

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 5-6  
FOR LOOP**



**Disusun oleh:**

**MOHAMAD ERLANGGA ZEIN**

**109082500020**

**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 a, b int

    fmt.Print("Masukkan nilai minimum: ")

    fmt.Scan(&a)

    fmt.Print("Masukkan nilai maksimum: ")

    fmt.Scan(&b)

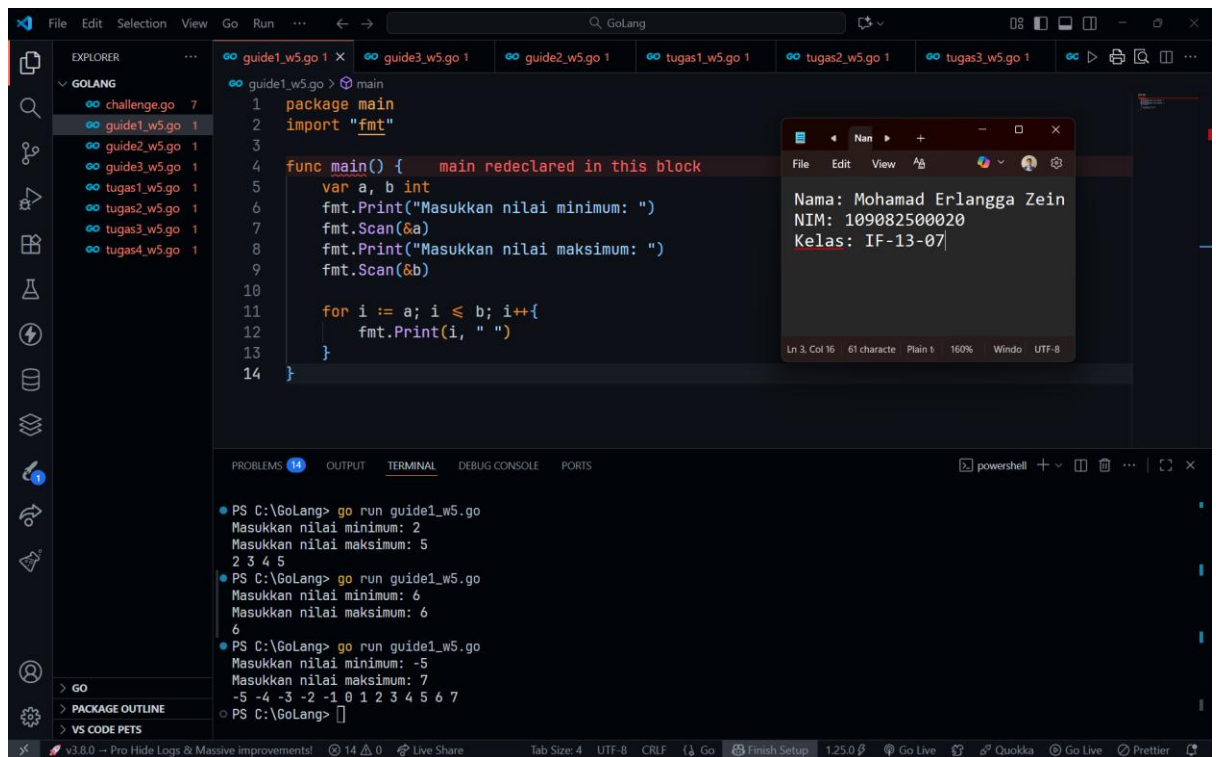
    for i := a; i <= b; i++){

        fmt.Print(i, " ")

    }

}
```

### Screenshoot program



### Deskripsi program

Pada program tersebut dijelaskan bahwa terjadi pengulangan terhadap nilai  $b$ , dengan nilai tertinggi pada variabel  $i$  tidak lebih kurang dari sama dengan nilai yang telah diinput pada variabel  $b$ , untuk langkah langkahnya yakni mula mula buat variabel  $a$  dan  $b$  untuk nantinya diisi sesuai dengan yang user inginkan, kemudian buat inisiasi  $for$  dengan  $for\ i := a$ , yang berarti nilai  $i$  sama dengan nilai yang diinputkan oleh user dengan variabel  $a$ , yang berarti juga nilai awalnya adalah nilai dari inisiasi pada  $for$ , kemudian  $i \leq b$ , yang artinya nilai dari  $i$  tidak boleh lebih kecil dari nilai  $b$  yang telah diinputkan oleh user, lalu  $i++$ , yang berarti perulangan/ $for$  loop ini selalu bertambah  $+1$  tiap barisnya atau tiap perintahnya, hingga disimpulkan bahwa nilai pada  $a$  tidak boleh lebih dari nilai  $b$ , dan ketika dimasukkan nilai dari  $a$  dan  $b$  maka hasilnya adalah perulangan dari mulai nilai yang telah dimasukkan pada variabel  $a$  hingga nilai maksimum pada nilai  $b$  yang telah diinputkan oleh user juga, seperti halnya pada gambar diatas.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
    var j, alas, tinggi, n int
    var luas float64

    fmt.Print("Masukkan perulangan sebanyak n: ")
    fmt.Scan(&n)

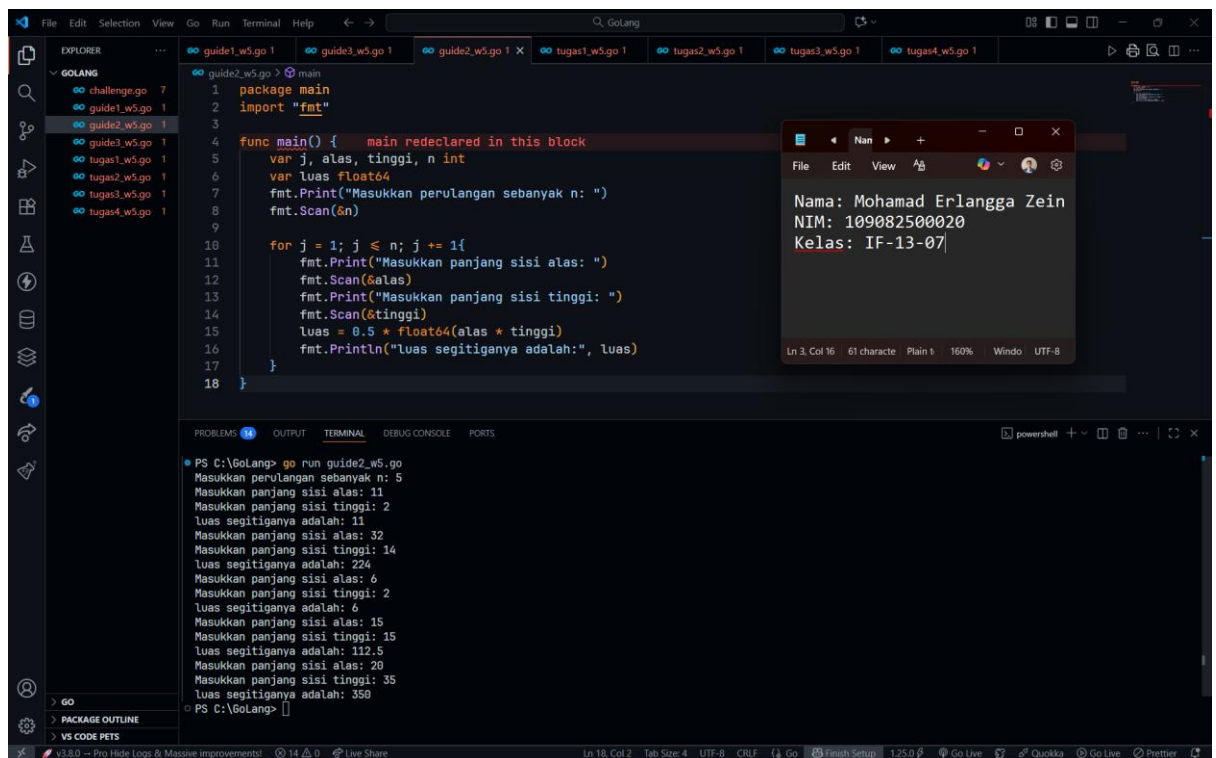
    for j = 1; j <= n; j += 1{
        fmt.Print("Masukkan panjang sisi alas: ")
        fmt.Scan(&alas)

        fmt.Print("Masukkan panjang sisi tinggi: ")
        fmt.Scan(&tinggi)

        luas = 0.5 * float64(alas * tinggi)

        fmt.Println("luas segitiganya adalah:", luas)
    }
}
```

## Screenshoot program



The screenshot shows a GoLang IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code in the editor is as follows:

```
1 package main
2 import "fmt"
3
4 func main() {
5     var j, alas, tinggi, n int
6     var luas float64
7     fmt.Println("Masukkan perulangan sebanyak n:")
8     fmt.Scan(&n)
9
10    for j = 1; j <= n; j += 1 {
11        fmt.Println("Masukkan panjang sisi alas:")
12        fmt.Scan(&alas)
13        fmt.Println("Masukkan panjang sisi tinggi:")
14        fmt.Scan(&tinggi)
15        luas = 0.5 * float64(alas * tinggi)
16        fmt.Println("Luas segitiganya adalah:", luas)
17    }
18 }
```

The terminal output shows the program running and calculating the area of a triangle 5 times based on the input n=5:

```
PS C:\GoLang> go run guide2_w5.go
Masukkan perulangan sebanyak n: 5
Masukkan panjang sisi alas: 11
Masukkan panjang sisi tinggi: 2
Luas segitiganya adalah: 11
Masukkan panjang sisi alas: 32
Masukkan panjang sisi tinggi: 14
Luas segitiganya adalah: 224
Masukkan panjang sisi alas: 6
Masukkan panjang sisi tinggi: 2
Luas segitiganya adalah: 6
Masukkan panjang sisi alas: 15
Masukkan panjang sisi tinggi: 15
Luas segitiganya adalah: 112.5
Masukkan panjang sisi alas: 20
Masukkan panjang sisi tinggi: 35
Luas segitiganya adalah: 350
PS C:\GoLang>
```

## Deskripsi program

Pada program ini output yang dikeluarkan adalah hasil dari perhitungan luas segitiga yang dimana luas segitiga itu memiliki rumus: " $\frac{1}{2} \times \text{Alas} \times \text{Tinggi}$ " serta, selain itu dalam program ini juga diminta agar user bisa memasukkan nilai alas dan tinggi nya berulang sebanyak n atau sebanyak yang user inginkan, seperti pada gambar diatas user memasukkan perulangan sebanyak 5, sehingga user bisa menghitung luas segitiga sebanyak 5 kali, bagaimana bisa seperti itu? Itu karena pada inisiasi for nya menggunakan `for j = 1; j <= n; j += 1`; yang artinya nomor/angka perulangan dimulai dari 1, kemudian `j <= n`; yang memiliki arti perulangan j sesuai dengan jumlah yang dimasukkan oleh user dan diawali dengan 1 sesuai dengan inisiasi `j = 1`, kemudian seterusnya `j += 1`; maksudnya adalah penugasan pada perulangan tersebut bertambah 1 kali sesuai dengan apa yang user masukkan nilainya pada n, setelah inisiasi for nya selesai, kemudian buat rumus dalam for, ini dimaksudkan agar user bisa melakukan perhitungan sebanyak n yang telah dimasukkan, dalam pembuatan rumusnya, sebelumnya harus buat variable terlebih dahulu sebelum inisiasi for loop nya, hingga kemudian dimasukkan dalam suatu for loop dengan rumus yang sesuai dengan rumus luas segitiga, karena disini variable luas menggunakan tipe data float64 sehingga nilai  $\frac{1}{2}$  dikonversikan ke dalam bentuk decimal menjadi 0.5, lalu variable alas dan tinggi dikonversi ke tipe data float64 karena sebelumnya variable alas dan tinggi itu tipe datanya int, sehingga diperlukan casting pada variable alas dan tinggi, setelah itu user bisa memasukkan nilai perulangan yang diinginkan, nilai alas dan tinggi dan nanti outputnya adalah hasil dari angka angka yang telah dimasukkan oleh user pada alas variable alas dan tinggi.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var a, b, hasil int

    fmt.Print("Masukkan bilangan pertama: ")

    fmt.Scan(&a)

    fmt.Print("Masukkan bilangan kedua: ")

    fmt.Scan(&b)

    hasil = 0

    for i := 1; i <= b; i+=1{

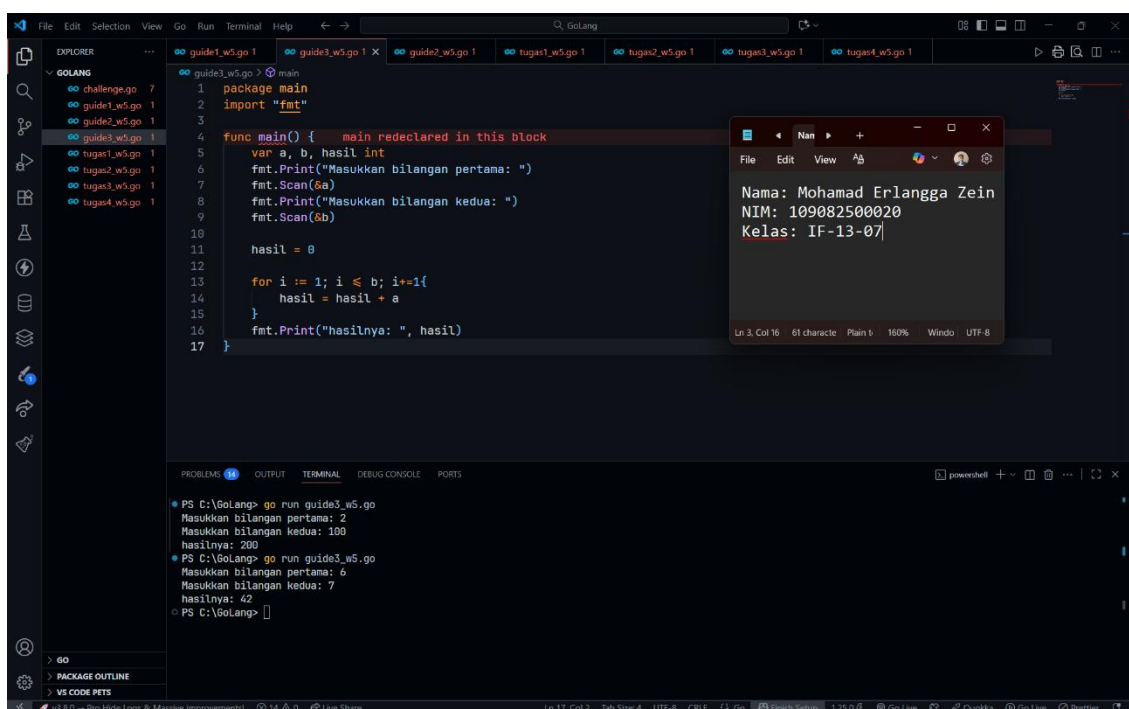
        hasil = hasil + a

    }

    fmt.Print("hasilnya: ", hasil)

}
```

#### Screenshoot program



## Deskripsi program

Program ini menghasilkan output berupa hasil perkalian dari bilangan pertama dengan bilangan kedua, namun dalam hal ini tidak diperbolehkan menggunakan operator perhitungan matematika "\*" namun harus menggunakan inisiasi for loop, pertama buat variable nilai a, b, dan hasil, kemudian masukkan nilai hasil = 0, kemudian lakukan inisiasi for dengan for i := 1; yang berarti nilai awalan dari suatu perulangan itu 1, lalu diikuti i <= b yang berarti nilai i tidak lebih kecil dari nilai b yang telah diinputkan oleh user, lalu i += 1; yang berarti ada penugasan dengan ditambah 1 tiap perulangan, lalu pada outputnya dipanggil lagi variable hasil dengan hasil = hasil + a, sehingga outputnya akan menghasilkan sebuah perkalian sesuai dengan nilai yang user masukkan.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {

    var i, n, hasil int

    fmt.Print("Masukkan nilai n: ")

    fmt.Scan(&n)

    for i = 1; i <= n; i++){

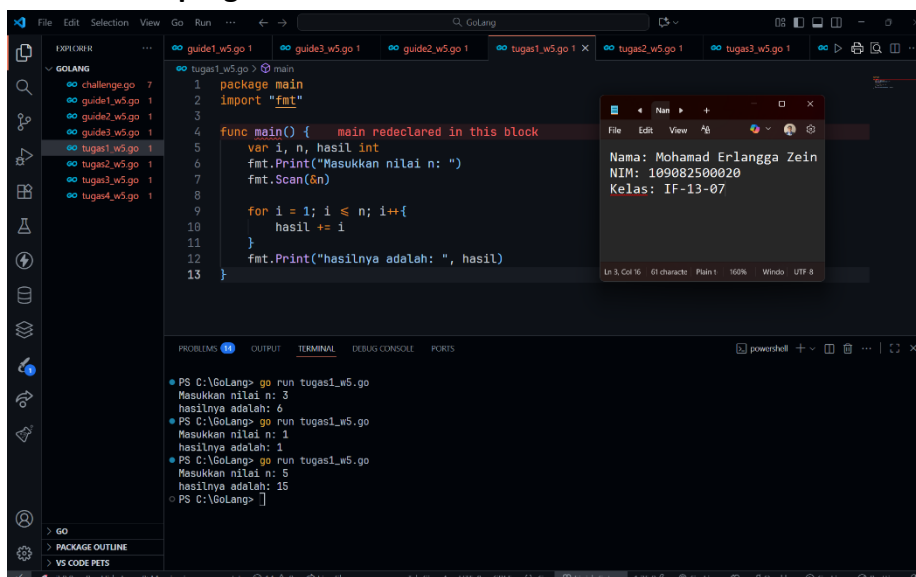
        hasil += i

    }

    fmt.Print("hasilnya adalah: ", hasil)

}
```

## Screenshoot program



### Deskripsi program

Program diatas merupakan program dengan output yang dimana inputnya itu dari penugasan tambahan dari += perulangan pada variable i, yang mula mula inisiasi variable nilainya yang diisi pada variable n, yang kemudian akan di deklarasi dengan for loop dengan  $i = 1; i \leq n; i++$ , sehingga hasilnya adalah perkalian atau penambahan sebanyak 3, sehingga outputnya akan mengalami perhitungan yang tertera pada deklarasi for nya.

## 2. Tugas 2

### Source code

```
package main

import (
    "fmt"
    "math"
)

func main(){
    var n int
    var jariJari, tinggi float64
    fmt.Print("Masukkan perulangan sebanyak n: ")
    fmt.Scan(&n)

    for i := 1; i <= n; i += 1{
        fmt.Print("Masukkan nilai jari jari: ")
        fmt.Scan(&jariJari)
        fmt.Print("Masukkan nilai tinggi: ")
        fmt.Scan(&tinggi)

        volumeKerucut := (1.0/3.0) * math.Pi *
jariJari * jariJari * tinggi
        fmt.Println("Volume kerucut:",
volumeKerucut)
    }
}
```

## Screenshoot program

The screenshot shows a GoLang IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code in the editor is as follows:

```
1 package main
2 import (
3     "fmt"
4     "math"
5 )
6
7 func main(){
8     var n int
9     var jariJari, tinggi float64
10    fmt.Print("Masukkan perulangan sebanyak n: ")
11    fmt.Scan(&n)
12
13    for i := 1; i <= n; i += 1{
14        fmt.Print("Masukkan nilai jari jari: ")
15        fmt.Scan(&jariJari)
16        fmt.Print("Masukkan nilai tinggi: ")
17        fmt.Scan(&tinggi)
18        volumeKerucut := (1.0/3.0) * math.Pi * jariJari * jariJari * tinggi
19        fmt.Println("Volume kerucut:", volumeKerucut)
20    }
21 }
```

The terminal output shows the program being run multiple times with different inputs:

```
PS C:\GoLang> go run tugas2_w5.go
Masukkan perulangan sebanyak n: 1
Masukkan nilai jari jari: 3
Masukkan nilai tinggi: 4
Volume kerucut: 37.699111843877524
PS C:\GoLang> go run tugas2_w5.go
Masukkan perulangan sebanyak n: 3
Masukkan nilai jari jari: 1
Masukkan nilai tinggi: 1
Volume kerucut: 1.0471975511965979
Masukkan nilai jari jari: 2
Masukkan nilai tinggi: 2
Volume kerucut: 8.377580409572783
Masukkan nilai jari jari: 3
Masukkan nilai tinggi: 3
Volume kerucut: 28.27433388230814
PS C:\GoLang>
```

## Deskripsi program

Pada program ini, outputnya adalah hasil dari perhitungan volume kerucut, dan selain itu program ini juga menggunakan perulangan sesuai yang user inginkan, jadi yang pertama adalah buat variable untuk perulangannya, disini menggunakan variable `n` bertipe data `int`, kemudian pada alas(jari jari) dan tinggi menggunakan tipe data `float64`, dan pada library ditambahkan library `"math"` karena disini menggunakan perhitungan matematika pada volume kerucut, kemudian setelah itu, masukkan variable alas(jari jari) dan tinggi pada inisiasi `for`, lalu kemudian buat operasi perhitungan rumus kerucutnya di dalam `for`, disini menggunakan library `"math"` yakni `math.Pi` untuk memasukkan bilangan pi untuk kebutuhan operasi perhitungan pada rumus volume kerucut, dan outputnya adalah nantinya user akan bisa melakukan perhitungan rumus volume kerucut dengan seenak usernya mau berapa kali, karena dideklarasikan oleh variable `n` untuk jumlah perulangannya.



### 3. Tugas 3

#### Source code

```
package main
import (
    "fmt"
    "math"
)

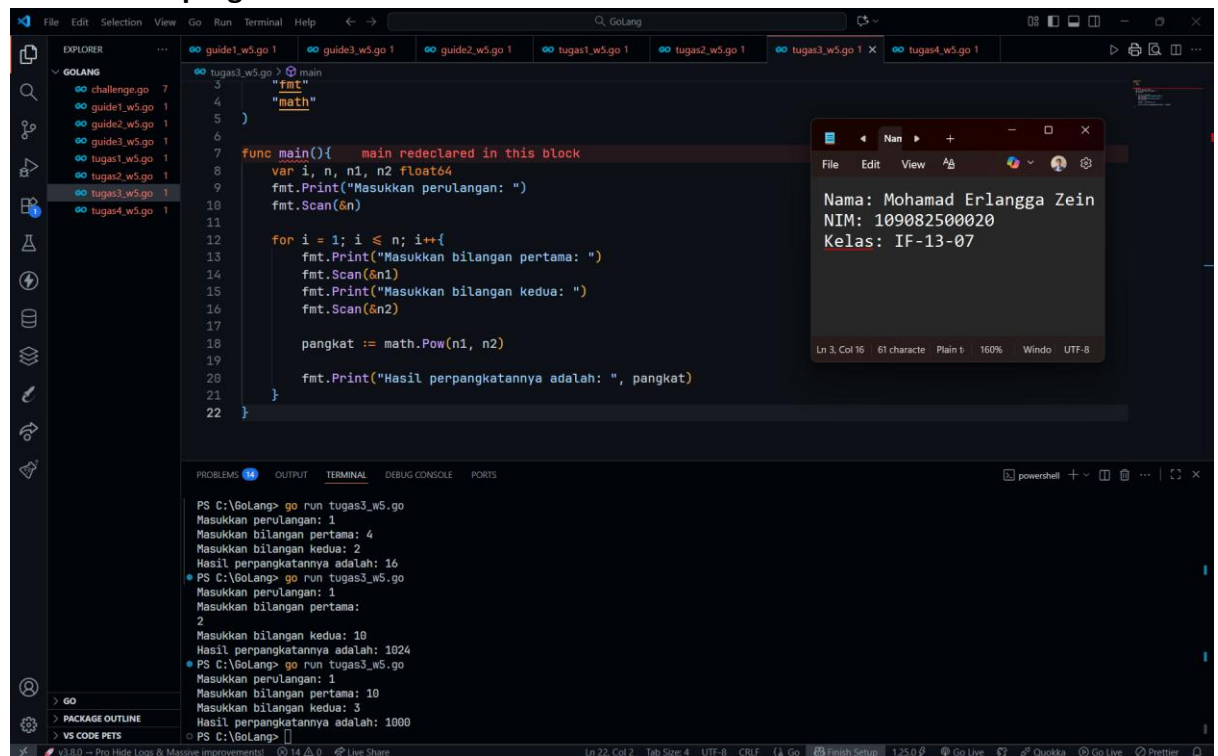
func main(){
    var i, n, n1, n2 float64
    fmt.Print("Masukkan perulangan: ")
    fmt.Scan(&n)

    for i = 1; i <= n; i++){
        fmt.Print("Masukkan bilangan pertama: ")
        fmt.Scan(&n1)
        fmt.Print("Masukkan bilangan kedua: ")
        fmt.Scan(&n2)

        pangkat := math.Pow(n1, n2)

        fmt.Print("Hasil perpangkatannya adalah: ",
pangkat)
    }
}
```

#### Screenshoot program



## Deskripsi program

Program ini menghasilkan output bilangan berpangkat dari bilangan pertama sebagai basis dan bilangan ke-dua sebagai eksponen/sebagai pangkatnya, yang pertama adalah menambahkan variable  $n$  dengan tipe data float64, variable  $n$  ini digunakan oleh user untuk menentukan seberapa kali perulangan dilakukan pada program, kemudian variable  $i$ ,  $n1$ , dan  $n2$  dengan tipe data float64 serta dalam program ini menggunakan library "math" untuk mempermudah dalam pemrogramannya, masukkan perulangan dengan  $\text{for } i = 1; i \leq n; i++$  yang maksudnya adalah perulangan dimulai dengan angka 1, kemudian jumlah perulangan tidak lebih kecil dari variable  $i$ , lalu perulangan bertambah 1, kemudian pada for diketik juga program untuk membuat bilangan berpangkat dengan menggunakan syntax pada library "math" yaitu  $\text{math.Pow}$ , dan hasilnya akan sesuai yang diinginkan user.

## 4. Tugas 4

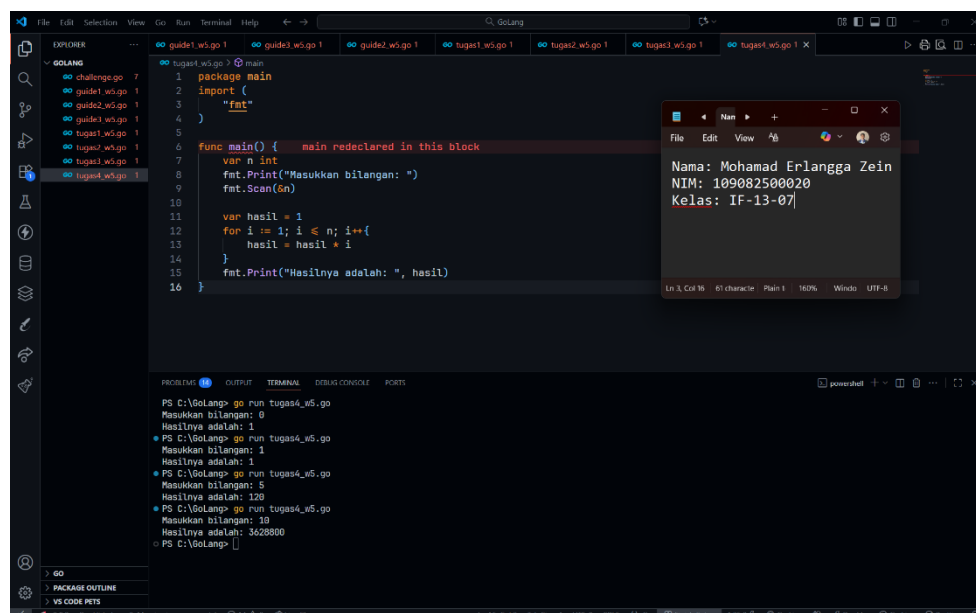
### Source code

```
package main
import (
    "fmt"
)

func main() {
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)

    var hasil = 1
    for i := 1; i <= n; i++){
        hasil = hasil * i
    }
    fmt.Print("Hasilnya adalah: ", hasil)
}
```

### Screenshoot program



**Deskripsi program**

Program ini menghasilkan output hasil dari factorial suatu bilangan, yang pertama adalah menambahkan variable n dan hasil dengan tipe data int dan pada variable hasil, nilainya adalah 1, kemudian inisiasi for dengan `for i := 1; i <= n; i++`, setelah itu dibawahnya masukkan hasil dengan hasil dikali dengan i, sehingga hasilnya seperti factorial, seperti pada gambar.