

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 4
TIPE DATA & VARIABEL**



Disusun oleh:

RAFI IMAM NASRULLAH

109082530010

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 jam, menit, detik int

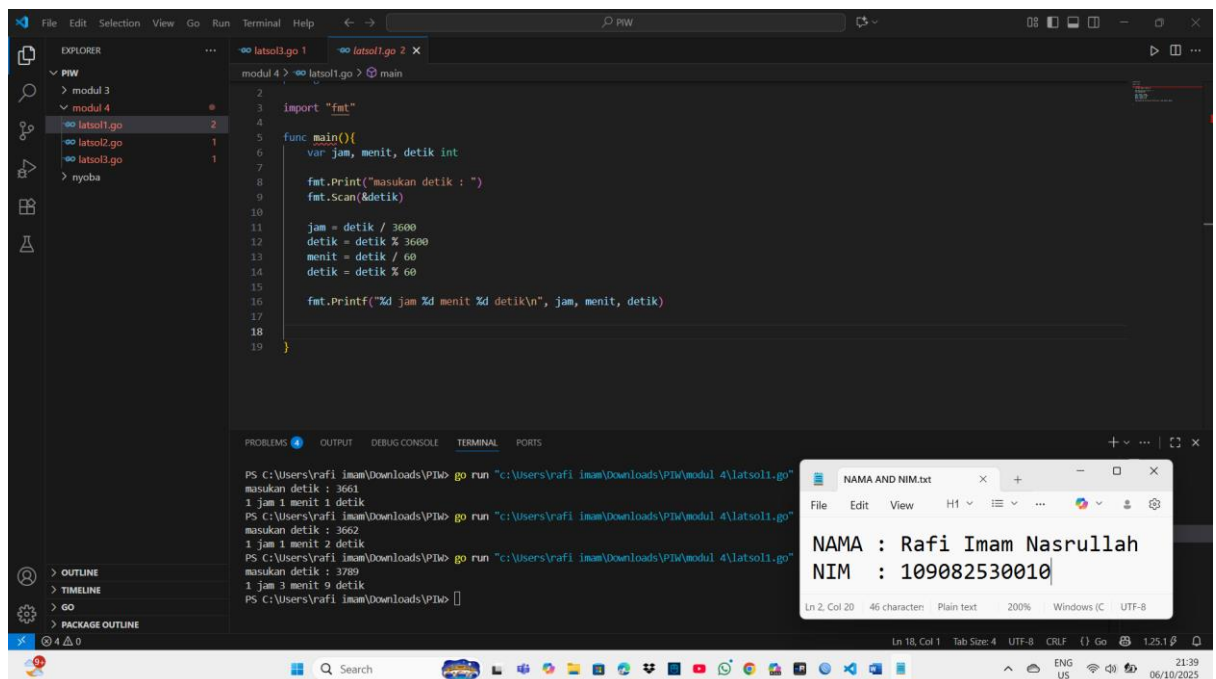
    fmt.Print("masukan detik : ")
    fmt.Scan(&detik)

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

    fmt.Printf("%d jam %d menit %d detik\n", jam, menit,
detik)

}
```

Screenshoot program



```
modul 4 > latsoli.go 1
modul 4 > latsoli.go 2 X

import "fmt"

func main() {
    var jam, menit, detik int

    fmt.Print("masukan detik : ")
    fmt.Scan(&detik)

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

    fmt.Printf("%d jam %d menit %d detik\n", jam, menit, detik)
}
```

```
PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\latsoli.go"
masukan detik : 3661
1 jam 1 menit 1 detik
PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\latsoli.go"
masukan detik : 3662
1 jam 1 menit 2 detik
PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\latsoli.go"
masukan detik : 3780
1 jam 3 menit 0 detik
PS C:\Users\rafi imam\Downloads\PIW>
```

NAMA AND NIM.txt

NAMA : Rafi Imam Nasrullah
NIM : 109082530010

Deskripsi program

Program ini mengubah jumlah detik menjadi format waktu yang lebih mudah dibaca: jam, menit, detik. Pada awal program dideklarasikan tiga variabel bertipe int: jam, menit, dan detik. Program meminta pengguna memasukkan satu angka (total detik) lewat `fmt.Scan(&detik)`. Setelah itu program menghitung jam dulu, lalu sisa detiknya dipakai untuk menghitung menit, dan sisa terakhir adalah detik.

Contoh konkret dengan input 3661: pertama program menghitung $\text{jam} = \text{detik} / 3600$. Karena $1 \text{ jam} = 3600 \text{ detik}$, operasi pembagian pada tipe int memberikan hasil pembulatan ke bawah (truncated). Jadi $\text{jam} = 3661 / 3600 = 1$. Selanjutnya program mengganti nilai detik dengan sisa setelah dikurangi jam penuh tadi menggunakan operator modulo: $\text{detik} = \text{detik} \% 3600$, sehingga $\text{detik} = 3661 \% 3600 = 61$ (sisa detik setelah diambil 1 jam). Dari sisa 61 detik itu dihitung $\text{menit} = \text{detik} / 60$, yaitu $61 / 60 = 1$ menit (lagi-lagi pembagian integer membuang desimal). Terakhir $\text{detik} = \text{detik} \% 60$ memberi sisa detik terakhir: $61 \% 60 = 1$. Jadi hasil akhirnya yang dicetak adalah 1 jam 1 menit 1 detik, ditampilkan oleh `fmt.Printf("%d jam %d menit %d detik\n", jam, menit, detik)`.

Beberapa hal teknis penting yang perlu diketahui: operator `/` pada tipe integer melakukan pembagian bulat (mis. $61/60 = 1$, bukan 1.0166), sedangkan operator `%`

memberi sisa pembagian. Program ini menerima nilai detik beberapa kali (pertama berisi total detik, lalu diubah menjadi sisa setelah jam, lalu sisa setelah menit) Contoh input lain untuk memperjelas: input 59 menghasilkan 0 jam 0 menit 59 detik; input 3600 menghasilkan 1 jam 0 menit 0 detik; input 7322 (atau 2 jam, 2 menit, 2 detik) melalui langkah yang sama menjadi 2 jam 2 menit 2 detik.

2. Guided 2

Source Code

```
package main

import "fmt"

func main(){

    var dig, dig1, dig2, dig3 int

    var hasil bool

    fmt.Print("masukan angka: ")

    fmt.Scanln(&dig)

    dig1 = dig / 100

    dig2 = (dig % 100) / 10

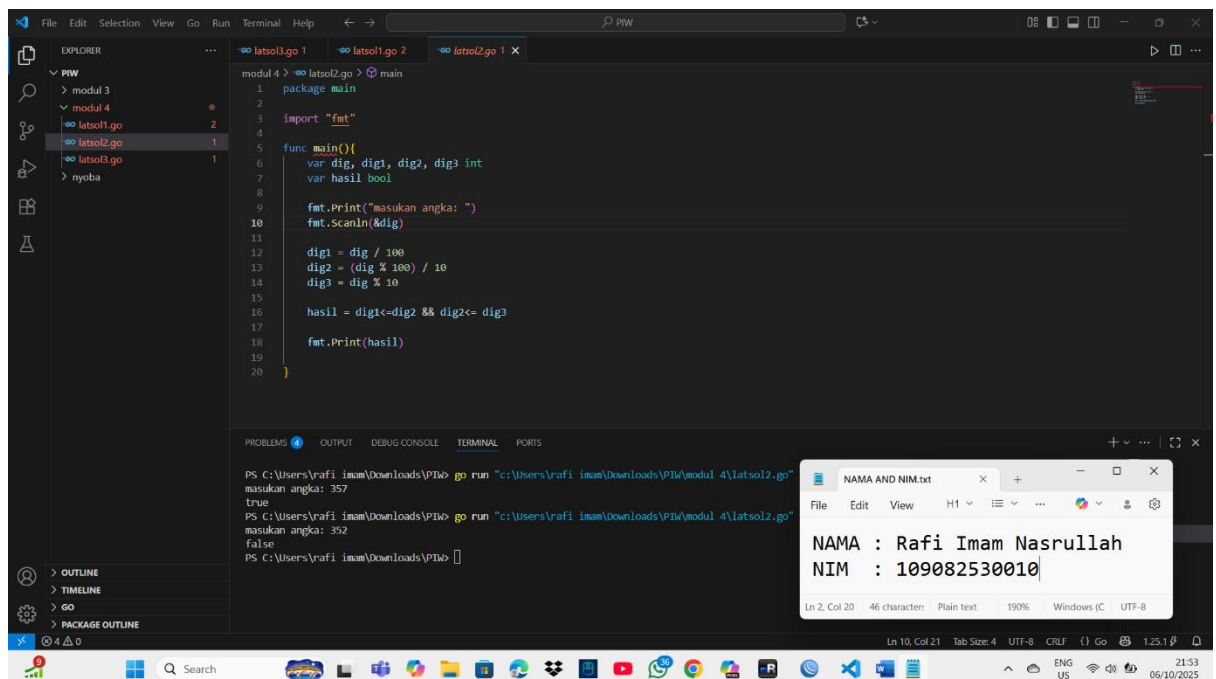
    dig3 = dig % 10

    hasil = dig1<=dig2 && dig2<= dig3

    fmt.Print(hasil)

}
```

Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func main(){
6     var dig, dig1, dig2, dig3 int
7     var hasil bool
8
9     fmt.Print("masukan angka: ")
10    fmt.Scanln(&dig)
11
12    dig1 = dig / 100
13    dig2 = (dig % 100) / 10
14    dig3 = dig % 10
15
16    hasil = dig1<=dig2 && dig2<= dig3
17
18    fmt.Print(hasil)
19
20 }
```

PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\latso2.go"

masukan angka: 357

true

PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\latso2.go"

masukan angka: 352

false

PS C:\Users\rafi imam\Downloads\PIW>

Deskripsi program

Program ini berfungsi untuk mengecek apakah tiga digit angka yang dimasukkan oleh pengguna tersusun secara menaik (dari kecil ke besar atau sama besar). Ketika program dijalankan, pengguna diminta memasukkan sebuah angka tiga digit melalui `fmt.Scanln(&dig)`. Misalnya pengguna mengetik angka 357. Program kemudian memecah angka itu menjadi tiga bagian: digit ratusan ($\text{dig1} = \text{dig} / 100$, hasilnya 3), digit puluhan ($\text{dig2} = (\text{dig} \% 100) / 10$, hasilnya 5), dan digit satuan ($\text{dig3} = \text{dig} \% 10$, hasilnya 7). Setelah itu, program melakukan perbandingan menggunakan logika `hasil = dig1 <= dig2 && dig2 <= dig3`, artinya digit pertama harus lebih kecil atau sama dengan digit kedua, dan digit kedua harus lebih kecil atau sama dengan digit ketiga. Jika kedua kondisi itu benar, maka hasilnya `true`, sebaliknya jika tidak berurutan menaik maka hasilnya `false`. Terakhir, program menampilkan nilai hasil tersebut ke layar. Jadi jika inputnya 357, output-nya adalah `true`, karena $3 \leq 5 \leq 7$, sedangkan jika inputnya 352, hasilnya `false`, karena urutannya menurun.

3. Guided 3

Source Code

```
package main

import (
    "fmt"
)

func main() {
    var berat, tinggi, BMI float64

    fmt.Print("Masukan berat : ")
    fmt.Scanln(&berat)

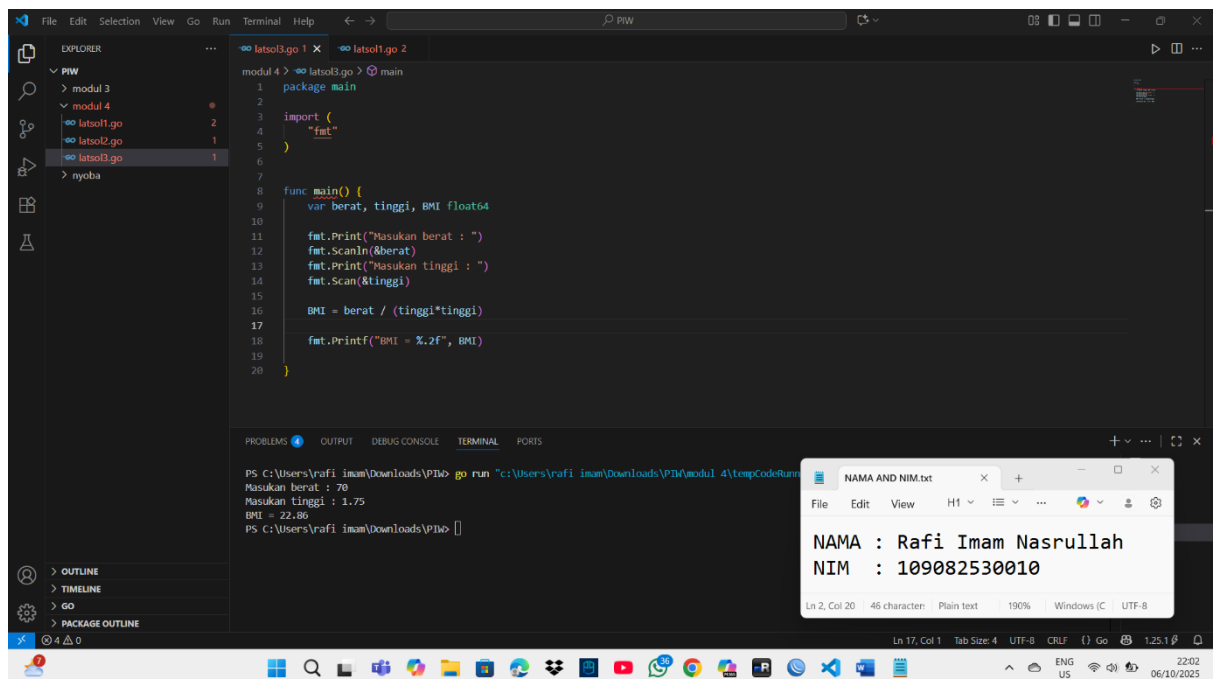
    fmt.Print("Masukan tinggi : ")
    fmt.Scan(&tinggi)

    BMI = berat / (tinggi*tinggi)

    fmt.Printf("BMI = %.2f", BMI)

}
```

Screenshoot program



```
1 package main
2
3 import (
4     "fmt"
5 )
6
7
8 func main() {
9     var berat, tinggi, BMI float64
10
11     fmt.Print("Masukan berat : ")
12     fmt.Scanln(&berat)
13     fmt.Print("Masukan tinggi : ")
14     fmt.Scan(&tinggi)
15
16     BMI = berat / (tinggi*tinggi)
17
18     fmt.Printf("BMI = %.2f", BMI)
19
20 }
```

PS C:\Users\rafi imam\Downloads\PTIK> go run "c:\Users\rafi imam\Downloads\PTIK\modul 4\tempCodeRunnerFile1.go"

Masukan berat : 70
Masukan tinggi : 1.75
BMI = 22.86
PS C:\Users\rafi imam\Downloads\PTIK>

NAMA AND NIM.txt

NAMA : Rafi Imam Nasrullah
NIM : 109082530010

Deskripsi program

Program ini digunakan untuk menghitung BMI (Body Mass Index) atau Indeks Massa Tubuh, Program ini menggunakan tiga variabel bertipe float64, yaitu berat, tinggi, dan BMI. Ketika dijalankan, program pertama-tama menampilkan pesan untuk meminta pengguna memasukkan berat badan dalam kilogram melalui `fmt.Print("Masukan berat : ")`, lalu membaca nilainya menggunakan `fmt.Scanln(&berat)`. Setelah itu, program meminta pengguna memasukkan tinggi badan dalam meter dengan `fmt.Print("Masukan tinggi : ")` dan membaca nilainya lewat `fmt.Scan(&tinggi)`. Setelah kedua nilai dimasukkan, program menghitung nilai BMI menggunakan rumus $BMI = \text{berat} / (\text{tinggi} * \text{tinggi})$. Rumus ini berarti berat badan dibagi dengan hasil perkalian tinggi badan dengan dirinya sendiri (tinggi kuadrat). Sebagai contoh, jika pengguna memasukkan berat badan 70 kg dan tinggi badan 1.75 meter, maka perhitungannya adalah $BMI = 70 / (1.75 * 1.75) = 70 / 3.0625 = 22.86$. Hasil perhitungan ini kemudian ditampilkan ke layar dengan perintah `fmt.Printf("BMI = %.2f", BMI)`, di mana format `%.2f` menunjukkan bahwa hasil akan ditampilkan dengan dua angka di belakang koma. Jadi output yang muncul adalah "BMI = 22.86".

TUGAS

1. Tugas 1

Source code

```
package main

import (
    "fmt"
)

func main() {
    var totalAwal int
    var diskon int

    fmt.Print("Masukkan total belanja awal: ")
    fmt.Scanln(&totalAwal)

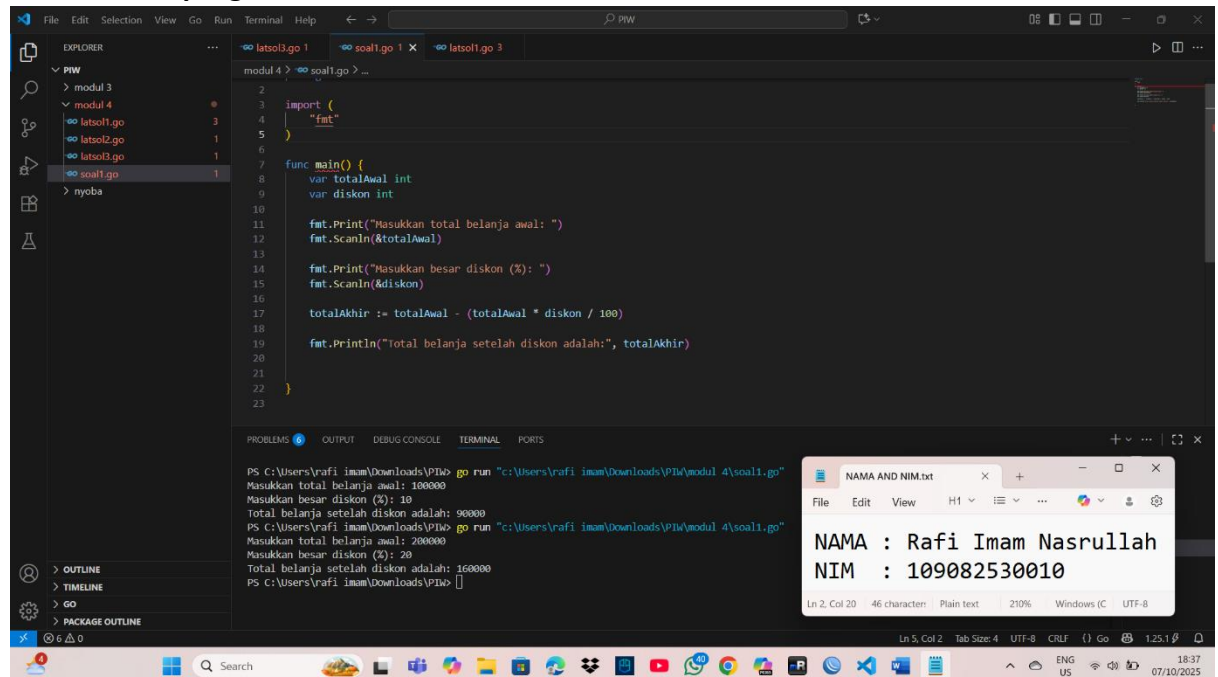
    fmt.Print("Masukkan besar diskon (%): ")
    fmt.Scanln(&diskon)

    totalAkhir := totalAwal - (totalAwal * diskon / 100)

    fmt.Println("Total belanja setelah diskon adalah:",
totalAkhir)

}
```


Screenshoot program



The screenshot shows a Go program in a VS Code editor. The program is located in `modul 4\soal1.go`. The code is as follows:

```
1 2
3  import (
4      "fmt"
5  )
6
7  func main() {
8      var totalAwal int
9      var diskon int
10
11     fmt.Print("Masukkan total belanja awal: ")
12     fmt.Scanln(&totalAwal)
13
14     fmt.Print("Masukkan besar diskon (X): ")
15     fmt.Scanln(&diskon)
16
17     totalAkhir := totalAwal - (totalAwal * diskon / 100)
18
19     fmt.Println("Total belanja setelah diskon adalah:", totalAkhir)
20
21 }
22
23
```

The terminal output shows the program being run and the user inputting values:

```
PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\soal1.go"
Masukkan total belanja awal: 100000
Masukkan besar diskon (X): 10
Total belanja setelah diskon adalah: 90000
PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\soal1.go"
Masukkan total belanja awal: 200000
Masukkan besar diskon (X): 20
Total belanja setelah diskon adalah: 160000
PS C:\Users\rafi imam\Downloads\PIW>
```

A separate window titled "NAMA AND NIM.txt" displays the user's name and NIM:

```
NAMA : Rafi Imam Nasrullah
NIM : 109082530010
```

Deskripsi program

Program ini dibuat menggunakan bahasa pemrograman Golang dan berfungsi untuk menghitung total harga akhir setelah mendapatkan diskon. Cara kerjanya sangat sederhana. Pertama, program akan meminta pengguna untuk memasukkan total belanja awal, yaitu jumlah uang sebelum potongan harga. Setelah itu, pengguna juga diminta untuk memasukkan besar diskon dalam persen. Nilai yang dimasukkan ini akan digunakan untuk menghitung berapa besar potongan harga yang diterima. Rumus yang digunakan dalam perhitungan adalah $\text{totalAkhir} = \text{totalAwal} - (\text{totalAwal} * \text{diskon} / 100)$. Pada rumus tersebut, bagian $(\text{totalAwal} * \text{diskon} / 100)$ merupakan nilai potongan harga yang dihitung berdasarkan persentase diskon dari total belanja awal, kemudian dikurangkan dari total belanja awal untuk mendapatkan hasil akhir. Setelah proses perhitungan selesai, program akan menampilkan total belanja akhir setelah diskon diterapkan.

Sebagai contoh, jika pengguna memasukkan total belanja awal 100000 dan diskon 10, maka perhitungannya menjadi $100000 - (100000 * 10 / 100)$ yang hasilnya adalah 90000. Begitu juga jika total belanja 200000 dengan diskon 20, maka hasilnya 160000, dan jika 150000 dengan diskon 15, maka hasilnya 127500.

2. Tugas 2

Source code

```
package main

import

"fmt"

func main() {

    var BMI, tinggi, berat float64

    fmt.Print("Masukkan nilai BMI: ")

    fmt.Scanln(&BMI)

    fmt.Print("Masukkan tinggi badan (meter): ")

    fmt.Scanln(&tinggi)

    berat = BMI * (tinggi * tinggi)

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

}
```

Screenshoot program

The screenshot shows a Go IDE with a project named 'PIW'. The file explorer on the left shows a directory structure with files like 'modul 3', 'modul 4', 'latso1.go', 'latso2.go', 'latso3.go', 'latso4.go', 'soal1.go', 'soal2.go', and 'nyoba'. The main editor displays the code for 'soal2.go', which is a Go program to calculate BMI. The code is as follows:

```
modul 4 > soal2.go > main
3 import
4
5 "fmt"
6
7
8 func main() {
9     var BMI, tinggi, berat float64
10
11     fmt.Print("Masukkan nilai BMI: ")
12     fmt.Scanln(&BMI)
13
14     fmt.Print("Masukkan tinggi badan (meter): ")
15     fmt.Scanln(&tinggi)
16
17     berat = BMI * (tinggi * tinggi)
18
19     fmt.Printf("Berat badan adalah: %.0f\n", berat)
20
21 }
22
```

The terminal at the bottom shows the execution of the program with three test cases:

```
PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\soal2.go"
Masukkan nilai BMI: 22.85
Masukkan tinggi badan (meter): 1.75
Berat badan adalah: 70
PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\soal2.go"
Masukkan nilai BMI: 23.42
Masukkan tinggi badan (meter): 1.6
Berat badan adalah: 60
PS C:\Users\rafi imam\Downloads\PIW> go run "c:\Users\rafi imam\Downloads\PIW\modul 4\soal2.go"
Masukkan nilai BMI: 24.69
Masukkan tinggi badan (meter): 1.8
Berat badan adalah: 80
```

Overlaid on the terminal is a text file named 'NAMA AND NIM.txt' containing the following text:

```
NAMA : Rafi Imam Nasrullah
NIM : 109082530010
```

Deskripsi program

Program ini dibuat untuk menghitung berat badan seseorang jika diketahui nilai BMI (Body Mass Index) dan tinggi badannya dalam meter. Jadi, cara kerja program ini adalah dengan meminta dua input dari pengguna, yaitu nilai BMI dan tinggi badan. Setelah itu, program akan menghitung berat badan menggunakan rumus matematika $\text{berat} = \text{BMI} \times (\text{tinggi} \times \text{tinggi})$. Rumus ini berasal dari rumus dasar BMI, yaitu $\text{BMI} = \text{berat} \div (\text{tinggi} \times \text{tinggi})$, yang kemudian diubah untuk mencari berat badan. Artinya, tinggi badan dikalikan dengan dirinya sendiri terlebih dahulu (untuk mendapatkan nilai tinggi kuadrat), lalu hasilnya dikalikan dengan nilai BMI. Dari hasil perkalian tersebut, kita bisa mengetahui berat badan seseorang dalam satuan kilogram. Misalnya, jika seseorang memiliki BMI 22.85 dan tinggi badan 1.75 meter, maka berat badannya dihitung dengan rumus $22.85 \times (1.75 \times 1.75)$ yang menghasilkan 70 kilogram. Begitu juga, jika BMI-nya 23.42 dan tinggi badannya 1.6 meter maka beratnya 60 kilogram, dan jika BMI-nya 24.69 dengan tinggi 1.8 meter maka beratnya 80 kilogram.

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var ax, ay float64
    var bx, by float64
    var cx, cy float64

    fmt.Print("Masukkan koordinat titik A (x y): ")
    fmt.Scan(&ax, &ay)

    fmt.Print("Masukkan koordinat titik B (x y): ")
    fmt.Scan(&bx, &by)

    fmt.Print("Masukkan koordinat titik C (x y): ")
    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))

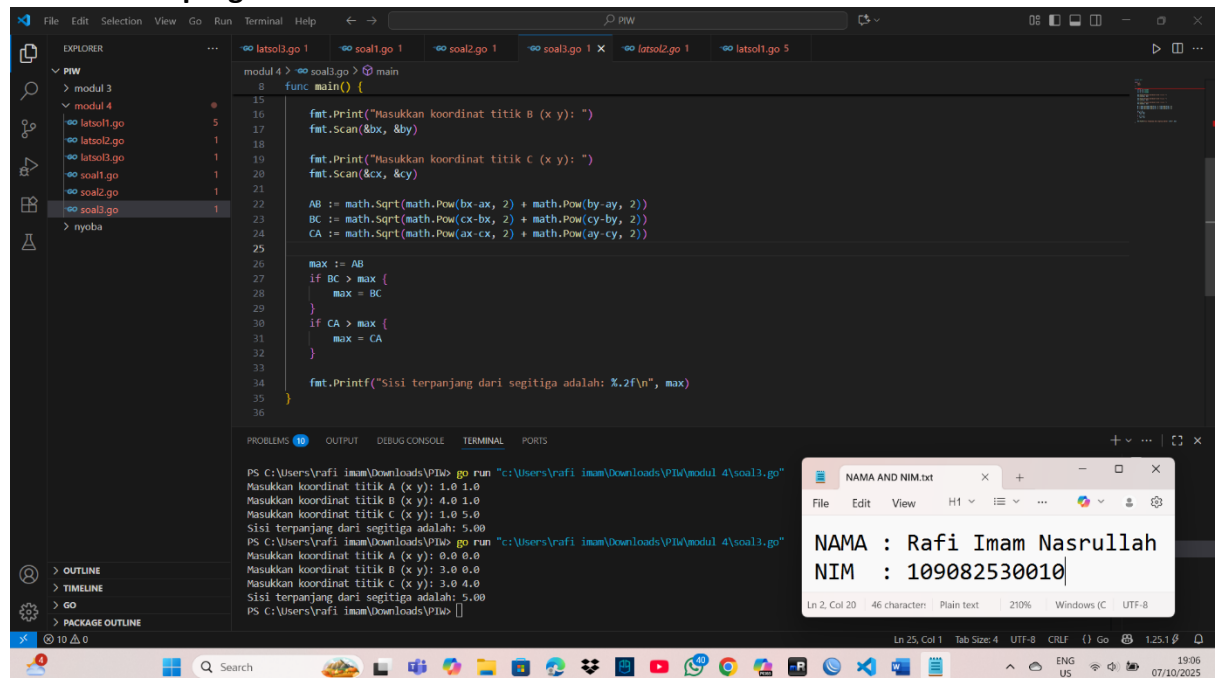
    max := AB
    if BC > max {
        max = BC
    }
    if CA > max {
        max = CA
    }
}
```

```

        fmt.Printf("Sisi terpanjang dari segitiga adalah:
%.2f\n", max)
    }
}

```

Screenshoot program



Deskripsi program

Program ini tujuannya adalah untuk mencari sisi terpanjang dari sebuah segitiga berdasarkan tiga titik koordinat yang dimasukkan oleh pengguna, yaitu titik A, B, dan C.

Pertama, program meminta pengguna memasukkan koordinat titik A, kemudian titik B, lalu titik C. Masing-masing titik terdiri dari dua nilai, yaitu posisi x dan y. Misalnya, pengguna bisa memasukkan "1 1" untuk titik A yang berarti $x = 1$ dan $y = 1$. Nilai-nilai ini kemudian disimpan ke dalam variabel ax , ay , bx , by , cx , dan cy .

Setelah semua titik dimasukkan, program menghitung panjang ketiga sisi segitiga. Sisi AB dihitung dari jarak antara titik A dan B, sisi BC dari titik B ke C, dan sisi CA dari titik C ke A. Rumus yang digunakan untuk menghitung jarak antar dua titik adalah akar dari $(x_2 - x_1)^2 + (y_2 - y_1)^2$. Dalam program ini, rumus tersebut dihitung menggunakan fungsi `math.Sqrt()` untuk akar dan `math.Pow()` untuk pangkat dua.

Hasil perhitungan panjang ketiga sisi disimpan dalam variabel AB, BC, dan CA. Kemudian program mencari nilai yang paling besar di antara ketiganya. Variabel max digunakan untuk menyimpan panjang sisi terpanjang. Awalnya, max diisi dengan nilai AB, lalu dibandingkan dengan BC dan CA. Jika ada yang lebih besar dari max, maka nilainya diganti dengan sisi yang lebih panjang.

Terakhir, program menampilkan hasilnya di layar dengan format dua angka di belakang koma. Pesan yang muncul akan menunjukkan sisi terpanjang dari segitiga tersebut. Misalnya, jika sisi terpanjangnya adalah 5, maka akan muncul tulisan: Sisi terpanjang dari segitiga adalah: 5.00