

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 1
“I/O TIPE DATA & VARIABEL”



DISUSUN OLEH:
ANGGUN WAHYU WIDIYANA
103112480280
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

Paradigma Perulangan

Perulangan merupakan salah satu struktur kontrol yang memungkinkan suatu instruksi yang sama dilakukan berulang kali dalam waktu atau jumlah yang lama. Tanpa instruksi perulangan, maka suatu instruksi akan ditulis dalam jumlah yang sangat banyak.

Sebagai contoh adalah menuliskan suatu teks "CAK1BAB3 Algoritma Pemrograman 1" sebanyak 1000 baris. Maka tidak mungkin kita menuliskan instruksi `fmt.Println("CAK1BAB3 Algoritma Pemrograman 1")` sebanyak 1000x (walaupun memungkinkan menggunakan copy-paste).

Berikut ilustrasinya:

Tanpa Perulangan	Dengan Perulangan
<pre>fmt.Println("CAK1BAB3 Algoritma Pemrograman 1") fmt.Println("CAK1BAB3 Algoritma Pemrograman 1") fmt.Println("CAK1BAB3 Algoritma Pemrograman 1") dst.</pre>	<pre>// instruksi perulangan 1000x fmt.Println("CAK1BAB3 Algoritma Pemrograman 1") // batas instruksi perulangan 1000x</pre>

Terlihat pada ilustrasi di atas, penulisan dengan instruksi perulangan jauh lebih singkat dan jelas dibandingkan dengan tanpa perulangan.

Penting! Pastikan bahwa instruksi perulangan yang digunakan pasti bisa membuat proses perulangan berhenti, apabila tidak maka program akan terus berjalan mengulangi instruksi tanpa akan pernah berhenti.

Karakteristik For-Loop (Perulangan berdasarkan iterasi)

Salah satu instruksi perulangan yang paling mudah adalah for-loop, yang mana dengan instruksi ini dapat digunakan untuk mengulangi instruksi sebanyak n kali (iterasi). Batasan besar nilai dari n menyesuaikan dengan batasan dari tipe data integer yang digunakan.

Tanpa Perulangan	Dengan Perulangan
<pre>for inisialisasi to n do // instruksi yang akan diulangi endfor</pre>	<pre>for inisialisasi; kondisi; update { // instruksi yang akan diulangi }</pre>

Keterangan:

- Gunakan tanda titik koma ";" (semicolon) untuk memisahkan inisialisasi, kondisi dan update pada instruksi for-loop.
- Berikan tab atau 4 spasi untuk instruksi yang akan diulangi. Tujuannya untuk mempermudah pembacaan kode dan mengetahui bagian kode mana yang akan diulangi.

Instruksi for-loop memiliki beberapa komponen, yaitu:

- 1) **inisialisasi** merupakan assignment variabel iterasi yang bertipe integer. Pada contoh diatas biasanya variabel iterasi = 0 atau 1, artinya iterasi dimulai dari 0 atau 1.

- 2) **kondisi** merupakan suatu operasi bernilai boolean yang menyatakan kapan perulangan harus dilakukan. Pada contoh di atas kondisi adalah variabel iterasi $\leq n$ (kurang dari atau sama dengan)
- 3) **update** merupakan ekspresi yang menyatakan perubahan nilai dari variabel iterasi.

Bentuk dasar `for` loop adalah:

```
```go
for i := 0; i < 5; i++ {
 (Kode yang dijalankan berulang-ulang)
}
```
```

Pada contoh di atas, loop akan dijalankan sebanyak 5 kali, dengan nilai `i` dimulai dari 0 dan diinkremen sebanyak 1 setiap iterasi.

```
```go
fruits := []string{"apel", "jeruk", "mangga"}
for i, fruit := range fruits {
 fmt.Printf("Index: %d, Fruit: %s\n", i, fruit)
}
```
```

Pada contoh di atas, loop akan berjalan sebanyak jumlah elemen dalam slice `fruits`. Setiap iterasi, nilai indeks (`i`) dan nilai elemen (`fruit`) akan dicetak.

CONTOH SOAL

Latihan1

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 = j+1 {
        fmt.Print(j, " ")
    }
}
```

Output:

```
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Cosomod5\praktik1.go"
2 5
2 3 4 5
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Cosomod5\praktik1.go"
6 6
6
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Cosomod5\praktik1.go"
-5 7
-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
PS C:\Users\User\Downloads\MODUL 4>
```

Deskripsi Program:

Program Go ini bertujuan untuk menampilkan barisan bilangan dari a hingga b secara berurutan.

Latihan2

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.

Contoh masukan dan keluaran:

| No | Masukan | Keluaran |
|----|---|--------------------------------|
| 1. | 5
11 2
32 14
6 2
15 15
20 35 | 11
224
6
112.5
350 |
| 2. | 3
12 32
231 234
43 34 | 192
27027
731 |

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:



```
5
11 2
32 14
224
6 2
6
112.5
20 35
350
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Cosomod5\praktik2.go"
3
12 32
192
231 234
27027
43 34
731
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Cosomod5\praktik2.go"
```

Deskripsi Program:

Program Go ini bertujuan untuk menghitung dan menampilkan luas beberapa segitiga berdasarkan jumlah segitiga n , dan panjang alas serta tinggi dari masing-masing segitiga yang diberikan.

Latihan3

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.

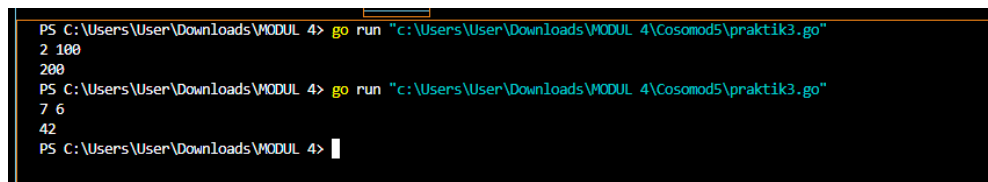
Contoh masukan dan keluaran:

| No | Masukan | Keluaran |
|----|---------|----------|
| 1. | 22 100 | 200 |
| 2. | 7 6 | 42 |

Source Code:

```
package main
import "fmt"
func main(){
    var hasil, v1, v2 int
    fmt.Scan(&v1, &v2)
    hasil=0
    for j := 1; j <= v2; j+=1 {
        hasil = hasil + v1
    }
    fmt.Println(hasil)
}
```

Output:



```
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Cosomod5\praktik3.go"
2 100
200
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Cosomod5\praktik3.go"
7 6
42
PS C:\Users\User\Downloads\MODUL 4> |
```

Deskripsi Program:

Program Go ini bertujuan untuk menghitung hasil perkalian dua bilangan bulat positif tanpa menggunakan operator kali (*). Program ini menggunakan penjumlahan berulang untuk melakukan perkalian.

SOAL LATIHAN (Statement Perulangan)

Latihan1

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:

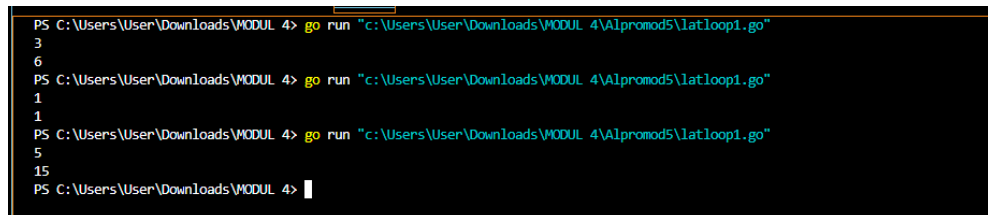
| No | Masukan | Keluaran |
|----|---------|----------|
| 1. | 3 | 6 |
| 2. | 1 | 1 |
| 3. | 5 | 15 |

Source Code:

```
package main
import "fmt"
func main(){
    var n, jumlah int
    fmt.Scan(&n)

    for i := 1; i <= n; i++ {
        jumlah += i
    }
    fmt.Printf("%d\n", jumlah)
}
```

Output:



```
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\A1promod5\latloop1.go"
3
6
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\A1promod5\latloop1.go"
1
1
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\A1promod5\latloop1.go"
5
15
PS C:\Users\User\Downloads\MODUL 4> |
```

Deskripsi Program:

Program Go ini bertujuan untuk menghitung jumlah total dari 1 hingga n untuk suatu bilangan bulat positif n yang diberikan.

Latihan2

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. program untuk menjumlahkan sekumpulan bilangan.

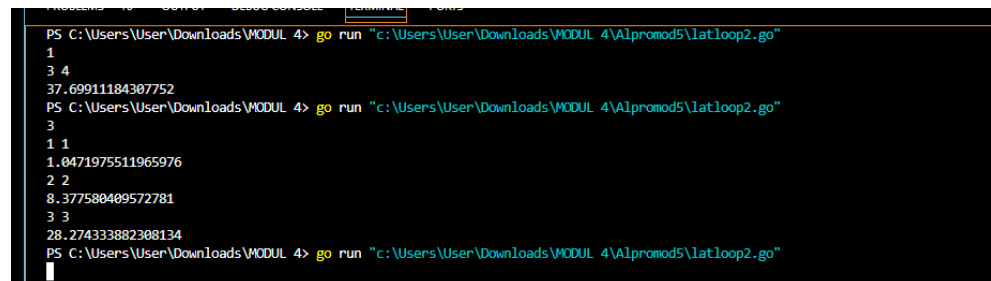
Contoh masukan dan keluaran:

| 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"
func main(){
    var n int
    var volume, jarijari, tinggikerucut float64
    fmt.Scan(&n)
    for i := 1; i <= n; i++ {
        fmt.Scan(&jarijari, &tinggikerucut)
        volume = (1.0/3.0) * (math.Pi * jarijari* jarijari *
tinggikerucut)
        fmt.Println(volume)
    }
}
```

Output:



```
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Alpromod5\latloop2.go"
1
3 4
37.69911184307752
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Alpromod5\latloop2.go"
3
1 1
1.0471975511965976
2 2
8.377580409572781
3 3
28.274333882308134
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Alpromod5\latloop2.go"
```

Deskripsi Program:

Program Go ini bertujuan untuk menghitung dan menampilkan volume beberapa kerucut berdasarkan nilai jari-jari alas dan tinggi kerucut yang diberikan untuk setiap kerucut.

Latihan3

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.

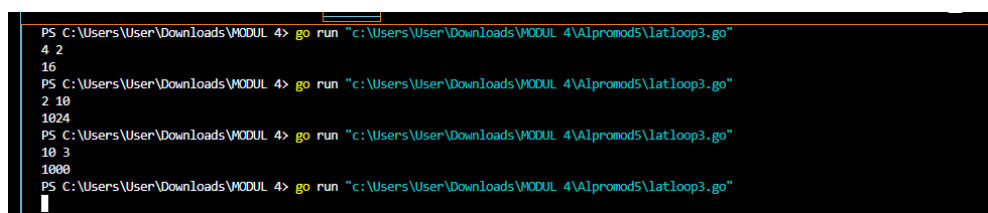
Contoh masukan dan keluaran:

| No | Masukan | Keluaran |
|----|---------|----------|
| 1. | 4 2 | 16 |
| 2. | 2 10 | 1024 |
| 3. | 10 3 | 1000 |

Source Code:

```
package main
import "fmt"
func main(){
    var basis, eksponen, hasil int
    fmt.Scan(&basis, &eksponen)
    hasil = 1
    for i := 1; i <= eksponen; i++ {
        hasil *= basis
    }
}
```

Output:



```
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\A1promod5\latloop3.go"
4 2
16
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\A1promod5\latloop3.go"
2 10
1024
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\A1promod5\latloop3.go"
10 3
1000
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\A1promod5\latloop3.go"
```

Deskripsi Program:

Program Go ini bertujuan untuk menghitung basis pangkat eksponen dengan mengalikan basis berulang kali, tanpa menggunakan fungsi bawaan pemangkatan.

Latihan4

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:

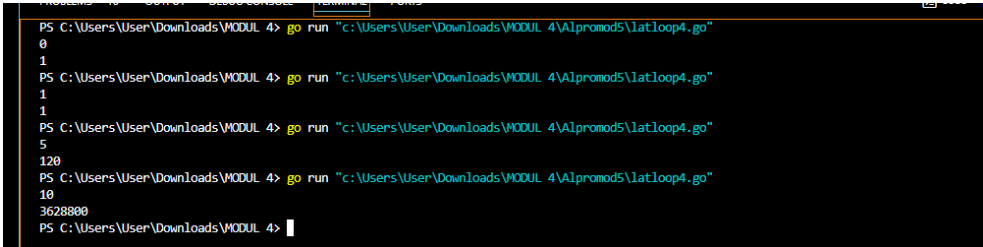
| No | Masukan | Keluaran |
|----|---------|----------|
| 1. | 0 | 16 |
| 2. | 1 | 1024 |
| 3. | 5 | 1000 |
| 4. | 10 | 3628800 |

Source Code:

```
package main
import "fmt"
func main(){
    var n, hasilFaktorial int
    fmt.Scan(&n)
    hasilFaktorial = 1

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

Output:



```
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Alpromod5\latloop4.go"
0
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Alpromod5\latloop4.go"
1
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Alpromod5\latloop4.go"
5
PS C:\Users\User\Downloads\MODUL 4> go run "c:\Users\User\Downloads\MODUL 4\Alpromod5\latloop4.go"
10
3628800
PS C:\Users\User\Downloads\MODUL 4>
```

Deskripsi Program:

Program Go ini bertujuan untuk menghitung faktorial dari suatu bilangan bulat non-negatif n. Program menerima input bilangan n, kemudian menghitung faktorial dengan mengalikan semua bilangan dari 1 hingga n menggunakan perulangan.

DAFTAR PUSTAKA

- 1) Modul Praktikum 5 & 6 - For Loop Algoritma dan Pemrograman 1 S1 Informatika. Published by School of Computing.
 - 2) Budi Raharjo. (2021). *Mudah Belajar Go: Pemrograman Golang untuk Pemula*. Bandung: Informatika.
-