

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 12**  
**“WHILE - LOOP”**



**DISUSUN OLEH:**  
**ICHYA ULUMIDDIIN**  
**103112400076**  
**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

### 1. Pendahuluan

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. Sebelumnya pada modul ke-5 dan 6 telah dipelajari instruksi perulangan dengan for-loop. Instruksi for-loop memungkinkan kita melakukan berulangan sebanyak  $n$  iterasi, akan tetapi pada banyak kasus yang melibatkan perulangan, tidak semua perulangan diketahui jumlah iterasinya di awal. Perulangan seperti ini disebut juga dengan istilah perulangan dengan kondisi.

- **Karakteristik While-Loop**

Struktur kontrol perulangan menggunakan while-loop memiliki bentuk yang hampir serupa dengan penulisan if-then pada percabangan, yaitu memiliki kondisi dan aksi. Hal yang membedakan adalah aksi akan dilakukan secara berulang-ulang selama kondisi bernilai true.

**1) Kondisi**, merupakan nilai atau operasi tipe data yang menghasilkan tipe data boolean. Kondisi ini merupakan syarat terjadinya perulangan. Artinya perulangan terjadi apabila kondisi bernilai true.

**2) Aksi**, merupakan kumpulan instruksi yang akan dieksekusi secara berulang-ulang selama kondisi bernilai true. Salah satu instruksi dari aksi harus bisa membuat kondisi yang awalnya bernilai true menjadi false, tujuannya adalah untuk membuat perulangan berhenti. Pada penulisan notasinya secara umum bahasa pemrograman menggunakan kata kunci while, tetapi khusus di bahasa pemrograman Go, kata kunci yang digunakan adalah for. Walaupun berbeda dari kata kunci yang digunakan, secara struktur penulisannya tetap sama, sehingga tetap mudah untuk membedakan instruksi for yang digunakan adalah for-loop atau while-loop.

## CONTOH SOAL

### 1. Latihan 1

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.

#### 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:

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\al
odul12\coso1\coso1.go"
5
5x4x3x2x1
```

#### Deskripsi Program:

Kode tersebut bertujuan untuk mencetak urutan perkalian dari angka n hingga 1 dalam format "n x (n-1) x ... x 1". Program ini pertama-tama meminta input nilai n, kemudian mencetak angka n diikuti oleh tanda "x". Setelah itu, nilai j (yang awalnya sama dengan n) dikurangi satu secara bertahap dalam sebuah loop, dan angka berikutnya dicetak hingga mencapai angka 1. Ketika j mencapai 1, program mencetak angka 1 tanpa tanda "x" di akhir, menyelesaikan urutan perkalian tersebut.

## 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".

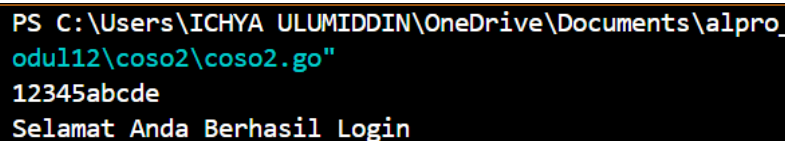
### Source Code:

```
package main

import "fmt"

func main() {
    var token string
    fmt.Scan(&token)
    for token != "12345abcde" {
        fmt.Scan(&token)
    }
    fmt.Println("Selamat Anda Berhasil Login")
}
```

### Output:



```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\alpro_
odu112\coso2\coso2.go"
12345abcde
Selamat Anda Berhasil Login
```

### Deskripsi Program:

Kode tersebut bertujuan untuk mensimulasikan proses login menggunakan token. Program pertama-tama meminta input token dari pengguna melalui `fmt.Scan(&token)`. Kemudian, program memeriksa apakah token yang dimasukkan tidak sama dengan nilai yang benar, yaitu "12345abcde". Selama token yang dimasukkan tidak sesuai, program akan terus meminta pengguna untuk memasukkan token lagi. Begitu token yang dimasukkan cocok dengan "12345abcde", program keluar dari loop dan menampilkan pesan "Selamat Anda Berhasil Login", yang menandakan bahwa proses login berhasil. Dengan kata lain, kode ini memastikan bahwa pengguna hanya bisa melanjutkan jika memasukkan token yang tepat.

### 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.

#### 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:

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\alpro_1>
odul12\coso3\coso3.go"
10
0 1 1 2 3 5 8 13 21 34
```

#### Deskripsi Program:

Kode tersebut bertujuan untuk mencetak deret angka Fibonacci sebanyak N kali. Deret Fibonacci dimulai dengan dua angka pertama, yaitu 0 dan 1, dan setiap angka berikutnya adalah jumlah dari dua angka sebelumnya. Program pertama-tama meminta input jumlah angka N yang ingin dicetak dalam deret Fibonacci. Kemudian, variabel s1 dan s2 diinisialisasi dengan nilai 0 dan 1, yang merupakan dua angka pertama dalam deret Fibonacci. Program memasuki sebuah loop yang akan berjalan sebanyak N kali. Dalam setiap iterasi, angka s1 dicetak, kemudian nilai s1 dan s2 diperbarui untuk mencetak angka berikutnya dalam deret, yaitu dengan cara menjadikan s2 sebagai s1, dan menjadikan jumlah dari s1 dan s2 sebagai s2. Loop berlanjut hingga jumlah angka Fibonacci yang dicetak mencapai N.

## SOAL LATIHAN

- 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

### *Source Code*

```
package main

import "fmt"

func main() {
    var username, password string
    var Perobaangagal int
    fmt.Scan(&username, &password)
    for username != "Admin" || password != "Admin" {
        Perobaangagal++
        fmt.Scan(&username, &password)
    }
    fmt.Println(Perobaangagal, "Percobaan gagal login")
}
```

### **Output:**

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\alpro_1
OMEWORK_WEEK12(LAB)\latsol1\latsol1.go"
Admin
admin
Admin Admin
1 Percobaan gagal login
```

### **Deskripsi Program:**

Kode tersebut bertujuan untuk mensimulasikan proses login dengan username dan password. Program pertama-tama meminta input username dan password dari pengguna. Kemudian, program memeriksa apakah kombinasi username dan password yang dimasukkan tidak sesuai dengan nilai yang benar, yaitu "Admin" untuk keduanya. Jika username atau password tidak sesuai, program akan menghitung jumlah percobaan gagal login dengan cara meningkatkan nilai variabel Perobaangagal setiap kali percobaan login gagal. Program akan terus meminta input username dan password hingga pengguna memasukkan kombinasi yang benar (yaitu "Admin" untuk keduanya). Setelah login

berhasil, program mencetak jumlah percobaan gagal yang terjadi sebelum login berhasil. Jadi, tujuan utama dari kode ini adalah untuk mengukur dan mencatat berapa kali pengguna gagal memasukkan username dan password yang benar.

- 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)

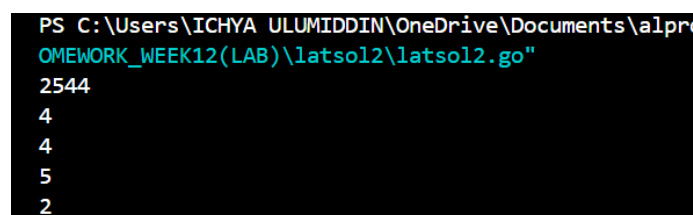
#### *Source Code*

```
package main

import "fmt"

func main() {
    var number int
    fmt.Scan(&number)
    for number > 0 {
        digit := number % 10
        fmt.Println(digit)
        number = number / 10
    }
}
```

#### **Output:**



```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\alpro\HOMEWORK_WEEK12(LAB)\latsol2\latsol2.go"
2544
4
4
5
2
```

#### **Deskripsi Program :**

Kode tersebut bertujuan untuk mencetak setiap digit dari sebuah angka secara terbalik. Program pertama-tama meminta input angka melalui `fmt.Scan(&number)`. Kemudian, dalam sebuah loop, program mengambil digit terakhir dari angka tersebut dengan menggunakan operasi modulus (`number % 10`) dan mencetak digit tersebut. Setelah mencetak digit, angka tersebut dibagi 10 (dengan operasi pembagian integer `number = number / 10`), yang menghilangkan digit terakhir, sehingga pada iterasi berikutnya, digit berikutnya akan diambil. Loop ini berlanjut hingga angka tersebut menjadi 0. Dengan

cara ini, angka yang awalnya diberikan akan dicetak digitnya satu per satu dari belakang ke depan, sehingga digit pertama yang dimasukkan akan dicetak terakhir.

- 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  $x \div y$ .

**Source Code:**

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x, &y)
    hasilBagi := 0
    for x >= y {
        x -= y
        hasilBagi++
    }
    fmt.Println(hasilBagi)
}
```

**Output:**

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\alpro_1>
OMework_WEEK12(LAB)\latsol3\latsol3.go"
10 7
1
```

**Deskripsi Program:**

Tujuan program tersebut adalah untuk menghitung hasil pembagian bulat antara dua angka, x dan y, menggunakan pendekatan pengurangan berulang. Program ini tidak menggunakan operator pembagian, melainkan mengurangi nilai x sebanyak nilai y hingga x menjadi lebih kecil dari y. Setiap kali pengurangan dilakukan, variabel hasilBagi akan bertambah 1, yang pada akhirnya mencatat berapa kali y dapat dikurangkan dari x. Setelah proses selesai, program mencetak hasil tersebut, yang menggambarkan hasil pembagian bulat (quotient) dari x dibagi y. Dengan kata lain, program ini menghitung seberapa banyak y bisa dimasukkan ke dalam x tanpa sisa.



## DAFTAR PUSTAKA

Amisha. (2023, May 4). *While Loop in Golang*. Retrieved from Scaler:  
<https://www.scaler.com/topics/golang/while-loop-in-golang/>