

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 5&6**  
**“FOR-LOOP”**



**DISUSUN OLEH:**  
**Muhammad Shabrian Fadly**  
**103112400087**  
**S1 IF-12-01**  
**DOSEN:**  
**Yohani Setiya Rafika Nur, M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024/2025**

## **DASAR TEORI**

### **For-Loop**

For loop adalah statement yang digunakan untuk mengeksekusi suatu blok kode secara berulang. Biasanya digunakan untuk melakukan iterasi terhadap nilai yang ada pada data slice, array, atau pada data yang bersifat iterable (dapat dilakukan perulangan).

### **Statement Pada For Loop**

Hampir sama seperti if dan switch, statement pada for loop. Ada dua statement yang bisa ditambahkan yaitu:

- 1) Init statement, merupakan sebuah statement sebelum for loop dieksekusi,
- 2) Post statement, merupakan statement yang akan selalu dieksekusi di akhir pada tiap-tiap perulangan.
- 3)

### **Range**

For loop bisa digunakan untuk melakukan iterasi pada data collection. Contoh data collection adalah map, array, dan slice. Untuk melakukan iterasi menggunakan for loop kita bisa menggunakan keyword ``range``.

## Contoh Soal:

### 1. Contoh Soal1

Input:

```
package main

import "fmt"

func main() {
    var a, b int
    var j int
    fmt.Scan(&a, &b)
    for j = a; j <= b; j = j + 1 {
        fmt.Print(j, " ")
    }
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul5\conso1\tempCodeRunnerFile.go"
2 5
2 3 4 5
PS D:\Coding manja> go run "d:\Coding manja\modul5\conso1\tempCodeRunnerFile.go"
6 6
6
PS D:\Coding manja> go run "d:\Coding manja\modul5\conso1\tempCodeRunnerFile.go"
-5 7
-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
```

Deskripsi Program:

#### 1. Deklarasi Variabel

- Dua variabel a dan b dideklarasikan untuk menyimpan input dari pengguna. Keduanya merupakan bilangan bulat (int).
- Variabel j dideklarasikan sebagai variabel yang digunakan dalam proses iterasi (perulangan).

#### 2. Pengambilan Input

- Fungsi `fmt.Scan(&a, &b)` digunakan untuk membaca dua bilangan bulat dari pengguna dan menyimpannya di variabel a dan b. Pengguna akan diminta memasukkan dua angka.

#### 3. Perulangan for

- Perulangan for digunakan untuk mencetak bilangan dari a hingga b.
- Loop dimulai dari nilai `j = a`, dan akan terus berjalan selama `j` kurang dari atau sama dengan `b`. Setiap kali iterasi selesai, nilai `j` akan ditambah 1 (`j = j + 1`).
- Pada setiap iterasi, nilai `j` akan dicetak dengan menggunakan `fmt.Print(j, " ")`, sehingga bilangan dicetak secara berurutan dengan spasi di antaranya.

## 2. Contoh Soal2

Input:

```
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)
    }
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul5\conso2\tempCodeRunnerFile.go"
5 11 2 32 14 6 2 15 15 20 35
11
224
6
112,5
350
PS D:\Coding manja> go run "d:\Coding manja\modul5\conso2\tempCodeRunnerFile.go"
3 12 32 231 234 43 34
192
27027
731
```

Deskripsi Program:

### 1. Deklarasi Variabel

- j: Variabel loop yang digunakan untuk iterasi.
- alas dan tinggi: Variabel untuk menyimpan nilai alas dan tinggi segitiga yang diinput oleh pengguna.
- n: Variabel yang menyimpan jumlah segitiga yang akan dihitung.
- luas: Variabel tipe float64 yang digunakan untuk menyimpan hasil perhitungan luas segitiga.

### 2. Pengambilan Input

- fmt.Scan(&n): Program meminta input n dari pengguna yang merepresentasikan jumlah segitiga yang akan dihitung.

### 3. Perulangan for

- for j = 1; j <= n; j += 1: Perulangan dimulai dari 1 hingga n. Setiap kali loop berjalan, program akan meminta pengguna memasukkan alas dan tinggi segitiga.
- fmt.Scan(&alas, &tinggi): Program meminta input untuk alas dan tinggi segitiga.

- `luas = 0.5 * float64(alas*tinggi)`: Program menghitung luas segitiga menggunakan rumus luas segitiga, yaitu  $\text{luas} = \frac{1}{2} \times \text{alas} \times \text{tinggi}$ . Perkalian antara alas dan tinggi dikonversi ke tipe float64 agar hasil perhitungan menjadi desimal.

### 3. Contoh Soal3

Input:

```
package main

import "fmt"

func main() {
    var j, hasil, v1, v2 int
    fmt.Scan(&v1, v2)
    for j = 1; j <= v2; j++ {
        hasil = hasil + v1
    }
    fmt.Print(hasil)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul5\conso3\contoh3.go"
2 100
200
PS D:\Coding manja> go run "d:\Coding manja\modul5\conso3\contoh3.go"
7 6
42
```

Deskripsi Program:

#### 1. Deklarasi Variabel

- j: Variabel loop yang digunakan sebagai penghitung iterasi.
- v1 dan v2: Variabel yang digunakan untuk menyimpan dua bilangan bulat yang akan diinput pengguna.
- hasil: Variabel yang digunakan untuk menyimpan hasil perkalian yang dilakukan dengan penjumlahan berulang.

#### 2. Pengambilan Input

- fmt.Scan(&v1, &v2): Program meminta pengguna memasukkan dua bilangan bulat, yaitu v1 (bilangan yang akan dijumlahkan) dan v2 (jumlah pengulangan atau faktor kali).

#### 3. Inisialisasi Variabel

- hasil = 0: Variabel hasil diinisialisasi dengan nilai 0 sebelum dilakukan operasi penjumlahan.

#### 4. Perulangan for

- for j = 1; j <= v2; j += 1: Program melakukan iterasi sebanyak v2 kali. Setiap iterasi, nilai v1 akan ditambahkan ke variabel hasil.
- hasil = hasil + v1: Setiap kali loop berjalan, nilai v1 ditambahkan ke hasil. Ini mensimulasikan operasi perkalian dengan penjumlahan berulang.

## Latihan Soal

### 1. Latihan Soal1

Buatlah program untuk menjumlahkan sekumpulan bilangan.

Masukan terdiri dari suatu bilangan bulat positif n.

Keluaran berupa bilangan hasil penjumlahan dari 1 sampai dengan n.

Input:

```
package main

import "fmt"

func main() {
    var input1 int
    var result int = 0

    fmt.Scan(&input1)

    for x := 1; x <= input1; x++ {
        result = result + x
    }
    fmt.Println(result)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal1.go\tempCodeRunnerFile.go"
3
6
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal1.go\tempCodeRunnerFile.go"
1
1
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal1.go\tempCodeRunnerFile.go"
5
15
```

Deskripsi Program:

#### 1. Deklarasi Variabel

- input1: Variabel tipe bilangan bulat (int) yang digunakan untuk menyimpan input dari pengguna, yaitu batas atas dari perhitungan.
- result: Variabel yang digunakan untuk menyimpan hasil perhitungan jumlah, diinisialisasi dengan nilai 0.

#### 2. Pengambilan Input

- fmt.Scan(&input1): Program meminta pengguna untuk memasukkan sebuah bilangan bulat (input1). Nilai ini akan digunakan sebagai batas atas dari perulangan.

### 3. Perulangan for

- `for x := 1; x <= input1; x++`: Perulangan for dimulai dari  $x = 1$  dan berlanjut hingga  $x$  mencapai nilai `input1`. Setiap kali loop berjalan,  $x$  akan bertambah 1.
- `result = result + x`: Pada setiap iterasi, nilai  $x$  ditambahkan ke `result`. Ini menyebabkan `result` menjadi akumulasi jumlah dari semua bilangan dari 1 hingga `input1`.



## 2. Latihan Soal2

Buatlah program yang digunakan untuk menghitung volume sejumlah n kerucut, apabila diketahui panjang jari-jari alas kerucut dan tinggi dari kerucut.

Masukan terdiri dari beberapa baris. Baris pertama adalah bilangan bulat n, selanjutnya n baris berikutnya masing-masing merupakan panjang jari-jari alas kerucut dan tinggi dari kerucut.

Keluaran terdiri dari beberapa baris, yang masing-masingnya menyatakan volume dari n kerucut.

Input:

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var input1, input2, input3 int
    var result float64

    fmt.Scan(&input1)

    for x := 0; x < input1; x++ {
        fmt.Scan(&input2, &input3)
        result = (1.0 / 3.0) * math.Pi * math.Pow(float64(input2), 2) * float64(input3)
        fmt.Println(result)
    }
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal2.go\soal2.go"
1 3 4
37.699111843077524
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal2.go\soal2.go"
3 1 1 2 2 3 3
1.0471975511965979
8.377580409572783
28.27433388230814
```

Deskripsi Program:

### 1. Deklarasi Variabel

- input1: Variabel tipe int yang digunakan untuk menyimpan jumlah kerucut yang akan dihitung.
- input2: Variabel tipe int yang menyimpan jari-jari alas kerucut (radius).
- input3: Variabel tipe int yang menyimpan tinggi kerucut (tinggi).
- result: Variabel tipe float64 yang digunakan untuk menyimpan hasil perhitungan volume kerucut.

## 2. Pengambilan Input

- `fmt.Scan(&input1)`: Program pertama-tama meminta pengguna untuk memasukkan jumlah kerucut (`input1`) yang akan dihitung.

## 3. Perulangan for

- `for x := 0; x < input1; x++`: Perulangan berjalan sebanyak `input1` kali. Setiap kali iterasi, program akan meminta input baru untuk jari-jari (`input2`) dan tinggi (`input3`) dari kerucut yang berbeda.
- `fmt.Scan(&input2, &input3)`: Program meminta pengguna untuk memasukkan dua nilai, yaitu jari-jari alas (`input2`) dan tinggi kerucut (`input3`).

## 4. Perhitungan Volume Kerucut

- Rumus volume kerucut adalah:  $V = \frac{1}{3} \times \pi \times r^2 \times t$  Dalam kode, rumus ini diimplementasikan sebagai:
- `math.Pi`: Konstanta  $\pi$  (Pi) dari paket `math`.
- `math.Pow(float64(input2), 2)`: Menghitung kuadrat dari jari-jari (`input2`).
- Hasil perhitungan dikonversi ke `float64` agar operasi dapat dilakukan dengan presisi desimal

### 3. Latihan Soal3

Buatlah program yang digunakan untuk menghitung hasil pemangkatan dari dua buah bilangan. Program dibuat dengan menggunakan operator perkalian dan struktur kontrol perulangan.

Masukan terdiri dari dua bilangan bulat positif.

Keluaran terdiri dari suatu bilangan yang menyatakan hasil bilangan pertama dipangkatkan dengan bilangan kedua.

Input:

```
package main

import (
    "fmt"
)

func main() {
    var input1, input2 int
    var result int

    fmt.Scan(&input1, &input2)

    for x := 0; x < input2; x++ {
        if x == 0 {
            result = input1
        } else {
            result = result * input1
        }
    }
    fmt.Println(result)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal3.go\tempCodeRunnerFile.go"
4 2
16
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal3.go\tempCodeRunnerFile.go"
2 10
1024
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal3.go\tempCodeRunnerFile.go"
10 3
1000
```

Deskripsi Program

## 2. Deklarasi Variabel

- input1: Variabel tipe int yang menyimpan bilangan dasar (basis) untuk perhitungan perpangkatan.
- input2: Variabel tipe int yang menyimpan pangkat (eksponen) dari perhitungan perpangkatan.

- `result`: Variabel tipe `int` yang digunakan untuk menyimpan hasil dari perkalian berulang.

### 3. Pengambilan Input

- `fmt.Scan(&input1, &input2)`: Program meminta pengguna memasukkan dua nilai: `input1` sebagai bilangan dasar dan `input2` sebagai eksponen (jumlah kali pengulangan perkalian).

### 3. Perulangan `for`

- `for x := 0; x < input2; x++`: Perulangan berjalan sebanyak `input2` kali, yang sesuai dengan nilai eksponen.
- **Kondisi dalam perulangan:**
  - Pada iterasi pertama (`x == 0`), `result` diinisialisasi dengan nilai `input1`. Ini berarti pada iterasi pertama, `result` hanya berisi nilai dasar.
  - Pada iterasi berikutnya (`x > 0`), `result` dikalikan dengan `input1` pada setiap iterasi, sehingga mensimulasikan operasi perpangkatan.

#### 4. Latihan Soal4

Buatlah program yang digunakan untuk menghitung hasil faktorial dari suatu bilangan. Masukan terdiri dari suatu bilangan bulat non negatif. Keluaran terdiri dari hasil faktorial dari bilangan bulat n.

Input:

```
package main

import "fmt"

func main() {
    var input1 int
    var result int = 1

    fmt.Scan(&input1)

    for x := 1; x <= input1; x++ {
        result = result * x
    }
    fmt.Println(result)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal4.go\tempCodeRunnerFile.go"
0
1
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal4.go\tempCodeRunnerFile.go"
1
1
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal4.go\tempCodeRunnerFile.go"
5
120
PS D:\Coding manja> go run "d:\Coding manja\modul5\soal4.go\tempCodeRunnerFile.go"
10
3628800
```

Deskripsi Program:

##### 1. Deklarasi Variabel

- input1: Variabel bertipe int yang digunakan untuk menyimpan input dari pengguna, yaitu bilangan bulat positif yang ingin dihitung faktorialnya.
- result: Variabel bertipe int yang digunakan untuk menyimpan hasil perhitungan faktorial. Variabel ini diinisialisasi dengan nilai 1 karena faktorial adalah hasil perkalian, dan memulai perkalian dengan 1 tidak akan memengaruhi hasil.

##### 2. Pengambilan Input

- fmt.Scan(&input1): Program meminta pengguna untuk memasukkan sebuah bilangan bulat (input1), yang akan dihitung faktorialnya.

##### 3. Perulangan for

- for x := 1; x <= input1; x++: Perulangan berjalan mulai dari x = 1 hingga x sama dengan nilai input1. Pada setiap iterasi, result dikalikan dengan nilai x, sehingga result mengakumulasi hasil faktorial dari bilangan tersebut.

- `result = result * x`: Pada setiap iterasi, nilai `x` dikalikan dengan `result`, sehingga variabel `result` menyimpan hasil faktorial yang diperbarui setiap kali loop berjalan.

## **Daftar Pustaka**

<https://blog.ruangdeveloper.com/golang-for-loop/>