

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 12**

**WHILE-LOOP**



**Disusun oleh:**

**MUHAMMAD NAUFAL AKMAL PEABOWO**

**109082500186**

**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 i int

    fmt.Scan(&i)

    hasil := 1

    for i >= 1 {
        fmt.Println(i)
        if i != 1 {
            fmt.Println(" x ")
        }
        hasil *= i
        i--
    }

    fmt.Println(" =", hasil)
}
```

#### Screenshot program

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

- Editor:** Displays a Go program named `moduli.go` with code for calculating factorials.
- Terminal:** Shows command-line output of the program running, displaying the factorial calculation process and the final result.
- File Viewer:** A separate window showing a file named `WOT` containing personal information.

```

package main

import "fmt"

func main() {
    var i int
    fmt.Scan(&i)
    hasil := 1
    for i >= 1 {
        fmt.Println(i)
        if i != 1 {
            fmt.print(" x ")
        }
        hasil *= i
        i--
    }
    fmt.Println(" = ", hasil)
}

```

```

PS C:\Users\naufal.akmal\OneDrive\Documents\WOT> go run moduli.go
0
= 1
PS C:\Users\naufal.akmal\OneDrive\Documents\WOT> go run moduli.go
5
5 x 4 x 3 x 2 x 1 = 120
PS C:\Users\naufal.akmal\OneDrive\Documents\WOT> go run moduli.go
10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 = 3628800
PS C:\Users\naufal.akmal\OneDrive\Documents\WOT> go run moduli.go
1
1 = 1
PS C:\Users\naufal.akmal\OneDrive\Documents\WOT>

```

## Deskripsi program

Program Go pada gambar tersebut merupakan kode sederhana untuk menghitung nilai faktorial dari sebuah angka yang dimasukkan oleh pengguna. Program dimulai dengan membaca input berupa bilangan integer, kemudian melakukan perhitungan faktorial menggunakan perulangan `for` secara menurun dari nilai input hingga mencapai 1. Setiap langkah perhitungan ditampilkan ke layar dalam format perkalian berurutan (misalnya “ $5 \times 4 \times 3 \times 2 \times 1$ ”), sambil mengalikan nilai hasil dengan angka saat ini. Setelah perulangan selesai, program mencetak hasil akhir dari faktorial tersebut. Program ini menunjukkan penggunaan dasar struktur kontrol, perulangan, input-output, serta operasi aritmatika dalam bahasa pemrograman Go.

## Guided 2

### Source Code

```

package main

import "fmt"

func main() {
    var token string

    tokenValid := "12345abcde"

    fmt.Print("Masukan Token: ")
    fmt.Scan(&token)

    for token != tokenValid {
        fmt.Print("Masukan Token: ")
    }
}

```

```

        fmt.Scan(&token)
    }
    fmt.Println("Selamat anda berhasil login")
}

```

## Screenshot program

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

- EXPLORER:** Shows files and folders: COBA1.GO, COBA2.GO, COBA3.GO, and WOI. WOI contains sub-files like modul1.go, modul2.go, and modul3.go.
- EDITOR:** Displays the main.go file content. The code asks for a token, checks if it's valid (12345abcde), and prints a success message if valid.
- TERMINAL:** Shows command-line interactions:
  - PS C:\Users\Naufal\akmal\OneDrive\Documents\WOI> go run modul2.go
 Masukan Token: Qwerty123
 Masukan Token: 231234
 Masukan Token: 13213
 Masukan Token: 12313jwe
 Masukan Token: 12345abcde
 Selamat anda berhasil login
  - PS C:\Users\Naufal\akmal\OneDrive\Documents\WOI> go run modul2.go
 Masukan Token: 12345abcde
 Selamat anda berhasil login
  - PS C:\Users\Naufal\akmal\OneDrive\Documents\WOI>
- OUTPUT:** Shows the terminal output from the last command.
- FILE VIEWER:** A separate window shows a file named "NAMA" with content "NAMA : M.NAUFAL AKMAL PRABONO" and "NIM : 109082500186".

## Deskripsi program

Program Go pada gambar tersebut merupakan simulasi sederhana proses login menggunakan token, di mana pengguna diminta memasukkan sebuah string yang harus sesuai dengan token valid yang telah ditentukan sebelumnya, yaitu "12345abcde". Program pertama-tama meminta input token dari pengguna, kemudian mengecek apakah input tersebut cocok dengan token valid; jika tidak, program akan terus mengulang permintaan input melalui perulangan `for token != tokenValid`, sehingga pengguna harus memasukkan token yang benar agar dapat melanjutkan. Setelah pengguna berhasil memasukkan token yang sesuai, program menampilkan pesan "Selamat anda berhasil login" sebagai tanda bahwa proses autentikasi berhasil. Program ini memperlihatkan penggunaan variabel string, input-output, perulangan, serta validasi sederhana dalam bahasa Go.

## Guided 3

### Source Code

```

package main

import "fmt"

func main() {
    var n int

```

```

fmt.Scan(&n)
a, b := 0, 1
i := 0

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

}

```

## Screenshoot program

The screenshot shows a code editor interface with several tabs open. The active tab is 'modul3.go' which contains the following Go code:

```

package main

import "fmt"

func main() {
    var n int

    fmt.Scan(&n)
    a, b := 0, 1
    i := 0

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

```

Below the code editor is a terminal window showing the command 'go run modul3.go' being run and the resulting Fibonacci sequence output.

## Deskripsi program

Program Go pada gambar tersebut adalah program untuk menampilkan deret Fibonacci sebanyak  $n$  suku berdasarkan input pengguna. Program dimulai dengan membaca nilai  $n$ , lalu menginisialisasi dua bilangan pertama deret Fibonacci, yaitu  $a = 0$  dan  $b = 1$ .

Selanjutnya, program menggunakan perulangan `for` untuk mencetak nilai  $a$  secara berurutan, kemudian memperbarui nilai  $a$  dan  $b$  dengan rumus Fibonacci ( $a, b \leftarrow b, a+b$ ). Setiap kali satu angka dicetak, nilai  $n$  dikurangi hingga perulangan selesai. Hasil akhirnya adalah deret Fibonacci sesuai jumlah suku yang diminta, misalnya untuk input 10, program akan mencetak: 0 1 1 2 3 5 8 13 21 34. Program ini menunjukkan penggunaan perulangan, manipulasi variabel, serta logika deret matematika dalam bahasa Go.

## TUGAS

## 1. Tugas 1

### Source code

```
package main

import "fmt"

func main() {
    var user string
    var password string

    userbenar := "Admin"
    passbenar := "Admin"

    gagal := 0

    fmt.Scan(&user, &password)

    for user != userbenar || password != passbenar {
        gagal++

        fmt.Scan(&user, &password)
    }

    fmt.Println(gagal, "percobaan gagal login")
}
```

### Screenshot program

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

- Code Editor:** The main window displays the source code of `tugas1.go`. The code defines a `main` function that reads user input for `user` and `password`. It then compares the input against pre-defined values (`"Admin"` for both). If they don't match, it increments a counter (`gagal`) and loops back to read another set of inputs. Finally, it prints the value of `gagal` followed by the message `"percobaan gagal login"`.
- Terminal:** Below the code editor, a terminal window shows the command `go run tugas1.go` being run. The output indicates that the user input was `User admin` and the password was `Admin admin`, which did not match the expected values, resulting in a value of `4` for `gagal` and the message `percobaan gagal login`.
- Output Window:** A separate window titled `NAM...` shows the results of the program execution. It displays the name `M.NAUFAL AKMAL PRABOWO` and the NIM number `109082500186`.

### Deskripsi program

*Program Go pada gambar tersebut merupakan simulasi login sederhana yang memverifikasi kecocokan antara username dan password yang dimasukkan pengguna dengan data login yang sudah ditentukan, yaitu username "Admin" dan password "Admin". Program diawali dengan membaca input username dan password, lalu menggunakan perulangan for untuk terus meminta pengguna memasukkan ulang kedua data tersebut selama salah satunya tidak sesuai. Setiap kali input salah, variabel penghitung gagal akan bertambah satu untuk mencatat jumlah percobaan login yang tidak berhasil. Ketika pengguna akhirnya berhasil memasukkan username dan password yang benar, perulangan berhenti, dan program menampilkan jumlah percobaan login yang gagal sebelum berhasil. Program ini menunjukkan penggunaan validasi login, perulangan, struktur kondisi, serta pencatatan jumlah kesalahan dalam bahasa Go.*

## Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var digit int

    fmt.Scan(&digit)

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

### Screenshot program

File Edit Selection View Go Run Terminal Help ↺ → Q WOI

EXPLORER COBA1.GO 1 tugas1.go 1 tugas2.go x tugas3.go 1 modul1.go 5 modul2.go 1 modul3.go 1 COBA2.GO 1 COBA3.GO

OPEN EDITORS

COBA1.GO 1 tugas1.go 1 tugas2.go > main 1 package main 2 3 import "fmt" 4 5 func main() { 6 var digit int 7 8 fmt.Scan(&digit) 9 10 for digit > 0 { 11 sisa := digit % 10 12 fmt.Println(sisa) 13 digit = digit / 10 14 } 15 }

W0I COBA1.GO 1 COBA2.GO 1 COBA3.GO modul1.go 5 modul2.go 1 modul3.go 1 tugas1.go 1 tugas2.go 1 tugas3.go 1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

PS C:\Users\naufal\_alkmal\OneDrive\Documents\WOI> go run tugas2.go  
2  
2  
PS C:\Users\naufal\_alkmal\OneDrive\Documents\WOI> go run tugas2.go  
2544  
4  
4  
5  
2  
PS C:\Users\naufal\_alkmal\OneDrive\Documents\WOI> go run tugas2.go  
342354654  
4  
5  
6  
4  
5  
5  
3  
2  
4  
3  
PS C:\Users\naufal\_alkmal\OneDrive\Documents\WOI>

> OUTLINE  
> TIMELINE  
> GO  
> PACKAGE OUTLINE

10 △ 2 ○ 22 🔍 BLACKBOX Agent Open Website

LN 15, COL 2 TAB SIZE: 4 UTF-8 CRLF ↴ Go 🔍 Signed out 1.25.1 🔍 BLACKBOX: Open Chat 🔍 Prettier

## Deskripsi program

Program Go pada gambar tersebut berfungsi untuk memecah dan menampilkan setiap digit dari sebuah bilangan yang dimasukkan pengguna, dimulai dari digit paling belakang. Program membaca input berupa sebuah angka, kemudian menggunakan perulangan `for digit > 0` untuk mengambil digit terakhir dengan operasi modulus (`digit % 10`) dan mencetaknya ke layar. Setelah itu, angka dibagi sepuluh (`digit = digit / 10`) untuk menghilangkan digit terakhir, dan proses ini terus berulang hingga seluruh digit habis. Sebagai hasilnya, setiap digit dari angka yang diinput ditampilkan satu per satu secara terbalik. Program ini menunjukkan penggunaan operasi aritmatika dasar, manipulasi angka, serta perulangan dalam bahasa Go.

## **Tugas 3**

### **Source code**

```
package main

import "fmt"

func main() {
    var x, y int

    fmt.Scan(&x, &y)

    hasil := 0
    temp := x

    for temp >= y {
        temp = temp %
        hasil++
    }
}
```

```

        }

        fmt.Println(hasil)
    }
}

```

## Screenshot program

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

- File Explorer:** Shows multiple Go files including COBA1.GO, COBA2.GO, COBA3.GO, and several modul files. The file `tugas3.go` is currently selected.
- Code Editor:** Displays the content of `tugas3.go`:

```

package main
import "fmt"

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

    hasil := 0
    temp := x

    for temp >= y {
        temp = temp - y
        hasil++
    }

    fmt.Println(hasil)
}

```
- Terminal:** Shows the command `go run tugas3.go` being run twice, once with inputs 5 2 and once with 10 7, both resulting in output 5.
- Output Window:** Displays the output of the program, showing the name and NIM of the user who ran it.

## Deskripsi program

ini adalah sebuah program sederhana dalam file `tugas3.go` yang berada dalam paket `main`. Program tersebut mendefinisikan sebuah fungsi `main` di mana dua variabel integer, `x` dan `y`, dideklarasikan. Program ini kemudian menggunakan `fmt.Scanf` untuk **membaca dua input integer** dari pengguna dan menyimpannya ke dalam `x` dan `y`. Selanjutnya, ada sebuah **loop for** yang diinisialisasi dengan `temp = x`. Loop ini akan terus berjalan **selama nilai temp kurang dari atau sama dengan y** (`temp <= y`), dan di setiap iterasi, variabel `hasil` akan **ditingkatkan (increment)** sebesar satu, serta `temp` juga **ditingkatkan sebesar y**. Akhirnya, program akan **mencetak nilai akhir dari variabel hasil** menggunakan `fmt.Println`, yang secara efektif menghitung berapa kali angka `y` harus ditambahkan ke `x` agar mencapai atau melampaui `y`, sambil secara bersamaan menghitung jumlah iterasi loop.