

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 10**  
**“PERULANGAN”**



**DISUSUN OLEH:**  
**DWI OKTA SURYANINGRUM**  
**103112400066**  
**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

### PERULANGAN

Perulangan adalah proses mengulang-ulang eksekusi blok kode tanpa henti, selama kondisi yang dijadikan acuan terpenuhi. Biasanya disiapkan variabel untuk iterasi atau variabel penanda kapan perulangan akan diberhentikan. Di Go keyword perulangan hanya **for** saja, tetapi meski demikian, kemampuannya merupakan gabungan **for**, **foreach**, dan **while** ibarat bahasa pemrograman lain.

#### 1. Perulangan Menggunakan Keyword **for**

Ada beberapa cara standar menggunakan **for**. Cara pertama dengan memasukkan variabel counter perulangan beserta kondisinya setelah keyword.

#### 2. Penggunaan Keyword **for** Dengan Argumen Hanya Kondisi

Cara ke-2 adalah dengan menuliskan kondisi setelah keyword **for** (hanya kondisi). Deklarasi dan iterasi variabel counter tidak dituliskan setelah keyword, hanya kondisi perulangan saja. Konsepnya mirip seperti **while** milik bahasa pemrograman lain.

#### 3. Penggunaan Keyword **for** Tanpa Argumen

Cara ke-3 adalah **for** ditulis tanpa kondisi. Dengan ini akan dihasilkan perulangan tanpa henti (sama dengan **for true**). Pemberhentian perulangan dilakukan dengan menggunakan keyword **break**.

#### 4. Penggunaan Keyword **for** – **range**

Cara ke-4 adalah perulangan dengan menggunakan kombinasi keyword **for** dan **range**. Cara ini biasa digunakan untuk me-looping data gabungan (misalnya string, array, slice, map)

#### 5. Penggunaan Keyword **break** & **continue**

Keyword **break** digunakan untuk menghentikan secara paksa sebuah perulangan, sedangkan **continue** dipakai untuk memaksa maju ke perulangan berikutnya.

#### 6. Perulangan Bersarang

Tak hanya seleksi kondisi yang bisa bersarang, perulangan juga bisa. Cara pengaplikasiannya kurang lebih sama, tinggal tulis blok statement perulangan di dalam perulangan. Di perulangan bersarang, **break** dan **continue** akan berlaku pada blok perulangan di mana ia digunakan saja. Ada cara agar kedua keyword ini bisa

tertuju pada perulangan terluar atau perulangan tertentu, yaitu dengan memanfaatkan teknik pemberian **label**.

## CONTOH SOAL

### 1. Latihan1

Buatlah program yang digunakan untuk menampilkan deret bilangan Faktorial dari suatu bilangan.

**Masukan** terdiri dari sebuah bilangan bulat non negatif n.

**Keluaran** berupa deret bilangan dari Faktorial n. Perhatikan contoh masukan dan keluaran yang diberikan.

**Contoh masukan dan keluaran:**

| No | Masukan | Keluaran                               |
|----|---------|--|
| 1. | 0       | 1                                      |
| 2. | 5       | 5 x 4 x 3 x 2 x 1                      |
| 3. | 10      | 10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 |
| 4. | 1       | 1                                      |

Source Code:

```
package main

import "fmt"

func main() {
    var n, j int
    fmt.Scan(&n)
    j = n
    for j>1 {
        fmt.Print(j, "x")
        j = j-1
    }
    fmt.Println(1)
}
```

Output:

```
go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
/Users/mymac/.zprofile:5: unmatched "
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
0
1
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
5
5x4x3x2x1
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
10
10x9x8x7x6x5x4x3x2x1
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
1
1
mymac@192 praktikum1 %
```

#### Deskripsi Program:

- var n dan j mendeklarasikan variabel dengan tipe data integer.
- `fmt.Scan(&n)` digunakan untuk menginputkan nilai var n.
- mendeklarasikan variabel j sama dengan n, artinya var j memiliki nilai yang sama seperti var n
- Menggunakan **for** loop, program menjalankan perintah selama kondisi `j > 1` terpenuhi:
- Mencetak nilai j diikuti tanda x tanpa pindah baris.
- Mengurangi nilai j sebanyak 1 di setiap iterasi (`j = j - 1`).
- Ketika kondisi `j > 1` tidak lagi terpenuhi (artinya `j == 1`), program keluar dari loop.

## 2. Latihan 2

Buatlah program Go yang digunakan untuk login ke dalam suatu aplikasi. Asumsi token untuk yang valid adalah "12345abcde".

**Masukan** terdiri dari suatu token. Selama token yang diberikan salah, maka program akan meminta token secara terus menerus hingga token yang diberikan benar.

**Keluaran** adalah teks yang menyatakan "Selamat Anda berhasil login".

**Contoh masukan dan keluaran:**

| No | Masukan   | Keluaran                    |
|----|---|-----------------------------|
| 1. | Qwe12312<br>231234<br>13213<br>123lijwe<br>12345abcde | Selamat Anda berhasil login |
| 2. | 12345abcde  | Selamat Anda berhasil login |

Source Code :

```
package main

import "fmt"

func main() {
    var token string
```

```

fmt.Scan(&token)
for token != "12345abcde" {
    fmt.Scan(&token)
}
fmt.Println("Selamat abda berhasil login!")
}

```

Output:

```

mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
Qwe12312
231234
13213
123lijwe
12345abcde
Selamat abda berhasil login!
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
12345abcde
Selamat abda berhasil login!
mymac@192 praktikum1 %

```

Deskripsi Program:

- var token mendeklarasikan variabel dengan tipe data string
- fmt.Scan(&token) digunakan untuk menginputkan isi dari variabel token.
- Melakukan perulangan dengan menggunakan **for loop** dengan kondisi token != "12345abcde":
- Jika token yang dimasukkan tidak sama dengan "12345abcde", program akan meminta input ulang.
- Perulangan ini akan terus berjalan hingga pengguna memasukkan token yang benar.
- Ketika token yang dimasukkan adalah "**12345abcde**", kondisi token != "12345abcde" menjadi **false**, dan perulangan berhenti.

### 3. Latihan 3

Buatlah program dalam bahasa Go yang digunakan untuk mencetak N bilangan pertama dalam deret Fibonacci.

**Masukan** terdiri dari bilangan bulat positif N dengan nilai besar atau sama dengan 2.

**Keluaran** terdiri dari sejumlah N bilangan yang menyatakan N deret bilangan Fibonacci yang pertama.

**Contoh masukan dan keluaran:**

| No | Masukan | Keluaran               |
|----|---------|------------------------|
| 1. | 5       | 0 1 1 2 3              |
| 2. | 2       | 0 1                    |
| 3. | 10      | 0 1 1 2 3 5 8 13 21 34 |

#### Source Code

```
package main

import "fmt"

func main() {
    var N, s1, s2, j, temp int
    fmt.Scan(&N)
    s1 = 0
    s2 = 1
    j = 0
    for j < N {
        fmt.Print(s1, " ")
        temp = s1 + s2
        s1 = s2
        s2 = temp
        j = j + 1
    }
}
```

#### Output:

```
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
5
0 1 1 2 3
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
2
0 1
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Coso/tempCodeRunnerFile.go"
10
0 1 1 2 3 5 8 13 21 34
mymac@192 praktikum1 %
```

#### Deskripsi Program:

- Var N, s1, s2, j, temp mendeklarasikan variabel dengan tipe data int
- Fmt.Scan(&N) digunakan untuk menginputkan nilai N
- **s1 dan s2**: Menyimpan dua bilangan terakhir dalam deret Fibonacci (diinisialisasi dengan 0 dan 1).

- **j**: Sebagai penghitung iterasi.
- Perulangan akan berjalan selama  $j < N$  (sebanyak  $N$  kali).
- Di setiap iterasi:
- Mencetak nilai  $s1$  (bilangan Fibonacci saat ini).
- Menghitung bilangan Fibonacci berikutnya:
  - **temp = s1 + s2** (bilangan berikutnya adalah jumlah dua bilangan sebelumnya).
  - Memperbarui nilai:
    - $s1 = s2$  (bilangan saat ini digeser ke  $s1$ ).
    - $s2 = \text{temp}$  (hasil penjumlahan digeser ke  $s2$ ).
- Menambah penghitung iterasi  $j = j + 1$ .



## SOAL LATIHAN

### Statement Percabangan

#### 1. Latihan Soal 1

Buatlah sebuah program dalam bahasa Go yang digunakan untuk menghitung berapa banyak seseorang pengguna gagal melakukan login, karena kesalahan memberikan username dan password.

**Masukan** terdiri dari dua teks yang berisi username dan password, apabila username dan password salah, maka program akan meminta masukan ulang. Apabila username dan password sudah benar maka program akan menampilkan informasi berapa kali percobaan login yang gagal dilakukan. Asumsi username dan password yang benar adalah "Admin" dan "Admin" tanpa tanda petik.

#### Contoh masukan dan keluaran

| No | Masukan   | Keluaran                |
|----|---|-------------------------|
| 1. | User123 user123<br>User admin<br>Admin admin<br>Admin Admin123<br>Admin Admin | 4 percobaan gagal login |
| 2. | Admin Admin   | 0 percobaan gagal login |

Source Code:

```
package main

import "fmt"

func main() {
    var usr, pwd string
    var salah int

    salah = 0
    fmt.Scan(&usr, &pwd)
    for usr != "Admin" || pwd != "Admin"{
        fmt.Scan(&usr, &pwd)
        salah++
    }
    fmt.Printf("%d percobaan gagal login\n", salah)
}
```

Output:

```
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Latsol/latsol1.go"
User123 user123
User admin
Admin admin
Admin Admin123
Admin Admin
4 percobaan gagal login
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Latsol/latsol1.go"
Admin Admin
0 percobaan gagal login
○ mymac@192 praktikum1 %
```

Deskripsi Program:

- var usr, pwd mendeklarasikan variabel dengan tipe data string
- var salah mendeklarasikan variabel dengan tipe data integer
- mendeklarasikan nilai salah = 0
- fmt.Scan(&usr, &pwd) untuk menginputkan nilai dari variabel
- akan terus melakukan perulangan jika kondisi usr yang diinputkan bukan "Admin" atau pwd juga bukan "Admin"
- var salah akan terus bertambah sebanyak perulangan yang dilakukan.

## 2. Latihan Soal 2

Buatlah program yang digunakan untuk mencacah setiap digit yang terdapat di dalam suatu bilangan bulat positif.

**Masukan** terdiri dari suatu bilangan bulat positif.

**Keluaran** terdiri dari nilai digit pada bilangan tersebut. Tampilkan dari digit terakhir (paling kanan) sampai dengan digit pertama (paling kiri).

**Contoh masukan dan keluaran**

| No | Masukan    | Keluaran         |
|----|------------|------------------|
| 1. | 2          | 2                |
| 2. | 2544       | 4<br>4<br>5<br>2 |
| 3. | 3423554654 | 4<br>5<br>6      |

|  |  |   |
|--|--|---|
|  |  | 4 |
|  |  | 5 |
|  |  | 5 |
|  |  | 3 |
|  |  | 2 |
|  |  | 4 |
|  |  | 3 |

Source Code:

```
package main

import "fmt"

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

    for n>0{
        digit := n % 10
        fmt.Println(digit)
        n = n / 10
    }
}
```

Output:

```
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Latsol/tempCodeRunnerFile.go"
2
2
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Latsol/tempCodeRunnerFile.go"
2544
4
4
5
2
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Latsol/tempCodeRunnerFile.go"
3423554654
4
5
6
4
5
5
3
2
4
3
mymac@192 praktikum1 %
```

Deskripsi Program :

- var n mendeklarasikan variabel dengan tipe data int
- fmt.Scan(&n) menginputkan nilai ke variabel n
- Perulangan berjalan selama nilai n lebih besar dari nol.

- Di setiap iterasi:
- **Mengambil Digit Terakhir:**
  - **digit := n % 10:** Mengambil digit paling kanan (sisa pembagian n dengan 10).
- **Mencetak Digit:**
  - **fmt.Println(digit):** Mencetak digit paling kanan.
- **Menghapus Digit Terakhir:**
  - **n = n / 10:** Membagi nilai n dengan 10 dan mengambil bagian bulatnya, sehingga digit terakhir dihilangkan.

### 3. Latihan Soal 3

Buatlah program untuk mencari hasil integer division dari dua bilangan.

Gunakan

perulangan dan tidak diperbolehkan menggunakan operator pembagian.

**Masukan** terdiri dari dua bilangan bulat positif x dan y, yang mana  $x \geq y$ .

**Keluaran** terdiri dari hasil dari operasi  $x \div y$  dengan y.

**Contoh masukan dan keluaran**

| No | Masukan | Keluaran |
|----|---------|----------|
| 1. | 5 2     | 2        |
| 2. | 10 7    | 1        |
| 3. | 120     | 30       |

Source Code

```
package main

import "fmt"

func main() {
    var x, y int

    fmt.Scan(&x, &y)

    i := 0
```

```

temp := x
for temp >= y{
    temp -=y
    i++
}
fmt.Println(i)
}

```

Output:

```

go run "/Users/mymac/praktikum1/Modul12/Latsol/latsol3.go"
/Users/mymac/.zprofile:5: unmatched "
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Latsol/latsol3.go"
5 2
2
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Latsol/latsol3.go"
10 7
1
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul12/Latsol/latsol3.go"
120 4
30
○ mymac@192 praktikum1 % █

```

Deskripsi Program :

- var v, y, l, temp mendeklarasikan variabel dengan tipe data integer
- fmt.Scan(&x, &y) menginputkan nilai dari var x dan y
- melakukan Perulangan (**for temp >= y**):
- Selama nilai temp masih lebih besar atau sama dengan y:
  1. Kurangi nilai temp dengan y (**temp -= y**).
  2. Tambahkan 1 ke variabel penghitung i (**i++**).
- Ketika nilai temp sudah kurang dari y, perulangan berhenti.
- Variabel i mencerminkan berapa kali y dapat masuk ke x secara utuh (hasil bagi bulat).

## DAFTAR PUSTAKA

*A.14. Perulangan. (n.d.). Retrieved from <https://dasarpemrogramangolang.novalagung.com/A-perulangan.html>*