

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 5-6  
FOR LOOP**



**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 a, b int

    fmt.Print("masukan a: ")

    fmt.Scan(&a)

    fmt.Print("Masukan b: ")

    fmt.Scan(&b)

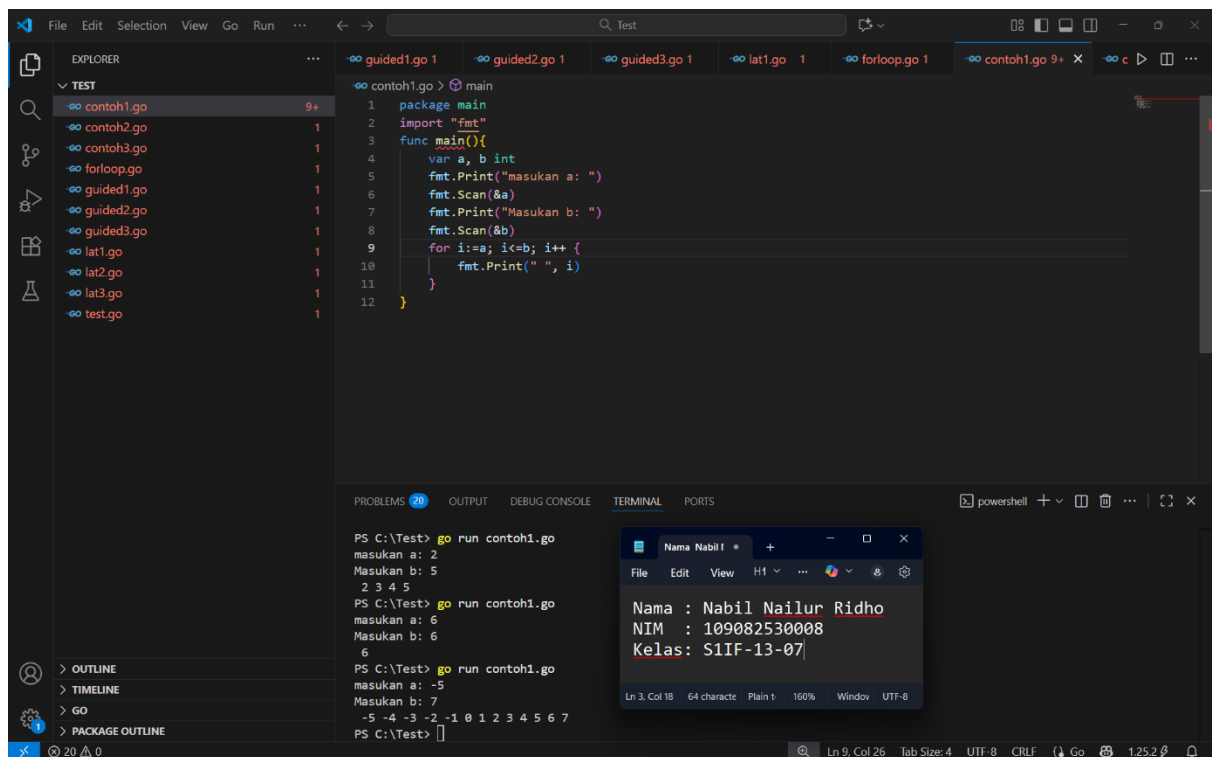
    for i:=a; i<=b; i++ {

        fmt.Print(" ", i)

    }

}
```

### Screenshoot program



### Deskripsi program

Program Go ini buat mencetak urutan angka. kita diminta memasukkan dua angka, yaitu 'a' dan 'b'. Setelah itu, programnya akan langsung menampilkan semua bilangan bulat secara berurutan, mulai dari angka 'a' sampai angka 'b'. Persis seperti di terminal, contohnya kita input -5 dan 7, dia langsung mencetak deretan angka -5, -4, -3, dan seterusnya sampai angka 7.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var j, alas, tinggi, n int

    var luas float64

    fmt.Scan(&n)

    for j = 1; j <=n; j+=1 {

        fmt.Scan(&alas, &tinggi)

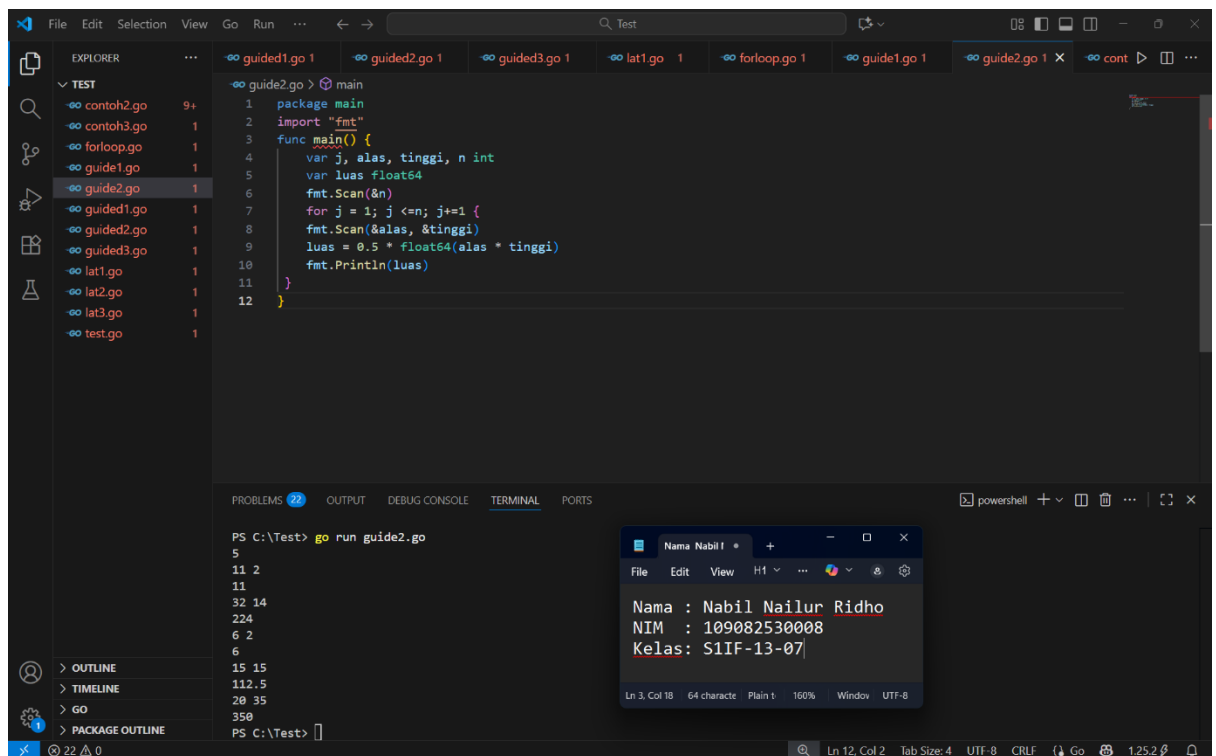
        luas = 0.5 * float64(alas * tinggi)

        fmt.Println(luas)

    }

}
```

## Screenshot program



## Deskripsi program

Program Go ini untuk menghitung luas segitiga. Pertama, program meminta kamu memasukkan satu angka (`n`) untuk menentukan berapa banyak segitiga yang mau dihitung (contohnya kamu menginput angka 5). Setelah itu, program akan berulang (looping) sebanyak `n` kali. Di setiap putaran, ia akan meminta kita memasukkan nilai "alas" dan "tinggi", lalu langsung menghitung luasnya dengan rumus  $0.5 \times \text{alas} \times \text{tinggi}$  dan mencetak hasilnya ke terminal, sebelum akhirnya meminta masukan untuk segitiga berikutnya.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var j, v1, v2 int

    var hasil int

    fmt.Print("Masukan angka pertama: ")

    fmt.Scan(&v1)
```

```

    fmt.Print("Masukan angka kedua: ")

    fmt.Scan(&v2)

    hasil = 0

    for j = 1; j <= v2; j+=1 {

        hasil = hasil + v1

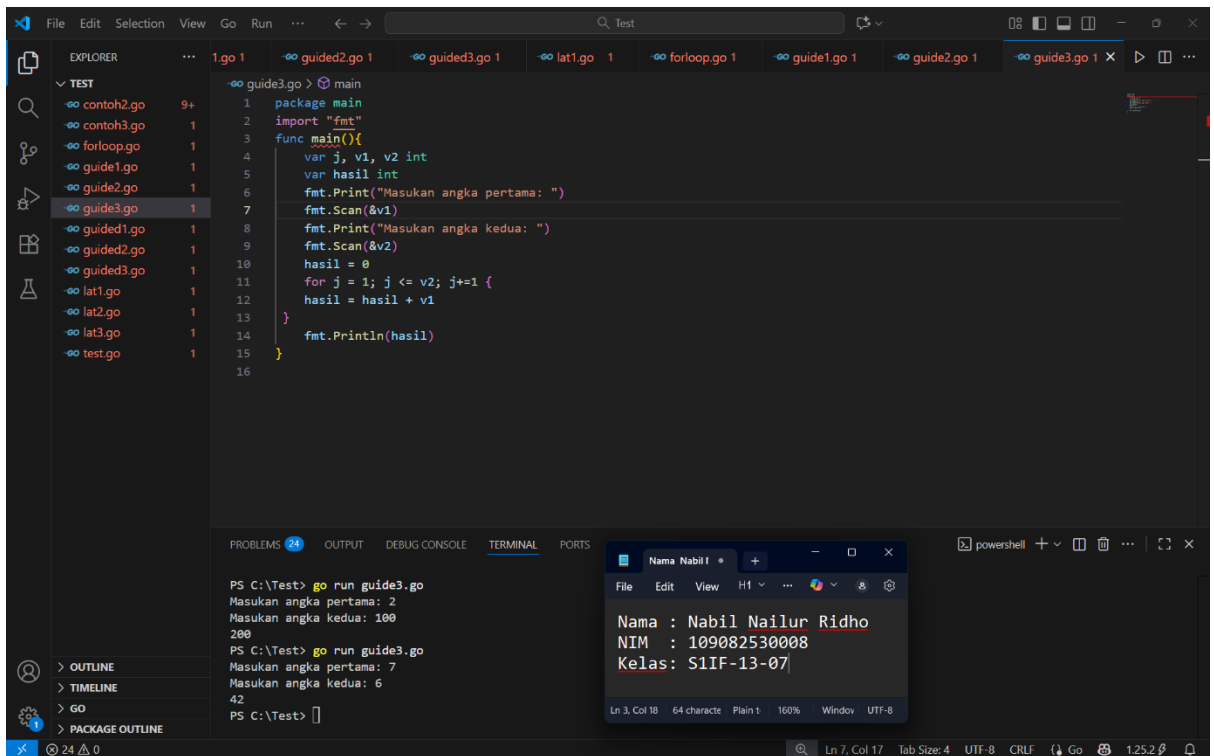
    }

    fmt.Println(hasil)

}

```

## Screenshoot program



## Deskripsi program

program Go ini adalah kalkulator perkalian ( $\text{angka1} \times \text{angka2}$ ), program ini melakukannya dengan cara yang sedikit manual. Program menggunakan operator perkalian (\*), program ini meminta kamu memasukkan "angka pertama" (v1) dan "angka kedua" (v2). Kemudian, program ini akan menambahkan angka pertama (v1) ke sebuah variabel hasil (yang awalnya 0) secara berulang-ulang, sebanyak v2 kali. Jadi, contohnya kita memasukkan 7 dan 6, program ini sebenarnya menghitung  $7 + 7 + 7 + 7 + 7 + 7$ , dan akhirnya mengeluarkan hasil 42.

## 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++){

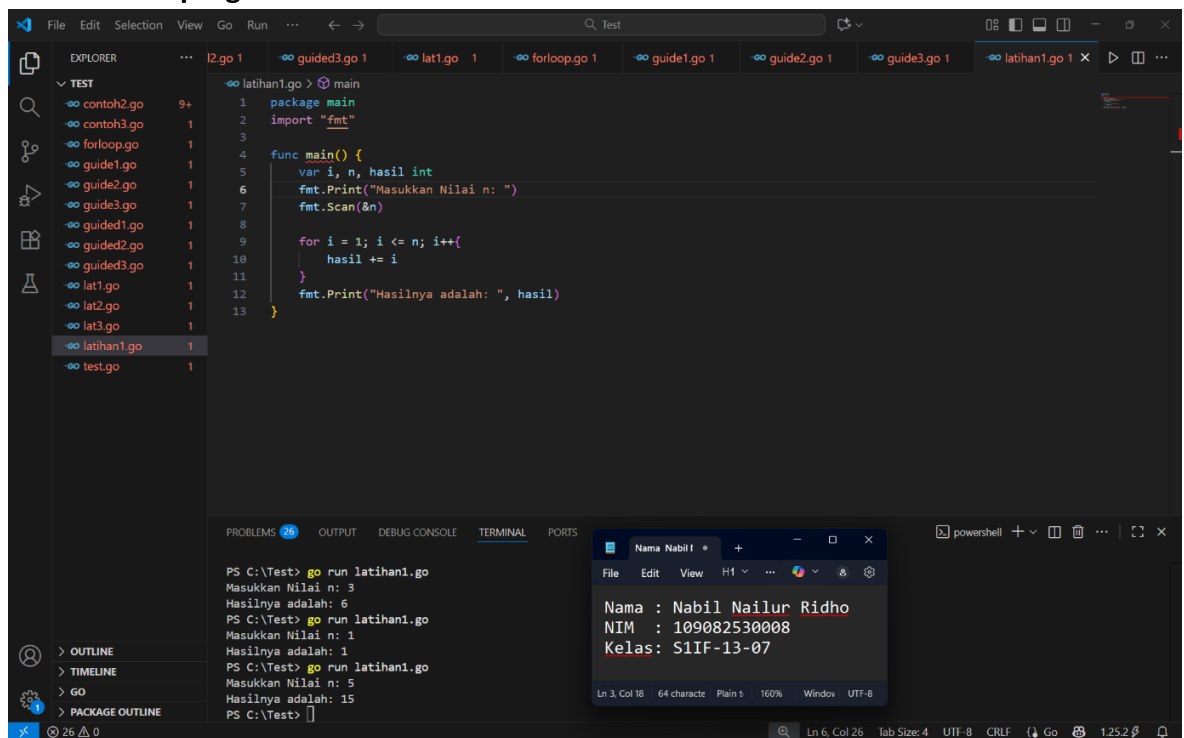
        hasil += i

    }

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

}
```

#### Screenshoot program



### Deskripsi program

Program Go ini dirancang untuk menghitung total jumlahan dari angka 1 hingga nilai  $n$  yang dimasukkan oleh pengguna. Program ini pertama-tama meminta pengguna untuk memasukkan sebuah angka ( $n$ ), kemudian menggunakan perulangan `for` untuk mengubah dari  $i = 1$  sampai  $i < n$ . Dalam setiap iterasi, nilai  $i$  saat ini ditambahkan ke variabel `hasil`. Setelah perulangan selesai, program akan mencetak nilai total yang tersimpan dalam variabel `hasil` ke terminal, seperti yang ditunjukkan oleh contoh output di mana input  $n=5$  menghasilkan 15.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var i, n, r, t int

    fmt.Print("Masukkan Banyaknya Kerucut : ")

    fmt.Scan(&n)

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

        fmt.Print("Masukkan jari-jari : ")

        fmt.Scan(&r)

        fmt.Print("Masukkan tinggi : ")

        fmt.Scan(&t)

        volume := (1.0 / 3.0) * 3.14 * float64(r * r * t)

        fmt.Printf("Volumenya adalah: %f", volume)
```

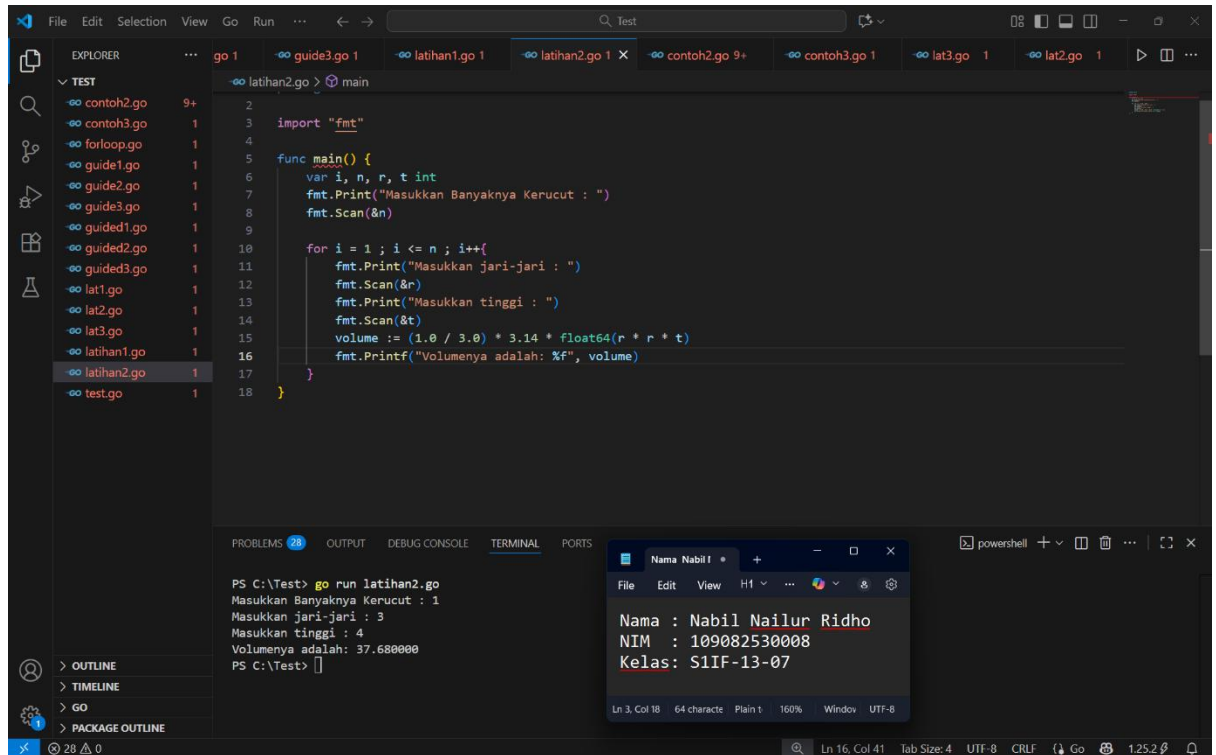
```

    }

}

```

## Screenshoot program



## Deskripsi program

Program Go ini untuk volume kerucut yang dapat menghitung volume untuk beberapa kerucut sekaligus. Pertama, program akan meminta pengguna memasukkan "Banyaknya Kerucut" (n) yang ingin dihitung. Setelah itu, program akan masuk ke dalam perulangan for yang akan berjalan sebanyak n kali. Di dalam setiap putaran, program akan meminta pengguna untuk memasukkan jari-jari (r) dan tinggi (t) kerucut. Program kemudian menghitung volume menggunakan rumus  $V = \frac{1}{3} \pi * (r * r * t)$  dan langsung mencetak hasil volume tersebut ke terminal.

## 3. Tugas 3

### Source code

```

package main

import "fmt"

func main() {
    var i, angka, pangkat, hasil int
}

```



```

    fmt.Print("Masukkan Angka : ")

    fmt.Scan(&angka)

    fmt.Print("Masukkan Pangkat : ")

    fmt.Scan(&pangkat)


    hasil = 1


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

        hasil *= angka

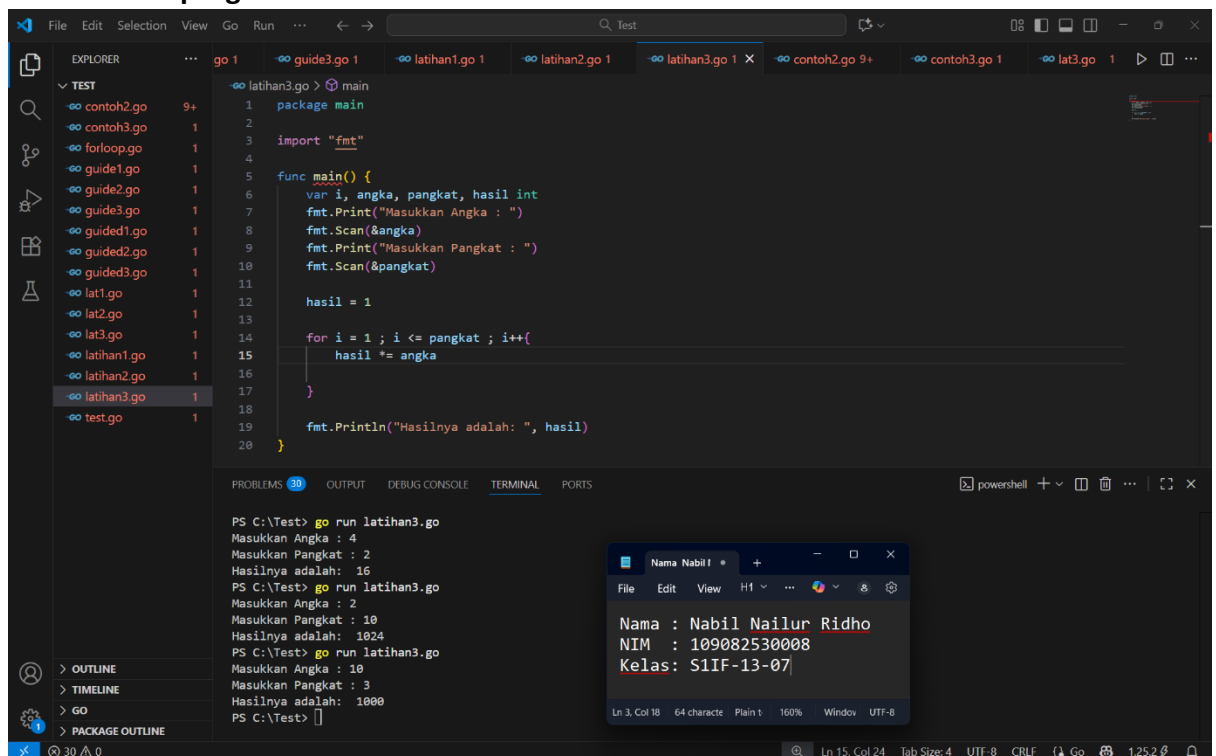
    }


    fmt.Println("Hasilnya adalah: ", hasil)

}

```

## Screenshoot program



### Deskripsi program

Program Go ini untuk menghitung hasil perpangkatan berdasarkan input dari pengguna. Program terlebih dahulu meminta pengguna memasukkan dua nilai integer sebuah "angka" dan "pangkat". Setelah itu, program menginisialisasi variabel hasil dengan nilai 1. Menggunakan perulangan for yang berjalan sebanyak nilai pangkat, program ini secara berulang mengalikan variabel hasil dengan angka yang dimasukkan. Setelah perulangan selesai, program akan menampilkan nilai akhir hasil ke terminal, yang secara efektif merupakan hasil dari angka dan pangkat, seperti yang ditunjukkan oleh contoh  $10^3$  menghasilkan 1000.

## 4. Tugas 4

### Source code

```
package main

import (
    "fmt"
)

func main() {
    var n int
    fmt.Print("Masukkan suatu bilangan: ")
    fmt.Scan(&n)
    var hasil = 1
    for i := 1; i <= n; i++){
        hasil = hasil * i
    }
    fmt.Print("Hasilnya adalah: ", hasil)
}
```

## Screenshoot program

The screenshot shows a Visual Studio Code editor with a Go file named `latihan4.go` open. The code implements a factorial function using a loop. The terminal window at the bottom shows the execution of the program with various inputs and their corresponding factorial results.

```
1 package main
2 import (
3     "fmt"
4 )
5
6 func main() {
7     var n int
8     fmt.Print("Masukkan suatu bilangan: ")
9     fmt.Scan(&n)
10    var hasil = 1
11    for i := 1; i <= n; i++{
12        hasil = hasil * i
13    }
14    fmt.Print("Hasilnya adalah: ", hasil)
15 }
```

Terminal Output:

```
PS C:\Test> go run latihan4.go
Masukkan suatu bilangan: 0
Hasilnya adalah: 1
PS C:\Test> go run latihan4.go
Masukkan suatu bilangan: 1
Hasilnya adalah: 1
PS C:\Test> go run latihan4.go
Masukkan suatu bilangan: 5
Hasilnya adalah: 120
PS C:\Test> go run latihan4.go
Masukkan suatu bilangan: 10
Hasilnya adalah: 3628800
PS C:\Test>
```

## Deskripsi program

Program Go ini merupakan faktorial ( $n!$ ). Jadi program ini meminta kamu memasukkan satu angka ( $n$ ), lalu ia akan menghitung total perkalian semua angka dari 1 sampai ke angka  $n$  tersebut. Caranya, ia menyiapkan variabel hasil bernilai 1, kemudian mengalikannya secara berulang ( $1 \times 1$ , lalu hasilnya  $\times 2$ , lalu hasilnya  $\times 3$ , dan seterusnya) sampai  $n$  kali. Seperti yang terlihat di terminal, jika kamu memasukkan 5, program akan menghitung  $1 \times 2 \times 3 \times 4 \times 5$  dan menampilkan output di terminal yaitu "Hasilnya adalah: 120".