

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

**MODUL 5 & 6**  
**FOR-LOOP**



**Disusun oleh:**

**HANAN FAHRI ABIYYU**

**109082500131**

**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 a, b, i int

    fmt.Scan(&a, &b)

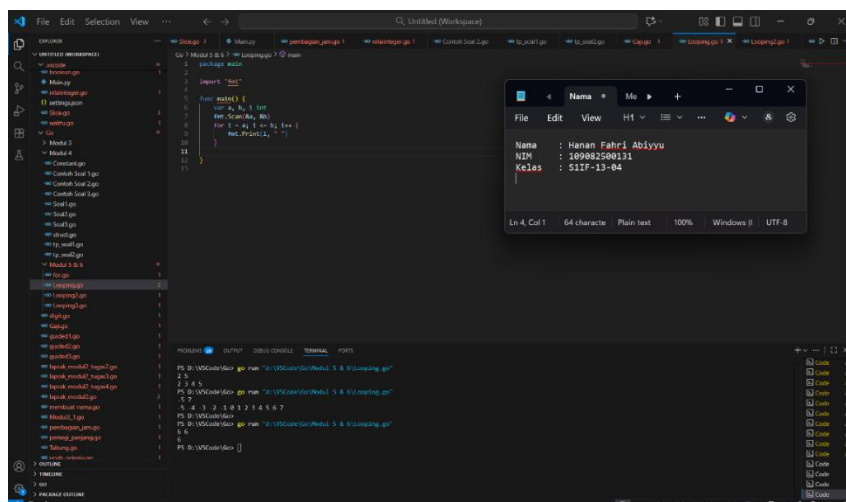
    for i = a; i <= b; i++ {

        fmt.Print(i, " ")

    }

}
```

### Screenshoot program



### Deskripsi program

Penjelasan tentang program di atas, membuat program dalam bahasa GO mengonversi total detik menjadi format jam:menit:detik. Paket yang digunakan yaitu "fmt" yang berfungsi untuk membaca suatu Input dan menghasilkan atau mengeluarkan suatu Output. Variabel yang digunakan yaitu terdapat a, b, i menggunakan tipe data **int** karena input dan output yang dihasilkan harus berupa bilangan bulat. Untuk kode ini berbeda dengan modul-modul sebelumnya, pada kode ini menggunakan (for) berfungsi sebagai perulangan kode dan bertujuan mempersingkat dan simpel dari penulisan kode.

Cara kerja, a dan b sebagai variabel Input, i sebagai perulangan.

for i = a; a <= b; i++ (i++ menandakan i bertambah 1 jika nilai kebenaran dari perbandingan bernilai True). Contoh diinputkan a = 2, b = 5.

For i = 2; 2 <= 5; i++ (jika benar a lebih kecil atau sama dengan b, maka iterasi bertambah 1) dan begitu seterusnya sampai nilai a **tidak** kurang/sama dengan b, tergantung pada operator juga.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var j, alas, tinggi, n int

    var luas float64

    fmt.Print("Masukkan Input Jumlah :")

    fmt.Scan(&n)

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

        fmt.Print("Masukkan angkanya :")

        fmt.Scan(&alas, &tinggi)

        luas = 0.5 * float64(alas*tinggi)

        fmt.Println(luas)
```

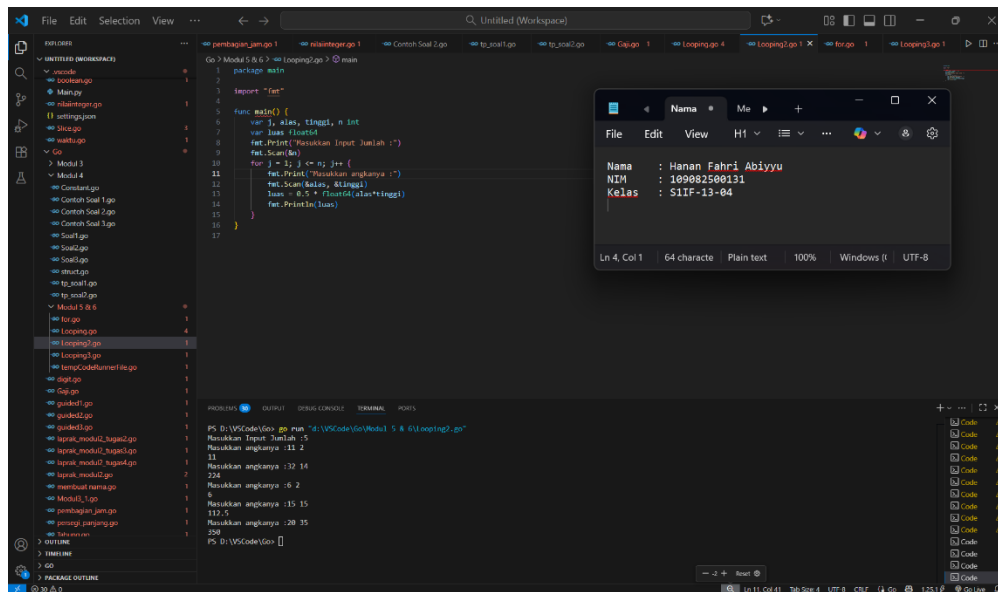
```

    }

}

```

## Screenshoot program



## Deskripsi program

Menghitung sejumlah  $n$  luas segitiga, apabila diketahui sisi alas dan tinggi dari masing-masing segitiga. Dengan variabel  $j$ , alas, tinggi bertipe data integer, luas bertipe data float64, atau dalam Pseudocode biasa disebut bilangan real.

Diinputkan jumlah segitiga yang akan dihitung, kemudian setiap segitiga di inputkan alas dan tinggi nya dan setiap dari Input alas dan tinggi, dikali dengan 0,5. Dan akan muncul berupa Output dari perhitungan dengan rumus tadi.

Pada syntax for, sudah ditetapkan bahwa  $j = 1$ , jika  $j$  lebih kecil/sama dengan  $n$ , maka nilai  $j$  bertambah 1 dan akan melakukan perulangan sebanyak  $n$  kali, tergantung pada nilai  $n$  (Input yang dimasukkan user).

## 3. Guided 3

### Source Code

```

package main

import "fmt"

func main() {

```

```

var a, b, hasil int

fmt.Print("Input :")

fmt.Scan(&a, &b)

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

    hasil += a

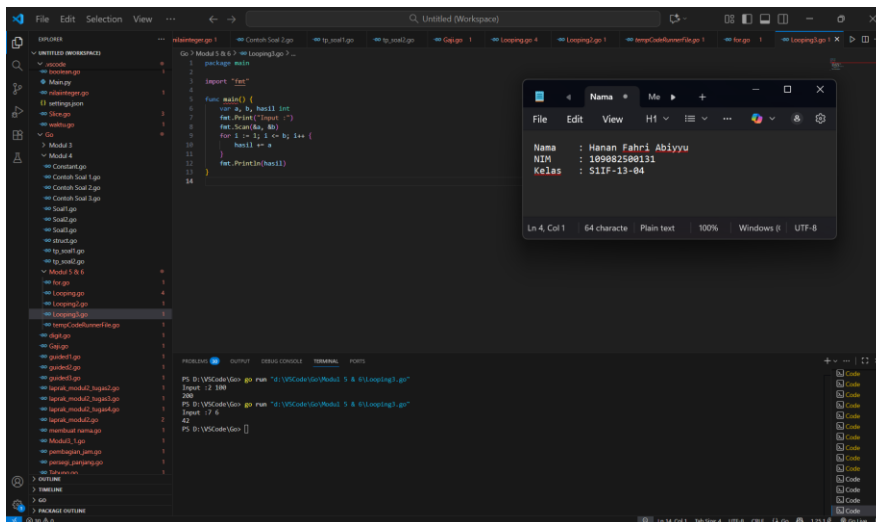
}

fmt.Println(hasil)

}

```

## Screenshoot program



## Deskripsi program

Membuat program perkalian tanpa menggunakan operator kali(\*). Terdapat 2 Input yang akan digunakan dengan variabel a, b bertipe data integer dan 1 Output dengan variabel hasil bertipe data integer.

Pengguna menginputkan angka yang akan dikali, kemudian akan dimasukkan dalam perulangan for, pada for terdapat variabel i yang sudah ditetapkan nilai nya = 1, dan i akan dicek nilai kebenarannya dengan cara dibandingkan dengan nilai Input b(jika i lebih kecil/sama dengan b, maka nilai i bertambah 1) dengan nilai awal 1 karena 1 memenuhi syarat bahwa kurang dari b maka akan melakukan perulangan 1 + 1 = 2, 2 + 1=3(sebanyak b kali).

Output ditampilkan menggunakan rumus hasil += a (+=) berarti menambahkan dirinya sendiri atau bisa juga ditulis hasil = hasil + a

# TUGAS

## 1. Tugas 1

### Source code

```
package main

import "fmt"

func main() {

    var n, i int

    var hasil int

    fmt.Scan(&n)

    for i = 0; i <= n; i++ {

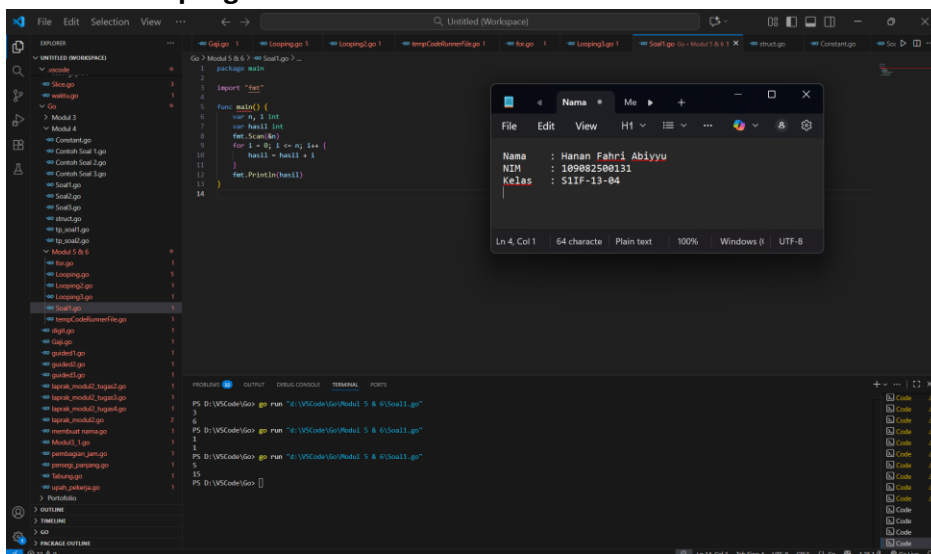
        hasil = hasil + i

    }

    fmt.Println(hasil)

}
```

### Screenshoot program



### Deskripsi program

Menjumlahkan sekumpulan bilangan menggunakan perulangan for. Terdapat Input yang akan digunakan dengan variabel n, bertipe data integer dan Output dengan variabel hasil bertipe data integer.

User menginputkan angka yang akan dihitung dan angka tersebut dimasukkan sebagai nilai dari variabel n. Pada bagian perulangan for, terdapat variabel i yang sudah ditetapkan nilai awalnya yaitu 0, step selanjutnya dibandingkan dengan nilai n menggunakan operator lebih kecil/sama dengan ( $\leq$ ), jika nilai  $i \leq n$  maka i bertambah 1, hingga nilai pada i lebih besar dari n, karena operator yang digunakan ( $\leq$ ) yang berarti jika  $i = n$  juga masih memenuhi syarat operator tersebut.

Output akan mengeluarkan hasil dengan rumus  $hasil = hasil + i$  (menambahkan dengan dirinya sendiri).

Contoh : Input 3 Output 6

Mengapa hasilnya 6? Karena Input = 3 cara kerjanya ialah,  $3+2+1 = 3 + 3 = 6$

Pada contoh kedua, Input = 1 mengapa hasilnya tetap 1? Karena bilangan yang diinputkan hanya satu, maka hasilnya 1 itu sendiri. Misal juga Input = 2, maka akan keluar 3, karena  $2+1 = 3$ . Ini mirip dengan konsep faktorial, dimana bilangan Input akan mengalikan dengan angka menurun/semakin mengecil.

## 2. Tugas 2

### Source code

```
package main

import (
    "fmt"
    "math"
)

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

    fmt.Print("Masukkan jumlah kerucut yang akan
    dihitung: ")

    fmt.Scan(&n)
```

```

for i = 0; i < n; i++ {

    fmt.Scan(&r, &t)

    jarijari := float64(r)

    tinggi := float64(t)

    volume := (1.0 / 3.0) * math.Pi * jarijari *
jarijari * tinggi

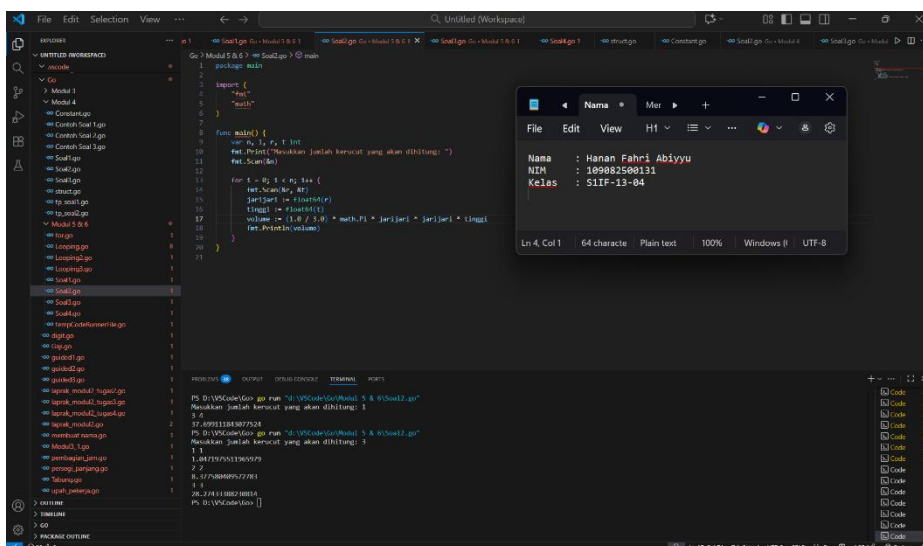
    fmt.Println(volume)

}

}

```

## Screenshoot program



## Deskripsi program

Program menghitung sebuah volume dari banyaknya Kerucut. Terdapat Input yang akan digunakan dengan variabel n dan variabel lain i, r, t bertipe data integer.

Pengguna menginputkan banyaknya Kerucut yang akan dimasukkan dalam variabel n. Selanjutnya akan masuk dalam perulangan for, dalam for terdapat variabel i yang sudah ditetapkan nilainya dengan angka 0, kemudian i dibandingkan dengan n menggunakan operator (<) tanpa sama dengan. Nilai i bertambah 1 sebanyak kurang dari nilai n, setelah nilai (false), akan dilanjutkan ke bawah. Terdapat variabel jarijari dan tinggi yang didefinisikan dengan r, t bertipe data float64.

Selanjutnya akan dihitung volume kerucut dengan rumus  $\text{volume} = (1.0 / 3.0) \times \text{math.Pi} \times \text{jarijari} \times \text{jarijari} \times \text{tinggi}$  dan akan menghasilkan Output dari hasil penghitungan volume.



Contoh : Input Kerucut 1

Input jarijari : 3

Input tinggi : 4

Output = 37.699111843077524

### 3. Tugas 3

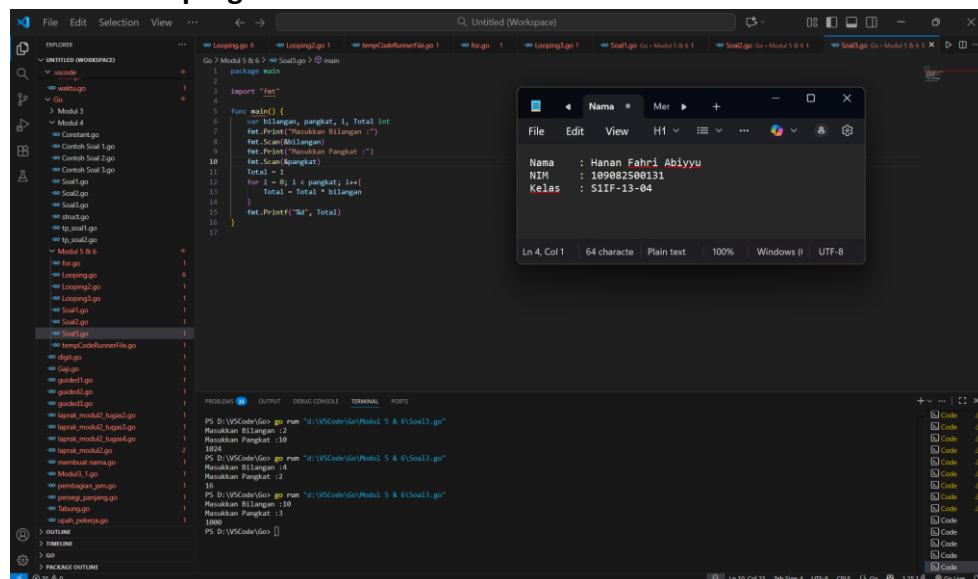
#### Source code

```
package main

import "fmt"

func main() {
    var bilangan, pangkat, i, Total int
    fmt.Print("Masukkan Bilangan :")
    fmt.Scan(&bilangan)
    fmt.Print("Masukkan Pangkat :")
    fmt.Scan(&pangkat)
    Total = 1
    for i = 0; i < pangkat; i++){
        Total = Total * bilangan
    }
    fmt.Printf("%d", Total)
}
```

#### Screenshoot program



#### Deskripsi program

Membuat program perkalian pangkat. Terdapat 3 Input yang akan digunakan dengan variabel bilangan, pangkat, i bertipe data integer dan 1 Output dengan variabel total bertipe data integer. Menggunakan paket "fmt" untuk membaca Input dan menampilkan Output.

Pengguna menginputkan 2 angka yang akan dihitung dan angka tersebut akan dimasukkan pada variabel bilangan dan pangkat, nilai total ditetapkan = 1.

Pada perulangan for, i juga sudah ditetapkan nilainya yaitu 0, kemudian i dibandingkan dengan pangkat menggunakan operator (<) tanpa sama dengan. Nilai i bertambah 1 sebanyak kurang dari nilai pangkat, setelah nilai (false), akan dilanjutkan ke bawah untuk dihitung menggunakan rumus  $Total = Total * bilangan$ . Dan akan menghasilkan Output berupa hasil perhitungan Input bilangan dan Input pangkat.

Contoh : Input 2 10 yang berarti 2 pangkat 10

Output 1024

Input 4 2

Output 16

Input 10 3

Output 1000

#### 4. Tugas 4

##### Source code

```
package main

import "fmt"

func main() {
    var n, total int

    fmt.Print("Masukkan angka :")

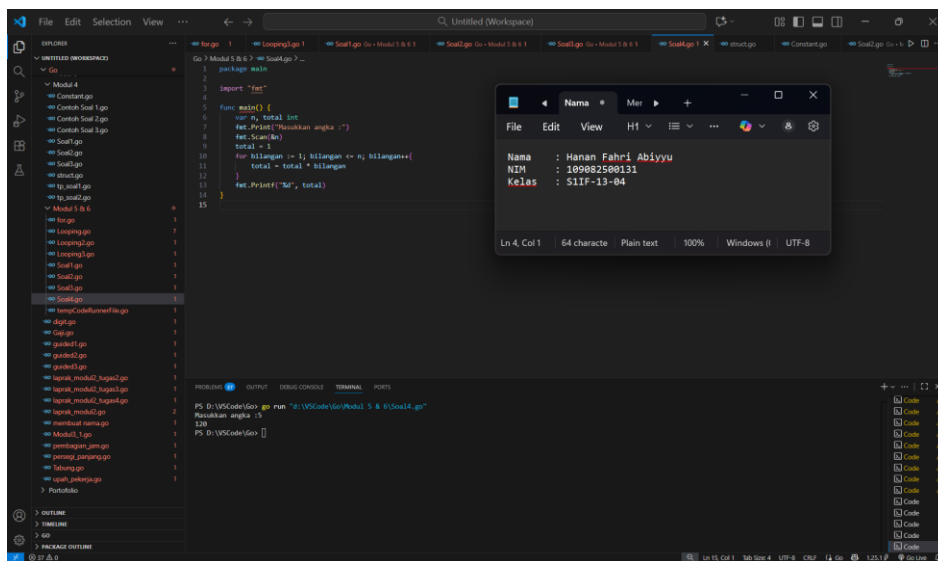
    fmt.Scan(&n)

    total = 1

    for bilangan := 1; bilangan <= n; bilangan++{
        total = total * bilangan
    }

    fmt.Printf("%d", total)
}
```

## Screenshoot



## Deskripsi program

Menghitung faktorial dari suatu Input, Terdapat Input yang akan digunakan dengan variabel  $n$ , Total bertipe data integer dan Output dengan variabel hasil bertipe data integer.

Misal diinputkan angka yang akan dihitung dan angka tersebut akan dimasukkan pada variabel  $n$ , nilai total ditetapkan = 1.

Dalam perulangan for, nilai bilangan sudah ditetapkan = 1, selanjutnya selanjutnya dibandingkan dengan nilai  $n$  menggunakan operator lebih kecil/sama dengan ( $\leq$ ), jika nilai  $i \leq n$  maka  $i$  bertambah 1, hingga nilai pada  $i$  lebih besar dari  $n$ , karena operator yang digunakan ( $\leq$ ) yang berarti jika  $i = n$  juga masih memenuhi syarat operator tersebut. Setelah nilai bilangan tidak memenuhi syarat operator perbandingan, maka akan dilanjutkan ke penghitungan. Menggunakan rumus  $\text{total} = \text{total} * \text{bilangan}$  (total dikalikan dengan dirinya sendiri).

Contoh : Input 5

Output 120, dihasilkan dari perkalian faktorial  $5 \times 4 \times 3 \times 2 \times 1 = 120$ .

