

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

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



Disusun oleh:
JEREMY CHRISTIAN SIHOMBING
109082500178
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 detik: ")

    fmt.Scan(&detik)

    jam = detik / 3600

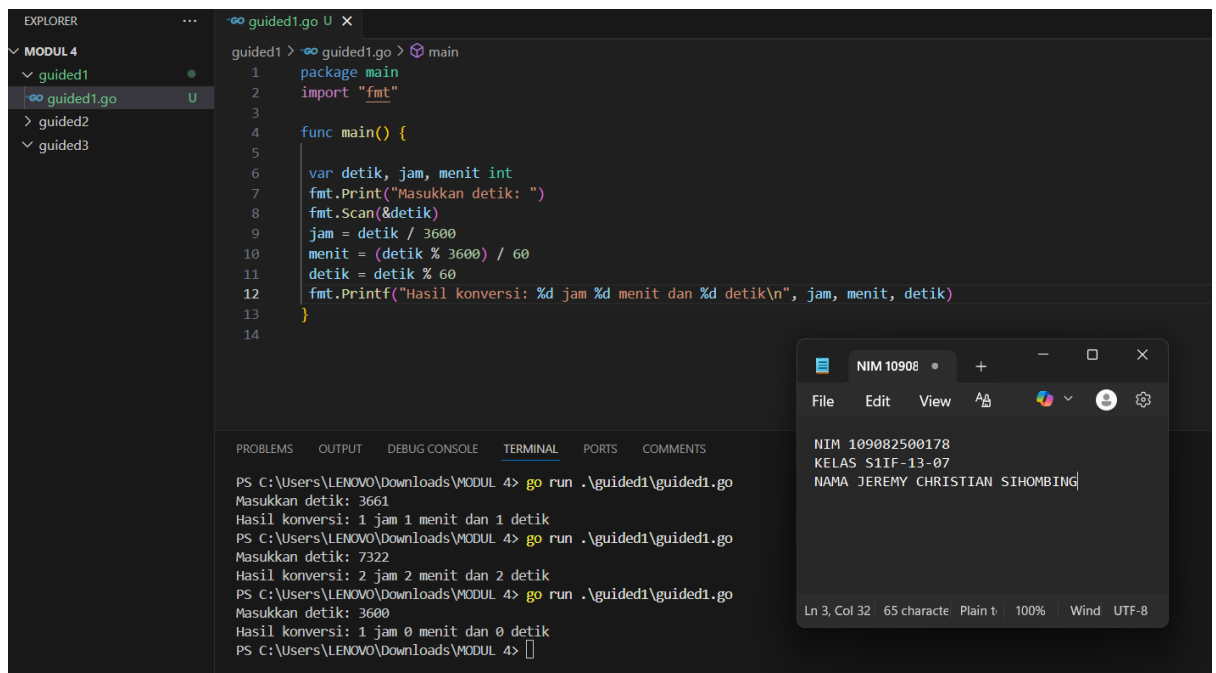
    menit = (detik % 3600) / 60

    detik = detik % 60

    fmt.Printf("Hasil konversi: %d jam %d menit dan %d detik\n", jam, menit, detik)

}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor. The Explorer panel on the left shows a project structure with 'MODUL 4' containing 'guided1', 'guided2', and 'guided3'. The 'guided1' directory is expanded, showing 'guided1.go'. The main editor displays the source code of 'guided1.go'.

```
1 package main
2 import "fmt"
3
4 func main() {
5
6     var detik, jam, menit int
7     fmt.Print("Masukkan detik: ")
8     fmt.Scan(&detik)
9     jam = detik / 3600
10    menit = (detik % 3600) / 60
11    detik = detik % 60
12    fmt.Printf("Hasil konversi: %d jam %d menit dan %d detik\n", jam, menit, detik)
13 }
14
```

The bottom panel shows the 'TERMINAL' output, which displays the program's execution results for three different inputs: 3661, 7322, and 3600.

```
PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\guided1\guided1.go
Masukkan detik: 3661
Hasil konversi: 1 jam 1 menit dan 1 detik
PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\guided1\guided1.go
Masukkan detik: 7322
Hasil konversi: 2 jam 2 menit dan 2 detik
PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\guided1\guided1.go
Masukkan detik: 3600
Hasil konversi: 1 jam 0 menit dan 0 detik
PS C:\Users\LENOVO\Downloads\MODUL 4>
```

A floating window titled 'NIM 10908' is also visible, containing the following text:

```
NIM 109082500178
KELAS SIIF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING
```

Deskripsi program

Program ini dibuat untuk mengubah waktu dalam satuan detik menjadi bentuk yang lebih mudah dimengerti, yaitu jam, menit, dan detik. Ketika dijalankan, pengguna akan diminta memasukkan jumlah detik. Nilai tersebut kemudian diolah dengan cara menghitung berapa jam yang bisa dibentuk dari total detik menggunakan pembagian 3600, karena satu jam sama dengan 3600 detik. Sisa dari hasil pembagian itu digunakan lagi untuk mencari jumlah menit dengan membaginya dengan 60, dan sisanya lagi menjadi detik. Setelah proses perhitungan selesai, program akan menampilkan hasil konversi dalam format "x jam y menit dan z detik". Secara sederhana, program ini membantu pengguna mengetahui waktu dalam bentuk yang lebih mudah dibaca dari total detik yang dimasukkan tanpa harus menghitung manual.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var bilangan, d1, d2, d3 int

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

    fmt.Scan(&bilangan)

    d1 = bilangan / 100

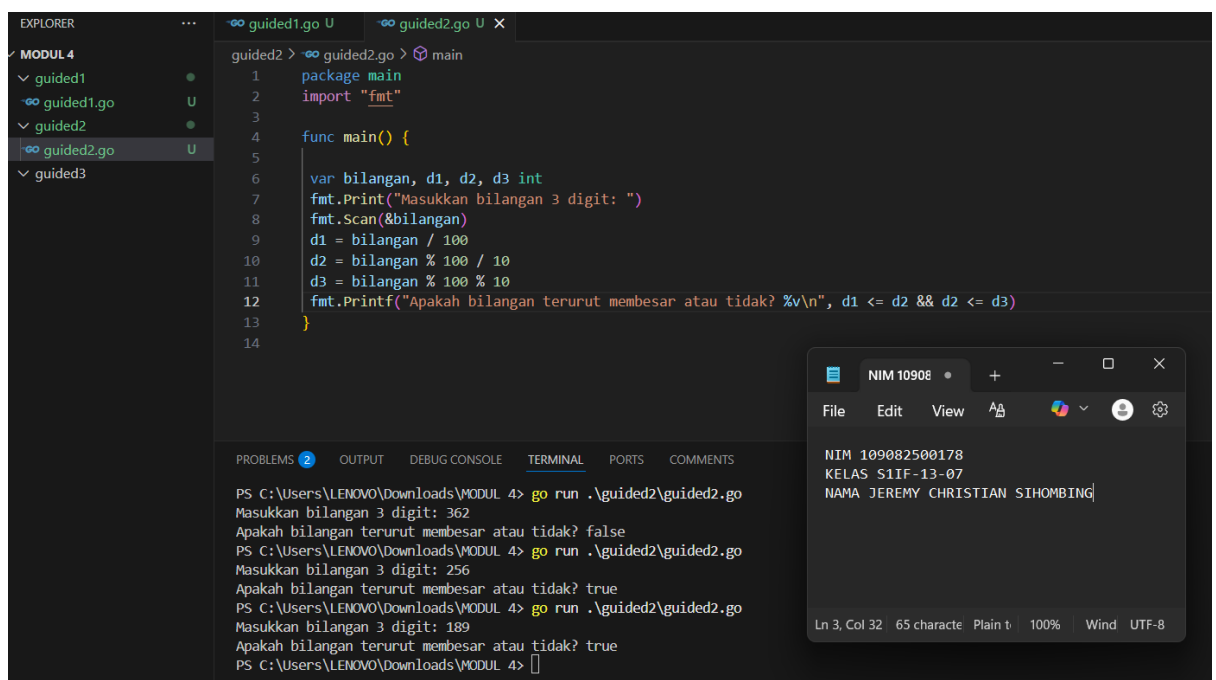
    d2 = bilangan % 100 / 10

    d3 = bilangan % 100 % 10

    fmt.Printf("Apakah bilangan terurut membesar atau
    tidak? %v\n", d1 <= d2 && d2 <= d3)

}
```

Screenshoot program



The screenshot displays a Go IDE interface. On the left, the Explorer pane shows a project structure with 'MODUL 4' containing 'guided1', 'guided2', and 'guided3'. The main editor shows the source code for 'guided2.go', which is identical to the code provided in the previous block. The bottom panel is split into 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', 'PORTS', and 'COMMENTS'. The 'TERMINAL' tab is active, showing the execution of the program. The output shows three test cases: 362 (returns false), 256 (returns true), and 189 (returns true). A small window titled 'NIM 10908' is also visible in the bottom right corner, displaying the student's information: NIM 109082500178, KELAS S1IF-13-07, and NAMA JEREMY CHRISTIAN SIHOMBING.

```
guided2 > package main
import "fmt"

func main() {
    var bilangan, d1, d2, d3 int
    fmt.Print("Masukkan bilangan 3 digit: ")
    fmt.Scan(&bilangan)
    d1 = bilangan / 100
    d2 = bilangan % 100 / 10
    d3 = bilangan % 100 % 10
    fmt.Printf("Apakah bilangan terurut membesar atau tidak? %v\n", d1 <= d2 && d2 <= d3)
}
```

PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\guided2\guided2.go
Masukkan bilangan 3 digit: 362
Apakah bilangan terurut membesar atau tidak? false
PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\guided2\guided2.go
Masukkan bilangan 3 digit: 256
Apakah bilangan terurut membesar atau tidak? true
PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\guided2\guided2.go
Masukkan bilangan 3 digit: 189
Apakah bilangan terurut membesar atau tidak? true
PS C:\Users\LENOVO\Downloads\MODUL 4>

NIM 10908
KELAS S1IF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING

Ln 3, Col 32 65 character Plain t 100% Wind UTF-8

Deskripsi program

Program ini digunakan untuk mengecek apakah sebuah bilangan dengan tiga digit memiliki urutan angka yang membesar atau tidak. Ketika program dijalankan, pengguna akan diminta untuk memasukkan sebuah bilangan yang terdiri dari tiga digit. Program kemudian memisahkan setiap digitnya, di mana digit pertama diperoleh dari hasil pembagian bilangan dengan 100, digit kedua dari sisa pembagian 100 yang dibagi 10, dan digit ketiga dari sisa pembagian 100 kemudian diambil satuannya. Setelah ketiga digit tersebut diperoleh, program akan memeriksa apakah urutannya membesar, yaitu apakah digit pertama lebih kecil atau sama dengan digit kedua, dan digit kedua lebih kecil atau sama dengan digit ketiga. Hasil pemeriksaan ditampilkan dalam bentuk nilai true jika bilangan tersebut terurut membesar, dan false jika tidak.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var beratBadan, tinggiBadan, bmi float64

    fmt.Print("Masukkan berat badan (kg) dan tinggi badan (m): ")

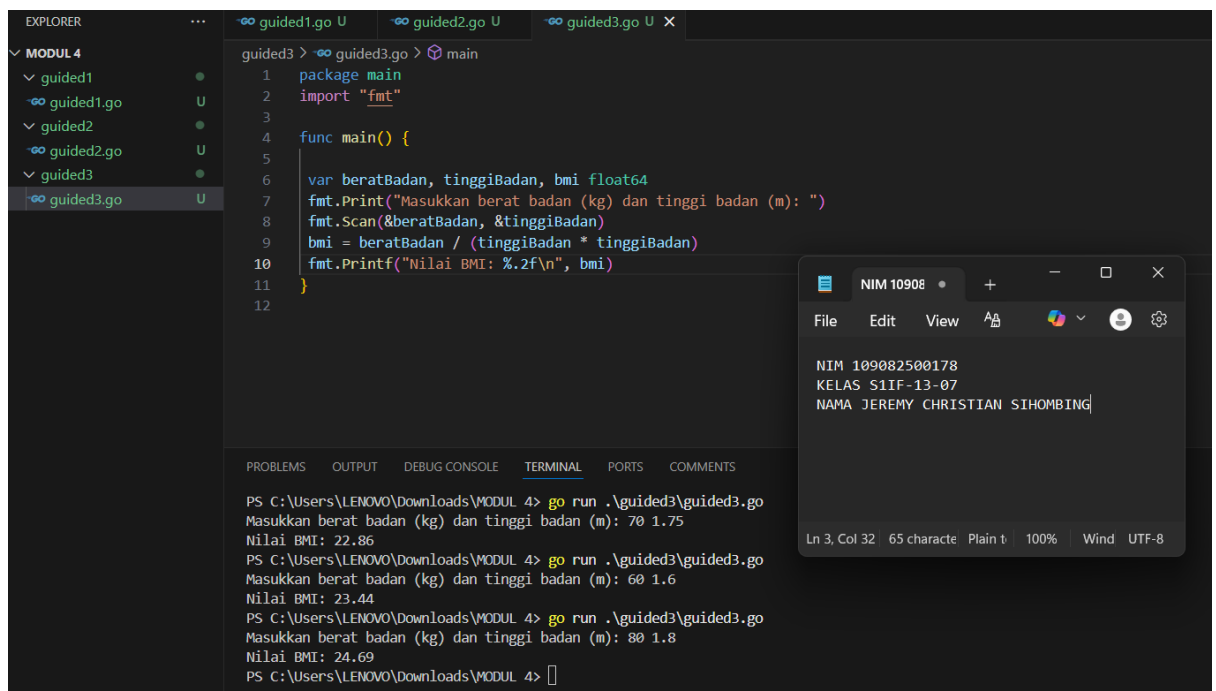
    fmt.Scan(&beratBadan, &tinggiBadan)

    bmi = beratBadan / (tinggiBadan * tinggiBadan)

    fmt.Printf("Nilai BMI: %.2f\n", bmi)

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menghitung nilai BMI (Body Mass Index) seseorang berdasarkan berat badan dan tinggi badan yang dimasukkan. Saat dijalankan, program akan meminta pengguna memasukkan berat badan dalam kilogram serta tinggi badan dalam meter. Kemudian, program menghitung nilai BMI dengan rumus berat badan dibagi dengan kuadrat tinggi badan. Setelah perhitungan selesai, hasilnya akan ditampilkan di layar dengan dua angka di belakang koma agar lebih akurat. Secara sederhana, program ini membantu pengguna mengetahui indeks massa tubuhnya, yang bisa digunakan untuk memperkirakan apakah berat badannya termasuk ideal, kurang, atau berlebih.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var totalBelanjaAwal, diskon int

    fmt.Print("Masukkan total belanja: ")

    fmt.Scan(&totalBelanjaAwal)

    fmt.Print("Masukkan besar diskon (%): ")

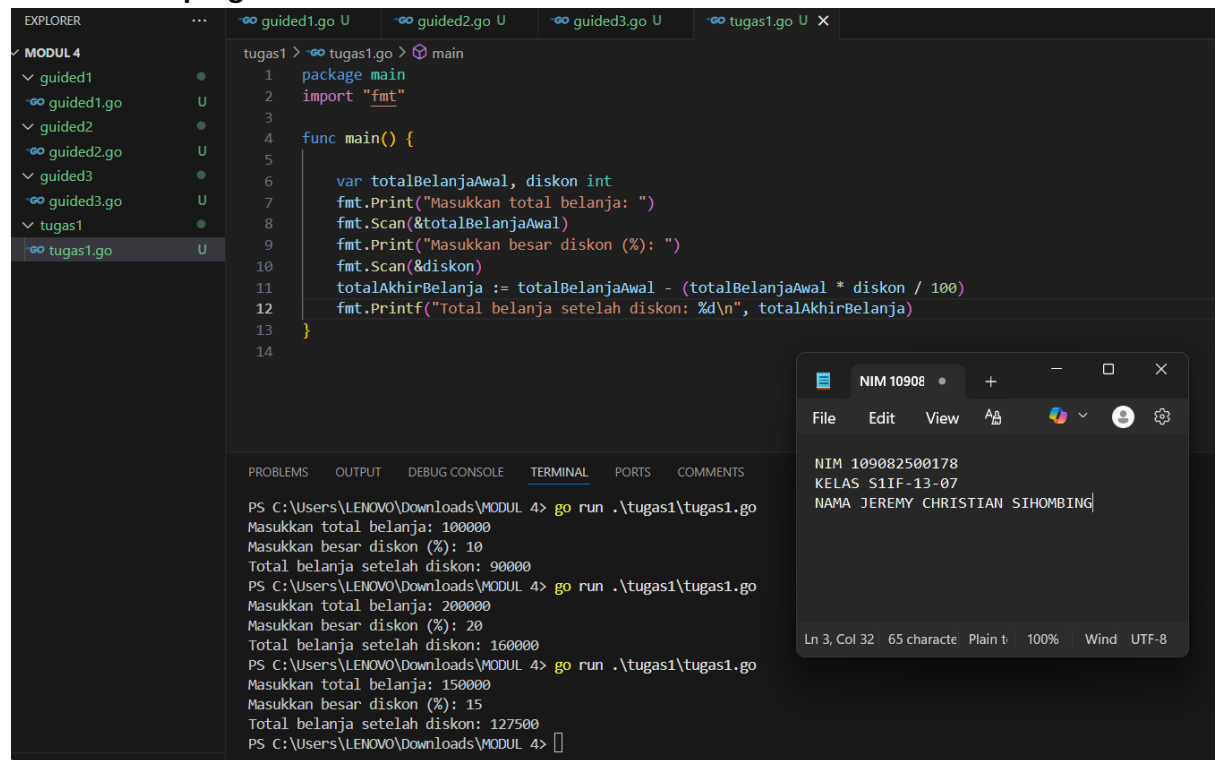
    fmt.Scan(&diskon)

    totalAkhirBelanja := totalBelanjaAwal - (totalBelanjaAwal
* diskon / 100)

    fmt.Printf("Total belanja setelah diskon: %d\n",
totalAkhirBelanja)

}
```


Screenshoot program



The screenshot shows a Go program in VS Code. The Explorer panel on the left shows a project structure with files `guided1.go`, `guided2.go`, `guided3.go`, and `tugas1.go`. The main editor displays the source code for `tugas1.go`, which defines a `main` function that takes total purchase and discount as input and calculates the final amount after discount.

```
tugas1 > .\tugas1.go > main
1 package main
2 import "fmt"
3
4 func main() {
5
6     var totalBelanjaAwal, diskon int
7     fmt.Print("Masukkan total belanja: ")
8     fmt.Scan(&totalBelanjaAwal)
9     fmt.Print("Masukkan besar diskon (%): ")
10    fmt.Scan(&diskon)
11    totalAkhirBelanja := totalBelanjaAwal - (totalBelanjaAwal * diskon / 100)
12    fmt.Printf("Total belanja setelah diskon: %d\n", totalAkhirBelanja)
13 }
14
```

The Terminal panel at the bottom shows the execution of the program with three different inputs:

```
PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\tugas1\tugas1.go
Masukkan total belanja: 100000
Masukkan besar diskon (%): 10
Total belanja setelah diskon: 90000
PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\tugas1\tugas1.go
Masukkan total belanja: 200000
Masukkan besar diskon (%): 20
Total belanja setelah diskon: 160000
PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\tugas1\tugas1.go
Masukkan total belanja: 150000
Masukkan besar diskon (%): 15
Total belanja setelah diskon: 127500
PS C:\Users\LENOVO\Downloads\MODUL 4>
```

A floating window in the bottom right corner displays the user's information:

```
NIM 109082500178
KELAS SIIF-13-07
NAMA JEREMY CHRISTIAN SIHOMBING
```

Deskripsi program

Program ini digunakan untuk menghitung total harga akhir setelah mendapatkan potongan diskon berdasarkan persentase yang diberikan. Ketika dijalankan, pengguna akan diminta memasukkan dua input, yaitu total belanja awal dan besar diskon dalam persen. Setelah itu, program akan menghitung jumlah potongan harga dengan cara mengalikan total belanja dengan persentase diskon, lalu hasilnya dikurangkan dari total belanja awal. Hasil akhirnya berupa total belanja yang sudah dipotong oleh diskon, kemudian ditampilkan di layar. Program ini membantu pengguna mengetahui berapa jumlah yang harus dibayar setelah mendapat potongan harga.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var bmi, tinggi, berat float64

    fmt.Print("Masukkan nilai BMI dan tinggi badan (meter): ")

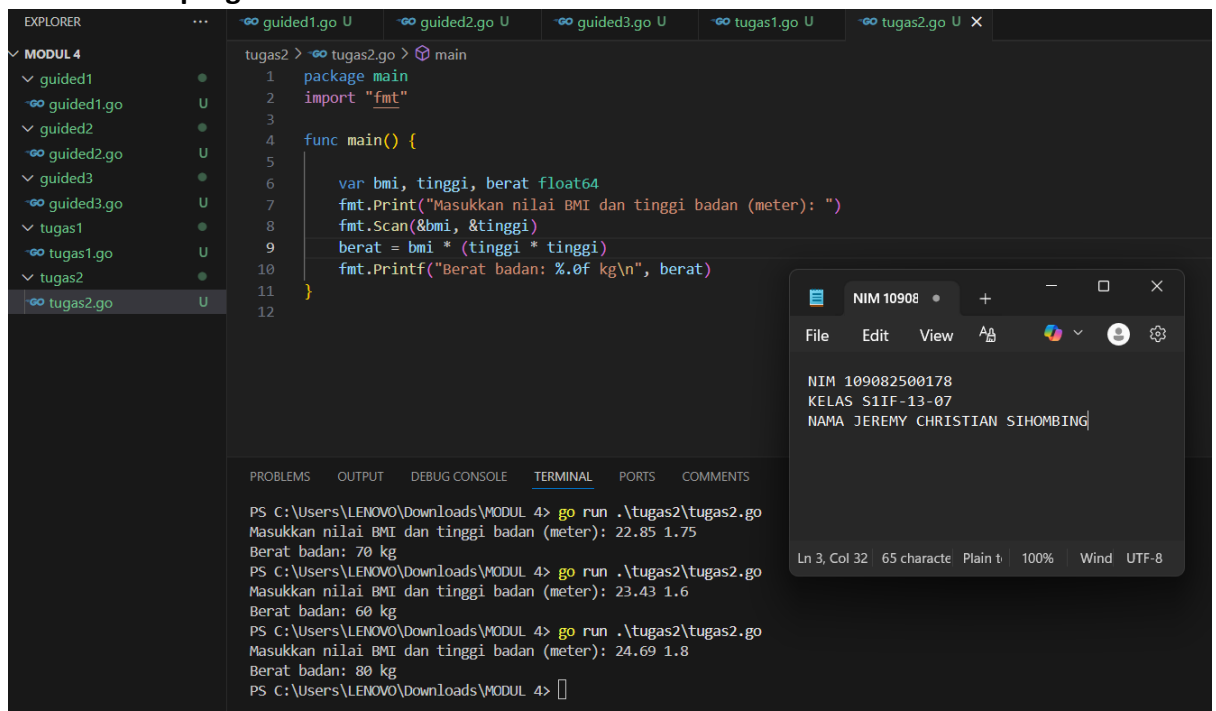
    fmt.Scan(&bmi, &tinggi)

    berat = bmi * (tinggi * tinggi)

    fmt.Printf("Berat badan: %.0f kg\n", berat)

}
```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk menghitung berat badan seseorang jika diketahui nilai BMI dan tinggi badannya dalam meter. Saat dijalankan, pengguna akan diminta untuk memasukkan dua data, yaitu nilai BMI dan tinggi badan. Rumus yang digunakan berasal dari rumus dasar BMI, yaitu $BMI = \text{berat} / (\text{tinggi}^2)$. Agar bisa mencari berat badan, rumus tersebut dibalik menjadi $\text{berat} = BMI \times (\text{tinggi}^2)$. Hasil dari perhitungan tersebut kemudian ditampilkan dalam satuan kilogram. Dengan program ini, pengguna dapat mengetahui berat badan berdasarkan nilai BMI dan tinggi badan yang dimasukkan.

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var x1, y1, x2, y2, x3, y3 float64
    fmt.Print("Masukkan koordinat titik A (x y): ")
    fmt.Scan(&x1, &y1)
    fmt.Print("Masukkan koordinat titik B (x y): ")
    fmt.Scan(&x2, &y2)
    fmt.Print("Masukkan koordinat titik C (x y): ")
    fmt.Scan(&x3, &y3)
    sisiab := math.Sqrt(math.Pow(x2-x1, 2) +
math.Pow(y2-y1, 2))
    sisibc := math.Sqrt(math.Pow(x3-x2, 2) +
math.Pow(y3-y2, 2))
    sisica := math.Sqrt(math.Pow(x1-x3, 2) +
math.Pow(y1-y3, 2))
    Terpanjang := sisiab
    if sisibc > Terpanjang {
        Terpanjang = sisibc
    }
    if sisica > Terpanjang {
        Terpanjang = sisica
    }
    fmt.Printf("Sisi terpanjang: %.2f\n", Terpanjang)
}
```

Screenshoot program

```
1 package main
2 import (
3     "fmt"
4     "math"
5 )
6 func main() {
7     var x1, y1, x2, y2, x3, y3 float64
8     fmt.Print("Masukkan koordinat titik A (x y): ")
9     fmt.Scan(&x1, &y1)
10    fmt.Print("Masukkan koordinat titik B (x y): ")
11    fmt.Scan(&x2, &y2)
12    fmt.Print("Masukkan koordinat titik C (x y): ")
13    fmt.Scan(&x3, &y3)
14    sisiab := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
15    sisibc := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
16    sisica := math.Sqrt(math.Pow(x1-x3, 2) + math.Pow(y1-y3, 2))
17    Terpanjang := sisiab
18    if sisibc > Terpanjang {
19        Terpanjang = sisibc
20    }
21    if sisica > Terpanjang {
22        Terpanjang = sisica
23    }
24    fmt.Printf("Sisi terpanjang: %.2f\n", Terpanjang)
25 }
```

PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\tugas3\tugas3.go
Masukkan koordinat titik A (x y): 1.0 1.0
Masukkan koordinat titik B (x y): 4.0 1.0
Masukkan koordinat titik C (x y): 2.0 5.0
Sisi terpanjang: 4.47

PS C:\Users\LENOVO\Downloads\MODUL 4> go run .\tugas3\tugas3.go
Masukkan koordinat titik A (x y): 0.0 0.0
Masukkan koordinat titik B (x y): 3.0 0.0
Masukkan koordinat titik C (x y): 3.0 4.0
Sisi terpanjang: 5.00

Deskripsi program

Program ini digunakan untuk mencari sisi terpanjang dari sebuah segitiga yang dibentuk oleh tiga titik koordinat dalam bidang kartesius dua dimensi. Pengguna diminta untuk memasukkan tiga titik, yaitu A, B, dan C, masing-masing berisi nilai koordinat x dan y. Setelah semua titik dimasukkan, program menghitung panjang setiap sisi segitiga menggunakan rumus jarak dua titik, yaitu $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Untuk melakukan perhitungan kuadrat dan akar, digunakan fungsi `math.Pow` dan `math.Sqrt`. Setelah ketiga sisi didapatkan, program membandingkan hasilnya untuk menentukan sisi yang paling panjang. Terakhir, hasil panjang sisi terpanjang tersebut ditampilkan dengan dua angka di belakang koma agar hasilnya terlihat lebih rapi dan mudah dibaca.