

LAPORAN PRAKTIKUM
MODUL: 12
ALGORITMA DAN PEMROGRAMAN



DISUSUN OLEH:

NAMA: JESIKA METANIA RAHMA ARIFIN

NIM: 103112400080

KELAS: 12 IF 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. 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. 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.

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*.

Notasi dalam pseudocode	Notasi dalam bahasa Go
<pre>while kondisi do // aksi endwhile</pre>	<pre>for kondisi { // aksi }</pre>

COSO: 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.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	0	1
2	5	5 x 4 x 3 x 2 x 1
3	10	10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
4	1	1

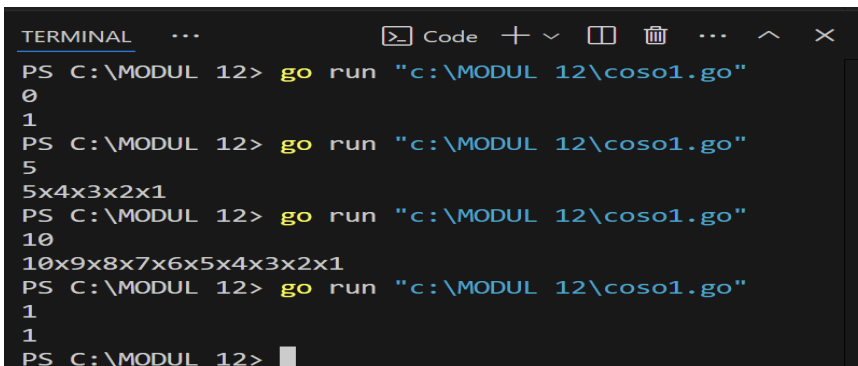
CODE:

```
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:



```
TERMINAL  ...  Code  + -  [ ]  [ ]  ...  ^  x
PS C:\MODUL 12> go run "c:\MODUL 12\coso1.go"
0
1
PS C:\MODUL 12> go run "c:\MODUL 12\coso1.go"
5
5x4x3x2x1
PS C:\MODUL 12> go run "c:\MODUL 12\coso1.go"
10
10x9x8x7x6x5x4x3x2x1
PS C:\MODUL 12> go run "c:\MODUL 12\coso1.go"
1
1
PS C:\MODUL 12>
```

DESKRIPSI PROGRAM:

Program ini dibuat untuk menampilkan deret bilangan Faktorial dari suatu bialangan.

COSO: 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".

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	Qwe12312 231234 13213 123lijwe 12345abcde	Selamat Anda berhasil login
2	12345abcde	Selamat Anda berhasil login

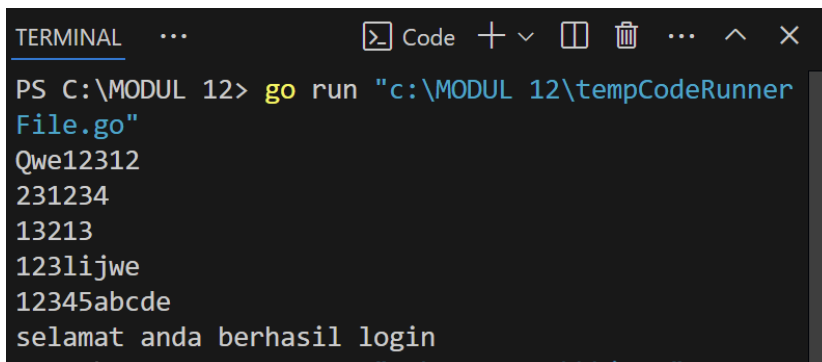
CODE:

```
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:



```
TERMINAL  ...  Code + - [ ] [ ] ... ^ x
PS C:\MODUL 12> go run "c:\MODUL 12\tempCodeRunner
File.go"
Qwe12312
231234
13213
123lijwe
12345abcde
selamat anda berhasil login
PS C:\MODUL 12>
```

DESKRIPSI PROGRAM:

Program ini dibuat untuk login ke dalam suatu aplikasi. Asumsi token yang valid adalah "12345abcde"

COSO : 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.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	0 1 1 2 3
2	2	0 1
3	10	0 1 1 2 3 5 8 13 21 34

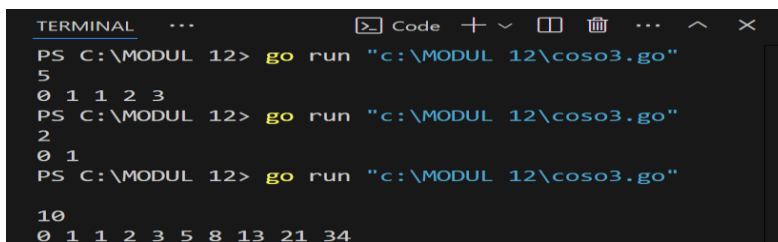
CODE:

```
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:



```
TERMINAL ... Code + - [ ] [ ] ... ^ X
PS C:\MODUL 12> go run "c:\MODUL 12\coso3.go"
5
0 1 1 2 3
PS C:\MODUL 12> go run "c:\MODUL 12\coso3.go"
2
0 1
PS C:\MODUL 12> go run "c:\MODUL 12\coso3.go"
10
0 1 1 2 3 5 8 13 21 34
```

DESKRIPSI PROGRAM:

Program ini digunakan untuk mencetak N bilangan dalam Fibonacci.

LATSOL: 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.

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

CODE:

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    const correctUsername = "Admin"
    const correctPassword = "Admin"
    var failedAttempts int
    for {
        var username, password string
        fmt.Scan(&username, &password)
        if strings.TrimSpace(username) == correctUsername &&
strings.TrimSpace(password) == correctPassword {
            break
        } else {
            failedAttempts++
        }
    }
    fmt.Printf("%d percobaan gagal login\n", failedAttempts)
}
```

OUTPUT:

```
TERMINAL ... Code + - [] [X] ... ^ X
PS C:\MODUL 12> go run "c:\MODUL 12\latsol1.go"
User123 user123
User admin
Admin admin
Admin Admin123
Admin Admin
4 percobaan gagal login
PS C:\MODUL 12> go run "c:\MODUL 12\latsol1.go"
Admin Admin
0 percobaan gagal login
```

LATSOL: 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

CODE:

```
package main

import "fmt"

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

```

    if bilangan <= 0 {
        fmt.Println("masukkan bilangan bulat positif: ")
    }
    for bilangan > 0 {
        digit := bilangan % 10
        fmt.Println(digit)
        bilangan /= 10
    }
}

```

OUTPUT:

```

TERMINAL  ...
Code + - [ ] [ ] ... ^ X
PS C:\MODUL 12> go run "c:\MODUL 12\latsol2.go"
2
2
PS C:\MODUL 12> go run "c:\MODUL 12\latsol2.go"
2544
4
4
5
2

```

DESKRIPSI PROGRAM:

Program ini digunakan untuk mencacah setiap digit yang terdapat di dalam suatu bilangan positif.

LATSOL: 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 $x \div y$ dengan y .

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5 2	2
2	10 7	1
3	120	30

CODE:

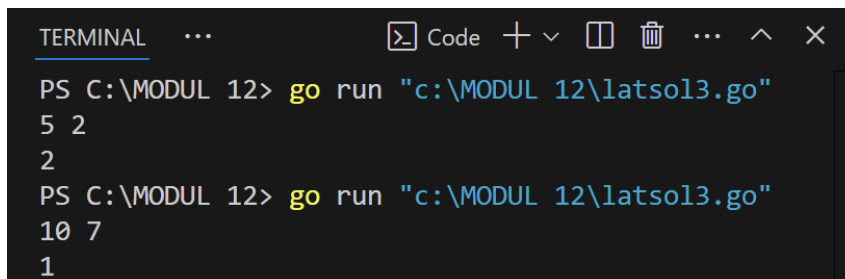
```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x)
    fmt.Scan(&y)

    if x < y // y <= 0 {
        fmt.Println("input tidak valid")
        fmt.Scan(&x, &y)
    }
    result := 0
    temp := x
    for temp >= y {
        temp -= y
        result++
    }
    fmt.Println(result)
}
```

OUTPUT:



```
TERMINAL ... Code + - [ ] [X] ... ^ X
PS C:\MODUL 12> go run "c:\MODUL 12\latsol3.go"
5 2
2
PS C:\MODUL 12> go run "c:\MODUL 12\latsol3.go"
10 7
1
```

DESKRIPSI PROGRAM:

Program ini di buat untuk mencari hasil integer division dari dua bilangan