# LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 1

MODUL No. 4

I/O, TIPE DATA & VARIABEL (LATIHAN 2)



Disusun oleh:

NADIFA AZKHIA SYARIF 109082530002 S1IF-13-02

**Asisten Praktikum** 

Adithana dharma putra

Alfin Ilham Berlianto

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

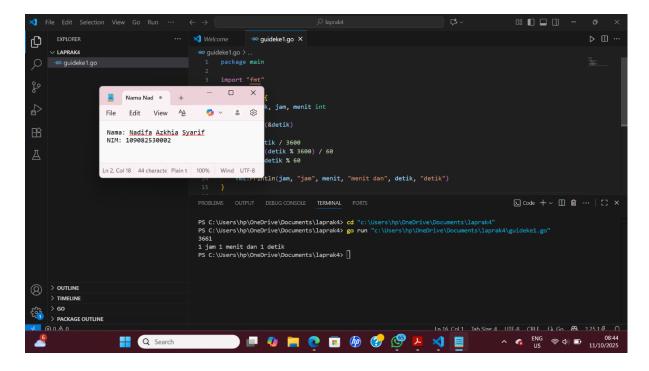
## **LATIHAN KELAS – GUIDED**

# 1. Guided 1 Source Code

```
// package main
import "fmt"
func main() {
     var 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")
}
```

## **Screenshoot program**

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan) CONTOH TANGKAPAN LAYAR:



Program di atas berfungsi untuk mengubah satuan waktu dari detik menjadi jam, menit, dandetik. Pertama, program meminta pengguna untuk memasukkan sebuah nilai dalam satuan detik. Nilai tersebut kemudian dihitung agar diketahui berapa jam, menit, dan detik yang setara. Perhitungannya dilakukan dengan cara: jumlah jam diperoleh dari hasil pembagian detik dengan 3600 (karena 1 jam = 3600 detik), kemudian sisa detiknya digunakan untuk menghitung menit dengan rumus (detik % 3600) / 60, dan sisa akhirnya (detik % 60) menjadi nilai detik yang tersisa.

# 2. Guided 2 Source Code

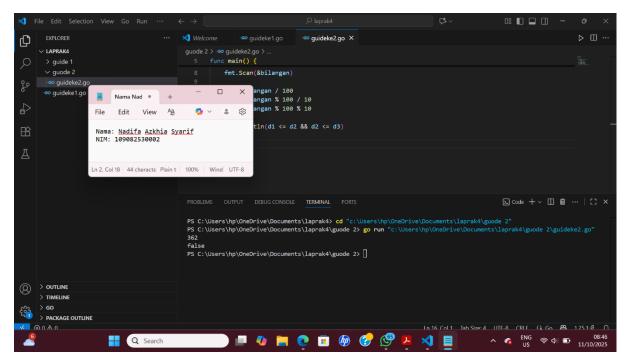
```
//package main
import "fmt"

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

fmt.Scan(&bilangan)
```

```
d1 = bilangan / 100
d2 = bilangan % 100 / 10
d3 = bilangan % 100 % 10

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



#### Deskripsi program

Program di atas digunakan untuk mengecek apakah tiga digit angka yang dimasukkan berurutan secara menaik (non-menurun).

Pertama, program meminta pengguna untuk memasukkan sebuah bilangan tiga digit. Nilai tersebut kemudian dipisahkan menjadi tiga bagian:

d1 adalah digit ratusan (diperoleh dari bilangan / 100),

d2 adalah digit puluhan (diperoleh dari bilangan % 100 / 10),

d3 adalah digit satuan (diperoleh dari bilangan % 100 % 10).

Setelah itu, program memeriksa apakah urutan digit memenuhi syarat d1 <= d2 && d2 <= d3, artinya setiap digit sama atau lebih kecil dari digit setelahnya. Jika urutannya benar (menaik), program akan menampilkan true, dan jika tidak, akan menampilkan false.

#### 3. Guided 3

## **Source Code**

```
package main

import "fmt"

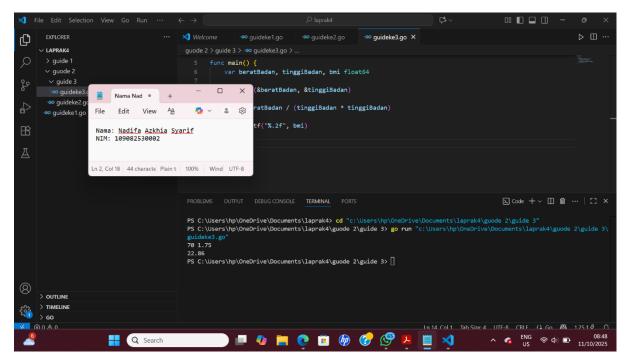
func main() {
    var beratBadan, tinggiBadan, bmi float64

    fmt.Scan(&beratBadan, &tinggiBadan)

bmi = beratBadan / (tinggiBadan * tinggiBadan)

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

# **Screenshoot program**



Program di atas digunakan untuk menghitung dan menampilkan nilai BMI (Body Mass Index) berdasarkan berat badan dan tinggi badan yang dimasukkan oleh pengguna. Pertama, program meminta pengguna untuk memasukkan dua nilai, yaitu berat badan (dalam kilogram) dan tinggi badan (dalam meter). Setelah pengguna mengetikkan nilainya, program akan memproses data tersebut dan menghasilkan nilai BMI.

#### **TUGAS**

# 1. Tugas 1

#### Source code

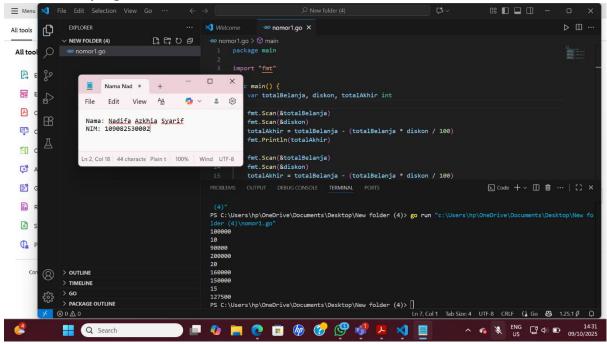
```
package main
import "fmt"
func main() {
     var totalBelanja, diskon, totalAkhir int
     fmt.Scan(&totalBelanja)
     fmt.Scan(&diskon)
     totalAkhir = totalBelanja - (totalBelanja * diskon /
100)
     fmt.Println(totalAkhir)
     fmt.Scan(&totalBelanja)
     fmt.Scan(&diskon)
     totalAkhir = totalBelanja - (totalBelanja * diskon /
100)
     fmt.Println(totalAkhir)
     fmt.Scan(&totalBelanja)
```

```
fmt.Scan(&diskon)

totalAkhir = totalBelanja - (totalBelanja * diskon /

100)

fmt.Println(totalAkhir)
}
```



## Deskripsi program

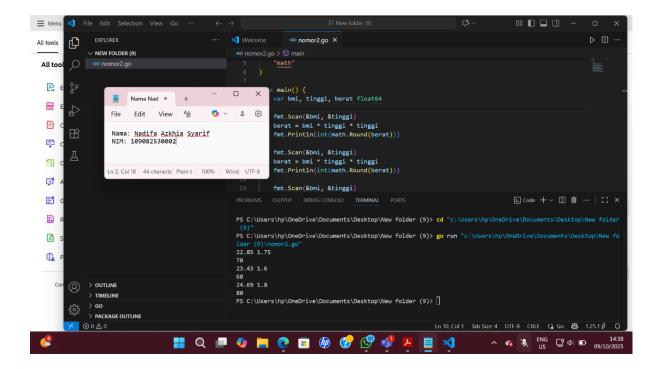
Program di atas digunakan untuk menghitung total harga akhir setelah mendapatkan diskon, dan dilakukan sebanyak tiga kali untuk tiga data yang berbeda. Pertama, program meminta pengguna memasukkan dua nilai, yaitu total belanja dan persentase diskon. Setelah kedua nilai dimasukkan, program menghitung total harga akhir dengan cara mengurangi total belanja dengan potongan diskon yang sesuai. Setelah itu, hasil akhir ditampilkan ke layar. Proses yang sama diulang lagi dua kali berikutnya untuk data kedua dan ketiga.

#### 2. Tugas 2

## Source code

```
package main
```

```
import (
     "fmt"
     "math"
)
func main() {
     var bmi, tinggi, berat float64
     fmt.Scan(&bmi, &tinggi)
     berat = bmi * tinggi * tinggi
     fmt.Println(int(math.Round(berat)))
     fmt.Scan(&bmi, &tinggi)
     berat = bmi * tinggi * tinggi
     fmt.Println(int(math.Round(berat)))
     fmt.Scan(&bmi, &tinggi)
     berat = bmi * tinggi * tinggi
     fmt.Println(int(math.Round(berat)))
}
```



Program di atas digunakan untuk menghitung berat badan berdasarkan nilai BMI dan tinggi badan, dan dilakukan sebanyak tiga kali untuk tiga data yang berbeda.

Pertama, program meminta pengguna memasukkan dua nilai, yaitu BMI (Body Mass Index) dan tinggi badan dalam meter. Setelah itu, program menghitung berat badan yang sesuai dengan nilai BMI dan tinggi tersebut. Hasil perhitungan kemudian dibulatkan ke bilangan bulat terdekat menggunakan math.Round, lalu ditampilkan di layar. Proses ini diulang untuk data kedua dan ketiga agar pengguna bisa menghitung tiga hasil sekaligus.

## 3. Tugas 3

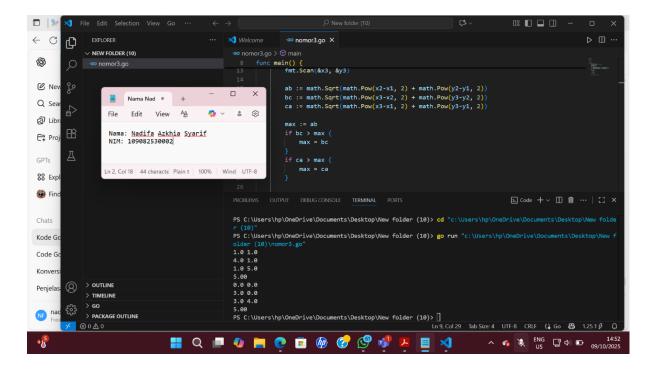
#### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    for i := 0; i < 2; i++ { // ngulang 2 kali buat dua data</pre>
```

```
var x1, y1, x2, y2, x3, y3 float64
           fmt.Scan(&x1, &y1)
           fmt.Scan(&x2, &y2)
           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(x3-x1, 2) +
math.Pow(y3-y1, 2))
           max := ab
           if bc > max {
                max = bc
           if ca > max {
               max = ca
           fmt.Printf("%.2f\n", max)
}
```



Program di atas digunakan untuk menentukan sisi terpanjang dari sebuah segitiga berdasarkan tiga titik koordinat, dan prosesnya dilakukan sebanyak dua kali. Setiap kali dijalankan, pengguna diminta memasukkan tiga titik koordinat, yaitu titik A, B, dan C. Setelah itu, program menghitung panjang masing-masing sisi segitiga, yaitu AB, BC, dan CA, dengan menggunakan rumus jarak antar dua titik. Dari ketiga sisi tersebut, program mencari sisi yang paling panjang menggunakan perbandingan nilai terbesar. Hasil akhirnya kemudian ditampilkan di layar dengan dua angka di belakang koma. Jika dijalankan dua kali dengan dua set titik berbeda, program akan menampilkan dua hasil yang masing-masing menunjukkan panjang sisi terpanjang dari segitiga yang terbentuk.