

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

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



**Disusun oleh:**  
**ALMA BONITA MIA WARDHANA**  
**109082500015**  
**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 int

    var j int

    fmt.Scan(&a, &b)

    for j = a; j <=b; j+=1 {

        fmt.Print(j, " ")

    }

}
```

#### Screenshoot program

```
File Edit Selection View Go Run Terminal Help
Modul 5&6_Almabonita

EXPLORER
MODUL 5&6_ALMABONITA
  guided1
  latsoli.go
  guided2
  soal1
  soal2
  soal3
  soal4

latsoli.go
1 package main
2 import "fmt"
3
4
5 func main() {
6     var a, b int
7     var j int
8     fmt.Scan(&a, &b)
9     for j = a; j <=b; j+=1 {
10         fmt.Print(j, " ")
11     }
12 }

TERMINAL
PS D:\Modul 5&6_Almabonita> go run "d:\Modul 5&6_Almabonita\guided1\latsoli.go"
2 5
2 3 4 5
PS D:\Modul 5&6_Almabonita> go run "d:\Modul 5&6_Almabonita\guided1\latsoli.go"
6 6
6
PS D:\Modul 5&6_Almabonita> go run "d:\Modul 5&6_Almabonita\guided1\latsoli.go"
-5 7
-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
PS D:\Modul 5&6_Almabonita>
```

#### Deskripsi program

Program di atas ditulis menggunakan bahasa Go dan berfungsi untuk menampilkan deretan angka dari nilai awal hingga nilai akhir yang dimasukkan oleh pengguna. Pertama, program mengimpor paket "fmt" yang digunakan untuk melakukan input dan output. Di dalam fungsi main(), terdapat dua variabel bertipe int yaitu *a* dan *b* sebagai batas awal dan batas akhir, serta variabel *j* sebagai penghitung dalam perulangan. Program meminta pengguna memasukkan dua angka melalui perintah `fmt.Scan(&a, &b)`.

Setelah itu, perulangan `for j = a; j <= b; j += 1` akan dijalankan, di mana nilai *j* dimulai dari *a* dan akan terus bertambah satu hingga mencapai atau melewati *b*. Di dalam perulangan, setiap nilai *j* dicetak menggunakan `fmt.Print(j, " ")`, sehingga hasil akhirnya adalah deretan angka dari *a* sampai *b* yang dipisahkan oleh spasi.

## 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)

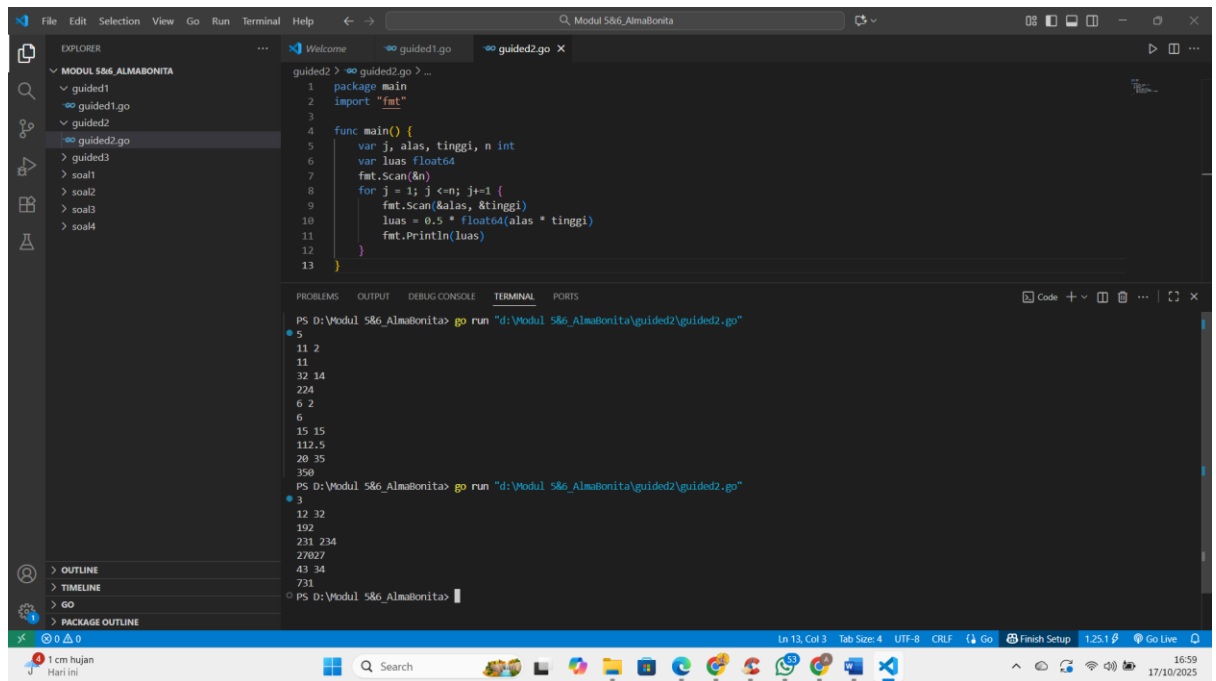
        luas = 0.5 * float64(alas * tinggi)

        fmt.Println(luas)

    }

}
```

**Screenshoot program**



### Deskripsi program

Program pada gambar tersebut merupakan program dalam bahasa Go yang digunakan untuk menghitung luas segitiga sebanyak beberapa kali sesuai jumlah data yang dimasukkan oleh pengguna. Pertama, program mengimpor paket "fmt" untuk menangani input dan output. Di dalam fungsi main(), terdapat variabel *j*, *alas*, *tinggi*, dan *n* bertipe integer, serta variabel *luas* bertipe float64 untuk menampung hasil perhitungan. Program meminta pengguna memasukkan nilai *n*, yaitu jumlah segitiga yang ingin dihitung.

Kemudian, dengan menggunakan perulangan for *j* = 1; *j* <= *n*; *j*++, program meminta input nilai alas dan tinggi untuk setiap segitiga. Rumus yang digunakan untuk menghitung luas segitiga adalah  $0.5 \times \text{alas} \times \text{tinggi}$ , dan hasilnya dikonversi ke tipe float64 agar mendukung hasil desimal. Setelah perhitungan selesai, hasil luas setiap segitiga ditampilkan menggunakan fmt.Println(luas). Dengan demikian, program ini secara otomatis menghitung dan menampilkan luas beberapa segitiga berdasarkan data yang diberikan pengguna.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
```

```

var j, v1, v2 int

var hasil int

fmt.Scan(&v1, &v2)

hasil = 0

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

    hasil = hasil + v1

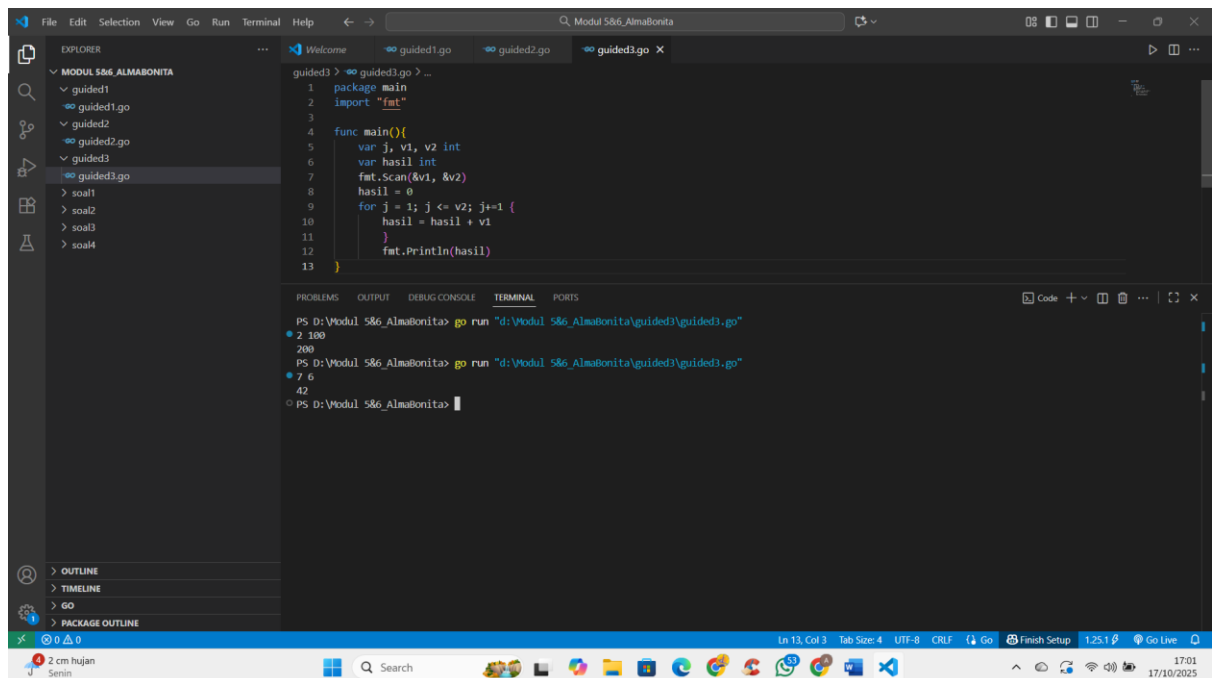
}

fmt.Println(hasil)

}

```

## Screenshoot program



## Deskripsi program

Program pada gambar tersebut ditulis menggunakan bahasa Go dan berfungsi untuk menghitung hasil perkalian antara dua bilangan bulat menggunakan konsep penjumlahan berulang. Program ini mengimpor paket "fmt" untuk menangani input dan output. Di dalam fungsi `main()`, terdapat variabel `v1` dan `v2` sebagai dua bilangan yang akan dikalikan, serta variabel `hasil` untuk menyimpan hasil perhitungan.

Program meminta pengguna memasukkan dua bilangan melalui `fmt.Scan(&v1, &v2)`. Selanjutnya, variabel `hasil` diinisialisasi dengan nilai 0, lalu perulangan `for j = 1; j <= v2;`

j++ dijalankan sebanyak nilai *v2* kali. Di dalam perulangan, nilai *v1* dijumlahkan ke dalam *hasil* secara berulang, sehingga pada akhir proses didapatkan hasil perkalian antara *v1* dan *v2*. Terakhir, hasil akhir ditampilkan menggunakan `fmt.Println(hasil)`. Dengan kata lain, program ini menghitung perkalian dua angka tanpa menggunakan operator perkalian langsung, melainkan melalui proses penjumlahan bertahap.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var n, i, jumlah int

    fmt.Print("Masukkan bilangan n: ")

    fmt.Scan(&n)

    jumlah = 0

    for i = 1; i <= n; i++ {
        jumlah = jumlah + i
    }

    fmt.Println("Hasil penjumlahan dari 1 sampai", n, "adalah",
jumlah)
}
```

#### Screenshoot program

## Deskripsi program

Program di atas merupakan program sederhana dalam bahasa Go yang digunakan untuk menghitung jumlah semua bilangan dari 1 hingga  $n$ , di mana  $n$  adalah bilangan bulat positif yang dimasukkan oleh pengguna. Program dimulai dengan mendeklarasikan variabel  $n$ ,  $i$ , dan  $jumlah$  bertipe integer. Setelah pengguna memasukkan nilai  $n$  melalui perintah `fmt.Scan(&n)`, program menginisialisasi  $jumlah$  dengan nilai 0.

Kemudian, perulangan `for i = 1; i <= n; i++` digunakan untuk menambahkan setiap nilai  $i$  ke variabel  $jumlah$  secara bertahap hingga mencapai nilai  $n$ . Setelah proses perulangan selesai, program menampilkan hasil akhir berupa total penjumlahan dari 1 sampai dengan  $n$  menggunakan perintah `fmt.Println`. Dengan demikian, program ini berfungsi sebagai kalkulator sederhana untuk menghitung deret penjumlahan bilangan positif secara otomatis.

## 2. Tugas 2

### Source code

```
package main

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

```

func main() {
    var n, jari, tinggi int
    var volume float64

    fmt.Print("Masukkan jumlah kerucut: ")
    fmt.Scan(&n)

    for i := 1; i <= n; i++ {
        fmt.Scan(&jari, &tinggi)

        volume = (1.0 / 3.0) * math.Pi *
math.Pow(float64(jari), 2) * float64(tinggi)

        fmt.Println(volume)
    }
}

```

## Screenshoot program

The screenshot shows a Go IDE with the following components:

- EXPLORER:** A file tree on the left showing a project named 'MODUL 5&6\_ALMABONITA' with subfolders 'guided1', 'guided2', 'guided3', 'soal1', 'soal2', 'soal3', and 'soal4'. The file 'soal2.go' is selected.
- EDITOR:** The main area showing the source code of 'soal2.go'. The code is identical to the one provided in the first block.
- TERMINAL:** The bottom panel showing the output of the program. It displays the prompt 'Masukkan jumlah kerucut: 3' followed by three lines of input and output:
 

```

1 1
1.0471975511965979
2 2
8.377580409572783
3 3
28.27433388230814
PS D:\Modul 5&6_AlmaBonita>

```

## Deskripsi program

Program di atas digunakan untuk menghitung volume sejumlah kerucut berdasarkan data jari-jari alas dan tinggi yang dimasukkan oleh pengguna. Program dimulai dengan meminta jumlah kerucut ( $n$ ), lalu menggunakan perulangan for untuk membaca pasangan nilai jari-jari dan tinggi setiap kerucut. Rumus yang digunakan untuk



menghitung volume adalah  $(1/3) \times \pi \times r^2 \times t$ , di mana  $r$  adalah jari-jari alas dan  $t$  adalah tinggi kerucut.

Fungsi `math.Pow` digunakan untuk menghitung kuadrat dari jari-jari, sedangkan konstanta `math.Pi` mewakili nilai  $\pi$ . Setelah perhitungan selesai, program menampilkan hasil volume tiap kerucut secara terpisah. Dengan demikian, program ini dapat menghitung volume beberapa kerucut sekaligus secara otomatis berdasarkan input pengguna.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

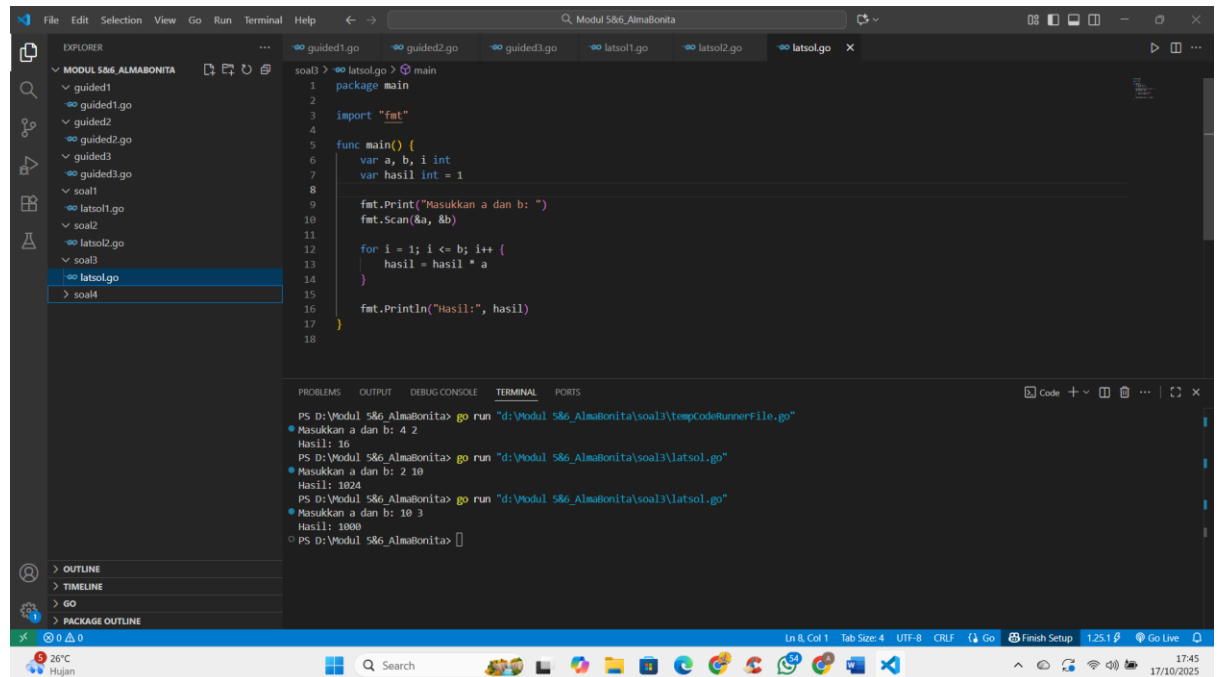
func main() {
    var a, b, i int
    var hasil int = 1

    fmt.Print("Masukkan a dan b: ")
    fmt.Scan(&a, &b)

    for i = 1; i <= b; i++ {
        hasil = hasil * a
    }

    fmt.Println("Hasil:", hasil)
}
```

## Screenshoot program



### Deskripsi program

Program di atas merupakan program sederhana dalam bahasa Go yang berfungsi untuk menghitung hasil pemangkatan dari suatu bilangan. Pertama, program mendeklarasikan variabel *a* dan *b* sebagai bilangan bulat, di mana *a* adalah bilangan pokok dan *b* adalah pangkatnya, serta variabel *hasil* yang diinisialisasi dengan nilai 1 untuk menyimpan hasil perkalian. Program meminta pengguna memasukkan dua nilai melalui perintah `fmt.Scan(&a, &b)`.

Selanjutnya, perulangan `for i = 1; i <= b; i++` dijalankan sebanyak *b* kali, dan pada setiap iterasi, nilai *hasil* dikalikan dengan *a*, sehingga setelah perulangan selesai, *hasil* menyimpan nilai dari *a* pangkat *b*. Terakhir, hasil perhitungan tersebut ditampilkan ke layar menggunakan perintah `fmt.Println("Hasil:", hasil)`.

## 4. Tugas 3

### Source code

```
package main

import "fmt"

func main() {
    var a, i int
    var hasil int = 1

    fmt.Print("masukan a: ")
    fmt.Scan(&a)

    for i = 1; i <= a; i++ {
```

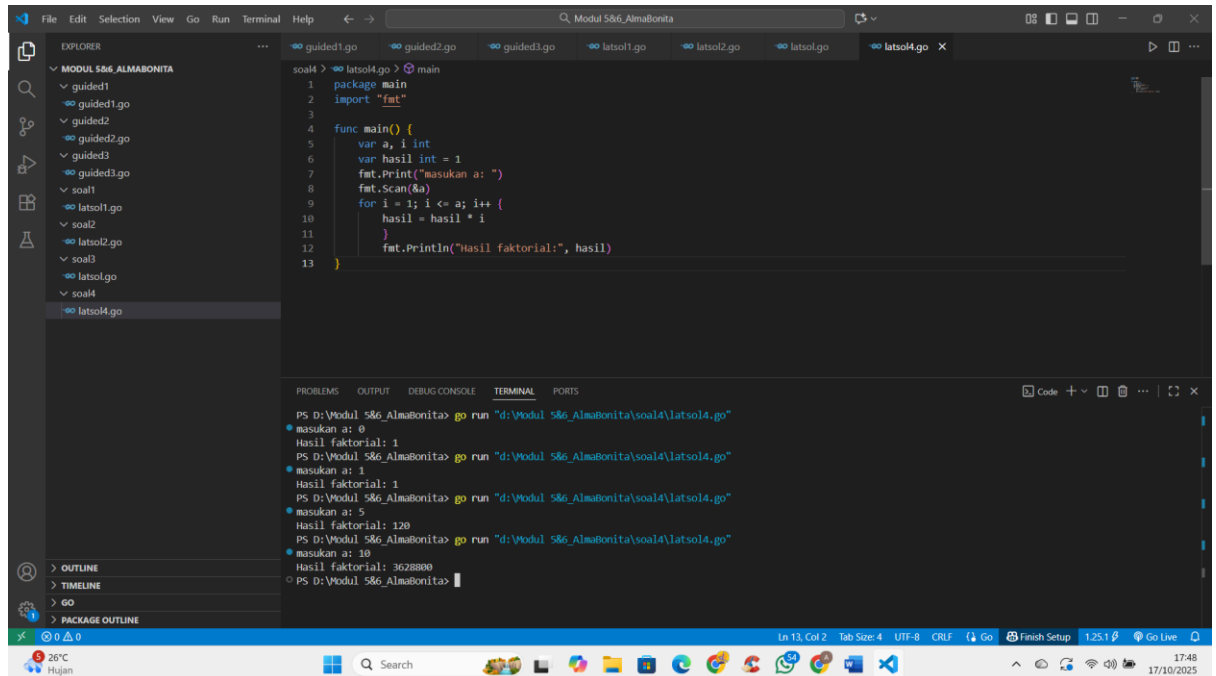
```

        hasil = hasil * i
    }

    fmt.Println("Hasil faktorial:", hasil)
}

```

## Screenshoot program



## Deskripsi program

Program di atas merupakan program dalam bahasa Go yang digunakan untuk menghitung faktorial dari suatu bilangan bulat positif yang dimasukkan oleh pengguna. Pertama, program mendeklarasikan variabel *a* untuk menampung input bilangan, variabel *i* sebagai penghitung dalam perulangan, serta variabel *hasil* yang diinisialisasi dengan nilai 1 untuk menyimpan hasil faktorial.

Melalui perintah `fmt.Scan(&a)`, pengguna diminta memasukkan nilai *a*. Selanjutnya, program menggunakan perulangan `for i = 1; i <= a; i++` untuk mengalikan nilai *hasil* dengan *i* secara berurutan, dimulai dari 1 hingga *a*. Proses ini menghasilkan nilai faktorial dari *a*, yaitu hasil perkalian semua bilangan dari 1 sampai *a*. Setelah perulangan selesai, program menampilkan hasil akhir menggunakan `fmt.Println("Hasil faktorial:", hasil)`.