

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**MODUL 12**

**WHILE-LOOP**



**Disusun oleh:**  
**EWALDO ARDIANSYAH WIDYADHANA**  
**109082500008**  
**S1IF-13-04**

**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

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var n, j int
7     fmt.Scan(&n)
8     j = n
9     for j > 1 {
10         fmt.Print(j, " x ")
11         j = j - 1
12     }
13     fmt.Println(1)
14 }
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

PS C:\Users\ASUS\vscode> go run "c:\tugas\alpro\guidelaprak12.1.go"  
0  
1  
PS C:\Users\ASUS\vscode> go run "c:\tugas\alpro\guidelaprak12.1.go"  
5  
5 x 4 x 3 x 2 x 1  
PS C:\Users\ASUS\vscode> go run "c:\tugas\alpro\guidelaprak12.1.go"  
10  
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1  
PS C:\Users\ASUS\vscode>

## Deskripsi program

Program Go ini berfungsi untuk menghitung dan mencetak faktorial dari suatu bilangan bulat non-negatif yang dimasukkan oleh pengguna. Program ini mendeklarasikan variabel `$n$` untuk input, lalu menggunakan `$fmt.Scan(&n)$` untuk membaca nilai tersebut. Kemudian, program menginisialisasi variabel `$x$` (faktorial) dengan nilai 1. Blok perulangan `for` dimulai dari `$j = 1$` hingga `$j \leq n$`, dan di setiap iterasi, ia mencetak perkalian saat ini (`$j \times x$`) dan memperbarui `$x$` dengan nilai faktorial yang baru. Setelah perulangan selesai, program mencetak nilai akhir dari `$x$`, yang merupakan `$n!$`.

## 2. Guided 2

### Source Code

```

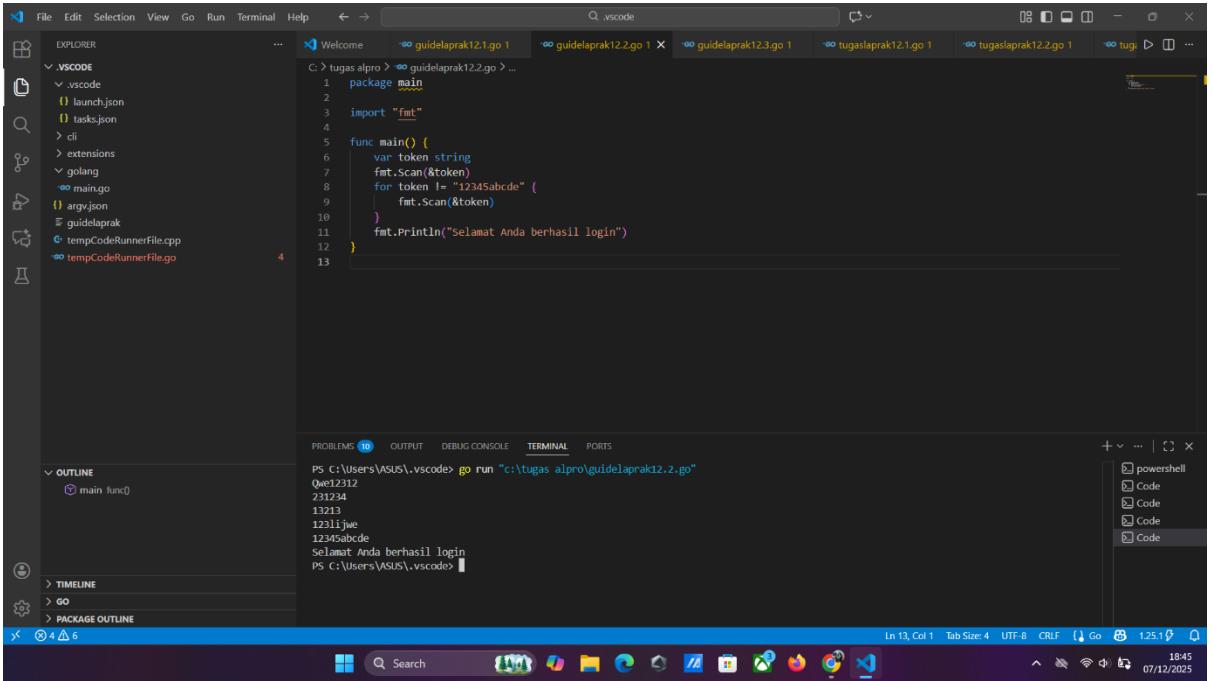
package main

import "fmt"

func main() {
    var token string
    fmt.Scan(&token)
    for token != "12345abcde" {
        fmt.Scan(&token)
    }
    fmt.Println("Selamat Anda berhasil login")
```

```
}
```

## Screenshot program



```
package main

import "fmt"

func main() {
    var token string
    fmt.Scan(&token)
    for token != "12345ABCDE" {
        fmt.Scan(&token)
    }
    fmt.Println("Selamat Anda berhasil login")
}
```

### Deskripsi program

program Go ini dirancang untuk melakukan validasi input token sederhana menggunakan perulangan for.

Program ini:

1. Mendeklarasikan sebuah variabel token bertipe string.
  2. Mendefinisikan sebuah perulangan iteratif (menggunakan range) pada string tetap "12345ABCDE".
  3. Di dalam perulangan, program membaca input string baru dari pengguna menggunakan `fmt.Scan(&token)` pada setiap iterasi.
  4. Perulangan ini akan berjalan sebanyak 10 kali (sesuai panjang string "12345ABCDE").
  5. Setelah perulangan selesai, program mencetak pesan konfirmasi: "Selamat Anda berhasil login".
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 the Visual Studio Code interface with the following details:

- Editor Area:** Displays the Go code for a Fibonacci sequence generator.
- Terminal:** Shows the command line output of running the program. It displays the first 10 numbers of the Fibonacci sequence: 0 1 1 2 3 5 8 13 21 34.
- Explorer:** Shows the project structure with files like `main.go`, `guidelaprapk12.3.go`, and `tempCodeRunnerFile.cpp`.
- Output:** Shows the results of the Go build command.
- Problems:** Shows 10 errors in the code.
- Outline:** Shows the function definition for `main func()`.
- Timeline:** Shows the history of changes made to the file.
- Go:** Shows the status of the Go extension.
- Package Outline:** Shows the package structure.

## Deskripsi program

program Go ini berfungsi untuk menghasilkan dan mencetak deret Fibonacci hingga batas yang ditentukan oleh input pengguna.

#### Analisis Logika

1. **Input Batas:** Program meminta input bilangan bulat ( $\$N\$$ ) dari pengguna menggunakan `fmt.Scan(&N)`.  $\$N\$$  menentukan berapa banyak suku (elemen) dari deret Fibonacci yang akan dicetak.
2. **Inisialisasi Variabel:** Tiga variabel integer, `s1`, `s2`, dan `temp`, digunakan untuk menghitung deret.
  - o `s1` diinisialisasi sebagai 0 (suku pertama).
  - o `s2` diinisialisasi sebagai 1 (suku kedua).
3. **Perulangan:** Perulangan `for` akan berjalan selama  $\$j < N\$$  (dimana  $\$j\$$  adalah *counter* yang diinisialisasi ke 0).
4. **Pencetakan dan Perhitungan:** Di dalam *loop*, program melakukan:
  - o Mencetak nilai `s1` saat ini.
  - o Menghitung suku berikutnya ( $\text{temp} = s1 + s2$ ).
  - o Memperbarui `s1` menjadi nilai `s2` yang lama ( $s1 = s2$ ).
  - o Memperbarui `s2` menjadi nilai `temp` yang baru dihitung ( $s2 = \text{temp}$ ).
  - o *Counter*  $\$j\$$  ditingkatkan ( $j = j + 1$ ).

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var usernameInput string
    var passwordInput string
    var failureCount int = 0

    const correctUsername string = "Admin"
    const correctPassword string = "Admin"

    for {
        fmt.Print("Masukkan Username: ")
        fmt.Scan(&usernameInput)
    }
}
```

```

        fmt.Print("Masukkan Password: ")

        fmt.Scan(&passwordInput)

        if usernameInput == correctUsername && passwordInput == correctPassword {
            fmt.Printf("\nLogin Berhasil!\n")
            fmt.Printf("%d percobaan gagal login\n", failureCount)
            break
        } else {
            failureCount++
            fmt.Println("Username atau Password salah. Silakan coba lagi.\n")
        }
    }
}

```

## Screenshoot program

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

- File Explorer:** Shows a .vscode folder containing launch.json and tasks.json, along with files main.go, argv.json, guidelaprapk, tempCodeRunnerFile.cpp, and tempCodeRunnerFile.go.
- Code Editor:** Displays the main.go file content. The code defines a main function that prints "Masukkan Username:" and "Masukkan Password:", scans user input, and then checks if it matches "Admin". If correct, it prints "Login Berhasil!" and breaks. If incorrect, it increments a failure count and prints an error message. The code editor has line numbers from 1 to 29.
- Terminal:** Shows the command "go run "c:\tugas\_alpro\tugaslaprak12.1.go"" being run, followed by the output: "Masukkan Username: Admin Admin" and "Masukkan Password: Login Berhasil! 0 percobaan gagal login".
- Output:** Shows the status bar indicating "In 20, Col 6 Tab Size: 4 UTF-8 CRLF ⌂ Go 1.25.1 1848 07/12/2025".

## Deskripsi program

### 1. Tujuan Program

Tujuan utama program ini adalah untuk memvalidasi input *username* dan *password* dari pengguna secara berulang dan menghitung total percobaan gagal *login* sebelum pengguna berhasil masuk.

## 2. Variabel Kunci

- `usernameInput` dan `passwordInput`: Menyimpan teks yang dimasukkan oleh pengguna.
  - `failureCount`: Variabel utama yang bertugas menghitung berapa kali *login* gagal. Diinisialisasi dari 0.
  - `correctUsername` dan `correctPassword`: Menyimpan nilai yang benar ("Admin" dan "Admin") sebagai acuan validasi.
- 3. Logika Perulangan (Loop)**
- Program menggunakan perulangan `for {} (loop tak terbatas)` agar terus meminta input sampai kondisi benar terpenuhi.
  - Di setiap iterasi:
    - Program meminta dan membaca `usernameInput` dan `passwordInput`.
    - Menggunakan `if statement` untuk membandingkan input dengan `correctUsername` dan `correctPassword`.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var number int

    fmt.Println("Masukkan bilangan bulat positif: ")
    fmt.Scan(&number)

    if number <= 0 {
        fmt.Println("Input harus bilangan bulat positif.")
        return
    }

    fmt.Println("\nKeluaran:")
```

```

for number > 0 {

    digit := number % 10

    fmt.Println(digit)

    number = number / 10

}

}

```

## Screenshot program

The screenshot shows the Visual Studio Code interface with a Go file named `tugaslaprak12.2.go` open in the editor. The code is as follows:

```

package main

import "fmt"

func main() {
    var number int

    fmt.Print("Masukkan bilangan bulat positif: ")
    fmt.Scan(&number)

    if number <= 0 {
        fmt.Println("Input harus bilangan bulat positif.")
        return
    }

    fmt.Println("\nKeluaran:")

    for number > 0 {
        digit := number % 10

        fmt.Println(digit)
        number = number / 10
    }
}

```

In the terminal tab, the user has run the program and inputted the number 2544. The output shows the digits 2, 5, 4, and 4 printed sequentially.

## Deskripsi program

### 1. Proses Inti

Logika utama program terletak pada perulangan `for` yang memanfaatkan dua operasi matematika kunci: **Modulo (%)** dan **Pembagian Integer (/)**.

### 2. Ekstraksi Digit (Modulo)

Untuk mendapatkan digit yang paling kanan (digit terakhir), kita menggunakan operator modulo 10.

- **$\$Digit = Bilangan \% 10$**
- Contoh: Jika bilangan adalah 2544, maka  $2544 \% 10$  menghasilkan 4. Digit ini kemudian langsung dicetak.

### 3. Pembaruan Bilangan (Pembagian Integer)

Setelah digit terakhir dicetak, kita perlu menghilangkannya dari bilangan asli agar bisa memproses digit berikutnya. Hal ini dilakukan dengan pembagian integer (pembagian yang mengabaikan sisa) dengan 10.

- $\$ \$ \text{Bilangan Baru} = \text{Bilangan Awal} / 10 \$ \$$
- Contoh: Jika bilangan adalah 2544, maka  $2544 / 10 \$$  menghasilkan 254.

#### 4. Pengulangan

Proses ini berlanjut sampai number menjadi 0. Ketika number mencapai 0, itu berarti semua digit sudah diekstrak dan dicetak, dan perulangan berhenti.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {
    var x int
    var y int
    var result int = 0

    fmt.Print("Masukkan bilangan X (pembilang/dividend): ")
    fmt.Scan(&x)

    fmt.Print("Masukkan bilangan Y (pembagi/divisor, Y harus <= X): ")
    fmt.Scan(&y)

    if x <= 0 || y <= 0 {
        fmt.Println("Error: X dan Y harus bilangan bulat positif.")
        return
    }
    if y > x {
        fmt.Println("Error: Y tidak boleh lebih besar dari X (ketentuan soal X >= Y).")
        return
    }
    for x >= y {
        x = x - y
        result++
    }

    fmt.Printf("\nKeluaran:\n")
    fmt.Printf("Hasil integer division adalah: %d\n", result)
}
```

## Screenshoot program

The screenshot shows the Visual Studio Code interface with a Go project open. The Explorer sidebar shows files like `main.go`, `guidelaprapk12.1.go`, `guidelaprapk12.2.go`, `guidelaprapk12.3.go`, `tugaslaprapk12.1.go`, `tugaslaprapk12.2.go`, and `tugaslaprapk12.3.go`. The terminal window shows the command `go run "c:\tugas\alpro\tugaslaprapk12.3.go"`. The code editor displays a Go function `main()` that prompts for two integers, `x` and `y`, and prints their quotient. The output window shows the execution of the program with inputs 5 and 2, resulting in an output of 2.

```
package main
import "fmt"
func main() {
    var x int
    var y int
    var result int = 0
    fmt.Println("Masukkan bilangan X (pembilang/dividend): ")
    fmt.Scan(&x)
    fmt.Println("Masukkan bilangan Y (pembagi/divisor, Y harus <= X): ")
    fmt.Scan(&y)
    if x <= 0 || y <= 0 {
        fmt.Println("Error: X dan Y harus bilangan bulat positif.")
        return
    }
    if y > x {
        fmt.Println("Error: Y tidak boleh lebih besar dari X (ketentuan soal X >= Y).")
        return
    }
    result = x / y
    fmt.Println("Hasil integer division adalah: ", result)
}
```

## Deskripsi program

### 1. Konsep Dasar

Pembagian integer (atau hasil bagi/*quotient*) secara fundamental adalah proses mencari tahu **berapa kali** bilangan pembagi (\$y\$) dapat dikurangkan dari bilangan pembilang (\$x\$) secara berulang hingga sisa pengurangannya kurang dari \$y\$.

### 2. Variabel Kunci

- $x$ : Bilangan yang dibagi (dividend).
- $y$ : Bilangan pembagi (divisor).
- $\text{result}$ : Variabel yang diinisialisasi **0** dan bertugas menghitung berapa kali pengurangan berhasil dilakukan. Ini adalah hasil akhir (*quotient*).

### 3. Logika Perulangan

- Program menggunakan perulangan  $\text{for } x \geq y$ .
- Perulangan akan terus berjalan selama nilai  $x$  (sisa sementara) **masih lebih besar atau sama dengan**  $y$ .