

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 12

WHILE-LOOP



Disusun oleh:

ELSA DWI RIZQIYANTI

109082500090

S1IF-13-02

Asisten Praktikum

Adithana dharma putra

Alfin Ilham Berlianto

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 int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)

    if n == 0 || n == 1 {
        fmt.Println("1")
        return
    }

    i := n
    for i >= 1 {
        if i == 1 {
            fmt.Print("1")
        } else {
            fmt.Print(i, " x ")
        }
        i--
    }
}
```

```
}
```

Screenshoot program

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

- File Explorer:** Shows a file named "guide1.go" under a folder "MODUL 12".
- Code Editor:** Displays the content of "guide1.go":

```
package main
import "fmt"
func main() {
    var n int
    fmt.Println("Masukkan bilangan: ")
    fmt.Scan(&n)

    if n == 0 || n == 1 {
        fmt.Println("1")
        return
    }

    i := n
    for i >= 1 {
        if i == 1 {
            fmt.Print("1")
        } else {
            fmt.Print(i, " x ")
        }
    }
}
```
- Terminal:** Shows the command "go run guide1.go" being run multiple times, resulting in the output:

```
Masukkan bilangan: 0
1
Masukkan bilangan: 5
5 x 4 x 3 x 2 x 1
Masukkan bilangan: 10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
Masukkan bilangan: 1
1
```
- Output Window:** Shows the output "Elsa Dwi Rizziyanti 109082500094".
- AI Assistant:** A sidebar titled "Ask about your code" with the message "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase." and a "Ask" button.
- System Tray:** Shows icons for battery (20°C), network, and date/time (01/12/2025).

Deskripsi program

```
if n == 0 || n == 1 {
    fmt.Println("1")
    return
}
```

Jika pengguna memasukkan 0 atau 1, program langsung mencetak 1.

Kemudian program menghentikan eksekusi dengan return.

Ini dibuat karena perkalian mundur dari 1 sampai 1 atau 0 akan menghasilkan nilai 1 (mirip konsep faktorial).

i := n

Membuat variabel i dan mengisinya dengan nilai n.

Variabel i digunakan sebagai counter dalam perulangan.

for i >= 1 {

Ini adalah while-loop gaya Go. Loop akan berjalan selama i >= 1.

Bagian dalam loop:

```
if i == 1 {
    fmt.Println("1")
```

```
    } else {
        fmt.Println(i, " x ")
    }
    i--
```

Penjelasannya:

1. Jika $i == 1$
Cetak hanya angka 1 tanpa tambahan " x " di belakangnya
(supaya output tidak berakhir dengan simbol " x ")
2. Jika $i > 1$

$i \times$

Contoh:

$5 \times 4 \times 3 \times 2 \times 1$

$i --$

Mengurangi nilai i satu per satu hingga mencapai 1.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    const tokenBenar = "12345abcde"
    var input string

    fmt.Print("Masukkan token: ")
    fmt.Scan(&input)

    for input != tokenBenar {
        fmt.Print("Masukkan token: ")
        fmt.Scan(&input)
    }
}
```

```

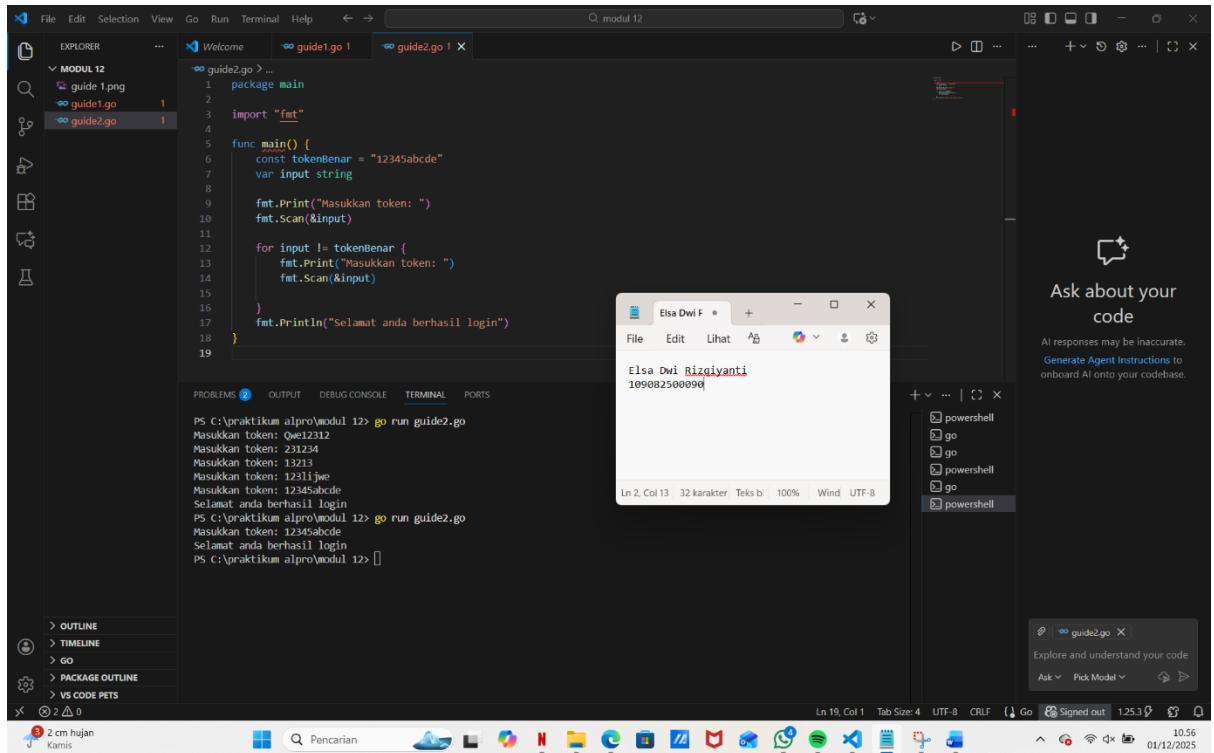
}

fmt.Println("Selamat anda berhasil login")

}

```

Screenshot program



Deskripsi program

`const tokenBenar = "12345abcde"`

- Membuat konstanta bernama tokenBenar.
- Konstanta adalah nilai yang *tidak bisa diubah* selama program berjalan.
- Token ini adalah nilai yang harus dimasukkan oleh pengguna agar login berhasil.

Perulangan (While-loop versi Go)

```

for input != tokenBenar {

    fmt.Println("Masukkan token: ")

    fmt.Scan(&input)

}

```

Penjelasan:

- for tanpa kondisi khusus seperti `{} bisa berfungsi sebagai while-loop.`

- Kondisi loop: input != tokenBenar
artinya: *ulang terus selama input tidak sama dengan token yang benar*
- Jika pengguna salah memasukkan token, loop akan:
 - meminta input ulang
 - membaca input baru
 - memeriksa lagi

Loop berhenti hanya ketika input == tokenBenar.

```
fmt.Println("Selamat anda berhasil login")  
Jika keluar dari loop, artinya token sudah benar.  
Program kemudian mencetak pesan sukses login.
```

3. Guided 3

Source Code

```
package main  
  
import "fmt"  
  
func main() {  
    var n int  
    fmt.Print("Masukkan N: ")  
    fmt.Scan(&n)  
  
    a := 0  
    b := 1  
  
    fmt.Print(a, " ", b)
```

```

i := 2

for i < n {

    c := a + b

    fmt.Println(" ", c)

    a = b

    b = c

    i++

}

}

```

Screenshot program

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "MODUL 12" containing files: guide1.png, guide2.png, guide3.png, guide1.go (2), guide2.go (1), and guide3.go (1).
- Code Editor:** Displays the content of guide3.go:

```

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

    a := 0
    b := 1

    fmt.Println(a, " ", b)

    i := 2

    for i < n {
        c := a + b
        fmt.Println(" ", c)

        a = b
        b = c
        i++
    }
}

```
- Terminal:** Shows command-line output for running the program:

```

PS C:\praktikum\alpro\modul 12> go run guide3.go
Masukkan N: 5
0 1 1 2 3
PS C:\praktikum\alpro\modul 12> go run guide3.go
Masukkan N: 2
0 1
PS C:\praktikum\alpro\modul 12> go run guide3.go
Masukkan N: 10
0 1 1 2 3 5 8 13 21 34
PS C:\praktikum\alpro\modul 12>

```
- AI Sidebar:** An "Ask about your code" sidebar is open, suggesting "powershell" and "go". It also includes a link to "Generate Agent Instructions to onboard AI onto your codebase".

Deskripsi program

Inisialisasi dua angka pertama Fibonacci

a := 0

b := 1

Deret Fibonacci selalu dimulai dari:

angka pertama = 0

angka kedua = 1

Variabel:

a menyimpan nilai Fibonacci sebelumnya

b menyimpan nilai Fibonacci saat ini

Menampilkan dua bilangan pertama

fmt.Println(a, " ", b)

Akan menghasilkan output seperti:

0 1

Ini adalah dua angka pertama deret Fibonacci.

Inisialisasi counter (i)

i := 2

Kenapa 2?

Karena deret Fibonacci sudah mencetak dua angka pertama ("0 1"), maka perhitungan berikutnya dimulai dari indeks ke-2.

Perulangan While-loop Fibonacci

for i < n {

 c := a + b

 fmt.Println(" ", c)

 a = b

 b = c

 i++

}

Ini adalah while-loop versi Go.

Penjelasan langkah demi langkah:

Kondisi loop:

i < n

Artinya, loop akan terus berjalan sampai jumlah bilangan yang dicetak mencapai N.

Menghitung bilangan Fibonacci berikutnya

c := a + b

- c adalah angka Fibonacci baru.

- Rumus Fibonacci:

bilangan berikutnya = dua bilangan sebelumnya dijumlahkan

Menampilkan bilangan ke-i
fmt.Println(" ", c)
Mencetak angka Fibonacci baru.

Geser nilai (update variabel)

a = b

b = c

Pemindahan nilai ini membuat:

- nilai b menjadi nilai Fibonacci sebelumnya pada iterasi selanjutnya
 - nilai c menjadi nilai Fibonacci saat ini
- Ini penting agar perhitungan Fibonacci terus berlanjut.

Increment counter

i++

Menambah nilai i agar loop mendekati kondisi berhenti.

Hasil Output Contoh

Jika input N = 7

Maka output:

0 1 1 2 3 5 8

Karena itu adalah 7 bilangan pertama Fibonacci.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var user, pass string
    gagal := 0

    for {
        fmt.Scan(&user, &pass)

        if user == "Admin" && pass == "Admin" {
            break
        }

        gagal++
    }

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

Screenshot program

The screenshot shows a Go development environment in a Windows operating system. On the left, the code editor displays a file named 'soal1.go' with the following content:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var user, pass string
7     gagal := 0
8
9     for {
10         fmt.Scan(&user, &pass)
11
12         if user == "Admin" && pass == "Admin" {
13             break
14         }
15
16         gagal++
17     }
18
19     fmt.Printf("%d percobaan gagal login\n", gagal)
20 }
21
```

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

```
PS C:\praktikum\apro\modul 12> go run soal1.go
User123 user123
User admin
Admin admin
Admin Admin123
Admin Admin
4 percobaan gagal login
PS C:\praktikum\apro\modul 12> go run soal1.go
Admin Admin
0 percobaan gagal login
PS C:\praktikum\apro\modul 12>
```

To the right of the terminal is a small window titled 'Elsa Dwi R' displaying the user information:

Elsa Dwi Rizqiyanti
109082500099

The taskbar at the bottom shows various pinned icons, including a weather icon for 'Berawan' (25°C) and several browser and utility icons.

Deskripsi program

Input username dan password

fmt.Scan(&user, &pass)

Fungsi ini membaca dua input sekaligus dari user:

1. username
2. password

Contoh input:

Admin Admin

Mengecek apakah login benar

if user == "Admin" && pass == "Admin" {

 break

}

Program melakukan pengecekan:

- username harus "Admin"
- password harus "Admin"

Jika kedua kondisi benar, maka:

- break menghentikan loop
- proses login selesai

Jika login salah: tambah hitungan gagal

gagal++

Setiap kali user memasukkan kombinasi yang salah, variabel gagal bertambah 1.

Setelah keluar dari loop: tampilkan jumlah kesalahan

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

Menampilkan jumlah berapa kali login salah dilakukan sebelum akhirnya berhasil.

Contoh output:

4 percobaan gagal login

2. Tugas 2

Source code

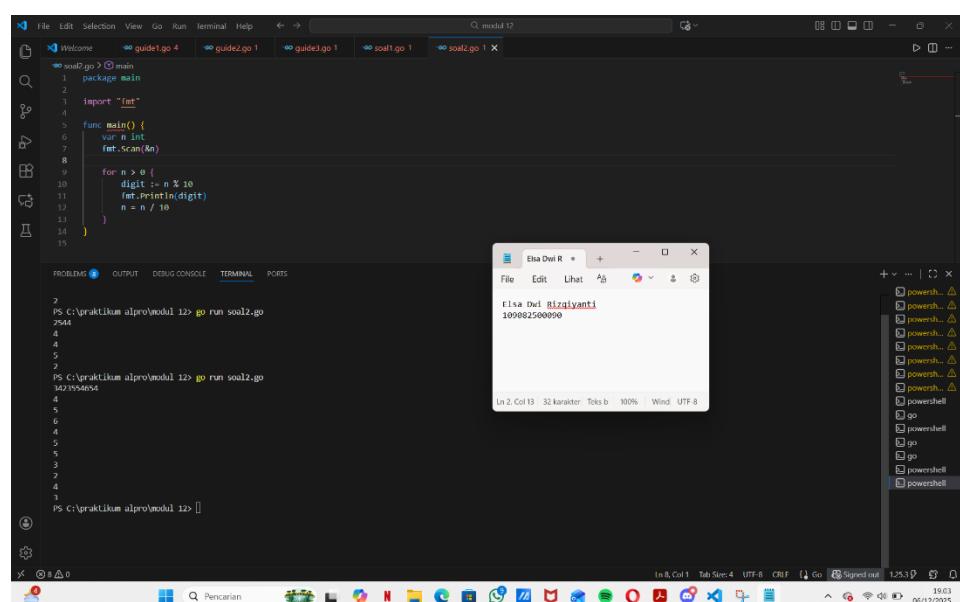
```
package main

import "fmt"

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

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

Screenshot program



Deskripsi program

Perulangan while-loop

for n > 0 {

Loop ini berjalan selama nilai n masih lebih besar dari 0.

Loop akan berhenti ketika n telah habis dibagi-bagi sehingga menjadi 0.

Mengambil digit terakhir

digit := n % 10

Operator modulus % 10 digunakan untuk mengambil digit paling belakang dari angka.

Contoh:

- n = 2544 digit = 4
- n = 254 digit = 4
- n = 25 digit = 5

Menampilkan digit

fmt.Println(digit)

Digit yang baru diambil ditampilkan satu per baris.

Menghapus digit terakhir

n = n / 10

Pembagian integer (/10) membuang digit paling belakang.

Contoh:

- n = 2544 254
- n = 254 25
- n = 25 2
- n = 2 0

Ketika n menjadi 0, loop berhenti.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var x, y int
    var hasil int

    fmt.Scan(&x, &y)
```

```

for x >= y {

    x = x - y

    hasil = hasil + 1

}

fmt.Println(hasil)
}

```

Screenshot program

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

- EXPLORER:** Shows a module named "MODUL 12" containing files: guide1.png, guide2.png, guide3.png, guide1.go, guide2.go, guide3.go, soal1.png, soal2.png, soal1.go, soal2.go, and soal3.go.
- CODE EDITOR:** Displays the content of soal3.go:

```

1 package main
2 import "fmt"
3 func main() {
4     var x, y int
5     var hasil int
6     fmt.Scan(&x, &y)
7     for x >= y {
8         x = x - y
9         hasil = hasil + 1
10    }
11    fmt.Println(hasil)
12}
13
14
15
16
17
18

```
- TERMINAL:** Shows command-line output for running the program:

```

PS C:\praktikum\alpro\modul 12> go run soal3.go
5 2
2
PS C:\praktikum\alpro\modul 12> go run soal3.go
10 7
1
PS C:\praktikum\alpro\modul 12> go run soal3.go
120
[]

```
- FILE VIEWER:** A separate window titled "Elsa Dwi Rizqiyanti" showing the content of soal3.go.

Deskripsi program

Perulangan while-loop (pembagian manual)

```

for x >= y {

    x = x - y

    hasil = hasil + 1

}

```

Kondisi loop

$x \geq y$

Loop terus berjalan selama x masih bisa dikurangi dengan y .

Ini meniru pembagian tanpa operator.

$x = x - y$

Setiap iterasi, program mengurangi x dengan y .

Ini seperti menghitung berapa kali y muat di dalam x .

hasil = hasil + 1

Setiap kali pengurangan dilakukan, itu berarti pembagi (y) masuk sekali ke dalam x .

Maka hasil bertambah 1.

Menampilkan hasil

fmt.Println(hasil)

Setelah loop berhenti, hasil berisi hasil pembagian x div y .

Input:

10 3

Proses:

- $10 \geq 3: 10 - 3 = 7$ hasil = 1
- $7 \geq 3: 7 - 3 = 4$ hasil = 2
- $4 \geq 3: 4 - 3 = 1$ hasil = 3
- $1 < 3$: stop