

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

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



**Disusun oleh:**

**Hassan Donny Darmawan**

**109082500030**

**S1IF-13-04**

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## Tugas Pendahuluan

**1. Pembuat soal :** Zhafif Iqbal Kurniawan - 109082500051

**Deskripsi:** buatlah sebuah program dengan input dalam satuan meter dan output satuan kilometer sampai milimeter

**Masukan:** satuan meter dengan bilangan rill

**Keluaran:** konversi satuan meter ke satuan kilometer sampai milimeter

Masukan	Satuan	Output
5	Km	0.50
	Hm	5.00
	Dam	50.00
	M	500
	Dm	5000
	Cm	50000
	mm	500000

### Source code

```
package main

import "fmt"

func main () {

    var km,hm,dam,m,dm,cm,mm float64

    fmt.Print("masukan nilai dalam satuan meter: ")

    fmt.Scanln(&m)

    km = m / 1000

    hm = m / 100

    dam = m / 10

    dm = m * 10

    cm = m * 100

    mm = m * 1000
```

```

    fmt.Printf("kilometer: %0.2f\n", km)

    fmt.Printf("hektometer: %0.2f\n", hm)

    fmt.Printf("deksameter: %0.2f\n", dam)

    fmt.Printf("meter: %0.0f\n", m)

    fmt.Printf("desimeter: %0.0f\n", dm)

    fmt.Printf("centimeter: %0.0f\n", cm)

    fmt.Printf("milimeter: %0.0f", mm)

}

```

## Screenshoot Program

The screenshot shows a Go IDE with a project named 'golang'. The Explorer panel on the left shows a directory structure with 'modul 2', 'modul 3', and 'modul 4'. Under 'modul 4', there are files 'guided1.go', 'guided2.go', 'guided3.go', 'tugas1.go', 'tugas2.go', 'tugas3.go', 'tugaspendahuluan1.go', and 'tugaspendahuluan2.go'. The file 'tugaspendahuluan1.go' is selected and open in the editor. The code in the editor is as follows:

```

1 package main
2 import "fmt"
3
4 func main() {
5     var km, hm, dam, m, dm, cm, mm float64
6     fmt.Print("masukan nilai dalam satuan meter: ")
7     fmt.Scanln(&m)
8     km = m / 1000
9     hm = m / 100
10    dam = m / 10
11    dm = m * 10
12    cm = m * 100
13    mm = m * 1000
14
15    fmt.Printf("kilometer: %0.2f\n", km)
16    fmt.Printf("hektometer: %0.2f\n", hm)
17    fmt.Printf("deksameter: %0.2f\n", dam)
18    fmt.Printf("meter: %0.0f\n", m)
19    fmt.Printf("desimeter: %0.0f\n", dm)
20    fmt.Printf("centimeter: %0.0f\n", cm)
21    fmt.Printf("milimeter: %0.0f", mm)
22 }

```

The OUTPUT panel at the bottom shows the following output:

```

desimeter: 5000
centimeter: 50000
milimeter: 500000
PS C:\Users\WYPE\AMD\OneDrive\Documents\golang> go run "c:\Users\WYPE\AMD\OneDrive\Documents\golang\modul 4\tugaspendahuluan1.go"
masukan nilai dalam satuan meter: 500
kilometer: 0.50
hektometer: 5.00
deksameter: 50.00
meter: 500
desimeter: 5000
centimeter: 50000
milimeter: 500000
PS C:\Users\WYPE\AMD\OneDrive\Documents\golang>

```

A small window titled 'Hassan Do' is also visible in the background, showing a name and ID.

## Deskripsi Program

Program ini merupakan program yang dirancang untuk konversi satuan meter ke satuan yang lain. Pengguna diminta menginputkan satuan meter ke dalam program yang nantinya akan di konversi olehs system dan akan ditampilkan kembali ke layar pengguna hasil dari konversinya.

**2. Pembuat soal :** Zhafif Iqbal Kurniawan - 109082500051

**Deskripsi:** buatlah sebuah program untuk mengurutkan angka dari kecil ke besar

**Masukan:** bilangan positif

**Keluaran:** bilangan yang sudah terurut

Input	output
18 7 10	7 10 18
21 3 2	2 3 21
12 25 24	12 24 25

**Source code**

```
package main

import

    "fmt"

func main () {

    var x,y,z int

    var temp int

    fmt.Print("masukan nilai 3 nilai: ")

    fmt.Scan(&x,&y,&z)

    if x>y {

        temp = x

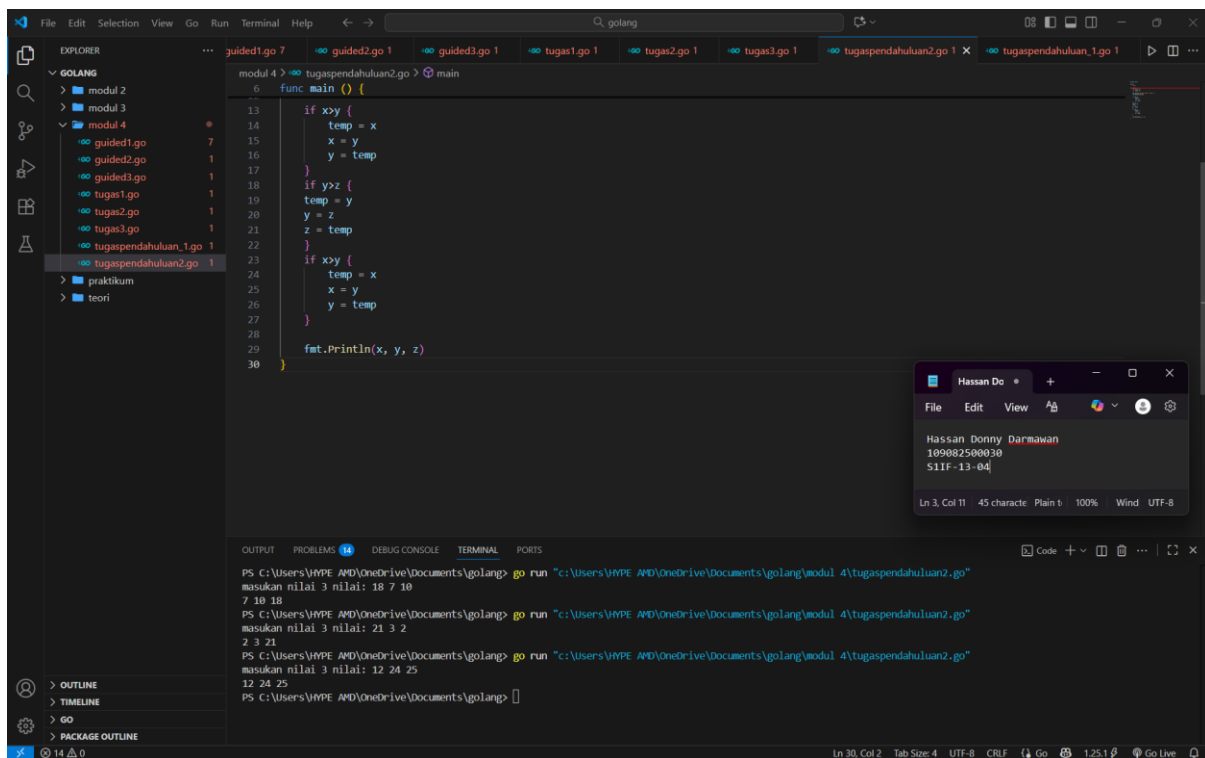
        x = y

        y = temp

    }
```

```
    if y>z {  
        temp = y  
        y = z  
        z = temp  
    }  
  
    if x>y {  
        temp = x  
        x = y  
        y = temp  
    }  
  
    fmt.Println(x, y, z)  
}
```

## Screenshoot program



The screenshot shows a Go IDE with a project named 'golang'. The Explorer panel on the left shows a directory structure with 'modul 4' containing 'tugaspendahuluan2.go'. The main editor displays the code for 'tugaspendahuluan2.go', which implements a bubble sort algorithm. The code is as follows:

```
6 func main() {  
13     if x > y {  
14         temp = x  
15         x = y  
16         y = temp  
17     }  
18     if y > z {  
19         temp = y  
20         y = z  
21         z = temp  
22     }  
23     if x > y {  
24         temp = x  
25         x = y  
26         y = temp  
27     }  
28     fmt.Println(x, y, z)  
29 }  
30 }
```

The OUTPUT panel at the bottom shows the results of running the program three times with different input values:

```
PS C:\Users\HYPE AND\OneDrive\Documents\golang> go run "C:\Users\HYPE AND\OneDrive\Documents\golang\modul 4\tugaspendahuluan2.go"  
masukan nilai 3 nilai: 18 7 10  
7 10 18  
PS C:\Users\HYPE AND\OneDrive\Documents\golang> go run "C:\Users\HYPE AND\OneDrive\Documents\golang\modul 4\tugaspendahuluan2.go"  
masukan nilai 3 nilai: 21 3 2  
2 3 21  
PS C:\Users\HYPE AND\OneDrive\Documents\golang> go run "C:\Users\HYPE AND\OneDrive\Documents\golang\modul 4\tugaspendahuluan2.go"  
masukan nilai 3 nilai: 12 24 25  
12 24 25  
PS C:\Users\HYPE AND\OneDrive\Documents\golang>
```

## Deskripsi Program

Program ini merupakan program mengurutkan 3 nilai yang diinput secara acak, system ini bekerja dengan membaca 3 nilai tersebut lalu dibandingkan jika tidak sesuai dengan yang diminta maka ada proses swap berulang ulang sampai dapat hasil yang diinginkan. Untuk outputnya merupakan 3 nilai yang dinput sudah terurut dari nilai terkecil ke besar.

## LATIHAN KELAS – GUIDED

### 1. Guided 1 Source Code

```
package main

import "fmt"

func main () {

var detik, jam, menit int

fmt.Print("masukan Print: ")

fmt.Scan(&detik)

jam = detik / 3600

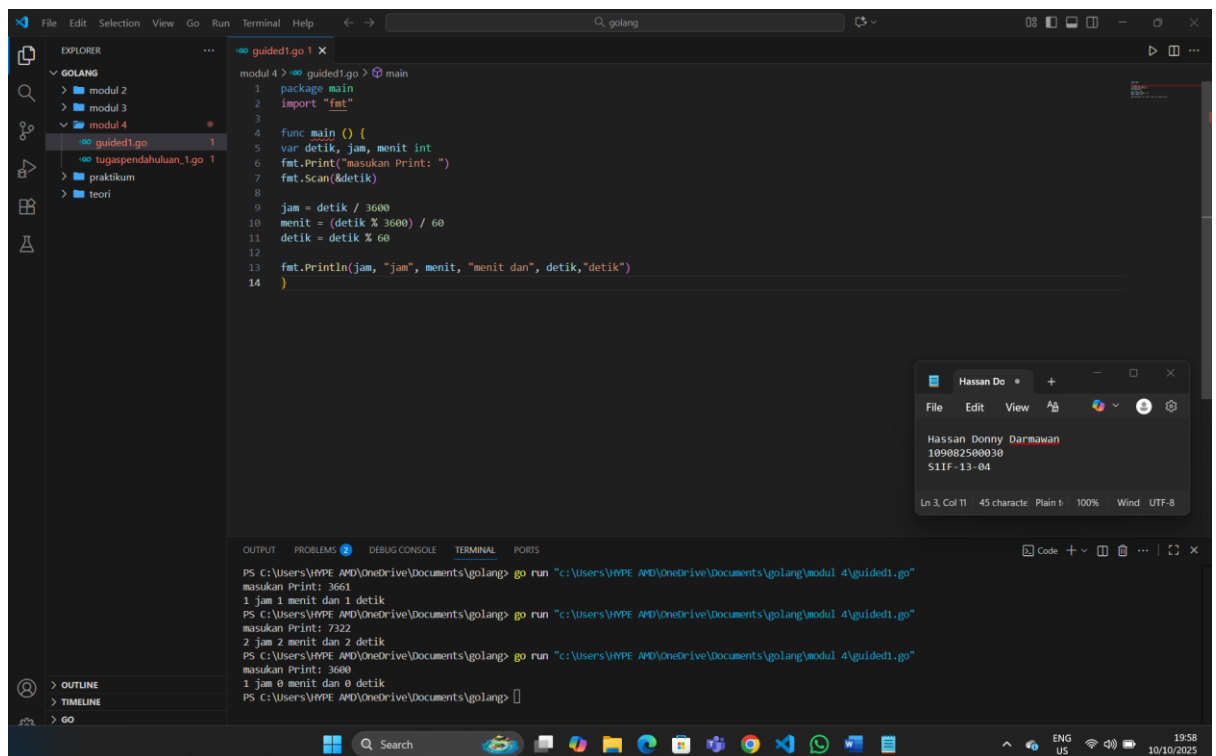
menit = (detik % 3600) / 60

detik = detik % 60

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

}
```

## Screenshoot program



## Deskripsi program

Pada guided 1 ini merupakan program yang dibuat untuk menghitung dari detik ke jam dan juga menit. Untuk proses awal program ini para user diminta menginputkan angka bilangan bulat satuan detik yang nantinya akan disimpan pada variable detik, untuk konversi dari detik ke jam maka nilai variable detik dibagi dengan 3600, sedangkan untuk menit yaitu variable x dimodulus 3600 karena kita ingin sisa dari jam itu berapa detik untuk menentukan menit, setelah itu di bagi 60. Untuk detik sendiri langsung dimodulus 60 karena 3600 habis dibagi 60 dan 60 habis dibagi 60, yang nantinya jika variable detik dimodulus 60 akan mendapatkan sisa detiknya saja. Diakhir program akan menampilkan kembali hasil dari proses konversi di layar user.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

var bilangan, d1, d2, d3 int

fmt.Print("masukan bilangan 3 digit: ")

fmt.Scan(&bilangan)
```



```

d1 = bilangan / 100

d2 = bilangan % 100 / 10

d3 = bilangan % 100 % 10

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

}

```

## Screenshoot program

The screenshot shows a Go IDE with the following code in `guided2.go`:

```

1 package main
2 import "fmt"
3
4 func main() {
5     var bilangan, d1, d2, d3 int
6
7     fmt.Print("masukan bilangan 3 digit: ")
8     fmt.Scan(&bilangan)
9
10    d1 = bilangan / 100
11    d2 = bilangan % 100 / 10
12    d3 = bilangan % 100 % 10
13
14    fmt.Println(d1 <= d2 && d2 <= d3)
15 }

```

The terminal output shows three test cases:

```

PS C:\Users\WYPE AND\OneDrive\Documents\golang> go run "c:\Users\WYPE AND\OneDrive\Documents\golang\modul 4\guided2.go"
masukan bilangan 3 digit: 362
false
PS C:\Users\WYPE AND\OneDrive\Documents\golang> go run "c:\Users\WYPE AND\OneDrive\Documents\golang\modul 4\guided2.go"
masukan bilangan 3 digit: 256
true
PS C:\Users\WYPE AND\OneDrive\Documents\golang> go run "c:\Users\WYPE AND\OneDrive\Documents\golang\modul 4\guided2.go"
masukan bilangan 3 digit: 189
true
PS C:\Users\WYPE AND\OneDrive\Documents\golang>

```

## Deskripsi program

Program ini untuk menentukan apakah 3 digit yang dimasukan membesar atau tidak, dan apakah digit itu tetap sama dengan dari kiri ke kanan.

Program meminta user input 3 digit dimana formatnya d1 sebagai ratusan, d2 sebagai puluhan, d3 sebagai satuan yang nantinya akan diproses untuk perbandingan antara d1,d2,d3 dengan  $(d1 \leq d2 \ \&\& \ d2 \leq d3)$  kedua perbandingan harus menghasilkan true untuk mendapatkan ouput true, dikarenakan memakai operator `&&` dan yang dimana dikatakan true jika keduanya benar. Diakhir program akan menampilkan hasil dari perbandingan di layar user.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var beratBadan, tinggiBadan, bmi float64

    fmt.Print("masukan berat badan dan tinggi badan: ")

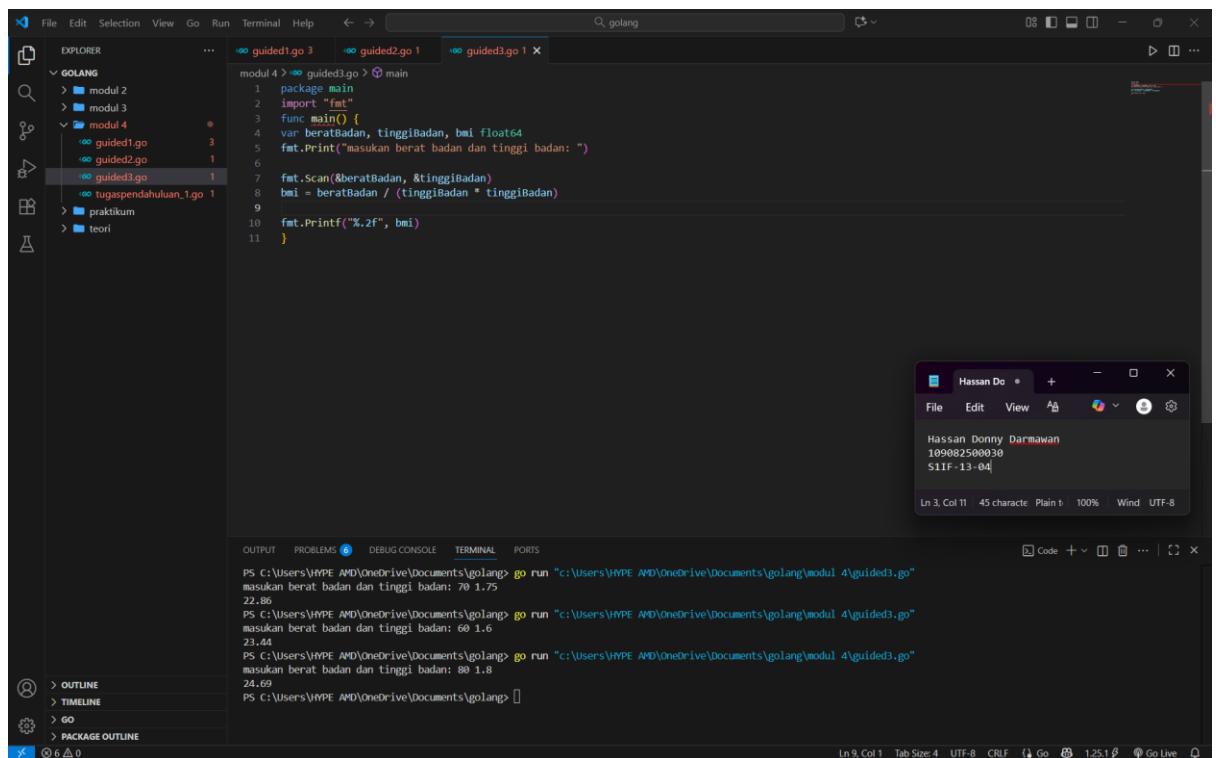
    fmt.Scan(&beratBadan, &tinggiBadan)

    bmi = beratBadan / (tinggiBadan * tinggiBadan)

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

}
```

#### Screenshoot program



The screenshot displays a Go IDE interface. The Explorer panel on the left shows a project structure with folders 'modul 2', 'modul 3', and 'modul 4'. Inside 'modul 4', there are files 'guided1.go', 'guided2.go', and 'guided3.go'. The 'guided3.go' file is selected and its content is shown in the main editor. The code is identical to the one provided in the 'Source Code' section. The terminal at the bottom shows the execution of the program with three different inputs: 70 1.75, 60 1.6, and 80 1.8. The output for each run shows the calculated BMI value rounded to two decimal places. A small window titled 'Hassan Do' is also visible in the background.

```
modul 4 > guided3.go > main
1 package main
2 import "fmt"
3 func main() {
4     var beratBadan, tinggiBadan, bmi float64
5     fmt.Print("masukan berat badan dan tinggi badan: ")
6
7     fmt.Scan(&beratBadan, &tinggiBadan)
8     bmi = beratBadan / (tinggiBadan * tinggiBadan)
9
10    fmt.Printf("%.2f", bmi)
11 }
```

OUTPUT

```
PS C:\Users\HYPE\AMD\OneDrive\Documents\golang> go run "c:\Users\HYPE\AMD\OneDrive\Documents\golang\modul 4\guided3.go"
masukan berat badan dan tinggi badan: 70 1.75
22.86
PS C:\Users\HYPE\AMD\OneDrive\Documents\golang> go run "c:\Users\HYPE\AMD\OneDrive\Documents\golang\modul 4\guided3.go"
masukan berat badan dan tinggi badan: 60 1.6
23.44
PS C:\Users\HYPE\AMD\OneDrive\Documents\golang> go run "c:\Users\HYPE\AMD\OneDrive\Documents\golang\modul 4\guided3.go"
masukan berat badan dan tinggi badan: 80 1.8
24.69
PS C:\Users\HYPE\AMD\OneDrive\Documents\golang>
```

### Deskripsi program

Program ini dibuat untuk menghitung nilai bmi, diawal program ini user diminta masukan nilai berat badan dan tinggi badan yang dimana kedua nilai itu akan disimpan pada variabelnya tersendiri. Di system akan di proses perhitungan dengan menggunakan nilai yang disimpan variable untuk mendapatkan hasil bmi. Diakhir program akan menampilkan dilayar user hasil dari perhitungan dengan format 2 angka dibelakang koma.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main () {

    var harga,diskon int

    fmt.Print("masukan total harga: ")

    fmt.Scanln(&harga)

    fmt.Print("masukan diskon: ")

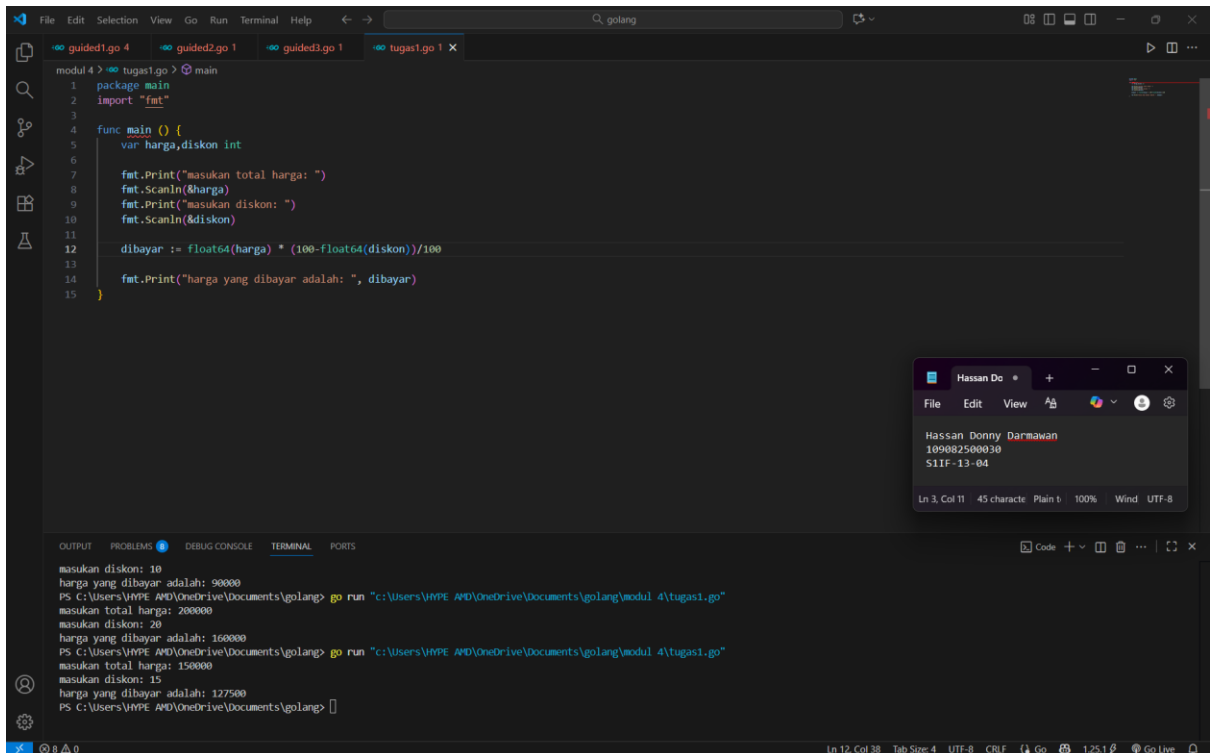
    fmt.Scanln(&diskon)

    dibayar := float64(harga) * (100-float64(diskon))/100

    fmt.Print("harga yang dibayar adalah: ", dibayar)

}
```

## Screenshoot program



```
modul 4 > tugasi.go > main
1 package main
2 import "fmt"
3
4 func main () {
5     var harga,diskon int
6
7     fmt.Print("masukan total harga: ")
8     fmt.ScanIn(&harga)
9     fmt.Print("masukan diskon: ")
10    fmt.ScanIn(&diskon)
11
12    dibayar := float64(harga) * (100-float64(diskon))/100
13
14    fmt.Print("harga yang dibayar adalah: ", dibayar)
15 }
```

OUTPUT PROBLEMS DEBUG CONSOLE TERMINAL PORTS

```
masukan diskon: 10
harga yang dibayar adalah: 90000
PS C:\Users\HYPE-AMD\OneDrive\Documents\golang> go run "c:\Users\HYPE-AMD\OneDrive\Documents\golang\modul 4\tugasi.go"
masukan total harga: 200000
masukan diskon: 20
harga yang dibayar adalah: 160000
PS C:\Users\HYPE-AMD\OneDrive\Documents\golang> go run "c:\Users\HYPE-AMD\OneDrive\Documents\golang\modul 4\tugasi.go"
masukan total harga: 150000
masukan diskon: 15
harga yang dibayar adalah: 127500
PS C:\Users\HYPE-AMD\OneDrive\Documents\golang>
```

## Deskripsi program

Program ini meminta user menginputkan 2 nilai berupa total harga dan persenan diskon, yang nantinya masing masing nilai akan disimpan pada variabelnya sendiri. Setelah itu sistem akan menghitung berapa yang harus dibayarkan pembeli dengan menggunakan nilai yang disimpan di variable, setelah selesai perhitungan hasil nya akan disimpan pada variable dibayar. Terakhir menampilkan hasil di layar user dengan memanggil variable dibayar pada `fmt.Print()`

## 2. Tugas 2

### Source code

```
package main

import (

    "fmt"

    "math"

)

func main () {

    var bmi,tinggi float64
```

```

    fmt.Print("masukan nilai bmi & tinggi badan (m): ")

    fmt.Scan(&bmi,&tinggi)

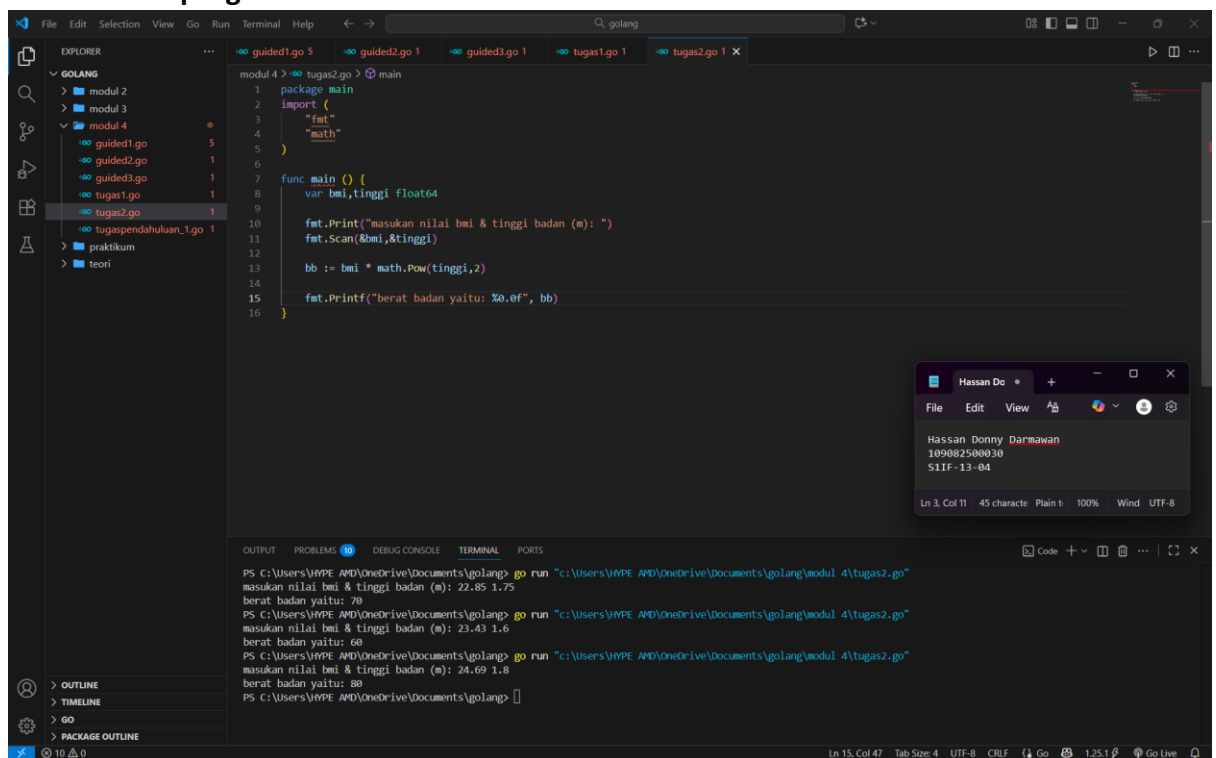
    bb := bmi * math.Pow(tinggi,2)

    fmt.Printf("berat badan yaitu: %0.0f", bb)

}

```

## Screenshoot program



## Deskripsi program

program ini digunakan untuk menghitung bmi dan tinggi badan untuk mendapatkan nilai berat badan, diawal program user diminta untuk menginputkan nilai bmi dan tinggi badan yang nantinya akan disimpan pada variable masing masing, setelah itu memasuki proses perhitungan dengan menggunakan rumus  $bb = bmi * \text{tinggi}^2$ , yang dimana hasil perhitungan akan disimpan pada variable bb. Diakhir program

menggunakan `fmt.Printf` untuk menampilkan hasil dengan menyebut variable `bb` dengan format tidak ada angka di belakang koma.

### 3. Tugas 3

#### Source code

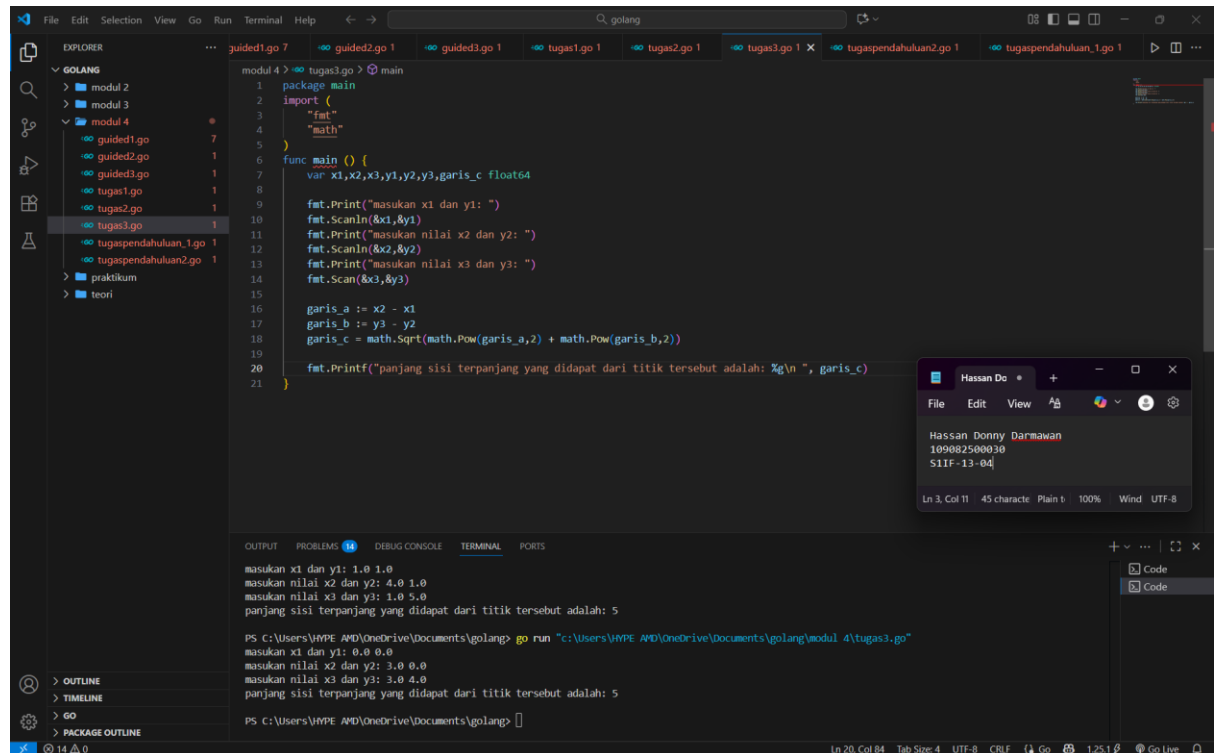
```
package main
import (
    "fmt"
    "math"
)
func main () {
    var x1,x2,x3,y1,y2,y3,garis_c float64

    fmt.Print("masukan x1 dan y1: ")
    fmt.Scanln(&x1,&y1)
    fmt.Print("masukan nilai x2 dan y2: ")
    fmt.Scanln(&x2,&y2)
    fmt.Print("masukan nilai x3 dan y3: ")
    fmt.Scan(&x3,&y3)

    garis_a := x2 - x1
    garis_b := y3 - y2
    garis_c = math.Sqrt(math.Pow(garis_a,2) +
    math.Pow(garis_b,2))

    fmt.Printf("panjang sisi terpanjang yang didapat
    dari titik tersebut adalah: %g\n ", garis_c)
}
```

## Screenshoot program



The screenshot shows a Go IDE with a project named 'golang'. The file explorer on the left shows a directory structure with 'modul 4' containing 'tugas3.go'. The editor displays the code for 'tugas3.go', which is a Go program that calculates the hypotenuse of a right triangle. The code uses the Pythagorean theorem:  $c = \sqrt{a^2 + b^2}$ . It prompts the user to input three coordinates (x1, y1, x2, y2, x3, y3) and calculates the distance between the first two points (x1, y1) and (x2, y2) to find the hypotenuse. The output shows the calculated hypotenuse for the input coordinates (1.0, 1.0) and (4.0, 1.0), which is 5.0.

```
modul 4 > tugas3.go > main
1 package main
2 import (
3     "fmt"
4     "math"
5 )
6 func main() {
7     var x1,x2,x3,y1,y2,y3,garis_c float64
8
9     fmt.Print("masukan x1 dan y1: ")
10    fmt.Scanln(&x1,&y1)
11    fmt.Print("masukan nilai x2 dan y2: ")
12    fmt.Scanln(&x2,&y2)
13    fmt.Print("masukan nilai x3 dan y3: ")
14    fmt.Scanln(&x3,&y3)
15
16    garis_a := x2 - x1
17    garis_b := y2 - y1
18    garis_c = math.Sqrt(math.Pow(garis_a,2) + math.Pow(garis_b,2))
19
20    fmt.Printf("panjang sisi terpanjang yang didapat dari titik tersebut adalah: %g\n ", garis_c)
21 }
```

OUTPUT

```
masukan x1 dan y1: 1.0 1.0
masukan nilai x2 dan y2: 4.0 1.0
masukan nilai x3 dan y3: 1.0 5.0
panjang sisi terpanjang yang didapat dari titik tersebut adalah: 5

PS C:\Users\HYPE AND\OneDrive\Documents\golang> go run "c:\Users\HYPE AND\OneDrive\Documents\golang\modul 4\tugas3.go"
masukan x1 dan y1: 0.0 0.0
masukan nilai x2 dan y2: 3.0 0.0
masukan nilai x3 dan y3: 3.0 4.0
panjang sisi terpanjang yang didapat dari titik tersebut adalah: 5

PS C:\Users\HYPE AND\OneDrive\Documents\golang>
```

## Deskripsi program

Program ini merupakan untuk mencari sisi terpanjang dari segitiga siku siku dengan phytagoras, segitiga ini dibentuk oleh titik kordinat kartesius yang dinput oleh pengguna. Dimana pengguna awalnya menginputkan sebuah titik yang akan diproses oleh system untuk mencari tahu Panjang dari setiap sisi yang nantinya akan digunakan untuk mencari sisi terpanjang dengan menggunakan rumus phytagoras yang akhirnya akan menampilkan hasil sisi terpanjang dari segitiga tersebut pada layar pengguna.