

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



Disusun oleh :
NAMA : Felix Pedrosa Valentino
NIM : 103112400056
S1 IF – 12 - 01

Dosen Pengampu :
Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

DASAR TEORI

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 suap demi suap selama saya masih lapar"*.

Terdapat kondisi "saya masih lapar" sebagai syarat perulangan.

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.

Bentuk While-Loop

- Secara struktur penulisannya mirip seperti If-then, yang mana <action> akan dieksekusi secara berulang selama <condition> bernilai true.
- Jumlah iterasi tidak dapat ditentukan, karena bergantung dengan perubahan nilai pada <condition>.
- Pada bahasa Go, keyword yang digunakan adalah "for" dengan struktur penulisan while-loop.

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

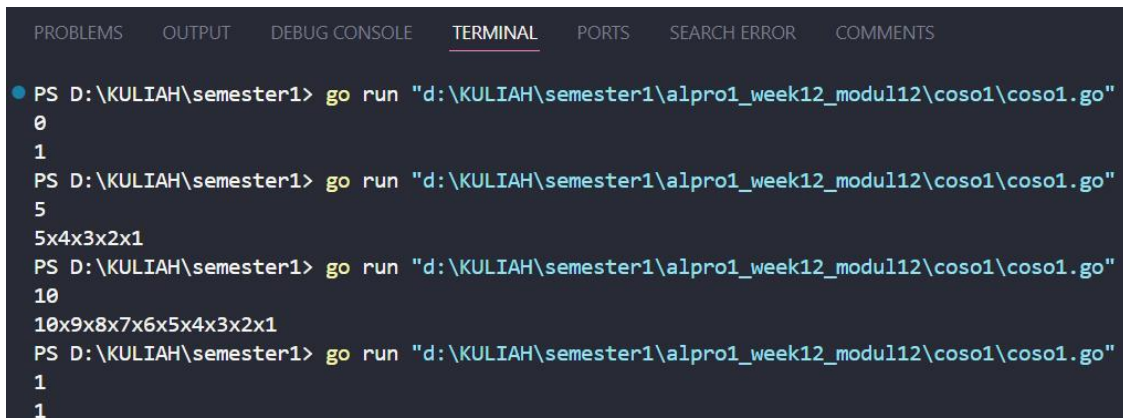
Source Code :

```
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 :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\coso1\coso1.go"
0
1
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\coso1\coso1.go"
5
5x4x3x2x1
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\coso1\coso1.go"
10
10x9x8x7x6x5x4x3x2x1
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\coso1\coso1.go"
1
1
1
```

Deskripsi Program :

Program di atas berfungsi untuk mencetak urutan angka dari nilai yang dimasukkan oleh pengguna hingga angka 1, dengan format yang diakhiri dengan huruf 'x' untuk setiap angka, kecuali angka terakhir. Pertama, program meminta pengguna untuk memasukkan sebuah bilangan bulat positif yang disimpan dalam variabel `n`. Kemudian, program

menginisialisasi variabel *j* dengan nilai *n* dan memulai sebuah loop yang akan terus berjalan selama *j* lebih besar dari 1. Di dalam loop, program mencetak nilai *j* diikuti dengan huruf 'x', dan kemudian mengurangi nilai *j* sebanyak 1. Setelah loop selesai, program mencetak angka 1 tanpa diikuti oleh huruf 'x'. Sebagai contoh, jika pengguna memasukkan angka 5, output yang dihasilkan akan menjadi "5x4x3x2x1".

2.) Contoh Soal 2

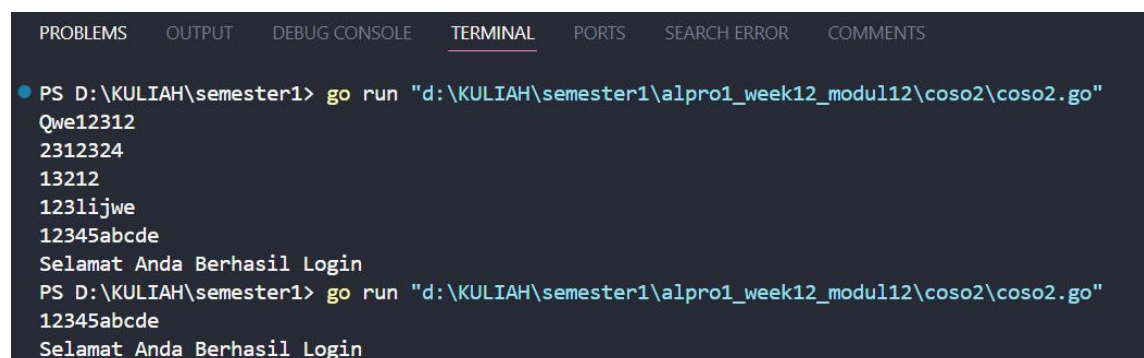
Source Code :

```
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 :



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\coso2\coso2.go"
Qwe12312
2312324
13212
1231ijwe
12345abcde
Selamat Anda Berhasil Login
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\coso2\coso2.go"
12345abcde
Selamat Anda Berhasil Login
```

Deskripsi Program :

Program di atas berfungsi sebagai sistem autentikasi sederhana yang meminta pengguna untuk memasukkan sebuah token. Program dimulai dengan mendeklarasikan variabel *token* bertipe *string* yang akan

menyimpan input dari pengguna. Selanjutnya, program menggunakan fungsi `fmt.Scan` untuk membaca input token dari pengguna. Program kemudian memasuki sebuah loop yang akan terus meminta input token hingga pengguna memasukkan token yang benar, yaitu "12345abcde". Jika token yang dimasukkan tidak sesuai, program akan terus meminta pengguna untuk memasukkan token yang baru. Setelah pengguna berhasil memasukkan token yang benar, program akan keluar dari loop dan mencetak pesan "Selamat Anda Berhasil Login".

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 :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\coso3\coso3.go"
5
0 1 1 2 3
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\coso3\coso3.go"
2
0 1
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\coso3\coso3.go"
10
0 1 1 2 3 5 8 13 21 34
```

Deskripsi Program :

Program di atas berfungsi untuk mencetak deret Fibonacci hingga jumlah elemen yang ditentukan oleh pengguna. Program dimulai dengan mendeklarasikan variabel N, s1, s2, j, temp bertipe integer untuk menyimpan jumlah elemen yang ingin dicetak, s1 dan s2 untuk menyimpan dua angka pertama dalam deret Fibonacci, serta j dan temp untuk mengontrol iterasi dan menyimpan nilai sementara. Setelah meminta pengguna untuk memasukkan nilai N, program menginisialisasi s1 dengan 0 dan s2 dengan 1, yang merupakan dua angka pertama dalam deret Fibonacci. Program kemudian memasuki sebuah loop yang akan berjalan selama j kurang dari N. Di dalam loop, program mencetak nilai s1, kemudian menghitung angka Fibonacci berikutnya dengan menjumlahkan s1 dan s2, dan memperbarui nilai s1 dan s2 untuk iterasi berikutnya. Variabel j juga diperbarui untuk melacak jumlah elemen yang telah dicetak. Dengan demikian, jika pengguna memasukkan angka 5, output yang dihasilkan akan menjadi "0 1 1 2 3", yang merupakan lima angka pertama dalam deret Fibonacci.

Latihan Soal

1.) Latihan Soal 1

Source Code :

```
package main

import "fmt"

func main() {
    var username, password string
    fmt.Scan(&username, &password)
    percobaan := 0
    for username != "Admin" || password != "Admin" {
        fmt.Scan(&username, &password)
        percobaan++
    }
    fmt.Println(percobaan, "Percobaan gagal login")
}
```

Output :

```
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\latsoal1\latsoal1.go"
User123 user123
User admin
Admin admin
Admin Admin123
Admin Admin
4 Percobaan gagal login
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\latsoal1\latsoal1.go"
Admin Admin
0 Percobaan gagal login
```

Deskripsi Program :

Program ini berfungsi untuk mengelola proses login dengan memeriksa kombinasi username dan password yang dimasukkan oleh pengguna. Setelah pengguna memasukkan username dan password, program akan memeriksa apakah salah satu dari keduanya adalah "Admin". Jika keduanya tidak sesuai, program akan terus meminta input username dan password yang baru, sambil menghitung jumlah percobaan login yang gagal menggunakan variabel percobaan. Setiap kali pengguna

memasukkan kombinasi yang salah, nilai percobaan akan bertambah satu. Ketika pengguna akhirnya berhasil memasukkan username atau password yang benar, program akan keluar dari loop dan mencetak jumlah percobaan gagal yang telah dilakukan, disertai dengan pesan "Percobaan gagal login".

2.) Latihan Soal 2

Source Code :

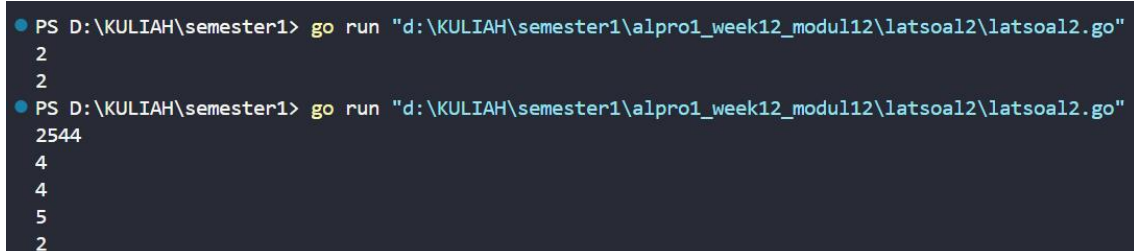
```
package main

import "fmt"

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

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

Output :



```
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\latsoal2\latsoal2.go"
2
2
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\latsoal2\latsoal2.go"
2544
4
4
5
2
```

Deskripsi Program :

Program ini berfungsi untuk membalik dan menampilkan digit dari sebuah bilangan bulat positif yang dimasukkan oleh pengguna. Setelah pengguna memasukkan bilangan tersebut, program akan menggunakan loop untuk memproses setiap digit dari bilangan tersebut. Dalam setiap

iterasi, program mengambil digit terakhir dengan menggunakan operasi modulus (%) dan mencetaknya ke layar. Kemudian, bilangan tersebut dibagi dengan 10 untuk menghapus digit terakhir, sehingga pada iterasi berikutnya, digit yang baru dapat diambil. Proses ini berlanjut hingga bilangan menjadi nol, yang menandakan bahwa semua digit telah diproses dan ditampilkan.

3.) Latihan Soal 3

Source Code :

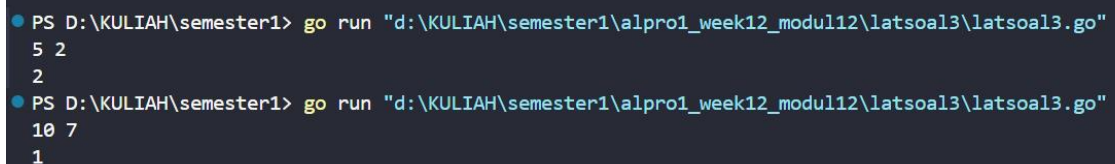
```
package main

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:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\latsoal3\latsoal3.go"
5 2
2
PS D:\KULIAH\semester1> go run "d:\KULIAH\semester1\alpro1_week12_modul12\latsoal3\latsoal3.go"
10 7
1
```

Deskripsi Program :

Program ini berfungsi untuk menghitung hasil bagi dari dua bilangan bulat yang diinputkan oleh pengguna. Di dalam fungsi main, program pertama-tama mendeklarasikan dua variabel integer, x dan y, yang akan menyimpan nilai yang dimasukkan oleh pengguna. Program kemudian

menggunakan `fmt.Scan` untuk membaca nilai-nilai tersebut dari input. Selanjutnya, variabel `hasilBagi` diinisialisasi dengan nilai 0, yang akan digunakan untuk menghitung berapa kali nilai `y` dapat dikurangkan dari `x` sebelum `x` menjadi kurang dari `y`. Program menggunakan loop `for` yang terus berjalan selama `x` lebih besar atau sama dengan `y`, di mana di dalam loop, nilai `y` dikurangkan dari `x` dan `hasilBagi` diinkrementasi. Setelah loop selesai, program mencetak nilai `hasilBagi`, yang merupakan jumlah kali `y` dapat dikurangkan dari `x`, atau dengan kata lain, hasil bagi dari `x` dibagi `y`.

DAFTAR PUSTAKA

Prasti Eko Yunanto, S.T., M.Kom. (2024). MODUL PRAKTIKUM 12 -
WHILE-LOOP
ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA (MODUL 12)