

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

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



Disusun oleh:

Farrasya Lazuardi

109082500040

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 angka dalam detik :")

    fmt.Scanln(&detik)

    jam = detik / 3600

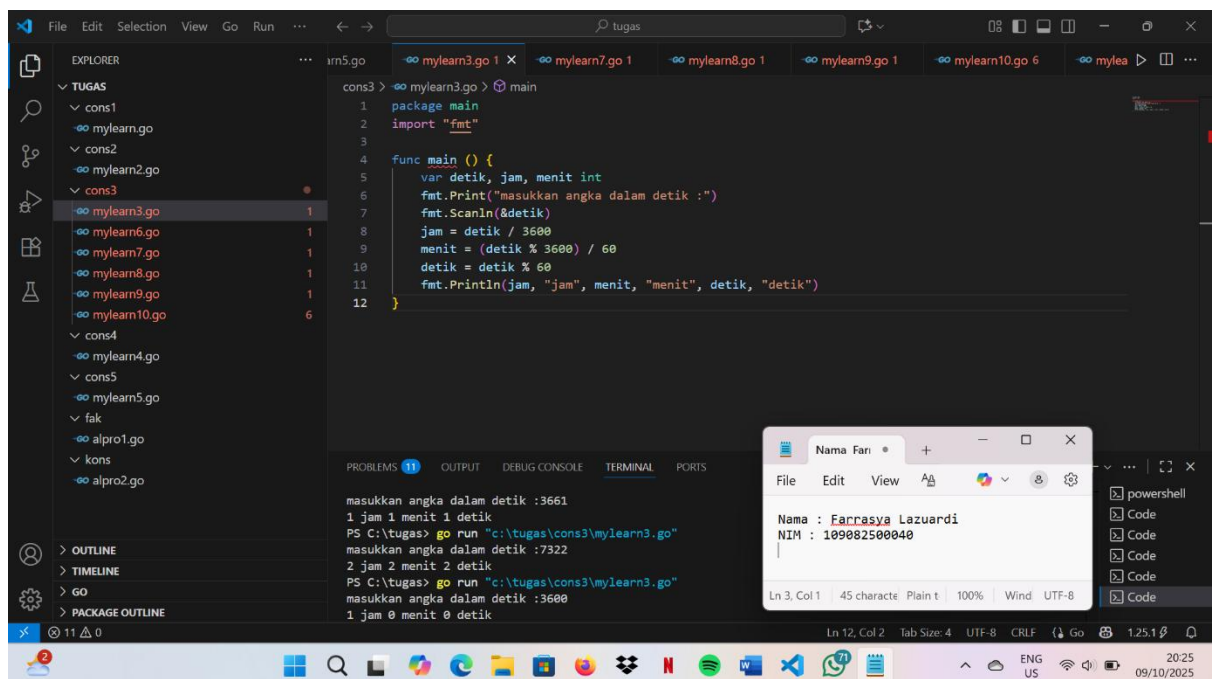
    menit = (detik % 3600) / 60

    detik = detik % 60

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

}
```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk **mengubah waktu dalam satuan detik menjadi jam, menit, dan detik.**

Dengan kata lain, pengguna memasukkan total detik, lalu program menghitung dan menampilkan berapa jam, menit, dan sisa detiknya.

Program ini:

- Menggunakan operasi pembagian (/) dan sisa bagi (%) untuk konversi waktu.
- Cocok untuk latihan pemahaman dasar **operator aritmatika, input/output, dan tipe data integer** di Go.

2. Guided 2

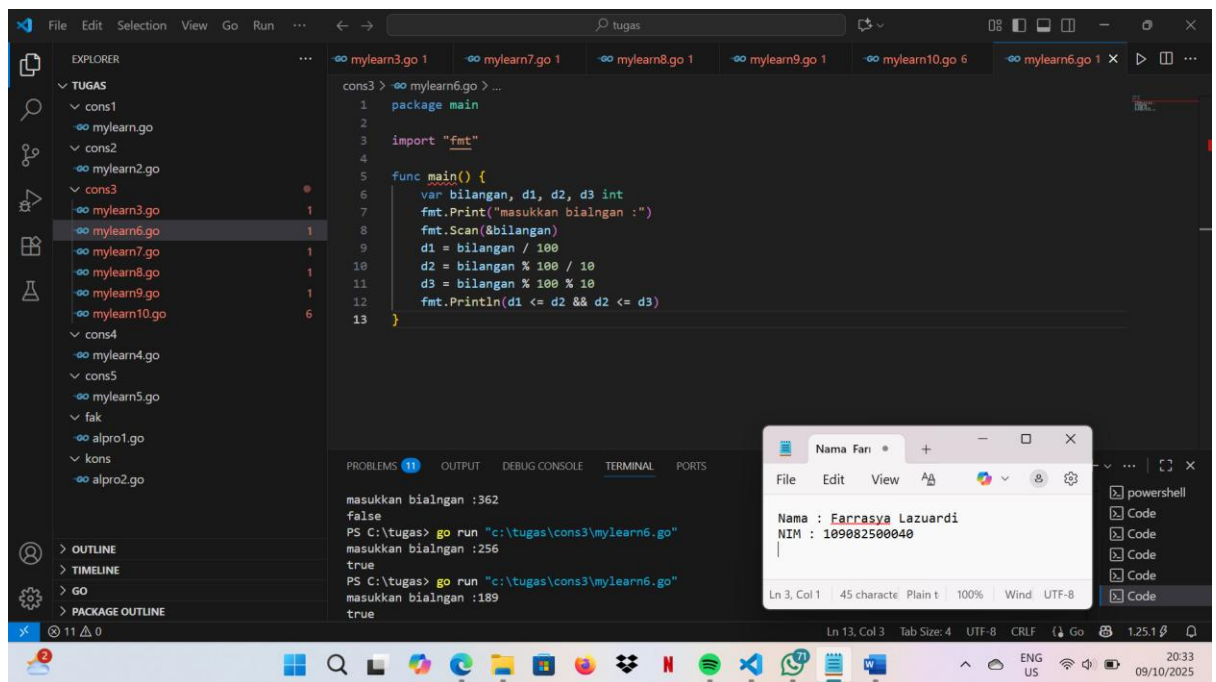
Source Code

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3 int
    fmt.Print("masukkan bialngan :")
    fmt.Scan(&bilangan)
    d1 = bilangan / 100
    d2 = bilangan % 100 / 10
    d3 = bilangan % 100 % 10
    fmt.Println(d1 <= d2 && d2 <= d3)
}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk memeriksa apakah tiga digit dari sebuah bilangan tiga angka berurutan secara tidak menurun (artinya dari kiri ke kanan nilainya semakin besar atau sama besar).

Program ini:

- Melatih logika perbandingan digit bilangan.
- Menggunakan **operator aritmatika (%) dan (/)** untuk memecah digit.
- Menghasilkan output **true** jika digit dari kiri ke kanan tidak menurun, dan **false** jika sebaliknya

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var beratBadan, tinggiBadan, bmi float64
    fmt.Print("masukkan berat badan : ")
    fmt.Scan(&beratBadan)
    fmt.Print("masukkan tinggi badan :")
    fmt.Scan(&tinggiBadan)
```

```

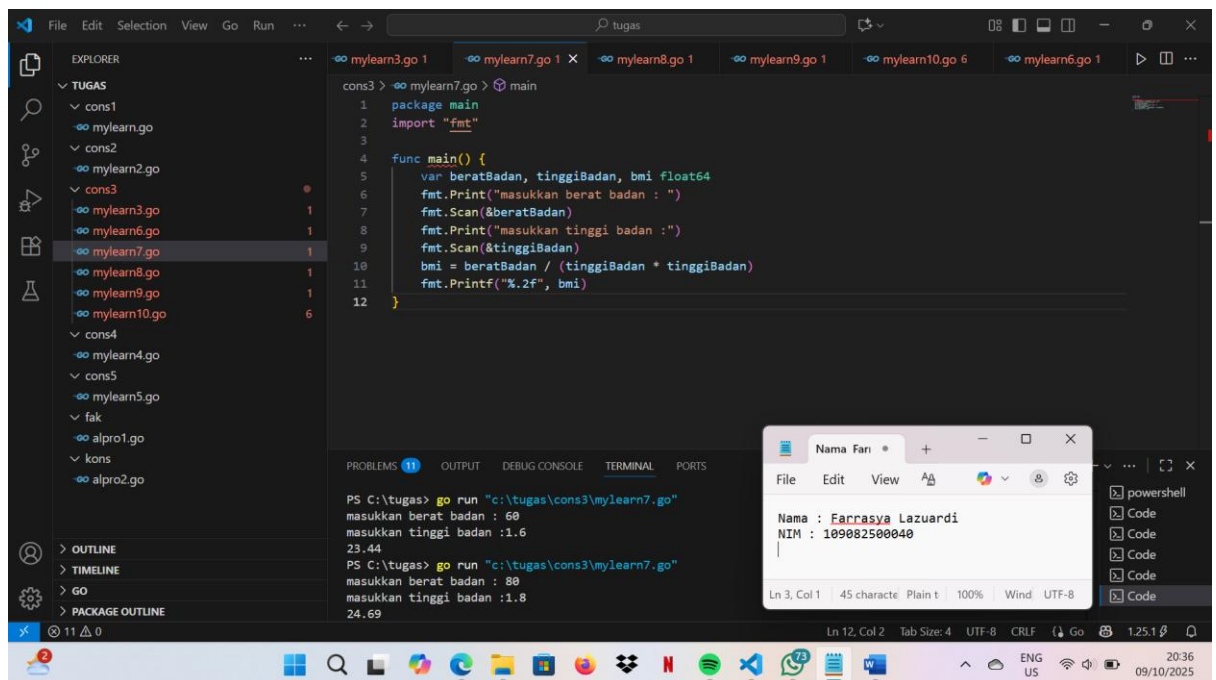
        bmi = beratBadan / (tinggiBadan * tinggiBadan)

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

    }
}

```

Screenshoot program



Deskripsi program

Program ini digunakan untuk **menghitung Indeks Massa Tubuh (BMI / Body Mass Index)** berdasarkan:

- berat badan (dalam kilogram),
- tinggi badan (dalam meter).

BMI merupakan ukuran sederhana untuk menilai apakah berat badan seseorang termasuk kurus, normal, gemuk, atau obesitas.

Program ini:

- Menggunakan **operator aritmatika** untuk menghitung kuadrat tinggi badan.
- Mengajarkan konsep **float64**, **input/output**, dan **formatting desimal**.
- Dapat dikembangkan lebih lanjut untuk menampilkan kategori BMI (misalnya "berat badan ideal" atau "obesitas").

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var totalBelanja, diskonBelanja, hasil float64

    fmt.Print("masukkan total harga belanja :")

    fmt.Scan(&totalBelanja)

    fmt.Print("masukkan diskon :")

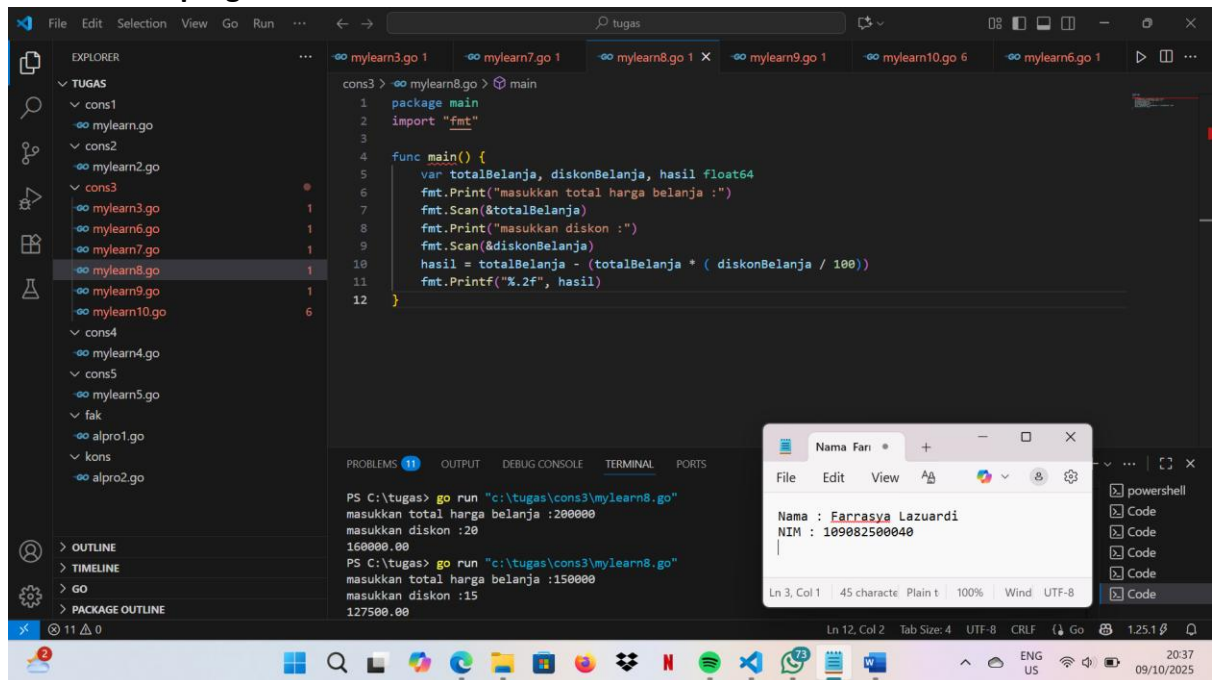
    fmt.Scan(&diskonBelanja)

    hasil = totalBelanja - (totalBelanja * (diskonBelanja / 100))

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

}
```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk **menghitung total harga yang harus dibayar pembeli setelah mendapat diskon.**

Kita memasukkan:

- Total harga belanja sebelum diskon.
- Persentase diskon (%).

Lalu program menghitung **harga akhir** setelah potongan diskon diterapkan.

2. Tugas 2

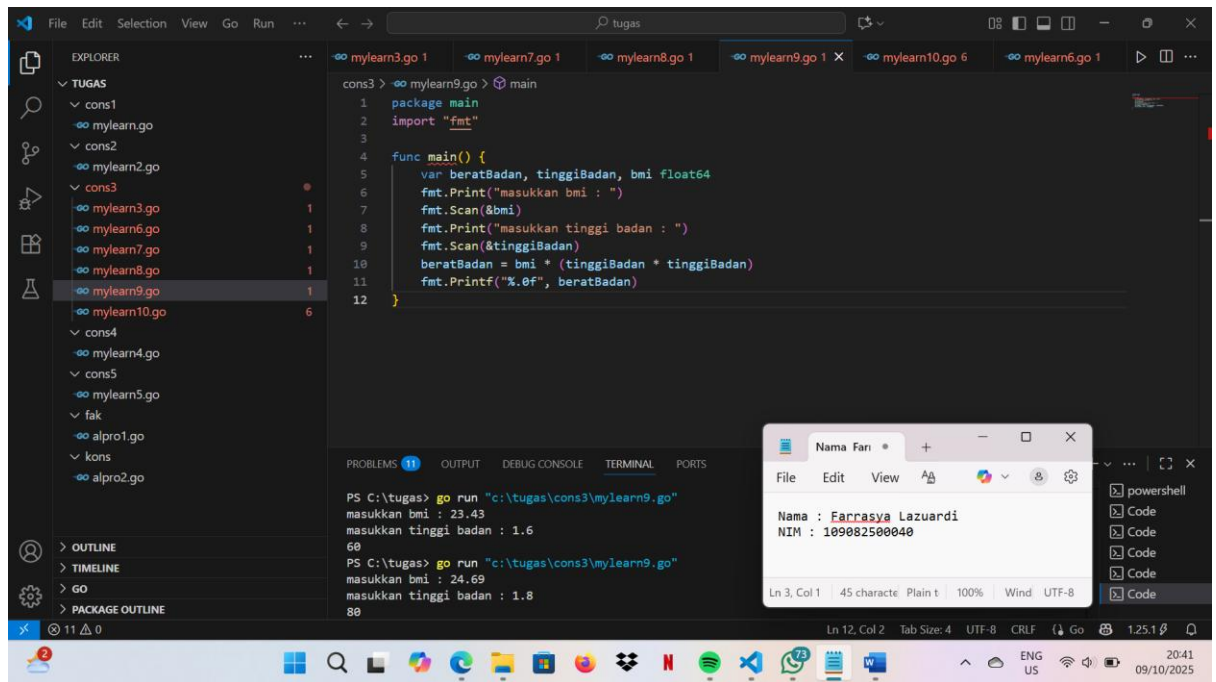
Source code

```
package main

import "fmt"

func main() {
    var beratBadan, tinggiBadan, bmi float64
    fmt.Print("masukkan bmi : ")
    fmt.Scan(&bmi)
    fmt.Print("masukkan tinggi badan : ")
    fmt.Scan(&tinggiBadan)
    beratBadan = bmi * (tinggiBadan * tinggiBadan)
    fmt.Printf("%.0f", beratBadan)
}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk **menghitung berat badan berdasarkan nilai BMI dan tinggi badan seseorang**.

Rumus yang digunakan adalah kebalikan dari rumus BMI:

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

Program ini:

- Menghitung **berat badan** berdasarkan nilai **BMI dan tinggi badan**.
- Menggunakan **operasi perkalian dan pangkat dua** untuk perhitungan.
- Melatih penggunaan **variabel float64, input/output, dan formatting hasil** di Go.

3. Tugas 3

Source code

```
package main
import (
    "fmt"
    "math"
)

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)

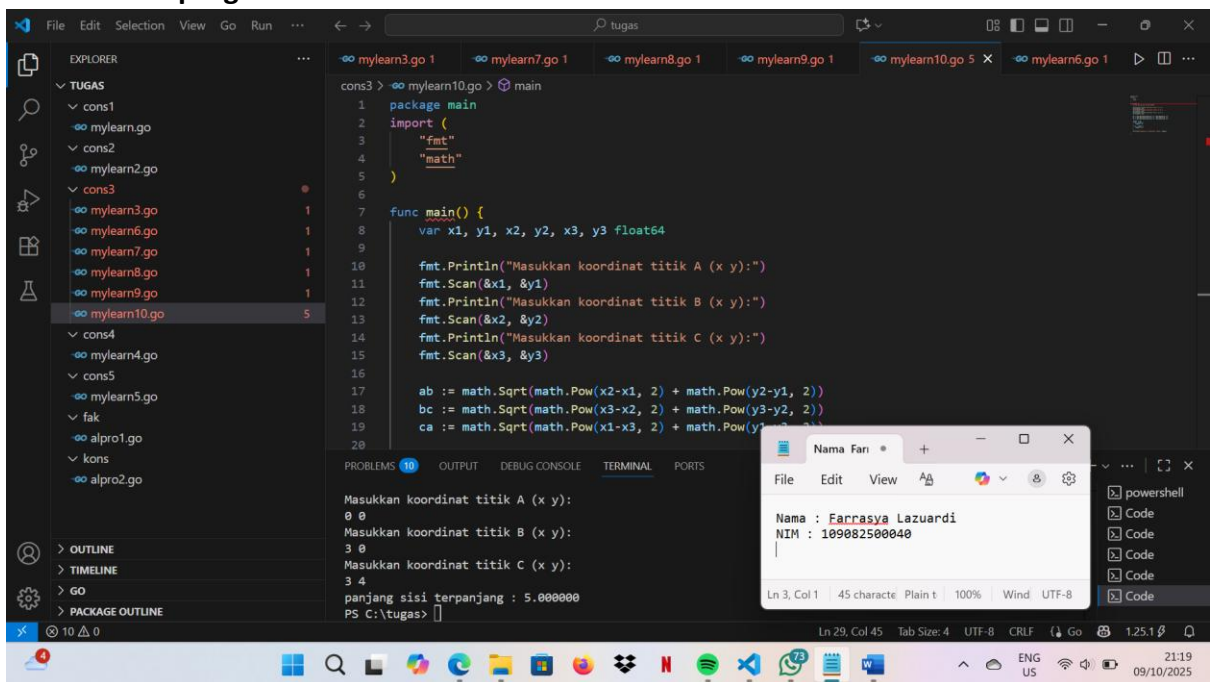
        ab := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1,
2))
        bc := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2,
2))
        ca := math.Sqrt(math.Pow(x1-x3, 2) + math.Pow(y1-y3,
2))

        longest := ab
        if bc > longest {
            longest = bc
        }
        if ca > longest {
            longest = ca
        }

        fmt.Printf("panjang sisi terpanjang : %2f\n", longest)
    }

```

Screenshoot program



Deskripsi program

1. Input Data

Program meminta pengguna memasukkan tiga titik koordinat:

- Titik **A** (x_1, y_1)
- Titik **B** (x_2, y_2)
- Titik **C** (x_3, y_3)

Semua disimpan sebagai tipe float64 karena bisa bernilai desimal.

2. Menghitung Panjang Setiap Sisi

Rumus jarak antara dua titik di bidang 2D digunakan:

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

Demikian juga untuk sisi lainnya:

$$BC = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}$$

$$CA = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2}$$

3. Menentukan Sisi Terpanjang

Program membandingkan ketiga panjang sisi (ab, bc, ca) dengan pernyataan if untuk menemukan yang paling besar, lalu menyimpannya ke variabel longest.

4. Output Hasil

Program menampilkan panjang sisi terpanjang dengan format dua angka di belakang koma:

Program ini:

- Menggunakan **rumus jarak antar dua titik** untuk menghitung panjang setiap sisi segitiga.
- Menentukan sisi terpanjang dengan membandingkan hasil jarak.
- Melatih konsep **geometri dasar** dan **penggunaan fungsi matematika (math.Sqrt, math.Pow) dalam Go**.