

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



DISUSUN OLEH:
RAIHAN ADI ARBA

103112400071

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 merupakan salah satu konstruksi pemrograman yang paling fundamental dan sering digunakan dalam Golang, serta dalam banyak bahasa pemrograman lainnya. Sebagai mekanisme pengulangan, for-loop memungkinkan pengembang untuk mengeksekusi blok kode berulang kali, yang sangat berguna dalam pengolahan data, pengulangan tugas, dan penerapan algoritma. Dalam Golang, for-loop memiliki beberapa bentuk sintaks yang berbeda, masing-masing dirancang untuk menyederhanakan berbagai kasus penggunaan pemrograman.

Bentuk yang paling sederhana dari for-loop hanya melibatkan satu kondisi, mirip dengan while-loop dalam bahasa pemrograman lain. Ini memungkinkan loop untuk terus berjalan selama kondisi tetap benar. Sebagai contoh, `for i < 10 {}` akan terus melakukan iterasi selama nilai `i` kurang dari 10. Bentuk lain yang umum melibatkan inisialisasi variabel, kondisi perulangan, dan increment atau decrement, yang terintegrasi dalam satu baris. Sintaks ini sangat efisien untuk melakukan iterasi melalui array, slice, atau jenis kumpulan data lainnya. Contohnya, `for i := 0; i < len(array); i++ {}` akan menjalankan loop sebanyak jumlah elemen dalam array tersebut.

For-loop juga mendukung iterasi langsung melalui elemen-elemen dari slice, map, atau channel menggunakan sintaks `range`. Penggunaan `range` tidak hanya mempermudah akses ke setiap elemen dalam koleksi, tetapi juga mengurangi risiko kesalahan yang umum terjadi dalam pengelolaan indeks loop. Misalnya, dengan `for index, value := range slice {}`, pengembang dapat memperoleh indeks dan nilai setiap elemen dalam slice tanpa perlu mengelola indeks secara manual.

Penggunaan efektif for-loop dalam Golang mengharuskan pemahaman mendalam tentang struktur data yang digunakan, serta skenario yang membutuhkan iterasi. Ini termasuk operasi yang melibatkan pengolahan batch data, algoritma pencarian dan pengurutan, serta tugas-tugas yang memerlukan pengulangan aksi sampai kondisi tertentu terpenuhi. Dengan kemampuan untuk menyederhanakan proses berulang sambil menyediakan kontrol yang ketat atas logika iterasi, for-loop berperan penting dalam mengembangkan software yang efisien dan mudah dipelihara. Memanfaatkan loop dengan benar dapat meningkatkan performa program secara signifikan dan memungkinkan penanganan kompleksitas data yang lebih tinggi dengan kode yang lebih bersih dan lebih mudah dimengerti.

A. GUIDED

1. Buatlah program untuk menampilkan baris bilangan dari a sampai dengan b .

Masukan terdiri dari dua bilangan bulat a dan b , yang mana $a \leq b$.

Keluaran berupa baris bilangan dari a sampai dengan b .

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	2 5	2 3 4 5
2	6 6	6
3	-5 7	-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7

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, " ")
    }
}
```

Output :

```
● raihan@Raihans-MacBook-Pro contoh soal % go run c2.go
2 9
2 3 4 5 6 7 8 9 %
```

2. Buatlah program yang digunakan untuk menampilkan sejumlah n luas segitiga, apabila diketahui sisi alas dan tinggi dari masing-masing segitiga.

Masukan terdiri dari $n + 1$ baris. Baris pertama merupakan suatu bilangan bulat n , selanjutnya n baris berikutnya masing-masing merupakan panjang sisi alas dan tinggi dari segitiga

Keluaran terdiri dari n baris, yang masing-masing menyatakan luas dari segitiga.

No	Masukan	Keluaran
1	5	11
	11 2	224
	32 14	6
	6 2	112.5
	15 15	350
	20 35	

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)
    }
}
```

Output :

```
● raihan@Raihans-MacBook-Pro contoh soal % go run c3.go
4
1 4
2
2 3
3
3 9
13.5
10 10
50
```

- Sebuah program digunakan untuk menghitung BMI atau Body Mass Indeks, yang mana BMI merupakan hasil bagi dari berat badan dengan kuadrat dari tinggi badan.

Buatlah program yang digunakan untuk menghitung hasil perkalian dua buah bilangan tanpa menggunakan operator kali "*".

Masukan terdiri dari dua bilangan bulat positif.

Keluaran terdiri dari sebuah bilangan yang menyatakan hasil perkalian dari dua bilangan pada masukan.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	2 100	200
2	7 6	42

Source code:

```
package main

import "fmt"

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

Output :

```
● raihan@Raihans-MacBook-Pro contoh soal % go run c3.go
3 9
1533
```

B. UNGUIDED

1. Latihan 1

Buatlah program untuk menjumlahkan sekumpulan bilangan.

Masukan terdiri dari suatu bilangan bulat positif n.

Keluaran berupa bilangan hasil penjumlahan dari 1 sampai dengan n.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	3	6
2	1	1
3	5	15

Source Code:

```
package main

import "fmt"

func main() {
    var X, hasil int
    fmt.Scan(&X)

    for i := 1; i <= X; i++ {
        hasil += i
    }

    fmt.Printf("%d\n", hasil)
}
```

Output :

```
● raihan@Raihans-MacBook-Pro latihan soal % go run p1.go
1
1
● raihan@Raihans-MacBook-Pro latihan soal % go run p1.go
15
120
● raihan@Raihans-MacBook-Pro latihan soal % go run p1.go
6
21
```

Penjelasan Program:

Program ini meminta pengguna untuk memasukkan sebuah nilai integer X. Setelah menerima input ini, program kemudian menghitung total dari semua bilangan bulat mulai dari 1 hingga X menggunakan loop iteratif. Dalam loop tersebut, setiap

bilangan integer secara berurutan ditambahkan ke dalam variabel hasil yang pada awalnya diinisialisasi dengan nilai nol.

2. Latihan 2

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.

Contoh masukan dan keluaran

No	Masukan	Keluaran
1	1 3 4	37.69911184307752
2	3 1 1 2 2 3 3	1.0471975511965976 8.377580409572781 28.274333882308138

Source Code:

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n int
    var vol, j, t float64
    fmt.Scan(&n)

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

Output:

```

● raihan@Raihans-MacBook-Pro latihan soal % go run p2.go
3
1 1
1.0471975511965976
2 2
8.377580409572781
3 3
28.274333882308134

```

Deskripsi Program:

Program meminta pengguna untuk memasukkan sebuah nilai integer, yang menentukan berapa banyak pasangan nilai jari-jari dan tinggi yang akan diproses. Setelah nilai ini diterima, program memasuki sebuah loop, di mana setiap iterasi meminta pengguna memasukkan sepasang nilai jari-jari dan tinggi. program menghitung volume sebuah kerucut dengan menggunakan formula volume $V = \frac{1}{3} \times \pi \times r \times r \times t$, di mana r adalah jari-jari dan t adalah tinggi kerucut. Hasil perhitungan ini kemudian ditampilkan langsung ke pengguna.

3. Latihan 3

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

Masukan terdiri dari dua bilangan bulat positif.

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

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	4 2	16
2	2 10	1024
3	10 3	1000

Source Code:

```

package main

import "fmt"

func main() {
    var x, y int
    var hasil = 1
    fmt.Scan(&x, &y)
    for i := 0; i < y; i++ {
        hasil *= x
    }
    fmt.Println(hasil)
}

```


Output:

```
● raihan@Raihans-MacBook-Pro latihan soal % go run p3.go
4 2
16
● raihan@Raihans-MacBook-Pro latihan soal % go run p3.go
10 3
1000
```

Deskripsi Program:

Program di atas dirancang untuk mengeksekusi operasi perpangkatan dengan memanfaatkan input dari pengguna. Setelah dijalankan, program ini meminta pengguna untuk memasukkan dua nilai integer, X dan Y. Nilai X akan dijadikan sebagai basis, sedangkan Y merupakan eksponen. Program ini menggunakan sebuah loop for untuk mengulang operasi perkalian sebanyak Y kali. Dalam setiap iterasi, nilai X dikalikan dengan variabel hasil, yang semula diinisialisasi dengan nilai 1. Proses ini berlanjut hingga loop selesai.

4. Latihan 4

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.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	0	1
2	1	1
3	5	120
4	10	3628800

Output :

```
● raihan@Raihans-MacBook-Pro latihan soal % go run p4.go
1
1
● raihan@Raihans-MacBook-Pro latihan soal % go run p4.go
5
120
● raihan@Raihans-MacBook-Pro latihan soal % go run p4.go
10
3628800
```

Source Code:

```
package main
```

```
import "fmt"

func main() {
    var n int
    var faktorial = 1
    fmt.Scan(&n)

    for i := 1; i <= n; i++ {
        faktorial *= i
    }
    fmt.Println(faktorial)
}
```

Deskripsi Program:

Program ini dirancang untuk menerima input berupa bilangan bulat positif dari pengguna. Setelah input tersebut diterima, program kemudian menghitung faktorial dari bilangan tersebut. Faktorial, yang dilambangkan dengan $n!$, adalah hasil perkalian semua bilangan bulat positif hingga n . Untuk mencapai perhitungan ini, program menggunakan sebuah loop iteratif yang dimulai dari 1 hingga n . Dalam setiap iterasi loop, nilai faktorial dikalikan secara berurutan dengan indeks loop saat ini, dimulai dari 1. Nilai awal dari variabel faktorial diinisialisasi dengan 1, karena perkalian apa pun dengan 0 akan menghasilkan 0. Setelah loop selesai, program kemudian menampilkan hasil faktorial ke pengguna.

DAFTAR PUSTAKA

Prayogo, N. A. (2021). *Dasar Pemrograman Go. Ebook*