

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**MODUL 5 & 6**  
**I/O, TIPE DATA & VARIABEL**



**Disusun oleh:**

**ZHAFIF YUSUF AL AMIN**

**109082500137**

**S1IF-13-04**

**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 n, m int

    fmt.Print("Input: ")

    fmt.Scan(&m, &n)

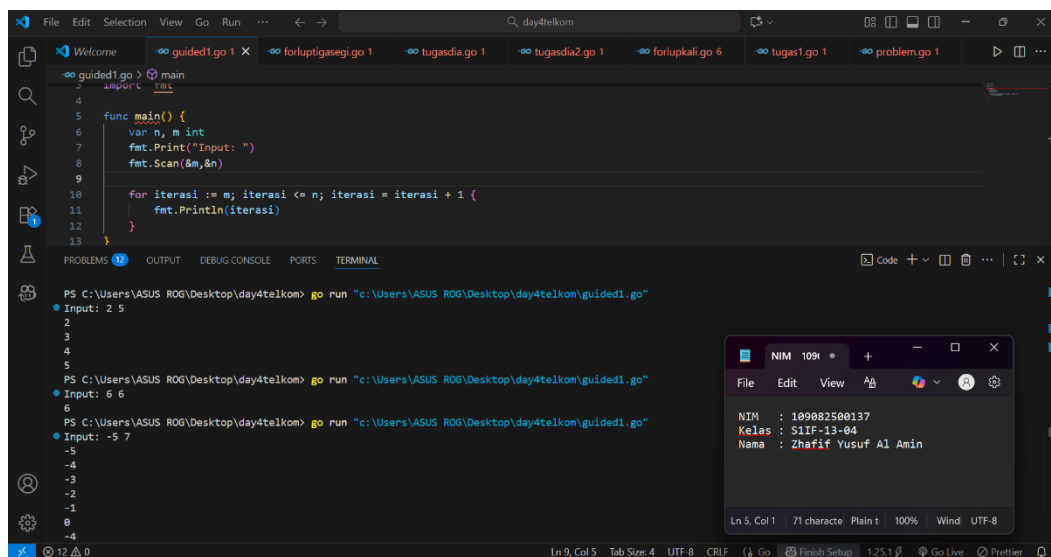
    for iterasi := m; iterasi <= n; iterasi = iterasi + 1 {

        fmt.Println(iterasi)

    }

}
```

#### Screenshoot program



#### Deskripsi program

package main: menandakan program utama ini ibaratnya kita akan membangun rumah & package main ini sebagai temboknya.

import "fmt": berfungsi memanggil package "fmt" ini ibarat bangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var n, m int`: deklarasi variabel n dan m menggunakan tipe data integer.

```
for iterasi := m; iterasi <= n; iterasi = iterasi + 1 {  
    fmt.Println(iterasi)
```

`}` : fungsi untuk mengulang program sampai suatu kondisi terpenuhi dalam kasus ini program akan menjalankan program berulang sebanyak 4 kali dan setiap perulangan akan ditambah 1. Jika inputan `m = 2` dan `n=5` maka outputnya adalah 2,3,4,5.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var n int

    var a, t float64


    fmt.Print("Input batas: ")

    fmt.Scan(&n)


    for iterasi := 2; iterasi <= n+1; iterasi = iterasi + 1
    {

        fmt.Print("Input alas dan tinggi: ")

        fmt.Scan(&a, &t)

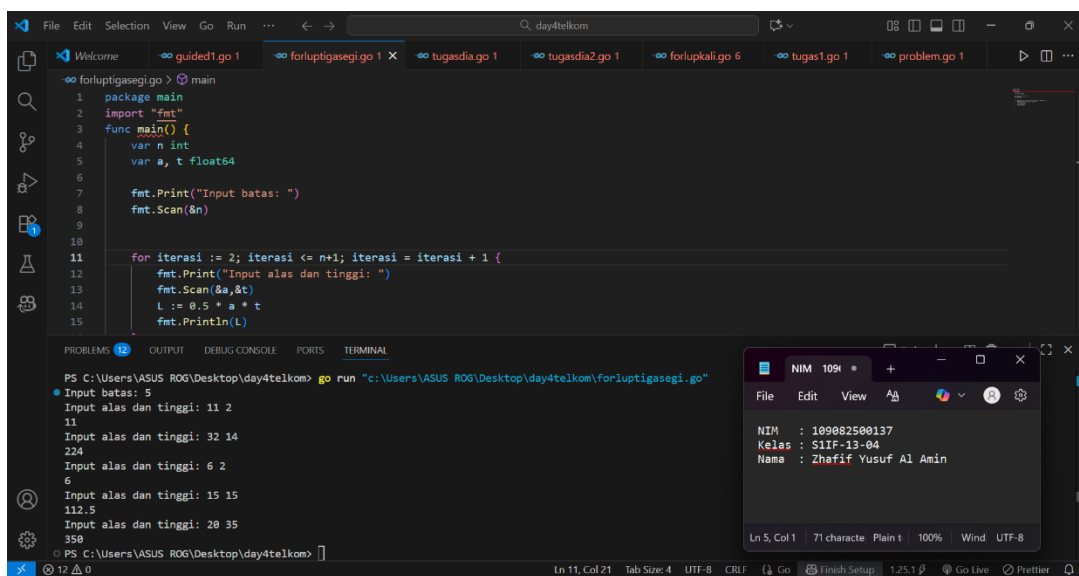
        L := 0.5 * a * t

        fmt.Println(L)

    }

}
```

### Screenshoot program



The screenshot shows a Go program running in VS Code. The code is as follows:

```
package main
import "fmt"
func main() {
    var n int
    var a, t float64
    fmt.Print("Input batas: ")
    fmt.Scan(&n)
    for iterasi := 2; iterasi <= n+1; iterasi = iterasi + 1 {
        fmt.Print("Input alas dan tinggi: ")
        fmt.Scan(&a, &t)
        L := 0.5 * a * t
        fmt.Println(L)
    }
}
```

The terminal output shows the program's execution for various inputs:

```
PS C:\Users\ASUS ROG\Desktop\day4telkom> go run "c:\Users\ASUS ROG\Desktop\day4telkom\forluptigasegi.go"
Input batas: 5
Input alas dan tinggi: 11 2
11
Input alas dan tinggi: 32 14
224
Input alas dan tinggi: 6 2
6
Input alas dan tinggi: 15 15
112.5
Input alas dan tinggi: 20 35
350
```

The status bar at the bottom indicates the file is at line 11, column 21, with a tab size of 4, UTF-8 encoding, and CR/LF line endings.

### Deskripsi program

`package main` : menandakan program utama ini ibaratnya kita akan membangun rumah & `package main` ini sebagai temboknya.

`import "fmt"` : berfungsi memanggil package "fmt" ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var n int` : deklarasi variabel x dengan tipe data int yang nantinya berguna untuk sebagai nilai n. n itu adalah berapa banyak kita ingin melakukan perulangan.

`var a, t float64`: deklarasi variabel a dan t yang nantinya berguna untuk input alas dan tinggi. Kenapa memakai float? Karena nanti di rumusnya kita akan menggunakan 0.5 sebagai pengganti 1/2.

```
fmt.Print("Input batas: ")
```

```
fmt.Scan(&n)
```

: berfungsi untuk menginput dan menyimpan isi dari variabel n. Yang dimana n itu sebagai batas perulangan.

```
for iterasi := 2; iterasi <= n+1; iterasi = iterasi + 1 {
```

```
fmt.Print("Input alas dan tinggi: ")
```

```
fmt.Scan(&a, &t)
```

```
L := 0.5 * a * t
```

```
fmt.Println(L)
```

} : fungsi untuk melakukan for atau perulangan dengan nilai iterasi adalah 2 dan nilai n adalah 5 ditambah 1 sehingga perulangan akan berulang sebanyak 5 kali. Fungsi print, scan, rumus luas segitiga dan print hasil rumus nya pun otomatis akan berulang sebanyak kondisi yang ditentukan. Dalam kasus ini yang di tentukan adalah 5 kali sehingga program akan berhenti pada perulangan ke 5.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var n int

    var a int

    var hasiliterasi int


    fmt.Print("Input pertama: ")

    fmt.Scan(&n)

    fmt.Print("Input kedua: ")

    fmt.Scan(&a)


    for iterasi := 1; iterasi <= n; iterasi++ {

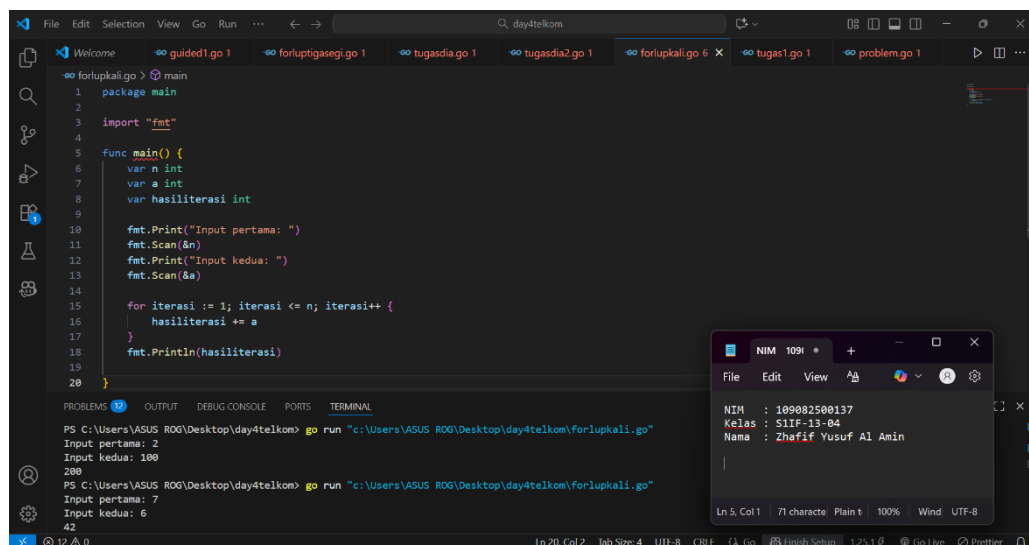
        hasiliterasi += a

    }

    fmt.Println(hasiliterasi)

}
```

#### Screenshoot program



### Deskripsi program

`package main` : menandakan program utama ini ibaratnya kita akan membangun rumah & `package main` ini sebagai temboknya.

`import "fmt"` : berfungsi memanggil package "fmt" ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

```
var n int
var a int
var hasiliterasi int
```

: deklarasi variabel `n`, `a` dan `hasiliterasi` dengan tipe data integer.

```
fmt.Print("Input pertama: ")
fmt.Scan(&n)
fmt.Print("Input kedua: ")
fmt.Scan(&a) : fungsi untuk menginput dan menyimpan isi dari variabel a dan n.
```

```
for iterasi := 1; iterasi <= n; iterasi++ {
    hasiliterasi += a
}
fmt.Println(hasiliterasi)
}
```

: berfungsi untuk melakukan perulangan dengan nilai iterasi ada 1 dan ingin di ulang hingga terpenuhi yaitu 2 kali perulangan. Setiap perulangan nilainya ditambah 1. Lalu hasil dari perulangan disimpan didalam variabel `hasiliterasi`. `Hasiliterasi` didapatkan dari perulangan `a` sebanyak `n`. lalu output berupa `hasiliterasi`.

# TUGAS

## 1. Tugas 1

### Source code

```
package main

import "fmt"

func main() {

    var n,hasil int

    fmt.Scan(&n)


    for iterasi := 1; iterasi <= n; iterasi = iterasi + 1 {

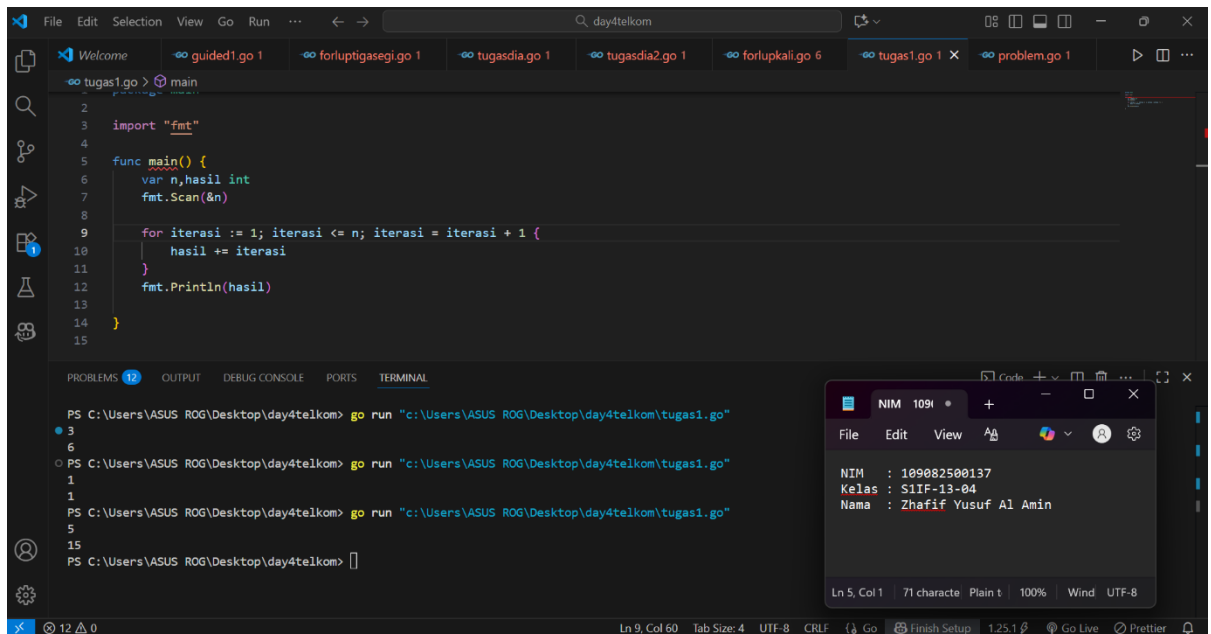
        hasil += iterasi

    }

    fmt.Println(hasil)

}
```

### Screenshoot program



### Deskripsi program

package main : menandakan program utama ini ibaratnya kita akan membangun rumah & package main ini sebagai temboknya.



`import "fmt"` : berfungsi memanggil package "fmt" ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var n, hasil int`: Pendeklarasian variabel n dan hasil menggunakan tipe data integer (bilangan bulat).

`fmt.Scan(&n)` : fungsi untuk menginput dan setelah di input isi dari variabel akan disimpan di &n. berfungsi untuk di iterasi/for/perulangan.

```
for iterasi := 1; iterasi <= n; iterasi = iterasi + 1 {  
    hasil += iterasi  
}  
fmt.Println(hasil)
```

: fungsi untuk melakukan operasi perulangan akan berhenti jika kondisi yang diinginkan sesuai. Berawalan dari 1 hingga nilai n sesuai yang kita inputkan Selama perulangan berlangsung, setiap nilai iterasi akan ditambahkan ke variabel hasil menggunakan operasi `hasil += iterasi`. dan akan di outputkan diluar perulangan.

## 2. Tugas 2

### Source code

```
package main

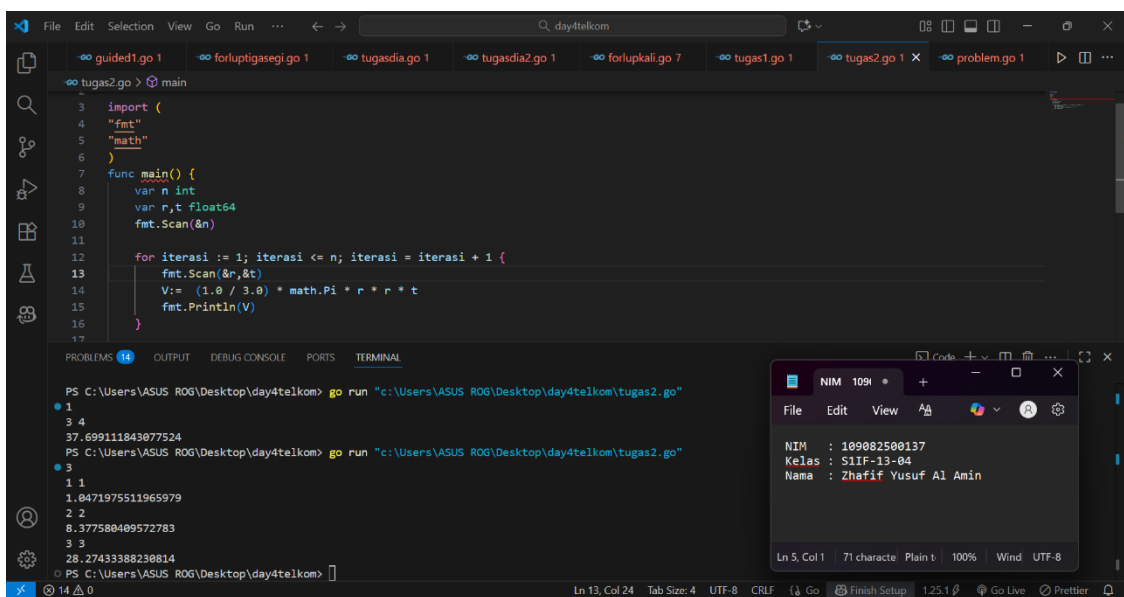
import (
    "fmt"
    "math"
)

func main() {
    var n int
    var r,t float64

    fmt.Scan(&n)

    for iterasi := 1; iterasi <= n; iterasi = iterasi + 1 {
        fmt.Scan(&r,&t)
        V:= (1.0 / 3.0) * math.Pi * r * r * t
        fmt.Println(V)
    }
}
```

### Screenshoot program



### Deskripsi program

`package main` : menandakan program utama ini ibaratnya kita akan membangun rumah & `package main` ini sebagai temboknya.

```
import (  
    "fmt"  
    "math"  
)
```

: berfungsi memanggil package “fmt” ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air. Dan juga memanggil package “math” yang nantinya berguna untuk operasi matematika. Disini kita menggunakan “math” untuk menentukan phi.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

```
var n int  
var r,t float64
```

: Deklarasi variabel `n` menggunakan tipe data integer. Deklarasi variabel jari jari dan tinggi menggunakan tipe data float64 atau bilangan bulat.

`fmt.Scan(&n)`: digunakan untuk membaca input `n` yang nantinya berguna untuk menentukan berapa kali perulangan di laksanakan.

```
for iterasi := 1; iterasi <= n; iterasi = iterasi + 1 {  
    fmt.Scan(&r,&t)  
    V:= (1.0 / 3.0) * math.Pi * r * r * t  
    fmt.Println(V)
```

} : fungsi untuk melakukan perulangan hingga suatu kondisi tercapai. Disini kita menggunakannya untuk menghitung volume kerucut sebanyak `n` kali dengan setiap perulangan ditambah 1. Lalu kita input jari jari dan tinggi dari kerucut kemudian dimasukkan kerumus didalam `V` hingga di hitung lalu di tampilkan menggunakan `fmt.Println(V)`. hasil akan di tampilkan sesuai nilai `n`.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

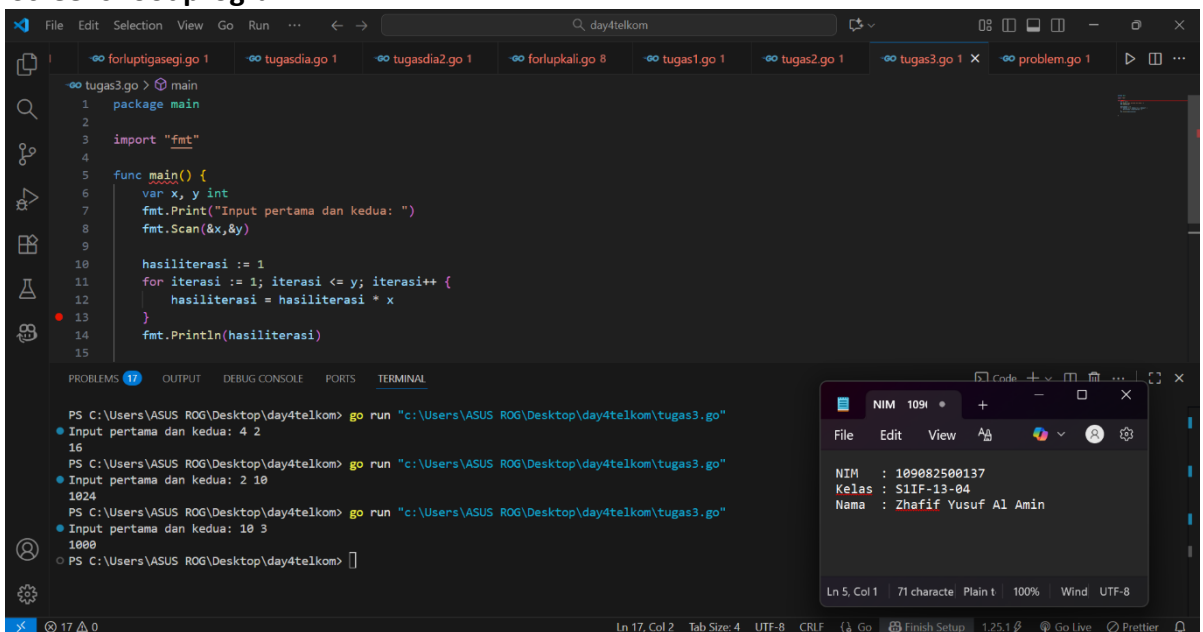
func main() {
    var x, y int

    fmt.Print("Input pertama dan kedua: ")
    fmt.Scan(&x, &y)

    hasiliterasi := 1
    for iterasi := 1; iterasi <= y; iterasi++ {
        hasiliterasi = hasiliterasi * x
    }

    fmt.Println(hasiliterasi)
}
```

#### Screenshoot program



### Deskripsi program

`package main` : menandakan program utama ini ibaratnya kita akan membangun rumah & `package main` ini sebagai temboknya.

`import "fmt"` : berfungsi memanggil package "fmt" ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var x, y int`: deklarasi variabel x dan y menggunakan tipe data integer.

`fmt.Print("Input pertama dan kedua: ")`

`fmt.Scan(&x, &y)` : berfungsi untuk menampilkan output berupa Input pertama dan kedua: kemudian membaca inputan yang telah kita input lalu disimpan ke variabel `&x, &y`.

`hasiliterasi := 1`

```
for iterasi := 1; iterasi <= y; iterasi++ {  
    hasiliterasi = hasiliterasi * x  
}
```

`fmt.Println(hasiliterasi)`

: kita deklarasi dulu isi dari variabel `hasiliterasi` itu nilainya berapa, disini kita isi 1 karena mau memangkatkan dari perulangan. selama iterasi masih kurang dari atau sama dengan `y`, maka `hasiliterasi` akan dikalikan terus dengan nilai `x`. kalau perulangan terdapat 3 kali maka `hasiliterasi` akan bernilai 3 kemudian akan dikalikan dengan `x`. nilai dari `hasiliterasi` akan bertambah seiring perulangan, misal 3 pangkat 3 maka  $1 \times 3, 3 \times 3, 9 \times 3$  dan seterusnya. Kemudian hasil dari perulangan akan di outputkan diluar perulangan.

#### 4. Tugas 4

##### Source code

```
package main

import "fmt"

func main() {

    var n int

    fmt.Scan(&n)

    hasil := 1

    for iterasi := n; iterasi >= 1; iterasi-- {

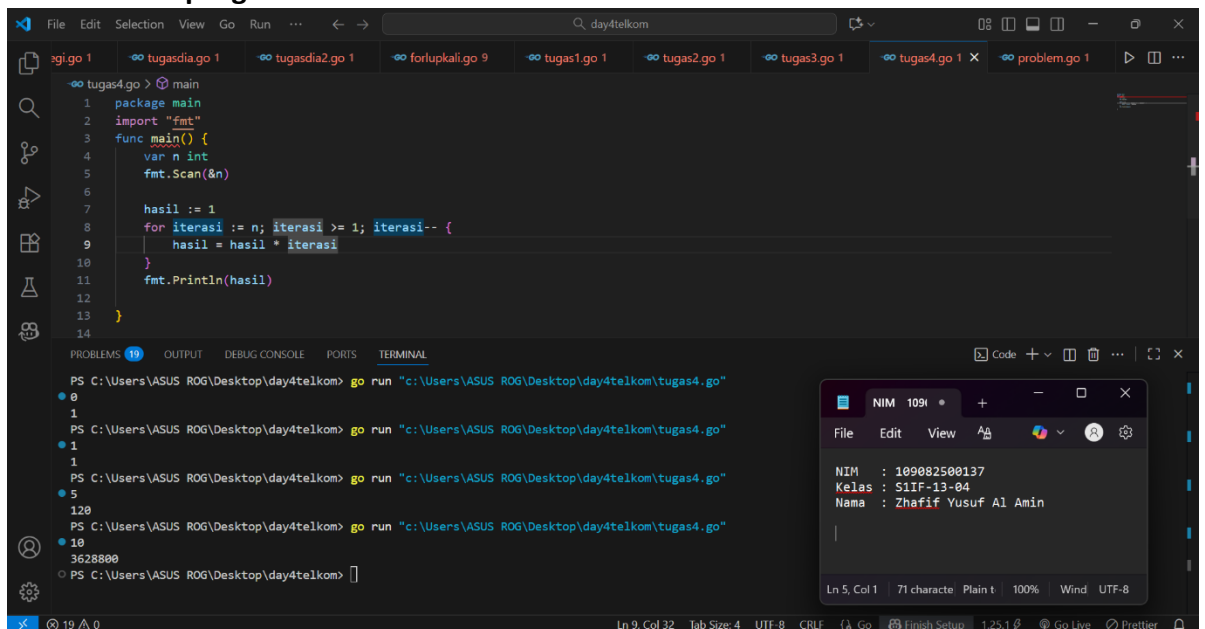
        hasil = hasil * iterasi

    }

    fmt.Println(hasil)

}
```

##### Screenshoot program



##### Deskripsi program

package main : menandakan program utama ini ibaratnya kita akan membangun rumah & package main ini sebagai temboknya.

`import "fmt"` : berfungsi memanggil package "fmt" ini ibarat dalam membangun rumah sebagai semen, pasir, batu, air.

`func main() {...}`: ini bagaikan akses dengan filosofi pintu rumah, sebuah rumah tidak akan bisa di masuki tanpa pintu, masa iya rumah full tembok.

`var n int`: deklarasi variabel n menggunakan tipe data integer.

`fmt.Scan(&n)` : berfungsi untuk membaca inputan yang telah kita input lalu disimpan ke variabel &n.

```
hasil := 1
```

```
for iterasi := n; iterasi >= 1; iterasi-- {
```

```
    hasil = hasil * iterasi
```

```
}
```

```
fmt.Println(hasil)
```

: kita deklarasi dulu isi dari variabel hasil itu nilainya berapa, disini kita isi 1 karena mau memfaktorialkan dari perulangan. Variabel hasil ini berfungsi sebagai tempat penyimpanan sementara dari hasil perkalian pada perulangan. Perulangan dimulai dari n lalu selanjutnya akan berkurang satu per satu sampai 1. Setiap perulangan hasilnya bakal terus bertambah lalu dikalikan dengan angka sebelumnya. Misal n=5 >= 1 jadinya akan berulang sampai 5x4x3x2x1. Jika kondisi sudah terpenuhi maka perulangan akan berhenti dan memberikan output dari variabel hasil didalam perulangan.