

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL [4]
[Variable Tipe Data]**



**Disusun oleh:
MUHAMMAD ZAKI AL-FIQRI
109082500085
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 int
    var menit int
    var jam, sec int

    fmt.Print(" masukan detik : ")
    fmt.Scan(&detik)
    jam = detik / 3600
    menit = (detik / 3600) % 60
    sec = detik % 60

    fmt.Println("jam", jam)
    fmt.Println("menit ", menit)
    fmt.Println("detik ", sec)
    fmt.Printf("%d jam %d menit %d detik", jam, menit,
sec)

}
```

Screenshoot program

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

The image shows a screenshot of a development environment. On the left, a Go program is visible in a code editor. The program is a function `main()` that takes an integer `detik` as input, calculates the equivalent time in hours (`jam`), minutes (`menit`), and seconds (`sec`), and prints the result. The program is run, and the output is shown in the terminal: `jam 1`, `menit 1`, `detik 1`, and `1 jam 1 menit 1 detik`. On the right, a browser window is open, displaying the name Muhammad Zaki Al-Fiqri and the ID Nim:109082500085.

```
4
5 func main() {
6     var detik int
7     var menit int
8     var jam, sec int
9
10    fmt.Print(" masukan detik : ")
11    fmt.Scan(&detik)
12    jam = detik / 3600
13    menit = (detik / 3600) % 60
14    sec = detik % 60
15
16    fmt.Println("jam", jam)
17    fmt.Println("menit ", menit)
18    fmt.Println("detik ", sec)
19    fmt.Printf("%d jam %d menit %d detik", jam, menit, sec)
20
21 }
22
```

jam 1
menit 1
detik 1
1 jam 1 menit 1 detik
PS C:\Users\USER\Desktop\tupas 1>

Muhammad Zaki Al-Fiqri
Nim:109082500085

Deskripsi program

Program ini berfungsi untuk mengkonversi total durasi waktu dalam satuan detik menjadi format Jam, Menit, dan Detik. Setelah pengguna memasukkan angka total detik, program akan menghitung jumlah jam penuh dengan membagi total detik dengan 3600, kemudian menghitung sisa menit dengan membagi total detik setelah dikonversi ke jam dengan 60, dan terakhir menghitung sisa detik dengan menggunakan operator modulo (3600) pada total detik awal, lalu menampilkan ketiga nilai tersebut secara terpisah dan dalam satu format kalimat.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var angka1, angka2, angka3, bilangan int

    fmt.Print("masukan bilangan : ")

    fmt.Scan(&bilangan)
```

```

    angka1 = bilangan / 100

    angka2 = (bilangan % 100) / 10

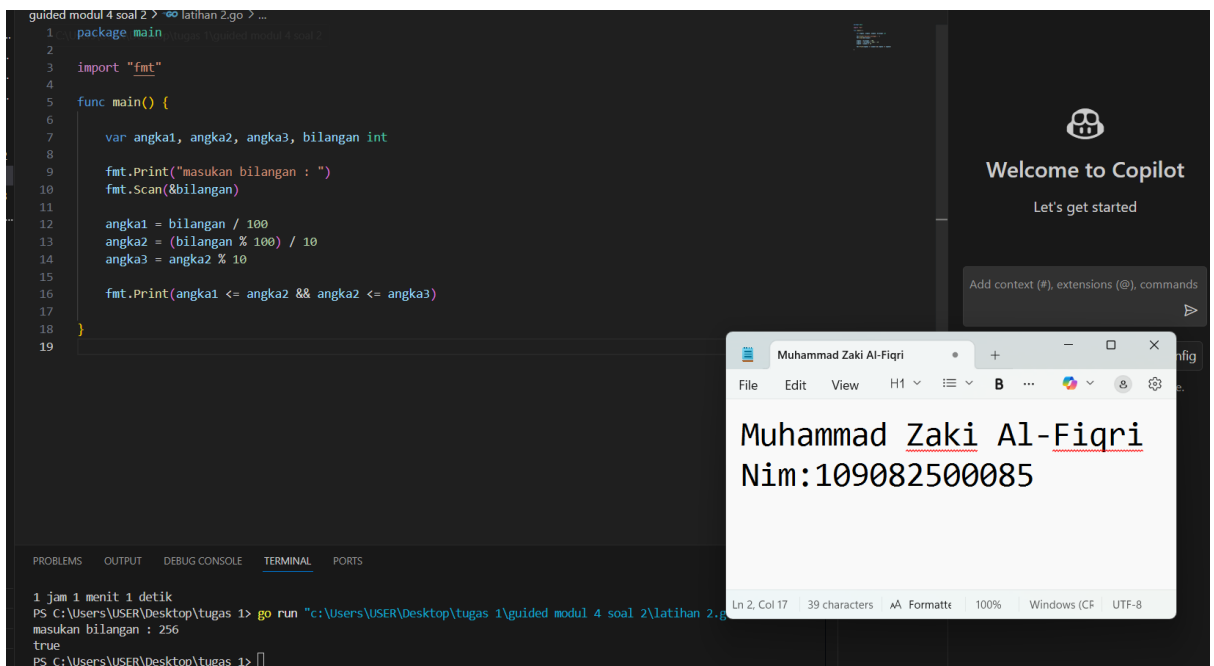
    angka3 = angka2 % 10

    fmt.Print(angka1 <= angka2 && angka2 <= angka3)

}

```

Screenshoot program



Deskripsi program

Program Go ini bertujuan untuk memeriksa apakah digit-digit dari suatu bilangan bulat tersusun dalam urutan yang tidak menurun. Program membaca input bilangan dari pengguna dan mencoba memecahnya menjadi tiga digit: digit ratusan (angka1), digit puluhan (angka2), dan digit satuan (angka3), meskipun logika perhitungan untuk angka3 salah. Setelah memisahkan (atau setidaknya mencoba memisahkan) digit-digit tersebut, program kemudian mengevaluasi dan mencetak hasil boolean (true atau false) dari ekspresi `angka1 <= angka2 && angka2 <= angka3`, yang bernilai benar jika setiap digit (dari kiri ke kanan) selalu lebih besar dari atau sama dengan digit sebelumnya.

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)

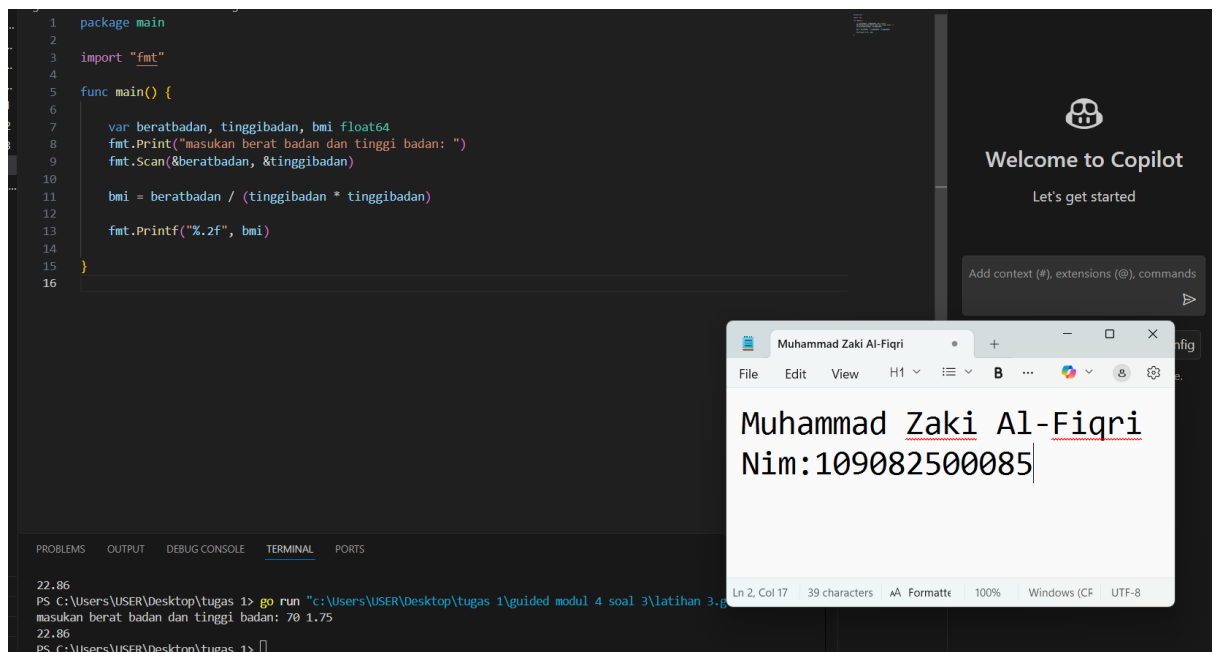
    bmi = beratbadan / (tinggibadan * tinggibadan)

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

}

```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk menghitung Indeks Massa Tubuh (IMT) atau Body Mass Index (BMI) berdasarkan input berat badan (kilogram) dan tinggi badan (meter) dari pengguna. Program membaca dua nilai floating-point (berat dan tinggi), kemudian menggunakan rumus $BMI = \text{berat badan} / (\text{tinggi badan} \times \text{tinggi badan})$ untuk melakukan

perhitungan. Hasil BMI, yang merupakan nilai Float , kemudian dicetak ke layar dengan diformat menjadi dua angka di belakang koma.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var hargaawal, hargasetelahdiskon, diskon, totaldiskon int

    fmt.Print("masukan harga awal : ")
    fmt.Scan(&hargaawal)
    fmt.Print("masukan diskon :")
    fmt.Scan(&diskon)

    totaldiskon = (hargaawal * diskon) / 100
    hargasetelahdiskon = hargaawal - totaldiskon

    fmt.Print(hargasetelahdiskon)

}
```

Screenshoot program

Deskripsi program

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var beratbadan, tinggibadan, bmi float64

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

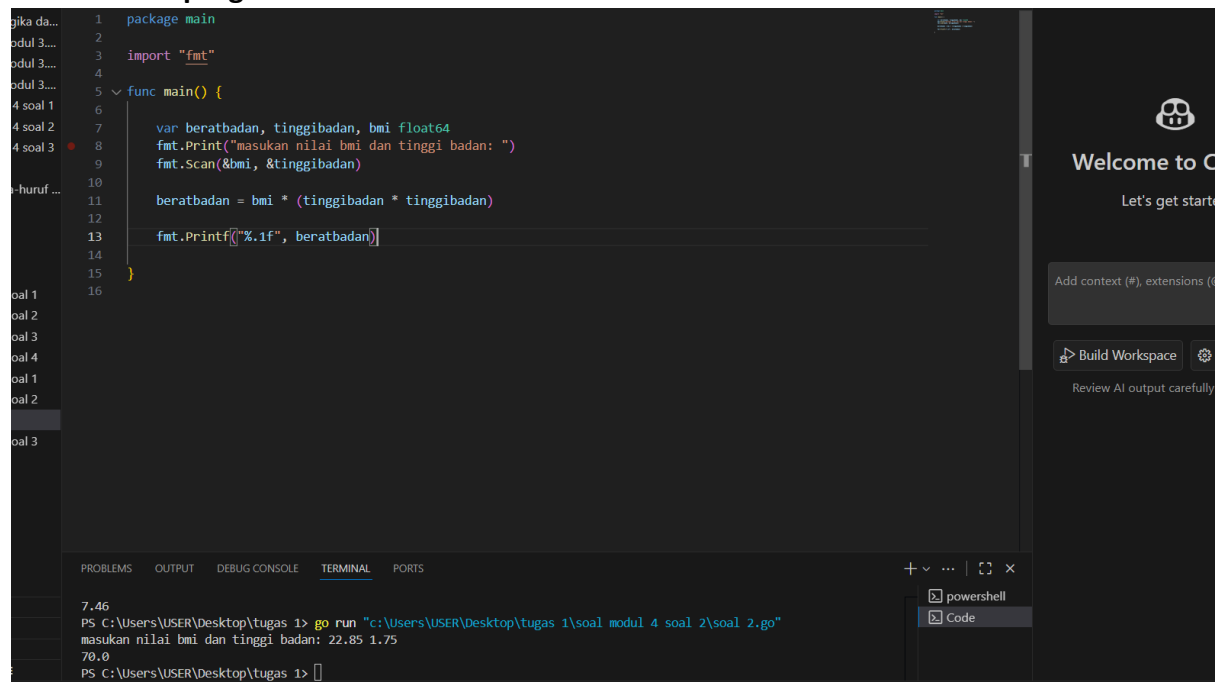
    fmt.Scan(&bmi, &tinggibadan)

    beratbadan = bmi * (tinggibadan * tinggibadan)

    fmt.Printf("%.1f", beratbadan)

}
```

Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The code calculates BMI based on input values. The terminal shows the command to run the program and the resulting output.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var beratbadan, tinggibadan, bmi float64
8     fmt.Print("masukan nilai bmi dan tinggi badan: ")
9     fmt.Scan(&bmi, &tinggibadan)
10
11     beratbadan = bmi * (tinggibadan * tinggibadan)
12
13     fmt.Printf("%.1f", beratbadan)
14
15 }
16
```

Terminal Output:

```
7.46
PS C:\Users\USER\Desktop\tugas 1> go run "c:\Users\USER\Desktop\tugas 1\soal modul 4 soal 2\soal 2.go"
masukan nilai bmi dan tinggi badan: 22.85 1.75
70.0
PS C:\Users\USER\Desktop\tugas 1>
```

Deskripsi program

Program ini berfungsi untuk menghitung berat badan (dalam kilogram) seseorang berdasarkan input nilai Indeks Massa Tubuh (BMI) dan tinggi badan (dalam meter). Program membaca kedua nilai *floating-point* tersebut dari pengguna, kemudian menggunakan rumus terbalik dari perhitungan BMI, yaitu Berat Badan=BMI×(Tinggi Badan×Tinggi Badan). Hasil akhir berat badan dicetak ke layar dengan diformat menjadi satu angka di belakang koma.

3. Tugas 3

Source code

```
package main

import (
    "math"

    "fmt"
)

func main() {

    var ax, ay, bx, by, cx, cy float64

    fmt.Print("masukan koordinat ax , ay : ")
    fmt.Scan(&ax, &ay)
    fmt.Print("masukan koordunat bx , by : ")
    fmt.Scan(&bx, &by)
    fmt.Print("masukan koordinat cx , cy : ")
    fmt.Scan(&cx, &cy)

    ab := math.Sqrt(math.Pow(bx-ax, 2) + math.Pow(by-
ay, 2))
    bc := math.Sqrt(math.Pow(cx-bx, 2) + math.Pow(cy-
by, 2))
    ca := math.Sqrt(math.Pow(ax-cx, 2) + math.Pow(ay-
cy, 2))

    sisiterpanjang := math.Max(ab, math.Max(bc, ca))

    fmt.Printf("sisi terpanjang: %.2f", sisiterpanjang)

}
```

Screenshoot program

The image shows a screenshot of a Go program in a code editor (VS Code) and a Copilot chat window. The Go program calculates the longest side of a triangle given three points A(xA, yA), B(xB, yB), and C(xC, yC). It uses the Euclidean distance formula to calculate the lengths of the three sides (AB, BC, and CA) and then uses the math.Max function to determine the longest side. The output shows the longest side is 5.00.

```
1 package main
2
3 import (
4     "math"
5     "fmt"
6 )
7
8 func main() {
9
10     var ax, ay, bx, by, cx, cy float64
11
12     fmt.Print("masukan koordinat ax , ay : ")
13     fmt.Scan(&ax, &ay)
14     fmt.Print("masukan koordinat bx , by : ")
15     fmt.Scan(&bx, &by)
16     fmt.Print("masukan koordinat cx , cy : ")
17     fmt.Scan(&cx, &cy)
18
19     ab := math.Sqrt(math.Pow(bx-ax, 2) + math.Pow(by-ay, 2))
20     bc := math.Sqrt(math.Pow(cx-bx, 2) + math.Pow(cy-by, 2))
21     ca := math.Sqrt(math.Pow(ax-cx, 2) + math.Pow(ay-cy, 2))
22
23     sisiterpanjang := math.Max(ab, math.Max(bc, ca))
24
25     fmt.Printf("sisi terpanjang: %.2f", sisiterpanjang)
26 }
27
28
29
```

Terminal output:

```
masukan koordinat ax , ay : 0.0 0.0
masukan koordinat bx , by : 0.0 3.0
masukan koordinat cx , cy : 3.0 4.0
sisi terpanjang: 5.00
PS C:\Users\USER\Desktop\tugas 1>
```

Copilot chat window:

Welcome to Copilot
Let's get started

Add context (#), extensions (@), commands

Muhammad Zaki Al-Fiqri

Muhammad Zaki Al-Fiqri
Nim:109082500085

Deskripsi program

Program Go ini berfungsi untuk **menghitung panjang sisi terpanjang** dari sebuah segitiga yang dibentuk oleh tiga titik koordinat A(xA,yA), B(xB,yB), dan C(xC,yC) yang dimasukkan oleh pengguna. Program menggunakan **rumus jarak Euclidean** (aplikasi dari Teorema Pythagoras) untuk menghitung panjang ketiga sisi (AB, BC, dan CA). Setelah menghitung ketiga panjang sisi, program menggunakan fungsi math.Max untuk menentukan sisi mana yang paling panjang, dan kemudian mencetak hasilnya ke layar dengan format **dua angka di belakang koma**.