

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



DISUSUN OLEH:
Muhammad Faris Rachmadi
103112400079
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

12.1 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. 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.

Contoh perulangan jenis ini di kehidupan dunia nyata adalah sebagai berikut:

- "Menulis teks tertentu selama tinta pena masih ada". Terdapat kondisi "tinta pena masih ada" sebagai syarat perulangan.
- "Saya makan sup demi sup selama saya masih lapar". Terdapat kondisi "saya masih lapar" sebagai syarat 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. Terdapat dua bentuk perulangan dengan kondisi ini, yaitu while-loop dan repeat-until. Masing masing kita akan bahas di modul yang berbeda.

12.2 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

CONTOH SOAL

12.1.1 Contoh Soal 1

Source Code:

```
coso1 > -go coso1.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n, j int
7      fmt.Scan(&n)
8      j = n
9      for j > 1 {
10         fmt.Print(j, " x ")
11         j = j - 1
12     }
13     fmt.Print(1)
14
15 }
16
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\coso1.go"
0
1
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\coso1.go"
5
5 x 4 x 3 x 2 x 1
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\coso1.go"
10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\coso1.go"
1
1
```

Deskripsi Program:

Tujuan program GO ini adalah untuk menampilkan deret bilangan Faktorial dari suatu bilangan.

12.1.2 Contoh Soal 2

Source Code:

```
coso2 > go coso2.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var token string
7      fmt.Scan(&token)
8      for token != "12345abcde" {
9          fmt.Scan(&token)
10     }
11
12     fmt.Println("Selamat anda berhasil login")
13 }
14
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Ma
Qwe12312
231234
13213
1231ijwe
12345abcde
Selamat anda berhasil login
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Ma
12345abcde
Selamat anda berhasil login
```

Deskripsi Program:

Tujuan program GO ini adalah untuk login ke dalam suatu aplikasi. Asumsi token untuk yang valid adalah "12345abcde".

12.1.3 Contoh Soal 3

Source Code:

```
coso3 > -go coso3.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var N, s1, s2, j, temp int
7      fmt.Scan(&N)
8      s1 = 0
9      s2 = 1
10     j = 0
11     for j < N {
12         fmt.Print(s1, " ")
13         temp = s1 + s2
14         s1 = s2
15         s2 = temp
16         j = j + 1
17     }
18 }
19
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Ku
5
0 1 1 2 3
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Ku
2
0 1
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Ku
10
0 1 1 2 3 5 8 13 21 34
```

Deskripsi Program:

Tujuan program GO ini adalah untuk mencetak N bilangan pertama dalam deret Fibonacci.

Latihan Soal

1. Latihan Soal 1

Source Code:

```
latsol1 > -o latsol1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var username, password string
7     fmt.Scan(&username, &password)
8     percobaan := 0
9     for username != "Admin" || password != "Admin" {
10         fmt.Scan(&username, &password)
11         percobaan++
12     }
13     fmt.Println(percobaan, "percobaan gagal login")
14 }
15
```

Output:

```
PROBLEMS OUTPUT TERMINAL PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go r
nts\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\latsol1\latsol1.go"
User123 user123
User admin
Admin admin
Admin Admin123
Admin Admin
4 percobaan gagal login
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go r
nts\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\latsol1\latsol1.go"
Admin Admin
0 percobaan gagal login
```

Deskripsi Program:

Tujuan program GO ini adalah untuk menghitung berapa banyak seseorang pengguna gagal melakukan login, karena kesalahan memberikan username dan password.

2. Latihan Soal 2

Source Code:

```
latsol2 > go latsol2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Scan(&n)
8     for n > 0 {
9         digit := n % 10
10        fmt.Println(digit)
11        n = n / 10
12    }
13 }
14
```

Output:

```
PROBLEMS OUTPUT TERMINAL PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\latsol2\latsol2.go"
2
2
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\latsol2\latsol2.go"
2544
4
4
5
2
```

Deskripsi Program:

Tujuan program GO Program untuk mencacah setiap digit yang terdapat di dalam suatu bilangan bulat positif.

3. Latihan Soal 3

Source Code:

```
latso3 > -go latso3.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var x, y int
7      fmt.Scan(&x, &y)
8      hasil := 0
9      for x >= y {
10         x = x - y
11         hasil++
12     }
13     fmt.Println(hasil)
14 }
15
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\latso3\latso3.go"
5 2
2
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul12\latso3\latso3.go"
10 7
1
```

Deskripsi Program:

Tujuan program GO ini adalah untuk mencari hasil integer division dari dua bilangan. Gunakan perulangan dan tidak diperbolehkan menggunakan operator pembagian

DAFTAR PUSTAKA

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

MODUL PRAKTIKUM 12 – WHILE-LOOP ALGORITMA DAN
PEMROGRAMAN 1 S1 INFORMATIKA