

MODUL PRAKTIKUM 5 & 6 - FOR-LOOP

ALGORITMA DAN PEMROGRAMAN 1

S1 INFORMATIKA

GO

Published by school of computing

Our official instagram



@informaticslab_telu

LEMBAR PENGESAHAN

Saya yang bertanda tangan di bawah ini:

Nama : Prasti Eko Yunanto, S.T., M.Kom.
NIP : 19890017
Koordinator Mata Kuliah : Algoritma dan Pemrograman 1
Prodi : S1 Informatika

Menerangkan dengan sesungguhnya bahwa modul ini digunakan untuk pelaksanaan praktikum di Semester Ganjil Tahun Ajaran 2024/2025 di Laboratorium Informatika, Fakultas Informatika, Universitas Telkom.



Bandung, 17 Agustus 2024



Mengesahkan,

Mengetahui,

Koordinator Mata Kuliah
Algoritma Pemrograman 1

Kaprodi S1 Informatika

A blue ink signature of Prasti Eko Yunanto, S.T., M.Kom.

Prasti Eko Yunanto, S.T., M.Kom.
NIP. 19890017

A blue ink signature of Dr. Erwin Budi Setiawan, S.Si., M.T.

Dr. Erwin Budi Setiawan, S.Si., M.T.
NIP. 00760045

MODUL 5 & 6. FOR-LOOP

5.1 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.

5.2 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.

Notasi dalam pseudocode	Notasi dalam bahasa Go
<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 di atas 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 <= n** (kurang dari atau sama dengan)
- 3) **update** merupakan ekspresi yang menyatakan perubahan nilai dari **variabel iterasi**. Pada contoh di atas update adalah **variabel iterasi = variabel iterasi + 1**.

5.3 Implementasi menggunakan Go

Sebagai contoh penulisan for-loop dengan menggunakan bahasa Go adalah sebagai berikut, yang mana kita akan gunakan contoh menampilkan teks "CAK1BAB3 Algoritma Pemrograman 1" sebanyak 1000x.

```
1 // filename: Alpro1.go
2 package main
3 import "fmt"
4
5 func main() {
6     var iterasi, n int
7     n = 1000
8
9     for iterasi = 1; iterasi <= n; iterasi = iterasi + 1 {
10         fmt.Println(iterasi, "CAK1BAB3 Algoritma Pemrograman 1")
11     }
12 }
```

```
C:\users\go\src\hello>go build Alpro1.go
```

```
C:\users\go\src\hello>.\Alpro1
```

```
1 CAK1BAB3 Algoritma Pemrograman 1
2 CAK1BAB3 Algoritma Pemrograman 1
3 CAK1BAB3 Algoritma Pemrograman 1
4 CAK1BAB3 Algoritma Pemrograman 1
5 CAK1BAB3 Algoritma Pemrograman 1
6 CAK1BAB3 Algoritma Pemrograman 1
7 CAK1BAB3 Algoritma Pemrograman 1
8 CAK1BAB3 Algoritma Pemrograman 1
...
...
1000 CAK1BAB3 Algoritma Pemrograman 1
C:\users\go\src\hello>
```

5.4 Contoh Soal Modul 5

- 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

Jawaban

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

```
E:\DEV\NATIVE\GO>go build Demo_Soal.go
E:\DEV\NATIVE\GO>.\Demo_Soal
2 5
2 3 4 5
E:\DEV\NATIVE\GO>.\Demo_Soal
6 6
6
E:\DEV\NATIVE\GO>.\Demo_Soal
-5 7
-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
```

- 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.

Contoh masukan dan keluaran:

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

Jawaban

```
1 package main
2 import "fmt"
3
4 func main() {
5     var j, alas, tinggi, n int
6     var luas float64
7     fmt.Scan(&n)
8     for j = 1; j <=n; j+=1 {
9         fmt.Scan(&alas, &tinggi)
10        luas = 0.5 * float64(alas * tinggi)
11        fmt.Println(luas)
12    }
13 }
14
```

E:\DEV\NATIVE\GO>go build Demo_Soal.go

E:\DEV\NATIVE\GO>.\Demo_Soal

5

11 2

11

32 14

224

6 2

6

15 15

112.5

20 35

350

E:\DEV\NATIVE\GO>.\Demo_Soal

3

12 32

192

231 234

27027

43 34

731

- 3) 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

Jawaban

```
1 package main
2 import "fmt"
3
4 func main(){
5     var j, v1, v2 int
6     var hasil int
7     fmt.Scan(&v1, &v2)
8     hasil = 0
9     for j = 1; j <= v2; j+=1 {
10         hasil = hasil + v1
11     }
12     fmt.Println(hasil)
13 }
```

```
E:\DEV\NATIVE\GO>go build Demo_Soal.go
E:\DEV\NATIVE\GO>.\Demo_Soal
2 10
20
E:\DEV\NATIVE\GO>.\Demo_Soal
7 6
42
```