

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



**DISUSUN OLEH:**  
**RAIHAN ADI ARBA**  
**103112400071**  
**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 adalah proses menjalankan blok kode secara berulang selama kondisi tertentu terpenuhi. Proses ini biasanya melibatkan variabel sebagai penanda iterasi atau pengontrol kapan perulangan harus dihentikan. Dalam bahasa Go, kata kunci yang digunakan untuk perulangan hanyalah `for`, tetapi fleksibilitasnya mencakup fungsi `for`, `foreach`, dan `while` seperti pada bahasa pemrograman lainnya.

Ada beberapa metode standar dalam menggunakan `for` di Go. Pertama, `for` dapat digunakan dengan mendeklarasikan variabel counter beserta kondisinya secara eksplisit setelah kata kunci `for`. Kedua, `for` juga dapat ditulis hanya dengan menyertakan kondisi tanpa deklarasi atau iterasi variabel counter, yang konsepnya menyerupai penggunaan `while` pada bahasa lain. Ketiga, `for` dapat digunakan tanpa argumen apa pun, menghasilkan perulangan tanpa henti yang hanya dapat dihentikan menggunakan kata kunci `break`.

Selanjutnya, terdapat kombinasi `for` dengan `range`, yang sering digunakan untuk mengiterasi struktur data seperti string, array, slice, atau map. Kata kunci lain yang sering digunakan dalam perulangan adalah `break` dan `continue`. `Break` berfungsi untuk menghentikan perulangan secara paksa, sedangkan `continue` digunakan untuk melompat langsung ke iterasi berikutnya tanpa menyelesaikan blok kode saat ini.

Go juga mendukung perulangan bersarang, di mana sebuah perulangan dapat ditempatkan di dalam perulangan lain. Dalam konteks ini, `break` dan `continue` hanya berlaku pada perulangan tempat mereka digunakan. Namun, dengan memanfaatkan teknik pemberian label pada perulangan, kedua kata kunci ini dapat diarahkan untuk mengontrol perulangan tertentu, termasuk perulangan terluar.

## A. UNGUIDED

### 1. Latihan 1

Buat program dalam bahasa Go untuk menghitung jumlah percobaan login yang gagal akibat kesalahan memasukkan username dan password. Masukan terdiri dari dua teks berupa username dan password. Jika salah, program meminta masukan ulang. Jika benar, tampilkan jumlah percobaan login gagal.

- Asumsi username dan password yang benar: `"Admin"` (tanpa tanda petik).
- Contoh Masukan dan Keluaran:

No	Masukan	Keluaran
1	User123 user123 User admin Admin admin Admin Admin123 Admin Admin	4 percobaan gagal login
2	Admin Admin	0 percobaan gagal login

Source Code:

```
package main

import "fmt"

func main() {
    var user, pass string
    var failed int

    failed = 0
    fmt.Scan(&user, &pass)
    for user != "Admin" || pass != "Admin" {
        fmt.Scan(&user, &pass)
        failed++
    }
    fmt.Printf("%d Percobaan gagal login\n", failed)
}
```

Output :

```

● raihan@Raihans-MacBook-Pro minggu 12 % go run l1.go
user admin
admin
pass
admin password
admin admin
admin pass
pass admin

admin admin
Admin Password
Admin Admin
8 Percobaan gagal login
○ raihan@Raihans-MacBook-Pro minggu 12 %
● raihan@Raihans-MacBook-Pro minggu 12 % go run l1.go
Admin Admin
0 Percobaan gagal login
○ raihan@Raihans-MacBook-Pro minggu 12 % █

```

#### Penjelasan Program:

Program ini merupakan implementasi sistem login sederhana yang ditulis dalam bahasa pemrograman Go. Program dimulai dengan mendeklarasikan dua variabel string untuk menyimpan usekredensialpassword, serta satu variabel integer untuk menghitung jumlah percobaan login yang gagal. Pada awalnya, program menginisialisasi counter kegagalan login dengan nilai 0 dan menerima input pertama dari pengguna untuk username dan password. Selanjutnya, program menggunakan struktur perulangan yang akan terus berjalan selama input username dan password tidak sesuai dengan nilai yang diharapkan, yaitu "Admin" untuk keduanya. Setiap kali terjadi kesalahan input, program akan meminta input baru dan menambah nilai counter kegagalan. Ketika akhirnya pengguna berhasil memasukkan kombinasi username dan password yang benar, program akan keluar dari loop dan menampilkan jumlah percobaan gagal yang telah dilakukan. Seperti yang terlihat pada contoh eksekusi, jika pengguna langsung memasukkan kredensial yang benar ("Admin Admin"), program akan menampilkan 0 percobaan gagal, namun jika terjadi beberapa kali kesalahan input sebelum berhasil login, program akan menampilkan jumlah total percobaan yang gagal tersebut.

## 2. Latihan 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 4 5 2
3	3423554654	4 5 6 4 5 5 3 2 4 3

Source Code :

```
package main

import "fmt"

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

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

Output :

```
● raihan@Raihans-MacBook-Pro minggu 12 % go run l2.go
2
2
● raihan@Raihans-MacBook-Pro minggu 12 % go run l2.go
24554
4
5
5
4
2
```

### Pembahasan :

Program ini adalah implementasi sederhana untuk memisahkan dan menampilkan setiap digit dari suatu bilangan bulat. Program dimulai dengan mendeklarasikan dua variabel integer: 'n' untuk menyimpan bilangan input dari pengguna dan 'digit' untuk menyimpan setiap digit yang akan dipisahkan. Setelah menerima input, program menggunakan perulangan for yang akan terus berjalan selama nilai n lebih besar dari 0. Di dalam loop, program menggunakan operator modulo (%) untuk mendapatkan digit terakhir dari bilangan tersebut ( $n \% 10$ ), kemudian menampilkan digit tersebut. Selanjutnya, program membagi nilai n dengan 10 ( $n = n / 10$ ) untuk menghapus digit terakhir yang sudah diproses dan melanjutkan ke digit berikutnya. Proses ini akan terus berulang sampai semua digit telah dipisahkan dan ditampilkan. Sebagai contoh, jika input adalah 123, maka program akan menampilkan 3, 2, dan 1 secara berurutan (dari digit terakhir ke digit pertama).

### 3. Latihan 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  $x \text{ div } y$ .

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, i, temp int

    fmt.Scan(&x, &y)

    i = 0
    temp = x
    for temp >= y {
        temp -= y
        i++
    }
}
```

```
}  
    fmt.Println(i)  
}
```

Output:

```
● raihan@Raihans-MacBook-Pro minggu 12 % go run l3.go  
5 2  
2  
● raihan@Raihans-MacBook-Pro minggu 12 % go run l3.go  
10 7  
1
```

Deskripsi Program:

Program ini merupakan implementasi pembagian bilangan menggunakan metode pengurangan berulang. Program dimulai dengan memensimulasikan tempat variabel integer: 'x' sebagai bilangan yang akan dibagi (dividend), 'y' sebagai pembagi (divisor), 'i' sebagai penghitung berapa kali pengurangan dilakukan, dan 'temp' sebagai variabel sementara untuk menyimpan hasil pengurangan. Setelah menerima input untuk x dan y, program menginisialisasi i dengan 0 dan menyalin nilai x ke dalam temp. Selanjutnya, program menggunakan perulangan for yang akan terus berjalan selama nilai temp lebih besar atau sama dengan y. Di dalam loop, program mengurangi temp dengan y dan menambah nilai i dengan 1. Proses ini akan terus berulang sampai temp menjadi lebih kecil dari y. Akhirnya, program menampilkan nilai i yang merepresentasikan hasil pembagian x dengan y ( $x \div y$ ). Metode ini pada dasarnya mensimulasikan proses pembagian manual dengan cara pengurangan berulang, di mana hasil bagi adalah jumlah pengurangan yang dapat dilakukan sebelum sisa menjadi lebih kecil dari pembagi.

## DAFTAR PUSTAKA

**Prayogo, N. A. (2021). *Dasar Pemrograman Go. Ebook***