

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



DISUSUN OLEH:
SAVILA NUR FADILLA
103112400031
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

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:

- a.) 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.
- b.) 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. Perulangan while akan terus dijalankan selama kondisinya bernilai benar. Kondisi diperiksa di awal setiap iterasi, jadi jika kondisinya bernilai salah saat perulangan pertama kali dijalankan, badan perulangan tidak akan pernah dijalankan.

While-Loop

While Loop adalah metode perulangan dimana ada kondisi yang harus dipenuhi supaya looping bisa berjalan terus. While Loop mengulangi eksekusi sub diagram didalamnya sampai terminal kondisi menerima nilai Boolean tertentu. Nilai Boolean tergantung dari sifat While Loop. Dengan cara klik kanan pada terminal kondisi dapat dilakukan pemilihan Stop if True atau Continue if True. While Loop akan selalu dieksekusi minimal satu kali. Jumlah pengulangan eksekusi dapat diambil dari terminal iterasi (i).

- Struktur while loop :

```
for condition {  
    // code block  
}
```

Perulangan while di sini mengevaluasi kondisi. Jika kondisinya seperti berikut:

benar: pernyataan dalam loop dijalankan, dan kondisinya dievaluasi ulang.

salah: loop dihentikan.

CONTOH SOAL

- 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.

Source Code:

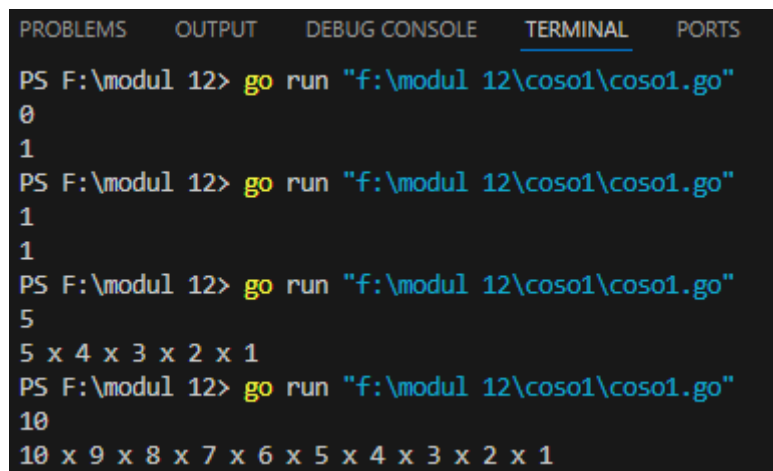
```
package main

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

PS F:\modul 12> go run "f:\modul 12\coso1\coso1.go"
0
1
PS F:\modul 12> go run "f:\modul 12\coso1\coso1.go"
1
1
PS F:\modul 12> go run "f:\modul 12\coso1\coso1.go"
5
5 x 4 x 3 x 2 x 1
PS F:\modul 12> go run "f:\modul 12\coso1\coso1.go"
10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menampilkan deret bilangan faktorial dari suatu bilangan. Program ini meminta kita untuk memasukkan sebuah bilangan bulat non negatif n . Variabel $j = n$ adalah j akan masuk dalam iterasi. Kemudian kita masuk ke struktur while loop. For $j > 1$ artinya perulangan akan terus terjadi selama nilai j lebih dari 1. Perulangan akan terus berjalan selama kondisi ini bernilai true. Jika kondisi tersebut true, program akan mencetak dan menampilkan nilai j diikuti string " x ". Variabel j ($j = j - 1$) akan berkurang 1 di setiap iterasi. Di akhir akan dicetak nilai 1 tanpa string " x " di belakangnya. Keluaran berupa deret bilangan dari faktorial n .

- 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".

Source Code:

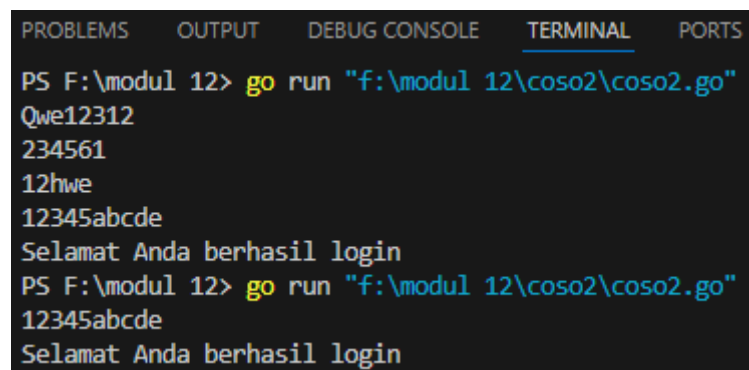
```
package main

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
PS F:\modul 12> go run "f:\modul 12\coso2\coso2.go"
Qwe12312
234561
12hwe
12345abcde
Selamat Anda berhasil login
PS F:\modul 12> go run "f:\modul 12\coso2\coso2.go"
12345abcde
Selamat Anda berhasil login
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk login ke dalam suatu aplikasi dengan asumsi token yang valid adalah "12345abcde". Program ini meminta kita untuk memasukkan suatu token berupa string. Kemudian kita masuk ke struktur while loop. For token != "12345abcde", perulangan akan terus berjalan selama token yang kita input salah atau tidak sama dengan "12345abcde". Perulangan akan terus berjalan selama kondisi ini bernilai true. Jika kondisi bernilai true, program akan terus meminta kita menginput ulang token. Jika nilai token yang diinput sudah benar yaitu "12345abcde" maka perulangan akan berhenti dan program akan mencetak dan menampilkan string yang menyatakan "Selamat Anda berhasil login". Keluaran berupa string tersebut.

- 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.

Source Code:

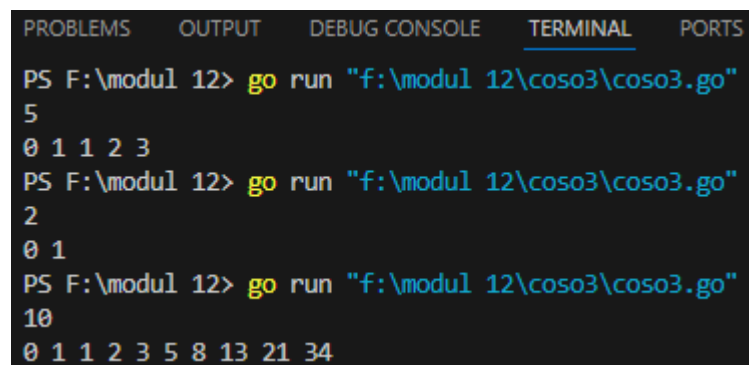
```
package main

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:



The screenshot shows a terminal window with the following content:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\modul 12> go run "f:\modul 12\coso3\coso3.go"
5
0 1 1 2 3
PS F:\modul 12> go run "f:\modul 12\coso3\coso3.go"
2
0 1
PS F:\modul 12> go run "f:\modul 12\coso3\coso3.go"
10
0 1 1 2 3 5 8 13 21 34
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan mencetak N bilangan pertama dalam deret Fibonacci. Program ini meminta kita untuk memasukkan bilangan bulat positif N dengan nilai lebih besar atau sama dengan 2. Variabel $s1 = 0$ (diinisialisasi dengan 0, bilangan pertama fibonacci). Variabel $s2 = 1$ (diinisialisasi dengan 1, bilangan kedua fibonacci). Variabel temp digunakan untuk menghitung bilangan fibonacci berikutnya. Kemudian kita masuk ke struktur while loop. For $j < N$, perulangan akan terus berjalan selama nilai j kurang dari N. Perulangan akan terus berjalan selama kondisi ini bernilai true. Jika kondisi bernilai true, program akan mencetak s1 diikuti spasi (s1, " "). Lalu $temp = s1 + s2$, program akan menghitung bilangan fibonacci berikutnya. Setelah dihitung, program akan memperbarui nilai s1 dan s2, dengan $s1 = s2$ yang artinya s1 menjadi bilangan fibonacci saat ini (s2) dan $s2 = temp$ artinya s2 menjadi bilangan fibonacci berikutnya. Variabel j ($j = j + 1$) akan terus bertambah 1 di setiap iterasi. Keluaran terdiri dari sejumlah N bilangan yang menyatakan N deret bilangan fibonacci yang pertama.

SOAL LATIHAN

- 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.

Source Code:

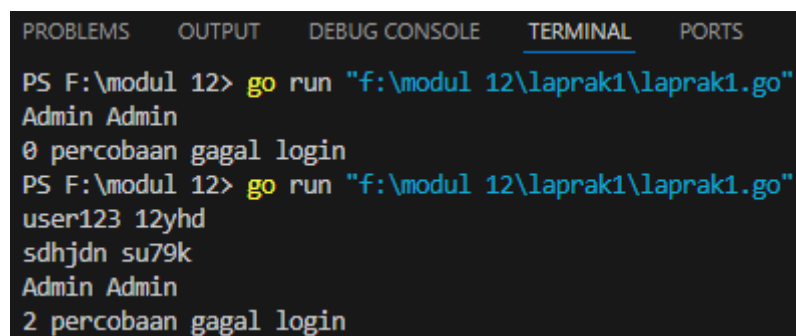
```
package main

import "fmt"

func main() {
    var usr, pass string
    fmt.Scan(&usr, &pass)

    i := 0
    for usr != "Admin" || pass != "Admin" {
        fmt.Scan(&usr, &pass)
        i++
    }
    fmt.Printf("%v percobaan gagal login", i)
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS F:\modul 12> go run "f:\modul 12\laprak1\laprak1.go"
Admin Admin
0 percobaan gagal login
PS F:\modul 12> go run "f:\modul 12\laprak1\laprak1.go"
user123 12yhd
sdhjdn su79k
Admin Admin
2 percobaan gagal login
```


Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan digunakan untuk menghitung berapa banyak seseorang pengguna gagal melakukan login karena kesalahan memberikan username dan password. Program ini meminta kita untuk memasukkan dua teks yang berisi username dan password. Variabel `i := 0` digunakan untuk menghitung perulangan dimana nilai awalnya 0. Kemudian kita masuk ke struktur `while loop`. `For usr != "Admin" || pass != "Admin"`, perulangan akan terus berjalan selama username atau password yang kita input salah atau tidak sama dengan "Admin". Perulangan akan terus berjalan selama kondisi ini bernilai `true`. Jika kondisi bernilai `true`, program akan meminta kita untuk menginput ulang username dan password. Variabel `i (i++)` akan bertambah 1 di setiap iterasi atau setiap kali percobaan login gagal. Keluaran berupa string yang menyatakan berapa kali pengguna gagal login, `"%v percobaan gagal login"`.

- 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).

Source Code:

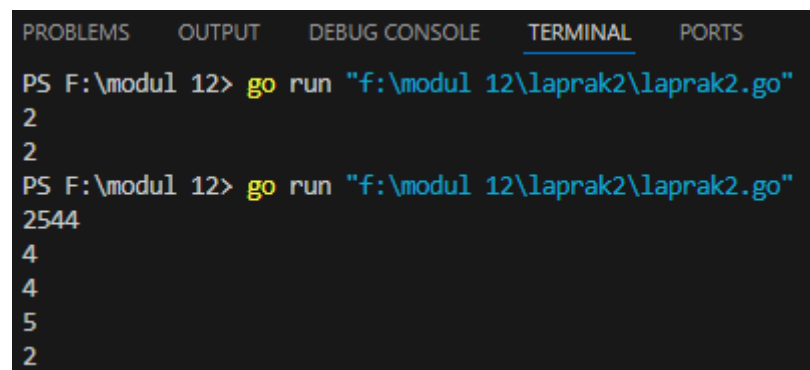
```
package main

import "fmt"

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

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

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\modul 12> go run "f:\modul 12\laprak2\laprak2.go"
2
2
PS F:\modul 12> go run "f:\modul 12\laprak2\laprak2.go"
2544
4
4
5
2
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk mencacah setiap digit yang terdapat di dalam suatu bilangan bulat positif. Program ini meminta kita untuk memasukkan suatu bilangan bulat positif. Kemudian kita masuk ke struktur while loop. For bilangan > 0, perulangan akan terus berjalan selama bilangan lebih besar dari 0. Perulangan akan terus berjalan selama kondisi ini bernilai true. Jika kondisi bernilai true, program akan menghitung digit = bilangan % 10, mengambil sisa pembagian bilangan dengan 10 (mengambil digit terakhir dari bilangan). Kemudian program akan mencetak digit. Lanjut ke bilangan = bilangan / 10, membagi bilangan dengan 10 (menghapus digit terakhir dari bilangan). Keluaran berupa nilai digit pada bilangan tersebut. Tampilan dari digit terakhir (paling kanan) sampai dengan digit pertama (paling kiri).

- 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 $x \text{ div } y$.

Source Code:

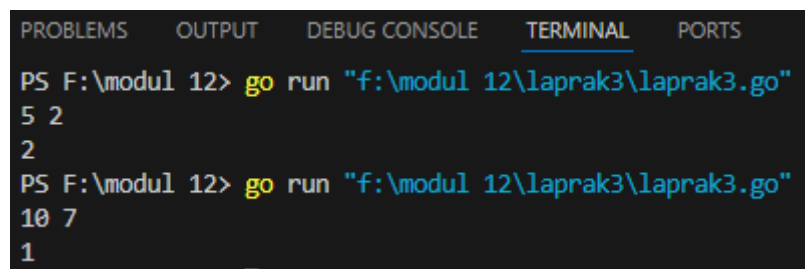
```
package main

import "fmt"

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

    for x >= y {
        x = x - y
        i++
    }
    fmt.Println(i)
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS F:\modul 12> go run "f:\modul 12\laprak3\laprak3.go"
5 2
2
PS F:\modul 12> go run "f:\modul 12\laprak3\laprak3.go"
10 7
1
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk mencari hasil integer division dari dua bilangan tanpa menggunakan operator pembagian. Program meminta kita untuk memasukkan dua bilangan bulat positif x dan y, yang mana $x \geq y$. Kemudian kita masuk ke struktur while loop. For $x \geq y$, perulangan akan terus berjalan selama nilai x lebih besar atau sama dengan y. Perulangan akan terus berjalan selama kondisi ini bernilai true. Jika kondisi bernilai true, program akan melanjutkan menghitung operasi pengurangan $x = x - y$, mengurangi nilai x dengan y di setiap iterasi. Variabel i ($i++$) akan terus bertambah 1 di setiap iterasi. Keluaran berupa i atau hasil dari operasi $x \text{ div } y$.

DAFTAR PUSTAKA

Amisha. (2023). *While Loop in Golang*. <https://www.scaler.com/topics/golang/while-loop-in-golang/>

pythonhello. (2022). *While Loop in Python (with best practices)*.
<https://www.pythonhello.com/fundamentals/python-while-loops>

Utama, J. (2020). *While Loops*.
[https://repository.unikom.ac.id/62967/1/Materi%20Pertemuan%204 Labview%201%20BWhile%20Loop%20%2B%20Shift%20Register.pdf](https://repository.unikom.ac.id/62967/1/Materi%20Pertemuan%204%20Labview%201%20BWhile%20Loop%20%2B%20Shift%20Register.pdf)

Prasti Eko Yunanto, S. T. , M. Kom. (2004). *MODUL PRAKTIKUM 12 – WHILE-LOOP ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA* .