

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 12

WHILE-LOOP



Disusun oleh:

HAFIZD SAMA'I SYAMSI

109082500183

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

1. Guided 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)
}
```

Screenshot program

The screenshot shows a code editor interface with several tabs at the top: Welcome, guided1.go 5, guided2.go 1, guided3.go 1, soal1.go 1, and another unnamed tab. The main editor area displays the following Go code:

```
5 package main
1 import "fmt"
1 func main() {
1     var n, j int
1     fmt.Scan(&n)
1     j = n
1     for j > 1 {
1         fmt.Print(j, " x ")
1         j = j - 1
1     }
1     fmt.Println(1)
1 }
```

Below the editor, there are tabs for PROBLEMS (11), OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the following command-line session:

```
PS D:\go\praktek12> go run guided1.go
0
1
PS D:\go\praktek12> go run guided1.go
5
5 x 4 x 3 x 2 x 1
PS D:\go\praktek12> go run guided1.go
10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
PS D:\go\praktek12> []
```

Deskripsi program

Cara kerjanya:

1. Program membaca satu angka dari pengguna dan menyimpannya ke variabel n.
2. Nilai n disalin ke variabel j sebagai penghitung mundur.
3. Program menjalankan perulangan selama j masih lebih besar dari 1.
 - o Pada setiap langkah, program mencetak nilai j diikuti teks " x " agar terlihat seperti operasi perkalian.
 - o Setelah mencetak, nilai j dikurangi 1.
4. Ketika perulangan selesai (j sudah 1), program mencetak angka 1 sebagai penutup deretan.

2. Guided 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")
}
```

Screenshot program

```
-GO guided2.go > main
1 package main
2 import "fmt"
3 func main() {
4     var token string
5     fmt.Scan(&token)
6     for token != "12345abcde" {
7         fmt.Scan(&token)
8     }
9     fmt.Println("Selamat Anda berhasil login")
10}
11
```

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\go\praktek12> go run guided2.go
12345
abcde
12345abcde
Selamat Anda berhasil login
PS D:\go\praktek12> 
```

Deskripsi program

Program Go ini berfungsi sebagai pemeriksa token login sederhana. Program akan terus meminta pengguna memasukkan token sampai token yang diberikan sesuai dengan token yang benar, yaitu "12345abcde". Jika token sudah cocok, program menampilkan pesan bahwa login berhasil.

- Deklarasi variable
Token bertipe string, digunakan untuk menyimpan input dari pengguna
- Input pertama
Program membaca satu token awal dari pengguna menggunakan `fmt.Scan(&token)`.

- Perulangan pengecekan
 - Program memakai perulangan for dengan kondisi token != "12345abcde"
 - Artinya, selama token yang dimasukan bukan "12345abcde", program akan terus meminta input baru.
 - Pada setiap perulangan, pengguna diminta mengisi token Kembali
- Jika token cocok
 - Ketika pengguna akhirnya masukkan "12345abcde", kondisi token != menjadi false, sehingga perulangan berhenti

3. Guided 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.Println(s1, " ")
        temp = s1 + s2
        s1 = s2
        s2 = temp
        j = j + 1
    }
}
```

Screenshot program

The screenshot shows a code editor with several tabs at the top: 'Welcome', 'guided1.go 5', 'guided2.go 1', 'guided3.go 1' (which is the active tab), and 'soal1.go 1'. The code in 'guided3.go' is as follows:

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

Below the code editor is a terminal window showing the execution of the program:

```
PS D:\go\praktek12> go run guided3.go
5
0 1 1 2 3
PS D:\go\praktek12> go run guided3.go
2
0 1
PS D:\go\praktek12> go run guided3.go
10
0 1 1 2 3 5 8 13 21 34
PS D:\go\praktek12>
```

Deskripsi program

Program Go ini digunakan untuk menampilkan deret Fibonacci sebanyak N suku, di mana nilai N dimasukkan oleh pengguna. Deret Fibonacci adalah urutan bilangan di mana setiap suku merupakan hasil penjumlahan dua suku sebelumnya, dimulai dari 0 dan 1.

1. Deklarasi variable

- N → Jumlah suku deret yang ingin ditampilkan
- s1 → menyimpan suku pertama (dimulai dari 0)
- s2 → menyimpan suku kedua (dimulai dari 1)
- j → sebagai penghitung berapa banyak suku yang dicetak
- Temp → variable untuk menampung penjumlahan sementara

2. Input

Program membaca nilai N dari pengguna

3. Inisialisasi awal

- $s1 = 0$
- $s2 = 1$
- $J = 0$

Nilai ini sesuai dengan dua suku pertama deret Fibonacci

4. Perulangan

Program ini menjalankan `for j < n` yang artinya terus mencetak suku sampai jumlahnya mencapai N.

Di dalam loop:

- Program ini mencetak nilai $s1$
- Menghitung suku berikutnya: $\text{temp} = s1 + s2$
- Pergeseran nilai:
 - $s1$ diisi nilai $s2$
 - $s2$ diisi nilai temp
- j ditambah 1 sebagai penghitung suku

5. Output

program menghasilkan N bilangan Fibonacci secara berurutan dimulai dari 0

TUGAS

1. Tugas 1

Source code

```
package main

import (
    "fmt"
)

func main() {
    var username, password string
    var gagal int
    gagal = 0
```

```

for {

    fmt.Scan(&username, &password)

    if username == "Admin" && password == "Admin" {

        break

    } else {

        gagal++

    }

}

fmt.Printf("%d percobaan gagal login\n", gagal)
}

```

Screenshoot program

The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for various files: Welcome, guided1.go, guided2.go, guided3.go, soal1.go (which is the active file), soal2.go, soal3.go, and main.go. The code in soal1.go is as follows:

```

func main() {
    var gagal int
    gagal = 0
    for {
        fmt.Scan(&username, &password)
        if username == "Admin" && password == "Admin" {
            break
        } else {
            gagal++
        }
    }
    fmt.Printf("%d percobaan gagal login\n", gagal)
}

```

Below the code editor is a terminal window with the following output:

```

PS D:\go\praktek12> go run soal1.go
User123 user123
User admin
Admin admin
Admin admin123
Admin Admin
4 percobaan gagal login
PS D:\go\praktek12> go run soal1.go
Admin Admin
0 percobaan gagal login
PS D:\go\praktek12>

```

To the right of the terminal is a small window displaying personal information:

```

NAMA : HAFIZD SAMA'I SYAMSI
KELAS : S1IF-13-07
NIM : 109082500183

```

Deskripsi program

Program Go ini berfungsi sebagai **simulator login sederhana** yang menghitung berapa kali pengguna gagal memasukkan username dan password sebelum akhirnya berhasil login. Program akan terus meminta input pasangan username password tanpa batas sampai pengguna memasukan kombinasu yang benar:

Username = “Admin” dan password = “Admin”

- Deklarasi variable
 - Username dan password bertipe string untuk menyimpan input pengguna
 - Gagal bertipe integer, digunakan untuk mencatat jumlah percobaan login yang salah
 - Gagal diinsialisasi dengan nilai 0
- Perulangan tak hingga (for {})
Program masuk ke loop yang berjalan terus menerus sampai kondisi break ditemukan
- Input login
Pada setiap iterasi program membaca dua input:
username password
- Pengecekan login
 - Jika username dan password cocok dengan “Admin”
program menjalankan break untuk keluar dari loop → artinya gagal
 - Jika Salah satu atau keduanya salah, variable gagal ditambah 1 → login dinyatakan gagal
- Akhir program
Setelah keluar dari loop (karena berhasil login), program ini mencetak jumlah percobaan gagal

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var bil int

    fmt.Scan(&bil)

    for bil > 0 {

        bill := bil % 10

        fmt.Println(bill)

        bil = bil / 10

    }

}
```

Screenshot program

The screenshot shows a Go development environment with the following components:

- Editor Area:** Displays the file `soal2.go` with the following code:

```
1 package main
2
3 func main() {
4     var bil int
5     fmt.Scan(&bil)
6     for bil > 0 {
7         bil1 := bil % 10
8         fmt.Println(bil1)
9         bil = bil / 10
10    }
11 }
```
- Terminal Area:** Shows the command `go run soal2.go` being run and its output:

```
PS D:\go\praktek12> go run soal2.go
3423554654
4
5
6
4
5
5
3
2
4
3
PS D:\go\praktek12>
```
- Status Bar:** Shows the following information:
 - Ln 3, Col 19 | 64 characters
 - Plain text
 - 100%

Deskripsi program

Program Go ini digunakan untuk memecah sebuah bilangan menjadi digit-digit penyusunnya, lalu menampilkannya satu per satu mulai dari digit paling belakang. Program bekerja dengan cara mengambil digit terakhir, mencetaknya, kemudian menghapus digit tersebut sampai tidak ada angka yang tersisa.

- Input bilangan
Program membaca sebuah bilangan dan menyimpan dalam variabel `bil`.
- Perulangan
Kondisi `for bil > 0` membuat program berjalan terus selama masih ada digit dalam bilangan
- Mengambil digit terakhir
 - `bil1 := bil % 10`
operator `%` (modulus) mengambil digit paling kanan dari bilangan
- menampilkan digit

- program menampilkan digit tersebut menggunakan fmt.Println(bil1)
- menghapus digit paling belakang
 - bil = bil / 10
pembagian bilangan bulat ini membuang digit paling kanan, sehingga program bisa memproses digit berikutnya.
- Proses berulang
Langkah-langkah tersebut diulang sampai bil menjadi 0

3. Tugas 3

Source code

```
package main
import (
    "fmt"
)
func main() {
    var x, y int
    var hasil int
    hasil = 0
    fmt.Scan(&x, &y)
    for x >= y {
        x = x - y
        hasil++
    }
    fmt.Println(hasil)
}
```

Screenshot program

The screenshot shows a Microsoft Visual Studio Code window. On the left, the code editor displays a Go file named 'soal3.go'. The code defines a package 'main' with a function 'main()' that performs integer division by repeated subtraction. The terminal tab at the bottom shows the execution of the program multiple times with different inputs, resulting in outputs like 5 2, 10 7, 120 30, and 120 1.

```
5 2
2
PS D:\go\praktek12> go run soal3.go
10 7
1
PS D:\go\praktek12> go run soal3.go
120
30
4
PS D:\go\praktek12> go run soal3.go
5 2
2
PS D:\go\praktek12> go run soal3.go
10 7
1
PS D:\go\praktek12> go run soal3.go
120
1
```

Deskripsi program

Program Go ini melakukan pembagian bilangan bulat menggunakan metode pengurangan berulang (integer division by repeated subtraction). Program tidak menggunakan operator pembagian (/), tetapi menghitung berapa kali y dapat dikurangkan dari x hingga x menjadi lebih kecil dari y. Nilai itulah yang menjadi hasil pembagian.

- Deklarasi variable
 - x dan y adalah dua bilangan bulat yang menjadi input
 - hasil digunakan untuk menghitung berapa kali pengurangan terjadi, diinisialisasi dengan 0
- input
 - program membaca dua bilangan dari pengguna:
fmt.Scan(&x, &y)
- perulangan pembagian
 - kondisi dor x >= y berarti program terus mengurangi x dengan y selama x masih cukup besar.
 - Setiap kali x = x - y, artinya satu “unit pembagian” terjadi.
 - Variable hasil ditambah 1 untuk menghitung jumlah pengurangan.

- Akhir perulangan
Ketika $x < y$, perulangan berhenti. Pada kondisi ini
 - Hasil berisi hasil pembagian bulat x/y
 - Sisa pembagian adalah nilai x terakhir (meskipun tidak dicetak pada program ini)
- Output
Program menampilkan nilai hasil, yaitu hasil pembagian bulat