

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



DISUSUN OLEH:
FEROS PEDROSA VALENTINO
103112400055
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

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.

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.

CONTOH SOAL

1. Coso1

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 C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\coso1\coso1.go"
0
1
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\coso1\coso1.go"
5
5 x 4 x 3 x 2 x 1
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\coso1\coso1.go"
10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\coso1\coso1.go"
1
1
```

Deskripsi program:

Program di atas ditulis dalam bahasa Go yang digunakan untuk mencetak urutan angka dari bilangan bulat non negatif yang dimasukkan oleh pengguna hingga angka satu. Program dimulai dengan mendeklarasikan dua variabel yaitu variabel *n*, *j* yang bertipe integer. Setelah itu, program meminta pengguna untuk memasukkan sebuah bilangan bulat non negatif, yang kemudian disimpan dalam variabel *n*. Lalu program menetapkan variabel *j* dengan nilai yang sama dengan *n*. Dengan while loop, program akan mencetak nilai *j* diikuti dengan mencetak “ x ” selama *j* masih lebih besar dari satu. Setiap kali loop dijalankan, nilai *j* akan dikurangi satu. Setelah loop selesai, program akan mencetak angka satu tanpa tambahan “ x ” di belakangnya.

2. Coso2

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 C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\coso2\coso2.go"
Qwe12312
231234
13213
1231ijwe
12345abcde
Selamat Anda berhasil login
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\coso2\coso2.go"
12345abcde
Selamat Anda berhasil login
```

Deskripsi program:

Program di atas ditulis dalam bahasa Go dan berfungsi sebagai sistem autentikasi sederhana yang meminta pengguna untuk memasukkan sebuah token. Pertama, deklarasikan variabel token bertipe string untuk menyimpan inputan pengguna. Kemudian, program akan meminta pengguna untuk memasukkan token. Selanjutnya, program menggunakan while loop yang akan terus meminta input token dari pengguna selama token yang dimasukkan tidak sama dengan string "12345abcde". Jika pengguna memasukkan token yang benar, yaitu "12345abcde", loop akan berhenti dan program akan mencetak pesan "Selamat Anda berhasil login".

3. Coso3

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 C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\coso3\coso3.go"
5
0 1 1 2 3
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\coso3\coso3.go"
2
0 1
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\coso3\coso3.go"
10
0 1 1 2 3 5 8 13 21 34
```

Deskripsi program:

Program di atas ditulis dalam bahasa Go dan digunakan untuk mencetak deret Fibonacci. Pertama deklarasikan variabel N, s1, s2, j, temp sebagai tipe data integer. Kemudian, program meminta pengguna untuk memasukkan sebuah bilangan bulat positif yang disimpan dalam variabel N. Lalu program menetapkan variabel s1 dengan nilai yang sama dengan 0, program juga menetapkan variabel s2 dengan nilai yang sama dengan 0, angka 0 dan 1 merupakan dua angka pertama dalam deret Fibonacci. Selanjutnya, program menggunakan while loop yang akan berjalan selama jumlah elemen yang dicetak j kurang dari N. Di dalam loop, program mencetak nilai s1, kemudian menghitung angka Fibonacci berikutnya dengan menjumlahkan s1 dan s2, dan menyimpan hasilnya dalam variabel temp. Setelah itu, nilai s1 diperbarui menjadi nilai s2, dan s2 diperbarui menjadi temp. Variabel j juga ditingkatkan satu untuk melanjutkan ke iterasi berikutnya.

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 C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\latsol1\latsol1.go"
● User123 user123
  User admin
  Admin admin
  Admin Admin123
  Admin Admin
  4 Percobaan gagal login
● PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\latsol1\latsol1.go"
  Admin Admin
  0 Percobaan gagal login
```

Deskripsi program:

Program di atas ditulis dalam bahasa Go dan berfungsi untuk mengelola proses login sederhana. Pertama deklarasikan dua variabel bertipe string, yaitu username dan password, yang akan digunakan untuk menyimpan input dari pengguna. Program kemudian menggunakan `fmt.Scan` untuk membaca input dari pengguna. Selanjutnya, program menggunakan while loop yang akan terus berjalan selama input username dan password yang dimasukkan tidak sama dengan “Admin” untuk username dan passwordnya. Setiap kali pengguna memasukkan input username dan password yang salah, variabel percobaan akan bertambah satu, mencatat jumlah percobaan login yang gagal. Setelah pengguna berhasil memasukkan inputan username dan password yang benar, program akan keluar dari loop dan mencetak jumlah percobaan gagal login yang telah dilakukan.

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 C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\latsol2\latsol2.go"
2
2
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\latsol2\latsol2.go"
2544
4
4
5
2
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\latsol2\latsol2.go"
3423554654
4
5
6
4
5
5
3
2
4
3
```

Deskripsi program:

Program di atas ditulis dalam bahasa Go dan berfungsi untuk mencetak setiap digit dari suatu bilangan bulat positif secara terbalik. Pertama deklarasikan variabel bilangan bertipe integer, yang akan menyimpan input dari pengguna. Program kemudian menggunakan `fmt.Scan` untuk membaca nilai bilangan yang dimasukkan. Selanjutnya, program menggunakan while loop yang akan terus berjalan selama nilai bilangan lebih besar dari nol. Di dalam loop, program menghitung digit terakhir dari bilangan dengan menggunakan operasi modulus (`% 10`) dan mencetak digit tersebut. Setelah itu, program menghapus digit terakhir dari bilangan dengan membagi bilangan dengan 10. Proses ini berlanjut hingga semua digit dari bilangan tersebut telah dicetak, sehingga hasilnya adalah setiap digit yang ditampilkan dalam urutan terbalik.

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 C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\latsol3\latsol3.go"
5 2
2
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\latsol3\latsol3.go"
10 7
1
PS C:\semester1_alpro1\alpro1_week12> go run "c:\semester1_alpro1\alpro1_week12\latsol3\latsol3.go"
120 4
30
```

Deskripsi program:

Program di atas ditulis dalam bahasa Go dan berfungsi untuk menghitung hasil bagi dari dua bilangan bulat positif x dan y dengan cara yang manual, yaitu melalui pengurangan berulang. Pertama deklarasikan variabel x dan y bertipe integer yang akan menyimpan input dari pengguna. Program kemudian menggunakan `fmt.Scan` untuk membaca nilai kedua bilangan inputan pengguna. Selanjutnya, program menginisialisasi variabel `hasilBagi` dengan nilai 0, yang akan digunakan untuk menyimpan jumlah berapa kali y dapat dikurangkan dari x. Program kemudian menggunakan `while` loop yang akan terus berjalan selama nilai x lebih besar atau sama dengan y. Di dalam loop, program mengurangi x dengan y dan menambah nilai `hasilBagi` sebanyak satu setiap kali pengurangan dilakukan. Setelah loop selesai, program mencetak nilai `hasilBagi`, yang merepresentasikan 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)