

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

**MODUL 4**  
**I/O, Tipe Data & Variabel**



**Disusun oleh:**

**Nama:Nafisah Salsabila**

**NIM:109082500063**

**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 detik, jam, menit int
    fmt.Scan(&detik)

    jam = detik / 3600
    menit = (detik % 3600) / 60
    detik = detik % 60

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

Output:

Input: 3661

Output: 1 jam 1 menit dan 1 detik

Input: 7322

Output: 2 jam 2 menit dan 2 detik

Input: 3600

Output: 1 jam 0 menit dan 0 detik

#### Screenshoot program

The screenshot shows the Visual Studio Code editor with a Go file named `guide1.go` open. The code is as follows:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var detik, jam, menit int
7     fmt.Scan(&detik)
8
9     jam = detik / 3600
10    menit = (detik % 3600) / 60
11    detik = detik % 60
12
13    fmt.Println(jam, "jam", menit, "menit dan", detik, "detik")
14 }
15
16
```

The Explorer sidebar on the left shows a project named "ALGORITMA DAN PEMOGRAMAN" with several files, including `guide1.go`. The bottom status bar indicates the file is at line 16, column 1, with a tab size of 4, UTF-8 encoding, and CRLF line endings.

The screenshot shows the Visual Studio Code editor with the same Go file `guide1.go` open. The terminal at the bottom displays the output of the program after running the command `go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide1.go"` multiple times:

```
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide1.go"
3661
3661
1 jam 1 menit dan 1 detik
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide1.go"
7322
7322
2 jam 2 menit dan 2 detik
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide1.go"
7322
7322
2 jam 2 menit dan 2 detik
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide1.go"
3600go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide1.go"
3600go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide1.go"
1 jam 0 menit dan 0 detik
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN>
```

The Explorer sidebar on the left shows the same project structure as the first screenshot. The bottom status bar indicates the file is at line 16, column 1, with a tab size of 4, UTF-8 encoding, and CRLF line endings.

## Deskripsi program

Program ini digunakan untuk mengonversi waktu dalam satuan detik menjadi jam, menit, dan detik.

Pengguna memasukkan satu bilangan bulat positif yang merepresentasikan jumlah detik. Program kemudian menghitung berapa jam, menit, dan detik yang setara dengan nilai tersebut.

Proses dilakukan dengan pembagian dan sisa hasil bagi menggunakan operator aritmetika (/ dan %).

Program ini bermanfaat untuk memahami cara kerja operator pembagian dan modulus, serta penggunaan tipe data integer dan fungsi input-output di bahasa Go.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3 int
    fmt.Scan(&bilangan)

    d1 = bilangan / 100
    d2 = (bilangan / 10) % 10
    d3 = bilangan % 10

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

Output:

Input: 362

Output: false

Input: 256

Output: true

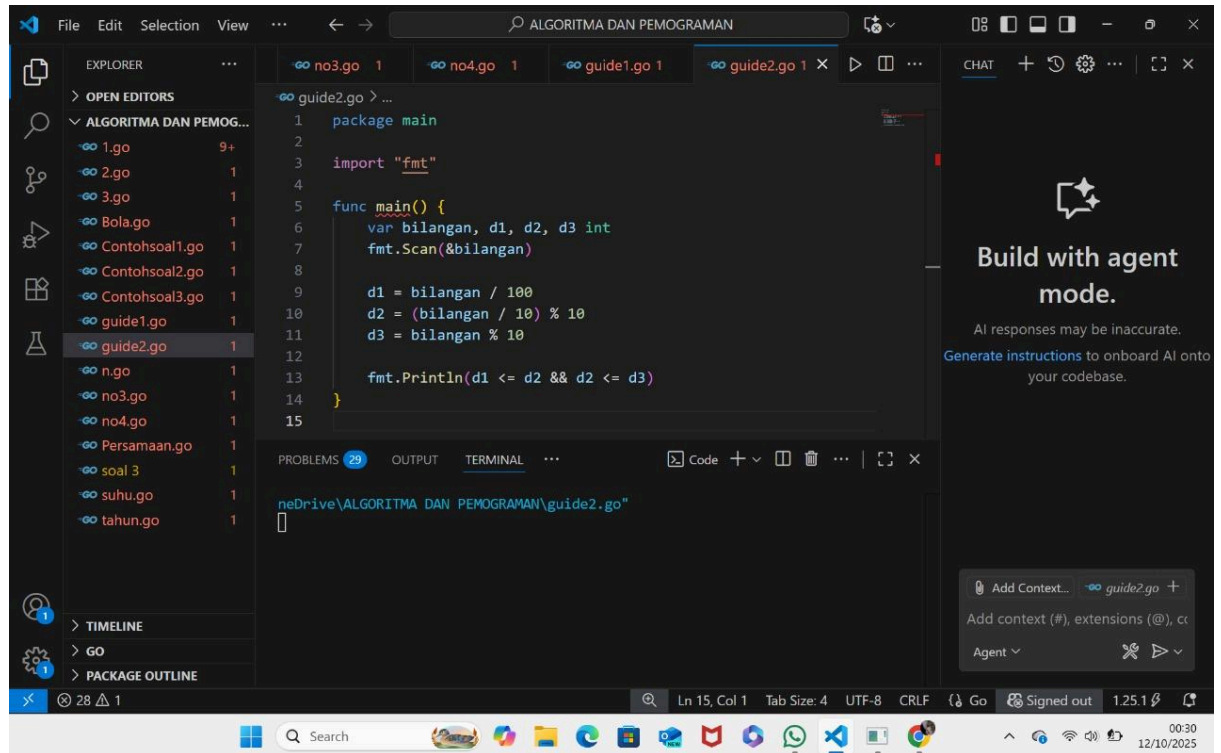
Input: 189

Output: true

Input: 555

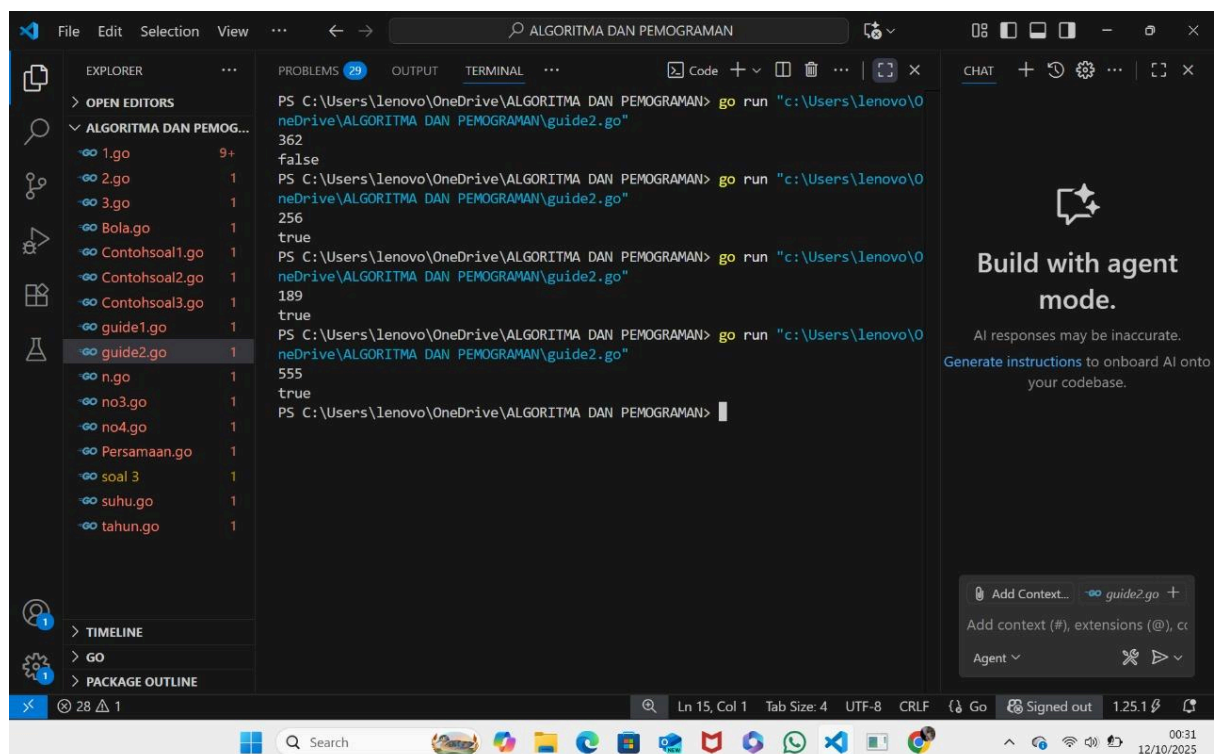
Output: true

## Screenshoot program



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project named 'ALGORITMA DAN PEMOGRAMAN'. The file 'guide2.go' is selected and its code is visible in the editor. The code defines a main function that takes an integer 'bilangan' and calculates three values: 'd1' (bilangan / 100), 'd2' ((bilangan / 10) % 10), and 'd3' (bilangan % 10). It then checks if 'd1 <= d2 && d2 <= d3' and prints the result. The terminal at the bottom is empty, and the chat panel on the right shows a 'Build with agent mode.' message.

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



The screenshot shows the Visual Studio Code editor with the same project. The terminal at the bottom now displays the output of running the program. The prompt 'PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN>' is followed by the command 'go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide2.go"'. The output shows the input '362' and the result 'false'. The next prompt shows the input '256' and the result 'true'. The third prompt shows the input '189' and the result 'true'. The fourth prompt shows the input '555' and the result 'true'. The chat panel on the right remains the same.

```
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide2.go"
362
false
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide2.go"
256
true
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide2.go"
189
true
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\guide2.go"
555
true
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN>
```

### Deskripsi program

Program ini berfungsi untuk mengecek apakah tiga digit dari sebuah bilangan tiga angka tersusun secara terurut membesar atau tidak.

Pengguna memasukkan satu bilangan bulat positif dengan tiga digit (antara 100 hingga 999).

Program kemudian memisahkan digit satuan, puluhan, dan ratusan menggunakan operasi pembagian dan modulus, lalu membandingkan urutannya.

Jika digit pertama  $\leq$  digit kedua dan digit kedua  $\leq$  digit ketiga, maka hasilnya true, sedangkan jika tidak berurutan membesar, hasilnya false.

Program ini melatih logika pemecahan bilangan menjadi digit-digit penyusunnya serta penggunaan operator logika pada 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\n", bmi)
}
```

Output:

Input: 70 1.75

Output: 22.85

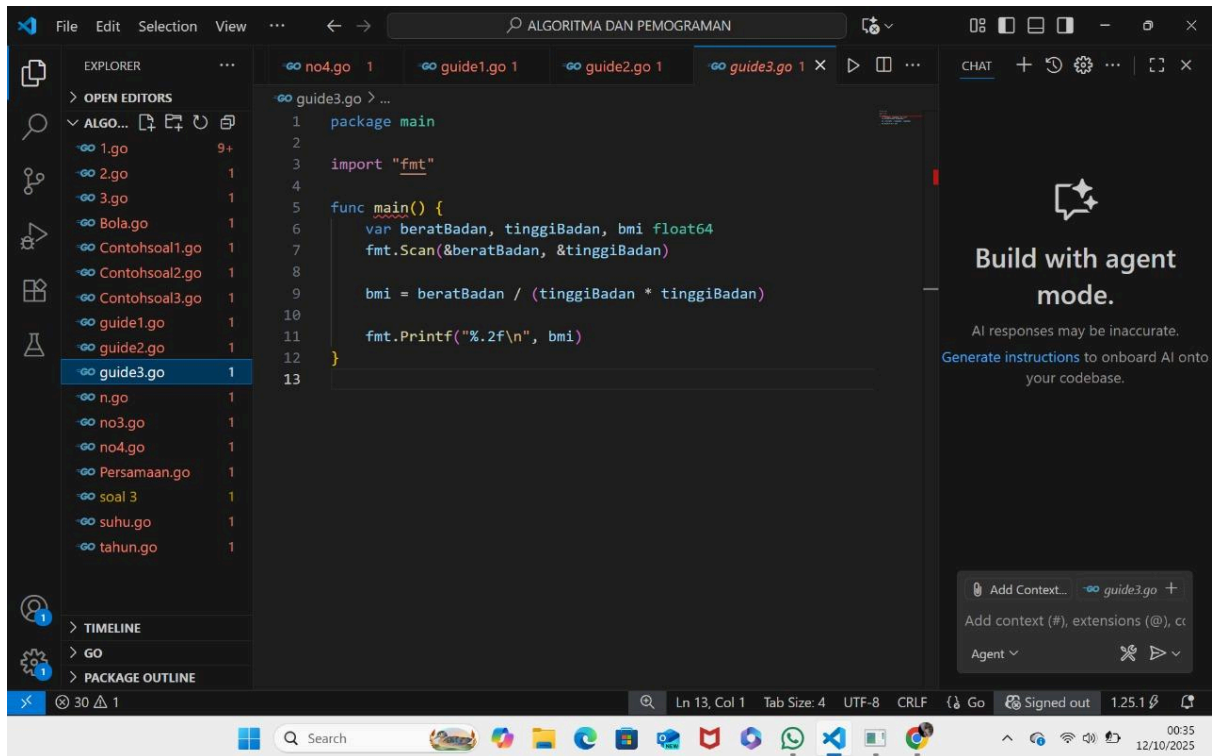
Input: 60 1.6

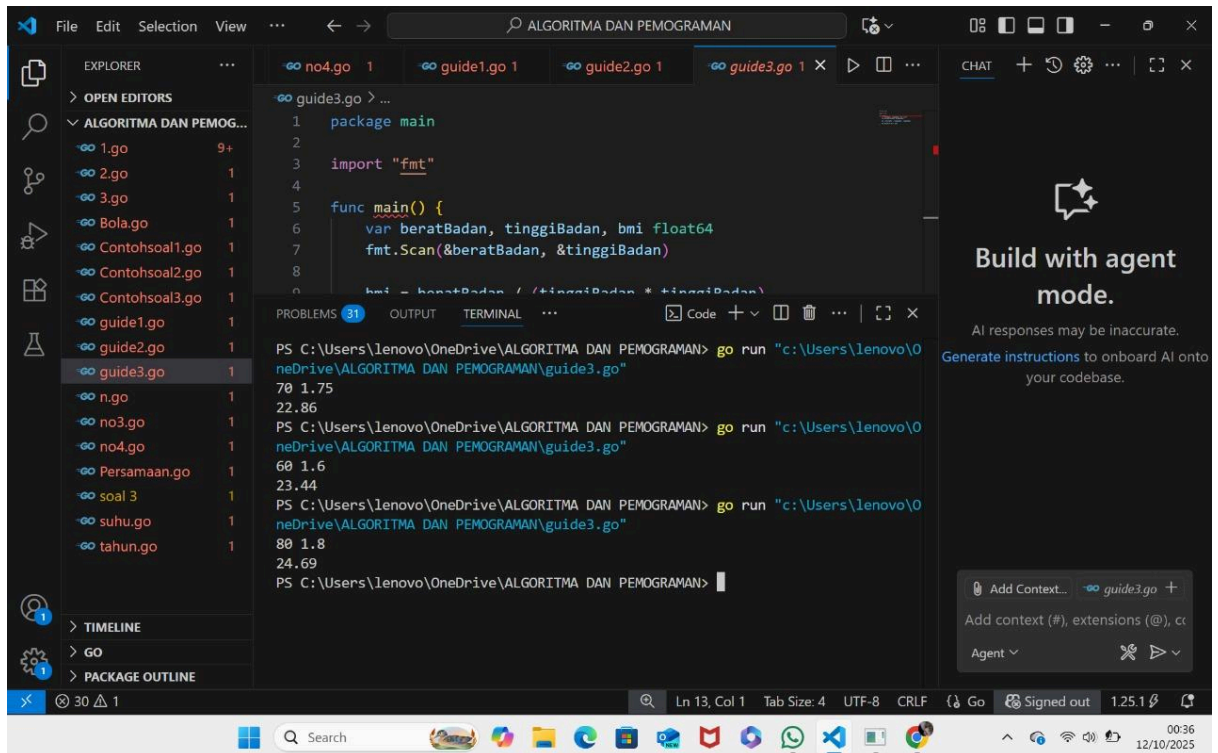
Output: 23.43

Input: 80 1.8

Output: 24.69

## Screenshoot program





## Deskripsi program

Program ini digunakan untuk menghitung nilai BMI (Body Mass Index) seseorang berdasarkan berat badan (kg) dan tinggi badan (m).

BMI dihitung dengan rumus:

$$\text{BMI} = \frac{\text{Berat \, Badan}}{(\text{Tinggi \, Badan})^2}$$

Pengguna memasukkan dua bilangan real (desimal) berupa berat badan dan tinggi badan.

Program kemudian menghitung nilai BMI dan menampilkannya dengan dua angka di belakang koma menggunakan `fmt.Printf("%.2f")`.

Program ini bertujuan untuk memperkenalkan penggunaan tipe data float, operasi aritmetika desimal, dan format output presisi di bahasa Go.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"
```



```
func main() {  
    var hargaAwal, diskon, total int  
    fmt.Scan(&hargaAwal)  
    fmt.Scan(&diskon)  
  
    total = hargaAwal - (hargaAwal * diskon / 100)  
  
    fmt.Println(total)  
}
```

Output:

Input:

100000

10

Output:

90000

Input:

200000

20

Output:

160000

Input:

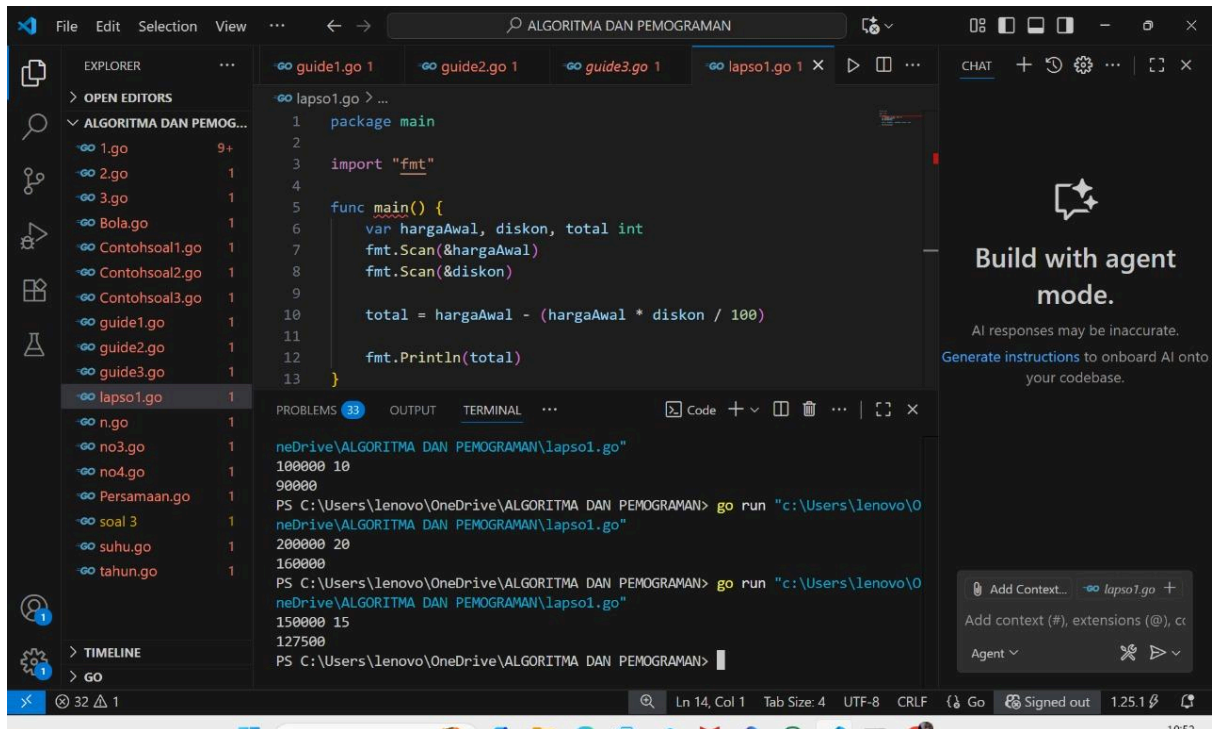
150000

15

Output:

127500

**Screenshoot program**



### Deskripsi program

Program ini digunakan untuk menghitung jarak antara dua titik di bidang kartesius  $(x_1, y_1)$  dan  $(x_2, y_2)$ .

Rumus yang digunakan adalah:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Tujuan:

Mengetahui jarak antara dua titik berdasarkan koordinat yang dimasukkan.

```
Package main
```

```
import "fmt"
```

```
func main() {
```

```
    var bmi, tinggiBadan, beratBadan float64
```

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

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

```
fmt.Printf("%.0f\n", beratBadan)
```

```
}
```

Output:

Input:

22.85 1.75

Output:

70

Input:

23.43 1.6

Output:

60

Input:

24.69 1.8

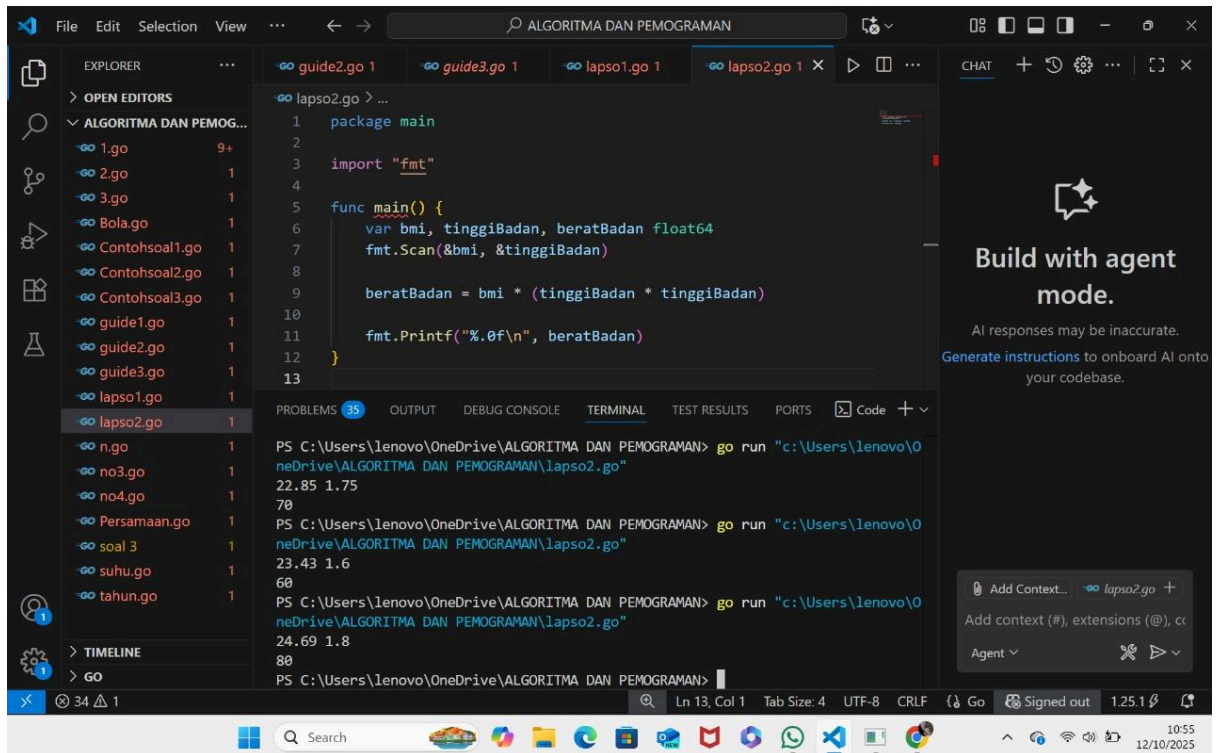
Output:

80

## Tugas 2

### Source code

### Screenshoot program



### Deskripsi program

Program ini digunakan untuk menghitung luas dan keliling segitiga berdasarkan tiga titik koordinat:  $(x_1, y_1)$ ,  $(x_2, y_2)$ , dan  $(x_3, y_3)$ .

Program menghitung tiga sisi segitiga menggunakan rumus jarak dua titik, lalu:

$$\text{Keliling} = a + b + c$$

$$\text{Luas} = \sqrt{s \times (s - a) \times (s - b) \times (s - c)} \text{ dengan } s = \text{keliling} / 2 \text{ (rumus Heron)}$$

Tujuan:

Menentukan luas dan keliling segitiga dari koordinat tiga titik di bidang datar.

## 2. Tugas 3

### Source code

```
package main

import (
    "fmt"
    "math"
)

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

```

// Input tiga titik (A, B, C)
fmt.Scan(&x1, &y1)
fmt.Scan(&x2, &y2)
fmt.Scan(&x3, &y3)

// Hitung 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(x3-x1, 2) + math.Pow(y3-y1, 2))

// Tentukan sisi terpanjang
maxSide := math.Max(AB, math.Max(BC, CA))

// Tampilkan hasil dengan dua angka di belakang koma
fmt.Printf("%.2f\n", maxSide)
}

```

Output:

```

1.0 1.0
4.0 1.0
1.0 5.0

```

4.47 (5.00 di terminal)

```

0.0 0.0
3.0 0.0
3.0 4.0

```

5.00

**Screenshoot program**

The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project named 'ALGORITMA DAN PEMOGRAMAN'. The 'lapso3.go' file is selected and its code is visible in the main editor. The code is a Go program that calculates the longest side of a triangle given three points (A, B, C). It uses the Pythagorean theorem to calculate the lengths of the sides AB, BC, and CA, and then determines the maximum of these three lengths.

```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var x1, y1, x2, y2, x3, y3 float64
10
11     // Input tiga titik (A, B, C)
12     fmt.Scan(&x1, &y1)
13     fmt.Scan(&x2, &y2)
14     fmt.Scan(&x3, &y3)
15
16     // Hitung panjang sisi-sisi segitiga
17     AB := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
18     BC := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
19     CA := math.Sqrt(math.Pow(x3-x1, 2) + math.Pow(y3-y1, 2))
20
21     // Tentukan sisi terpanjang
22     maxSide := math.Max(AB, math.Max(BC, CA))
23
24     // Tampilkan hasil dengan dua angka di belakang koma
25     fmt.Printf("%.2f\n", maxSide)
26 }
27
```

The screenshot shows the same Visual Studio Code editor with the 'lapso3.go' file selected. The terminal window at the bottom displays the output of the program after running the command 'go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\lapso3.go"'. The output shows the calculated longest side of the triangle for three different sets of input coordinates.

```
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\lapso3.go"
1.0 1.0
4.0 1.0
1.0 5.0
5.00
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\lapso3.go"
1.0 1.0
4.0 1.0
1.0 5.0
5.00
PS C:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN> go run "c:\Users\lenovo\OneDrive\ALGORITMA DAN PEMOGRAMAN\lapso3.go"
1.0 1.0
4.0 1.0
1.0 5.0
5.00
```

## Deskripsi program

Program ini digunakan untuk menentukan sisi terpanjang dari sebuah segitiga berdasarkan koordinat ketiga titiknya (A, B, dan C).

Pengguna memasukkan enam buah nilai berupa koordinat titik:

Titik A (x1, y1)

Titik B (x2, y2)

Titik C (x3, y3)

Selanjutnya program akan menghitung panjang masing-masing sisi menggunakan rumus jarak dua titik:

$$\text{Jarak} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Program menghitung tiga sisi:

AB = jarak antara titik A dan B

BC = jarak antara titik B dan C

CA = jarak antara titik C dan A

Kemudian program membandingkan ketiga sisi tersebut dan menentukan nilai terbesar sebagai sisi terpanjang.

Hasil akhirnya ditampilkan dengan dua angka di belakang koma.

## Tugas Pendahuluan

Soal:

1. Jelaskan perbedaan `fmt.Print()`, `fmt.Println()`, dan `fmt.Printf()` di Go!
2. Jelaskan penggunaan tipe data `int`, `float64`, `bool`, dan `string` di Go!
3. Bagaimana Cara mendeklarasikan variabel dengan kata kunci `var` dan dengan acara singkat? Berikan contoh! Hint `":="`
4. Apa perbedaan antara operator `==` dan `=` dalam bahasa Go?
5. Buatlah program Go sederhana untuk meminta input nama kalian, lalu menampilkan nama.

### Jawaban:

1.

- `fmt.Print()` adalah untuk mencetak teks ke layar tanpa menambah baris baru.

→ Contoh:

Go:

```
fmt.Print("Halo ")  
fmt.Print("Billa")
```

Output:

Halo Billa

- `fmt.Println()` → mencetak teks lalu otomatis pindah baris setelahnya.

→ Contoh:

Go:

```
fmt.Println("Halo")  
fmt.Println("Billa")
```



Output:

Halo  
Billa

- `fmt.Println()` adalah mencetak teks lalu otomatis pindah baris setelahnya.

→ Contoh:

Go:

```
fmt.Printf("Nilai saya: %.2f", 90.555)
```

Output:

Nilai saya: 90.56

2. `int` → untuk bilangan bulat, misalnya 10, -7, 0.

`float64` → untuk bilangan desimal, misalnya 3.14, 5.0.

`bool` → untuk nilai benar/salah, yaitu `true` atau `false`.

`string` → untuk teks, misalnya "Halo Dunia".

Contoh:

```
var umur int = 18  
var tinggi float64 = 168.5  
var sehat bool = true  
var nama string = "Billa"
```

### 3. Dengan var:

```
var nama string = "Billa"  
var umur int = 18
```

Dengan cara singkat (:=):

```
nama := "Billa"  
umur := 18
```

Ini otomatis menentukan tipe data berdasarkan nilainya (string, int, dll).

### 4. = untuk memberikan nilai (assignment) ke variabel.

Contoh: x = 5

== untuk membandingkan dua nilai.

Contoh: x == 5 (hasilnya true kalau x memang 5)

### 5. Program sederhana untuk memasukkan nama:

```
package main  
import "fmt"  
  
func main() {  
    var nama string  
    fmt.Print("Masukkan nama kamu: ")  
    fmt.Scan(&nama)  
    fmt.Println("Halo,", nama, "selamat datang di Golang!")  
}
```

Output:  
Masukkan nama kamu: Billa

Halo, Billa selamat datang di Golang