

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 04

I/O Tipe Data & Variabel



Disusun oleh:

Nabil Nailur Ridho

109082530008

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.Scan(&detik)

    jam = detik / 3600

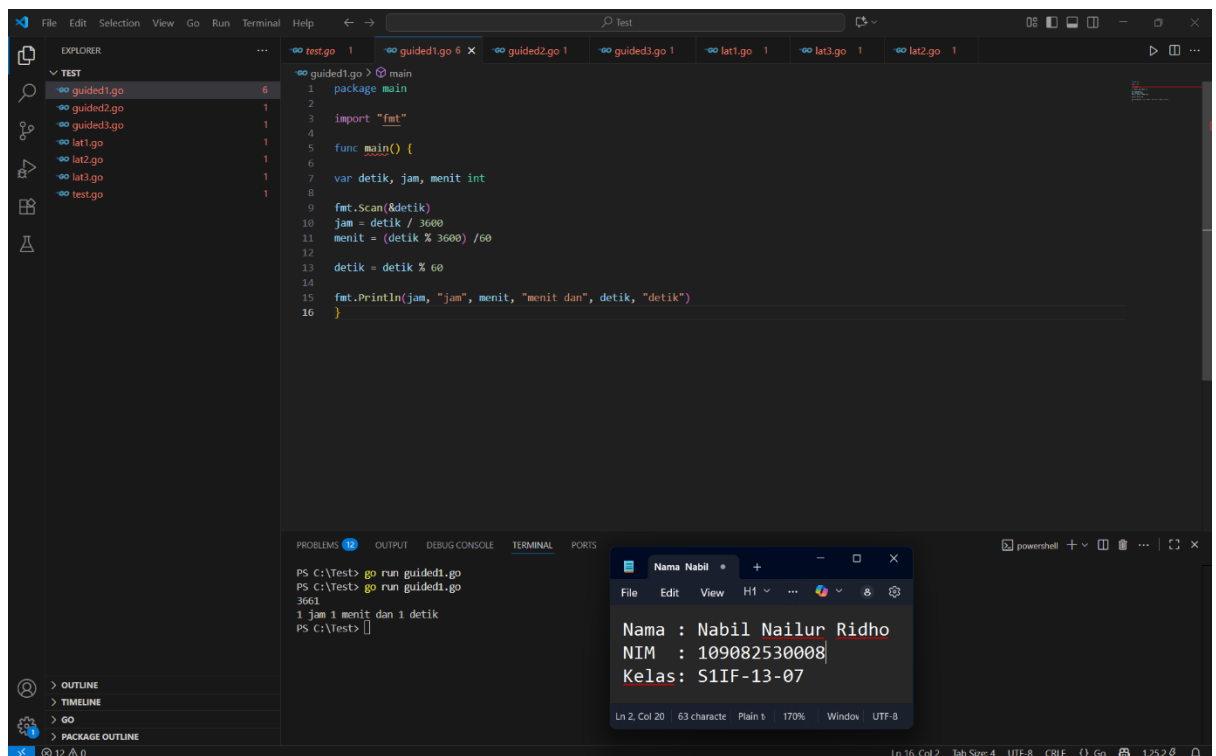
    menit = (detik % 3600) / 60

    detik = detik % 60

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

}
```

Screenshoot program



Deskripsi program

Program go ini berfungsi untuk mengubah durasi waktu yang di input atau di masukkan sama kita sendiri dalam satuan perak, lalu mengubah nya dari satuan perak menjadi jam menit dan detik.

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
```

```

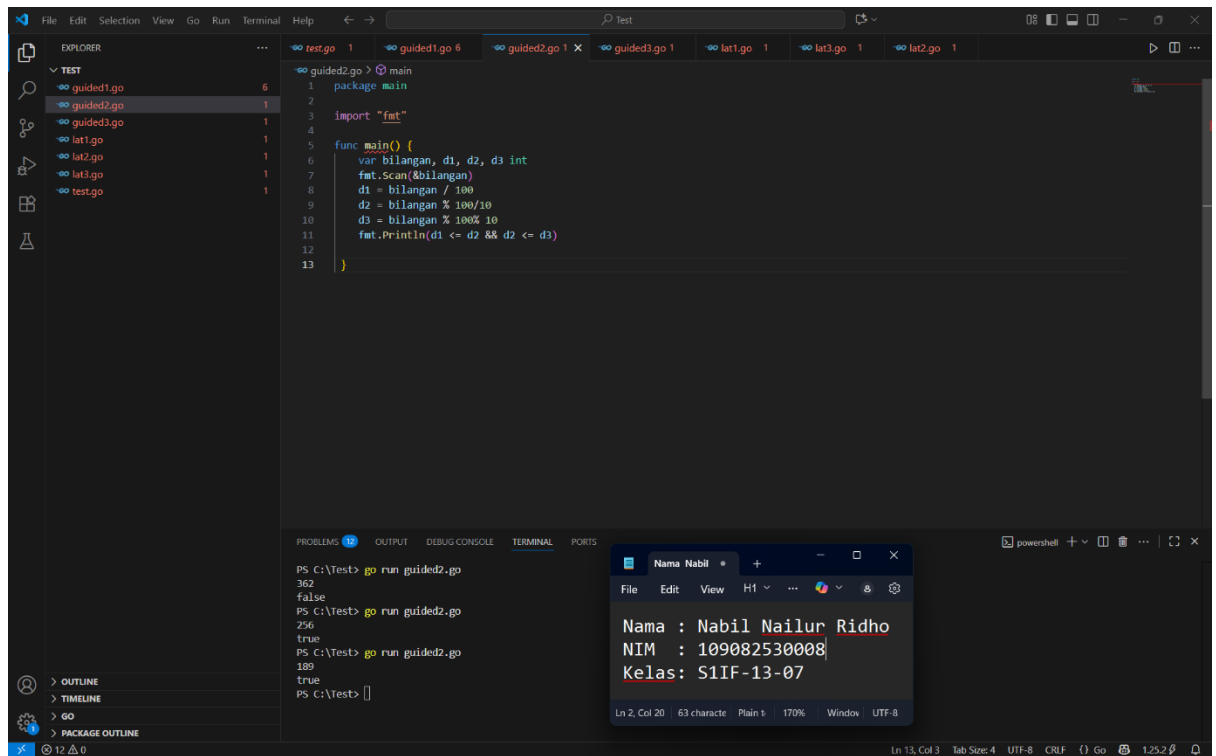
d3 = bilangan % 100% 10

fmt.Println(d1 <= d2 && d2 <= d3)

}

```

Screenshoot program



Deskripsi program

Program Go ini berfungsi untuk memeriksa urutan digit dari sebuah bilangan bulat tiga angka. Program ini pertama-tama akan meminta pengguna untuk memasukkan sebuah bilangan. Selanjutnya, program akan memecah bilangan tersebut menjadi tiga digit terpisah: ratusan, puluhan, dan satuan. Akhirnya, program akan mengevaluasi apakah digit-digit tersebut tersusun dalam urutan menaik, yaitu apakah digit pertama lebih kecil atau sama dengan digit kedua, dan digit kedua lebih kecil atau sama dengan digit ketiga, lalu mencetak hasilnya apakah true atau false.

Guided 3

Source Code

```

package main

import "fmt"

```

```

func main() {

    var beratBadan, tinggiBadan, bmi float64

    fmt.Scan(&beratBadan, &tinggiBadan)

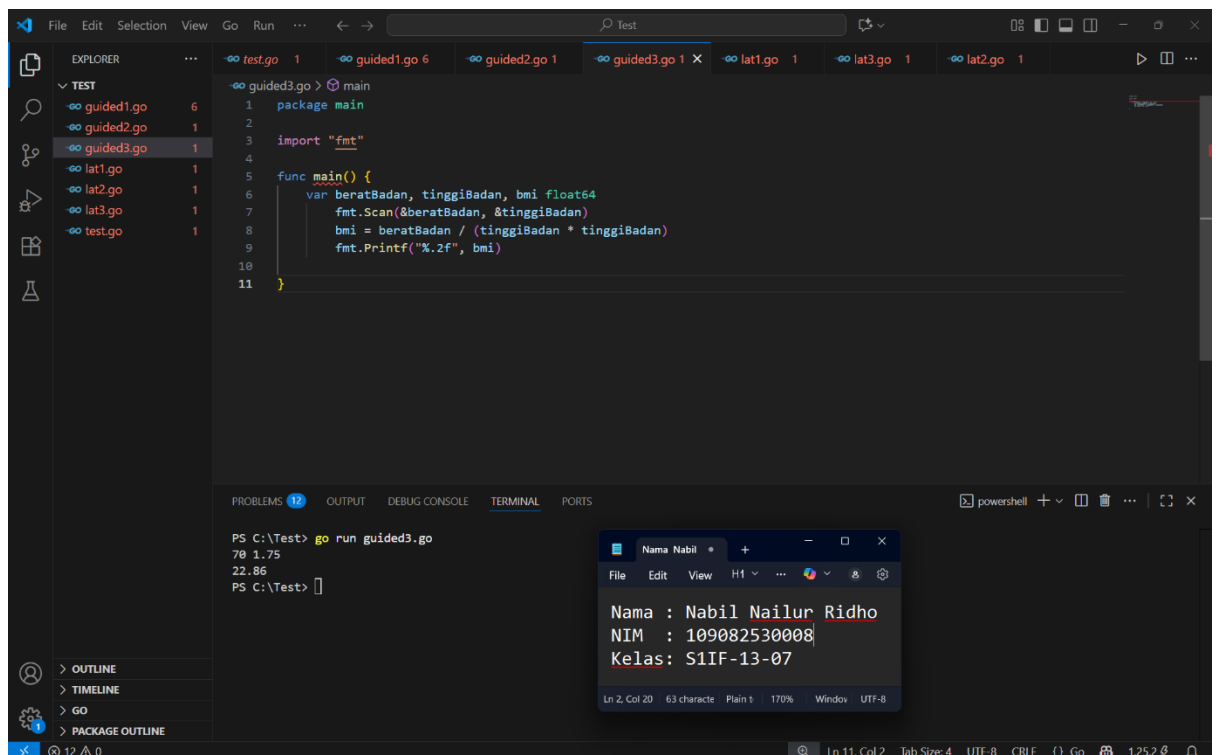
    bmi = beratBadan / (tinggiBadan * tinggiBadan)

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

}

```

Screenshoot program



Deskripsi program

program Go ini tuh kayak kalkulator buat ngitung BMI. Jadi kita cuma perlu masukan berat badan sama tinggi badan. Setelah itu, programnya bakal langsung ngitung BMI kamu pake rumus biasa: $\text{berat} / \text{tinggi} * \text{tinggi}$. Lalu hasilnya nanti ditampilin di layar, tapi angkanya dibuletin jadi dua digit aja di belakang koma biar rapi, contoh di output terminal nya menjadi 22.86.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var hargabarang, diskon float64

    fmt.Print("Masukkan jumlah harga barang: ")

    fmt.Scan(&hargabarang)

    fmt.Print("Masukan Harga diskon: ")

    fmt.Scan(&diskon)

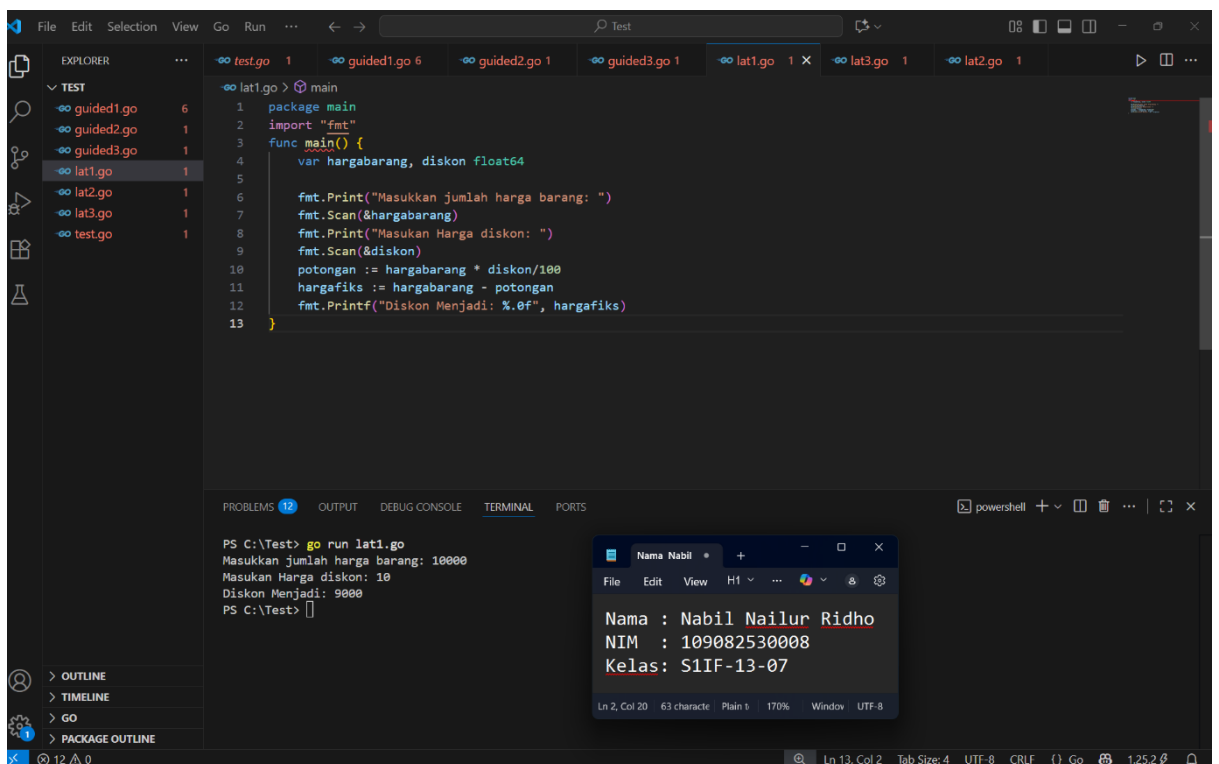
    potongan := hargabarang * diskon/100

    hargafiks := hargabarang - potongan

    fmt.Printf("Diskon Menjadi: %.0f", hargafiks)

}
```

Screenshoot program



Deskripsi program

Program Go yang ini berfungsi sebagai kalkulator diskon sederhana. Cara kerjanya itu, program akan meminta kamu memasukkan dua input an yaitu harga awal barang dan persentase diskonnya. Lalu program akan menghitung berapa besar potongannya, lalu mengurangkannya dari harga awal. Hasil akhirnya, yaitu harga barang setelah didiskon, akan langsung ditampilkan di layar sebagai angka bulat.

2. Tugas 2

Source code

```
package main

import "fmt"

func main(){

    var bmi, tinggiBadan float64

    fmt.Print("Masukan bmi: ")

    fmt.Scan(&bmi)

    fmt.Print("Masukkan Tinggi Badan: ")

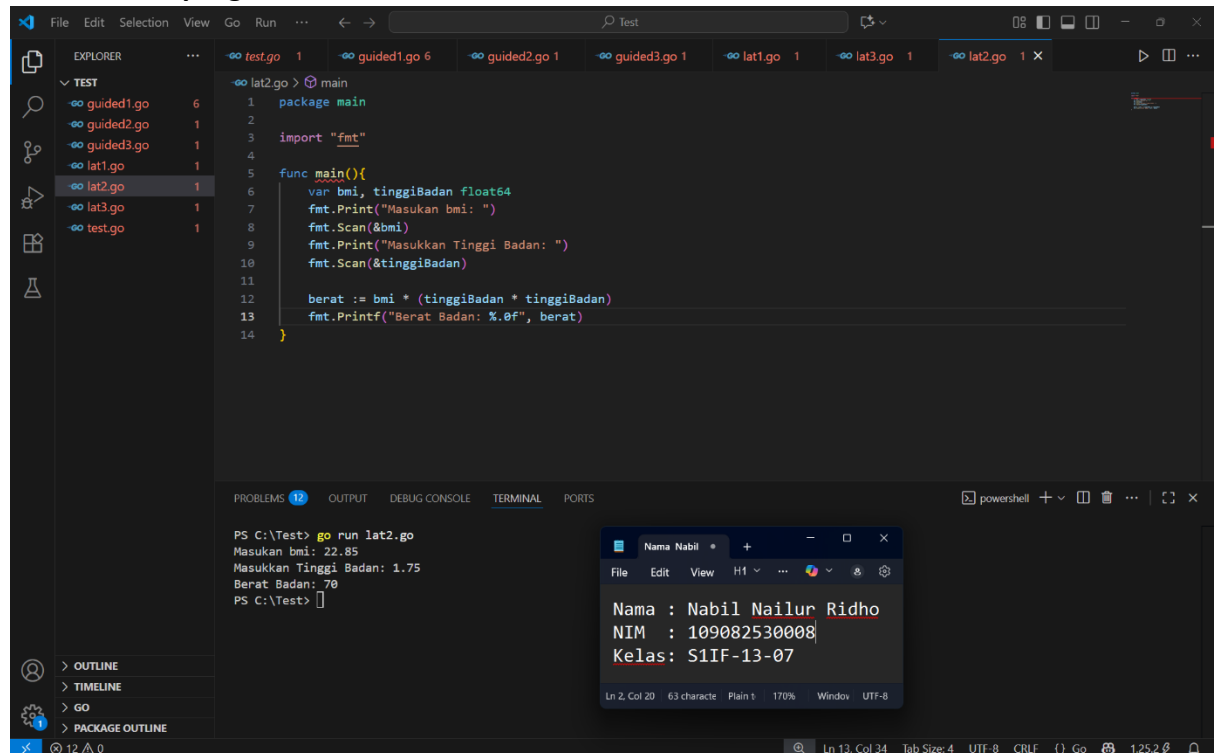
    fmt.Scan(&tinggiBadan)

    berat := bmi * (tinggiBadan * tinggiBadan)

    fmt.Printf("Berat Badan: %.0f", berat)

}
```

Screenshoot program



Deskripsi program

Program Go ini berfungsi untuk menghitung berat badan ideal terbalik. Berbeda dari kalkulator BMI biasa yang menghitung BMI dari berat dan tinggi. Program ini untuk menghitung berat badan dari nilai BMI dan tinggi badan yang kamu masukkan. Jadi program akan meminta kamu buat input target BMI dan tinggi badanmu. Kemudian, program akan menghitung berapa berat badan yang seharusnya kamu miliki untuk mencapai BMI tersebut dengan rumus: $\text{berat} = \text{bmi} * (\text{tinggi} * \text{tinggi})$. Lalu output nya ditampilkan sebagai angka bulat.

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

func main(){
    var x1, y1, x2, y2, x3, y3 float64
```



```

fmt.Println("Masukan nilai A: ")

fmt.Scan(&x1, &y1)

fmt.Println("Masukan nilai B: ")

fmt.Scan(&x2, &y2)

fmt.Println("Masukan nilai C: ")

fmt.Scan(&x3, &y3)

sisiAB := math.Sqrt(math.Pow(x2 - x1, 2) + math.Pow(y2 - y1, 2))
sisiAC := math.Sqrt(math.Pow(x3 - x1, 2) + math.Pow(y3 - y1, 2))
sisiBC := math.Sqrt(math.Pow(x3 - x2, 2) + math.Pow(y3 - y2, 2))

sisiTerpanjang := sisiAB
if sisiAC > sisiTerpanjang {
    sisiTerpanjang = sisiAC
} else if sisiBC > sisiTerpanjang {
    sisiTerpanjang = sisiBC
}

fmt.Printf("Sisi Terpanjang Adalah: %.0f", sisiTerpanjang)
}

```

Screenshoot program

```

func main(){
    var x1, y1, x2, y2, x3, y3 float64
    fmt.Println("Masukan nilai A: ")
    fmt.Scan(&x1, &y1)
    fmt.Println("Masukan nilai B: ")
    fmt.Scan(&x2, &y2)
    fmt.Println("Masukan nilai C: ")
    fmt.Scan(&x3, &y3)
    sisiAB := math.Sqrt(math.Pow(x2 - x1, 2) + math.Pow(y2 - y1, 2))
    sisiAC := math.Sqrt(math.Pow(x3 - x1, 2) + math.Pow(y3 - y1, 2))
    sisiBC := math.Sqrt(math.Pow(x3 - x2, 2) + math.Pow(y3 - y2, 2))

    sisiTerpanjang := sisiAB
    if sisiAC > sisiTerpanjang {
        sisiTerpanjang = sisiAC
    } else if sisiBC > sisiTerpanjang {
        sisiTerpanjang = sisiBC
    }
    fmt.Printf("Sisi Terpanjang Adalah: %.0f", sisiTerpanjang)
}

```

PS C:\Test> go run lat3.go
 Masukan nilai A: 1.0 1.0
 Masukan nilai B: 4.0 1.0
 Masukan nilai C: 1.0 5.0
 Sisi Terpanjang Adalah: 4
 PS C:\Test> go run lat3.go
 Masukan nilai A: 0.0 0.0
 Masukan nilai B: 3.0 0.0
 Masukan nilai C: 3.0 4.0
 Sisi Terpanjang Adalah: 5
 PS C:\Test>

Nama : Nabil Nailur Ridho
 NIM : 109082530008
 Kelas: S1IF-13-07

Deskripsi program

Program Go ini berfungsi untuk menemukan sisi terpanjang dari sebuah segitiga dengan cara meminta pengguna memasukkan koordinat (x, y) dari tiga titik sudutnya (A, B, dan C). Setelah mendapatkan input, program menggunakan rumus jarak matematis untuk menghitung panjang setiap sisi AB, AC, dan BC. Akhirnya, program membandingkan ketiga panjang sisi tersebut dan mencetak nilai dari sisi yang paling panjang ke layar.