

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



**DISUSUN OLEH:**  
**Muhammad Shabrian Fadly**  
**103112400087**  
**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

### WHILE-LOOP

#### 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. 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. Contoh soal 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:

```
PS D:\Coding manja> go run "d:\Coding manja\modul12\conso1\conso1.go"
0
1
PS D:\Coding manja> go run "d:\Coding manja\modul12\conso1\conso1.go"
5
5x4x3x2x1
PS D:\Coding manja> go run "d:\Coding manja\modul12\conso1\conso1.go"
10
10x9x8x7x6x5x4x3x2x1
PS D:\Coding manja> go run "d:\Coding manja\modul12\conso1\conso1.go"
1
1
```

Deskripsi Program:

Program ini mencetak urutan bilangan mulai dari angka yang dimasukkan oleh pengguna hingga angka 1, dengan setiap angka dipisahkan oleh tanda "x". Program mengurangi angka satu per satu setiap kali dan mencetaknya sampai mencapai angka 1. Misalnya, jika pengguna memasukkan angka 5, program akan mencetak "5x4x3x2x1".

## 2. Contoh soal 2

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 D:\Coding manja> go run "d:\Coding manja\modul12\conso2\conso2.go"
12345abcde
Selamat Anda Berhasil Login
```

Deskripsi Program:

Program ini digunakan untuk memastikan bahwa pengguna hanya bisa login jika mereka memasukkan token yang benar. Selama token yang dimasukkan salah, program akan terus meminta input hingga token yang benar dimasukkan, kemudian memberi pesan bahwa login berhasil.

### 3. Contoh soal 3

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 D:\Coding manja> go run "d:\Coding manja\modul12\conso3\conso3.go"
5
0 1 1 2 3
PS D:\Coding manja> go run "d:\Coding manja\modul12\conso3\conso3.go"
2
0 1
PS D:\Coding manja> go run "d:\Coding manja\modul12\conso3\conso3.go"
10
0 1 1 2 3 5 8 13 21 34
```

Deskripsi Program:

Program ini digunakan untuk mencetak deret Fibonacci sebanyak N angka yang diminta oleh pengguna. Deret Fibonacci dimulai dengan angka 0 dan 1, dan angka berikutnya adalah hasil penjumlahan dari dua angka sebelumnya. Program terus mencetak angka dalam deret tersebut sampai mencapai jumlah yang diinginkan. Misalnya, jika pengguna memasukkan angka 5, program akan mencetak 0 1 1 2 3. Program menggunakan perulangan untuk menghasilkan dan mencetak deret Fibonacci berdasarkan jumlah yang diinginkan oleh pengguna.

## Latihan Soal

### 1. Latihan soal 1

Statement Perulangan:

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 D:\Coding manja> go run "d:\Coding manja\modul12\latsol1\latsol1.go"
admin
admin
User
Admin
User123
user123
Admin
admin
Admin
Admin
4 Percobaan gagal login
PS D:\Coding manja> go run "d:\Coding manja\modul12\latsol1\latsol1.go"
Admin
Admin
0 Percobaan gagal login
```

Deskripsi Program: Program ini meminta pengguna untuk memasukkan username dan password. Jika username atau password yang dimasukkan tidak sesuai dengan "Admin", program akan terus meminta input hingga keduanya benar. Setiap kali pengguna salah memasukkan username atau password, program akan menghitung percobaan gagal dan menampilkannya setelah login berhasil. Sebagai contoh, jika pengguna memasukkan username dan password yang salah tiga kali, program akan menampilkan jumlah percobaan gagal tersebut setelah berhasil login.

## 2. Latihan soal 2

Statement Perulangan:

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 D:\Coding manja> go run "d:\Coding manja\modul12\latsol2\latsol2.go"
2
2
PS D:\Coding manja> go run "d:\Coding manja\modul12\latsol2\latsol2.go"
2544
4
4
5
2
```

Deskripsi Program:

Program ini digunakan untuk membalikkan angka yang dimasukkan oleh pengguna dan mencetak setiap digitnya satu per satu dari belakang. Program ini bekerja dengan cara mengambil sisa hasil bagi angka dengan 10 untuk mendapatkan digit terakhir, kemudian membagi angka tersebut dengan 10 untuk menghilangkan digit terakhir, dan mengulang proses tersebut hingga angka menjadi 0.



### 3. Latihan soal 3

Statement Perulangan:

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 dengan 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 D:\Coding manja> go run "d:\Coding manja\modul12\latsol3\latsol3.go"
5
2
2
PS D:\Coding manja> go run "d:\Coding manja\modul12\latsol3\latsol3.go"
10
7
1
```

Deskripsi Program: Program ini menghitung hasil pembagian antara dua angka x dan y dengan cara mengurangi angka x sebanyak y hingga x menjadi lebih kecil dari y. Setiap kali pengurangan dilakukan, variabel hasilBagi akan bertambah 1. Program ini akan menampilkan berapa kali angka y dapat dikurangkan dari angka x, yang pada dasarnya adalah hasil pembagian tanpa sisa. Sebagai contoh, jika x adalah 10 dan y adalah 3, maka hasil yang ditampilkan adalah 3, karena 10 dapat dikurangi 3 sebanyak 3 kali sebelum angka x menjadi lebih kecil dari y.