  $\leq$  puluhan dan puluhan  $\leq$  satuan).
3. Hasil pemeriksaan urutan tersebut akan ditampilkan sebagai `true` (Benar) jika urutannya naik/sama, atau `false` (Salah) jika urutannya melompat turun.

Contoh Masukkan 1:

1. Program meminta pengguna memasukkan angka (100-199)
2. Pengguna memasukkan angka 134
3. Program akan memproses digit  $1 \leq 3 \leq 4$
4. Maka hasilnya **True**, karena digitnya berurutan naik.

Contoh Masukkan 2:

1. Program meminta pengguna memasukkan angka (100-199)
2. Pengguna memasukkan angka 163
3. Program akan memproses digit  $1 \leq 6 \geq 3$
4. Maka hasilnya **False**, karena digitnya tidak berurutan naik.

### 3. Guided 3

#### Source Code

```
package main

import (
    "fmt"
)

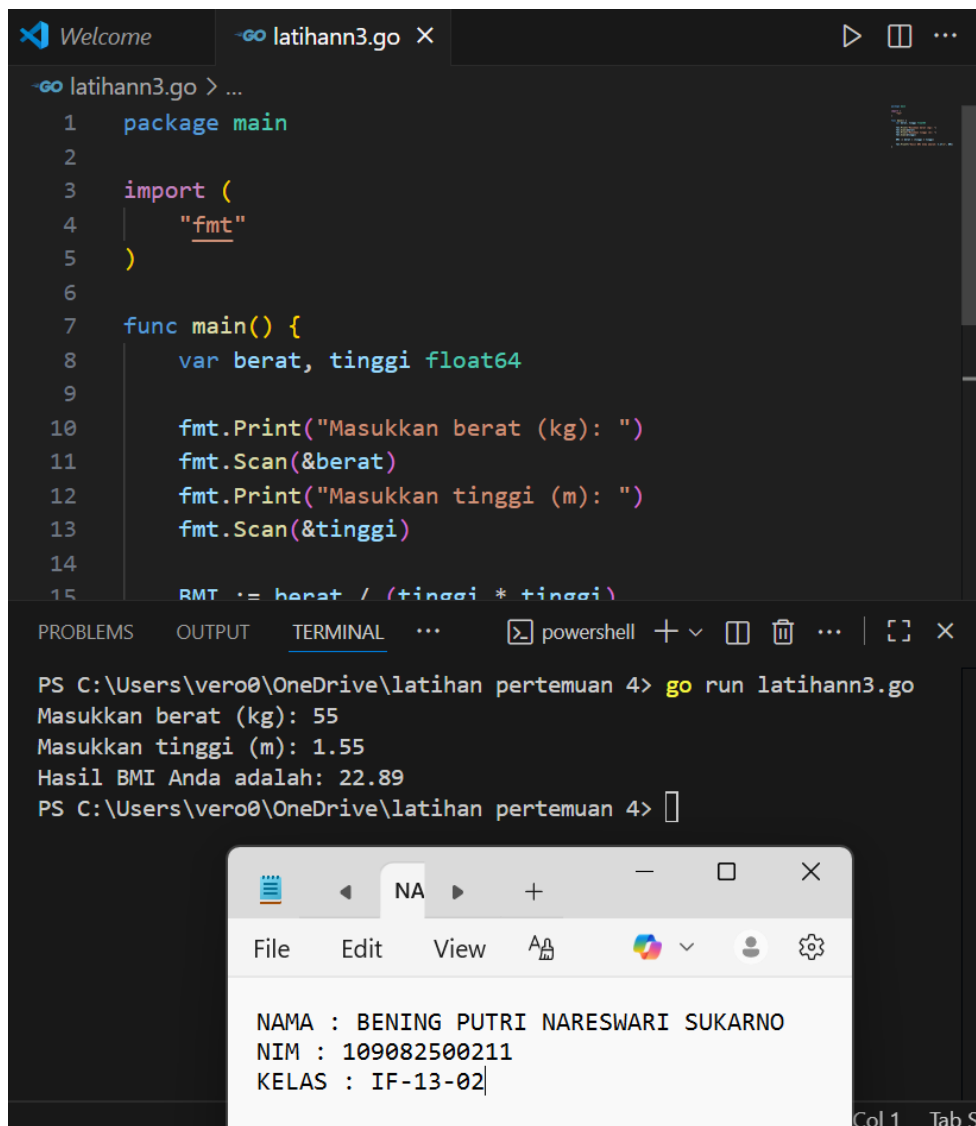
func main() {
    var berat, tinggi float64

    fmt.Print("Masukkan berat (kg): ")
    fmt.Scan(&berat)
    fmt.Print("Masukkan tinggi (m): ")
    fmt.Scan(&tinggi)

    BMI := berat / (tinggi * tinggi)

    fmt.Printf("Hasil BMI Anda adalah: %.2f\n", BMI)
}
```

## Screenshoot program



The screenshot shows a Go program in a VS Code editor and its execution in a PowerShell terminal. The program is a simple BMI calculator. The code in the editor is as follows:

```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var berat, tinggi float64
9
10    fmt.Print("Masukkan berat (kg): ")
11    fmt.Scan(&berat)
12    fmt.Print("Masukkan tinggi (m): ")
13    fmt.Scan(&tinggi)
14
15    BMI := berat / (tinggi * tinggi)
```

The terminal output shows the program being run and the user's input:

```
PS C:\Users\vero0\OneDrive\latihan pertemuan 4> go run latihann3.go
Masukkan berat (kg): 55
Masukkan tinggi (m): 1.55
Hasil BMI Anda adalah: 22.89
PS C:\Users\vero0\OneDrive\latihan pertemuan 4>
```

Below the terminal, a small window titled 'NA' displays the user's personal information:

```
NAMA : BENING PUTRI NARESWARI SUKARNO
NIM  : 109082500211
KELAS : IF-13-02
```

## Deskripsi program

Program tersebut adalah kalkulator sederhana untuk menghitung Indeks Massa Tubuh (BMI).

Cara Kerja:

1. Mendeklarasikan dua variabel dengan tipe data float64 yaitu berat (kg) dan tinggi (meter).
2. Mencetak pesan "Masukkan berat (kg): " ke terminal, kemudian menunggu dan membaca input angka dari pengguna untuk variabel berat.
3. Mencetak pesan "Masukkan tinggi (m): " ke terminal, kemudian menunggu dan membaca input angka dari pengguna untuk variabel tinggi.
4. Perhitungan BMI: Menghitung nilai BMI menggunakan rumus:  
$$\text{BMI} = \text{berat} / (\text{tinggi} * \text{tinggi})$$
  
Hasil perhitungan disimpan dalam variabel BMI.
5. Output mencetak hasil perhitungan BMI ke terminal dengan pesan "Hasil BMI Anda adalah: ".



## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {

    var totalBelanja float64

    var diskonPersen float64


    fmt.Print("Masukkan totalBelanja: ")

    fmt.Scan(&totalBelanja)

    fmt.Print("Masukkan diskonPersen: ")

    fmt.Scan(&diskonPersen)


    potonganHarga := totalBelanja * (diskonPersen / 100)

    hargaAkhir := totalBelanja - potonganHarga

    fmt.Println(int(hargaAkhir))

}
```

[illegible]

Program tersebut adalah sebuah kalkulator untuk menghitung harga akhir suatu belanjaan setelah dipotong diskon.

1. Program meminta pengguna memasukkan total belanja dan disimpan dalam variabel `totalBelanja`.
2. Program kemudian meminta pengguna memasukkan persentase diskon dan disimpan dalam variabel `diskonPersen`.
3. Program kemudian menghitung jumlah potongan harga dengan rumus:  

$$\text{potonganHarga} = \text{totalBelanja} * (\text{diskonPersen} / 100)$$
4. Program menghitung harga akhir dengan mengurangi total belanja dengan potongan harga:  $\text{hargaAkhir} = \text{totalBelanja} - \text{potonganHarga}$
5. Output program menampilkan nilai `hargaAkhir` ke terminal.

1. Program meminta pengguna memasukkan totalBelanja, kemudian pengguna memasukkan 265000
2. Program meminta pengguna memasukkan persentase diskon, kemudian pengguna memasukkan 30
3. Program akan menghitung potongan harga dengan rumus:  

$$\text{potonganHarga} = 265000 * (30 / 100) = 265000 \times 0.3 = 79500$$
 dan menghitung harga akhir dengan rumus:  $\text{hargaAkhir} = 265000 - 79500 = 185500$
4. Outputnya hargaAkhir **185500**.

## 2. Tugas 2

### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var BMI float64
    var tinggi float64

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

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

    var berat = BMI * (tinggi * tinggi)

    fmt.Println(math.Round(berat))
}
```

## Screenshoot program

The screenshot shows a Go program in a VS Code editor and its execution in a terminal. The program is located in `soal2modul4.go` and is in the `main` package. It imports `fmt` and `math` packages. The `main` function declares two variables, `BMI` and `tinggi`, both of type `float64`. It then prompts the user to enter BMI and height, reads the input, and calculates the weight using the formula  $berat = BMI * (tinggi * tinggi)$ . The result is rounded to the nearest integer using `math.Round` and printed.

```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var BMI float64
10    var tinggi float64
11
12    fmt.Print("Masukkan BMI: ")
13    fmt.Scan(&BMI)
14    fmt.Print("Masukkan tinggi: ")
15    fmt.Scan(&tinggi)
```

The terminal output shows the program being run with `go run soal2modul4.go`. The user enters `22.89` for BMI and `1.55` for height. The program outputs `55`.

```
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\internal\language\modul 4> go run soal2modul4.go
Masukkan BMI: 22.89
Masukkan tinggi: 1.55
55
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\internal\language\modul 4>
```

## Deskripsi program

program tersebut digunakan untuk menentukan berat badan seseorang dengan memasukkan BMI dan tinggi badan.

Cara Kerja:

1. Program meminta pengguna memasukkan nilai BMI ("Masukkan BMI: ") dan menyimpannya.
2. Program meminta pengguna memasukkan tinggi badan dalam meter ("Masukkan tinggi: ") dan menyimpannya.
3. Variabel berat dihitung menggunakan rumus:  $berat = BMI * (tinggi * tinggi)$
4. Nilai berat yang telah dihitung kemudian dibulatkan ke bilangan bulat terdekat menggunakan `math.Round(berat)`.
5. Hasil pembulatan tersebut kemudian dicetak (`fmt.Println`) ke terminal.

Contoh Masukkan:

1. Pengguna memasukkan BMI = 22.89
2. Pengguna memasukkan tinggi badan(m) = 1.55
3. Program akan menghitung dengan rumus:  
 $berat = BMI * (tinggi * tinggi)$   
 $berat = 22.89 \times (1.55 \times 1.55)$   
 $berat = 22.89 \times 2.4025$   
 $berat = 54.998025$
4. Nilai berat (54.998025) kemudian dibulatkan menggunakan `math.Round()` menjadi 55.
5. output 55.

### 3. Tugas 3

#### Source code

```
package main

import (
    "fmt"
    "math"
)

func hitungJarak(x1, y1, x2, y2 float64) float64 {
    jarakKuadrat := math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2)
    return math.Sqrt(jarakKuadrat)
}

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

    fmt.Println("Masukkan koordinat Titik A (x y):")
    fmt.Scan(&x1, &y1)
    fmt.Println("Masukkan koordinat Titik B (x y):")
    fmt.Scan(&x2, &y2)
    fmt.Println("Masukkan koordinat Titik C (x y):")
    fmt.Scan(&x3, &y3)

    sisiAB := hitungJarak(x1, y1, x2, y2)
    sisiBC := hitungJarak(x2, y2, x3, y3)
    sisiCA := hitungJarak(x3, y3, x1, y1)

    sisiTerpanjangSementara := math.Max(sisiAB, sisiBC)
    sisiTerpanjang := math.Max(sisiTerpanjangSementara,
sisiCA)

    fmt.Printf("\nPanjang sisi terpanjang adalah: %.2f\n",
sisiTerpanjang)
}
```

## Screenshoot program

```
soal3modul4.go > ...
14 func main() {
15     var x1, y1, x2, y2, x3, y3 float64
16
17     fmt.Println("Masukkan koordinat Titik A (x y):")
18     fmt.Scan(&x1, &y1)
19     fmt.Println("Masukkan koordinat Titik B (x y):")
20     fmt.Scan(&x2, &y2)
21     fmt.Println("Masukkan koordinat Titik C (x y):")
22     fmt.Scan(&x3, &y3)
23
24     sisiAB := hitungJarak(x1, y1, x2, y2)
```

```
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\internal\language\modul 4> go run soal3modul4.go
Masukkan koordinat Titik A (x y):
1 2
Masukkan koordinat Titik B (x y):
3 4
Masukkan koordinat Titik C (x y):
5 6

Panjang sisi terpanjang adalah: 5.66
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\internal\language\modul 4> 
```

NAMA : BENING PUTRI NARESWARI SUKARNO  
NIM : 109082500211  
KELAS : IF-13-02

## Deskripsi program

Program berfungsi sebagai kalkulator untuk menentukan panjang sisi terpanjang dari sebuah segitiga yang dibentuk oleh tiga titik koordinat A, B, dan C dalam sistem Kartesius 2 dimensi.

Cara Kerja:

1. Program meminta pengguna memasukkan enam nilai (koordinat x dan y untuk Titik A, B, dan C) dan menyimpannya dalam variabel.
2. Program menggunakan fungsi pembantu `hitungJarak` yang didasarkan pada Teorema Pythagoras untuk menghitung panjang ketiga sisi segitiga: Sisi AB, Sisi BC, dan Sisi CA.
3. Program menggunakan fungsi `math.Max` secara berulang untuk membandingkan ketiga panjang sisi tersebut, lalu memilih nilai yang terbesar sebagai sisi terpanjang, dan menampilkan hasilnya dengan dua angka di belakang koma.

Contoh Masukkan:

1. Input yang diberikan pengguna adalah koordinat untuk tiga titik: A (1, 2), B (3, 4), dan C (5, 6).

2. Perhitungan menggunakan koordinat tersebut menghasilkan panjang sisi CA sebagai yang terbesar 5.66.
3. Output yang ditampilkan program adalah 5.66, yang merupakan jarak lurus antara Titik C dan Titik A.