

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 12**

**WHILE-LOOP**



**Disusun oleh:**

**MUHAMMAD FIRDAUS ARDIANSYAH**

**109082500126**

**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 int
    fmt.Print("Masukkan bilangan bulat non negatif: ")
    fmt.Scan(&n)
    for n > 1 {
        fmt.Print(n, " x ")
        n--
    }
    fmt.Println(1)
}
```

#### Screenshot program

```

package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat non negatif: ")
    fmt.Scan(&n)
    for n > 1 {
        fmt.Print(n, " x ")
        n--
    }
    fmt.Println(1)
}

```

MUHAMMAD FIRDAUS ARDIANSYAH  
S11F-13-07  
109082500126

PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\guided1\main.go"  
Masukkan bilangan bulat non negatif: 0  
1  
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\guided1\main.go"  
Masukkan bilangan bulat non negatif: 5  
5 x 4 x 3 x 2 x 1  
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\guided1\main.go"  
Masukkan bilangan bulat non negatif: 10  
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1  
PS D:\Coding\Latihan GO 1> []

## Deskripsi program

Program ini berfungsi untuk mencetak deret angka menurun dari input user hingga angka 1, dengan format visual yang menyerupai notasi matematika faktorial. Logika utamanya menggunakan perulangan for yang terus berjalan selama nilai input (n) lebih besar dari 1; di setiap langkah, program mencetak angka tersebut diikuti simbol "x" lalu mengurangi nilai n sebanyak satu angka.

Proses ini berhenti ketika nilai n mencapai 1, di mana program kemudian mencetak angka "1" sebagai penutup baris. Hasil akhirnya hanya berupa tampilan teks urutan angka (seperti "5 x 4 x 3 x 2 x 1" untuk input 5) tanpa melakukan perhitungan hasil total perkalian dari angka-angka tersebut.

## 2. Guided 2

### Source Code

```

package main

import "fmt"

func main() {

```

```

var token string

fmt.Print("Masukkan token akses: ")

fmt.Scan(&token)

for token != "12345abcde" {

    fmt.Scan(&token)

}

fmt.Println("Selamat anda berhasil login")

}

```

## Screenshot program

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

- File Explorer:** Shows a project structure under "LATIHAN GO 1". The "main.go" file is selected in the "guided2" package.
- Code Editor:** Displays the source code for "main.go" in the "guided2" package. The code checks if the input token is "12345abcde" and prints a success message if it is.
- Terminal:** Shows two terminal sessions. The first session runs the program and prompts for input. The second session shows the output where the user inputs "12345abcde" and the program responds with "Selamat anda berhasil login".
- Modal Window:** A dark-themed modal window titled "Lahiran GO 1" displays the student's information: MUHAMMAD FIRDAUS ARDIANSYAH, S1TF-13-07, and 109082500126.

## Deskripsi program

Program ini melakukan simulasi pemeriksaan akses token sederhana di mana input user sebenarnya diabaikan oleh program. Program pertama kali meminta user memasukkan token, tetapi nilai variabel token segera ditimpas secara *hardcoded* menjadi "12345abcde" di baris 9.

Meskipun user memasukkan token apa pun di awal, program kemudian melakukan fmt.Scan kedua kalinya dan langsung mencetak pesan keberhasilan: "Selamat anda berhasil login". Intinya, kode ini adalah contoh *guided exercise* yang menunjukkan manipulasi variabel input, yang mana pada bentuk aslinya selalu menghasilkan login yang berhasil terlepas dari input user yang sebenarnya.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var a, s1, s2, b, c int
    fmt.Scan(&a)
    s1 = 0
    s2 = 1
    b = 0
    for b < a {
        fmt.Println(s1, " ")
        c = s1 + s2
        s1 = s2
        s2 = c
        b = b + 1
    }
}
```

#### Screenshot program

The screenshot shows the Visual Studio Code interface with several tabs open. The main editor tab contains the following Go code:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, s1, s2, b, c int
7     fmt.Scan(&a)
8     s1 = 0
9     s2 = 1
10    b = 0
11    for b < a {
12        fmt.Println(s1, " ")
13        c = s1 + s2
14        s1 = s2
15        s2 = c
16        b = b + 1
17    }
18 }

```

The terminal tab shows the command `go run "d:\Coding\Latihan GO 1\modul12\guided3\main.go"` and its output:

```

PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\guided3\main.go"
5
0 1 1 2 3

```

## Deskripsi program

Program ini membaca sebuah angka sebagai batas banyaknya deret yang ingin ditampilkan, lalu menghasilkan urutan bilangan Fibonacci secara berurutan . Dua nilai awal deret diset sebagai 0 dan 1, kemudian program mencetak nilai pertama dan terus menjumlahkan dua nilai sebelumnya untuk mendapatkan nilai berikutnya. Setiap iterasi, nilai-nilai digeser ke depan sehingga proses perhitungan berulang hingga jumlah elemen yang diminta terpenuhi.

Secara keseluruhan, logika utamanya berada pada loop for yang terus berjalan selama penghitung b masih lebih kecil dari input pengguna. Di dalam loop, nilai Fibonacci dicetak, dihitung ulang, dan variabel-variabel diperbarui agar deret tetap berkelanjutan hingga selesai . Program ini menampilkan deret Fibonacci dalam satu baris dengan spasi sebagai pemisah setiap angkanya.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main
```

```
import "fmt"

func main() {

    const Username = "Admin"

    const Password = "Admin"

    var username, password string

    var gagal int

    for {

        fmt.Print("Masukkan Username: ")

        fmt.Scan(&username)

        fmt.Print("Masukkan Password: ")

        fmt.Scan(&password)

        if username == Username && password == Password {

            break

        } else {

            gagal++

            fmt.Println("Username atau Password salah,"

coba lagi.")

        }

    }

}
```

```

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

    }

```

## Screenshoot program

The screenshot shows a code editor with multiple tabs open, displaying Go code. The main code file contains a for loop that prints user input and increments a counter if the credentials are incorrect. A terminal window to the right shows the execution of the program and its output.

```

package main

import "fmt"

func main() {
    const Username = "Admin"
    const Password = "Admin"

    var username, password string
    var gagal int

    for {
        fmt.Print("Masukkan Username: ")
        fmt.Scan(&username)

        fmt.Print("Masukkan Password: ")
        fmt.Scan(&password)

        if username == Username && password == Password {
            break
        } else {
            gagal++
            fmt.Println("Username atau Password salah, coba lagi.")
        }
    }

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

```

Terminal Output:

```

MUHAMMAD FIRDAUS ARDIANSYAH
S1IF-13-07
109082500126

Ln 3, Col 13 51 character Plain b 150% Wind UTF-8
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\soall\main.go"
Masukkan Username: Admin
Masukkan Password: Admin
Username atau Password salah, coba lagi.
Masukkan Username: admin
Masukkan Password: admin
Username atau Password salah, coba lagi.
Masukkan Username: Admin
Masukkan Password: Admin
2 percobaan gagal login
PS D:\Coding\Latihan GO 1> []

```

## Deskripsi program

Program ini mengimplementasikan sistem *login* dasar berbasis perulangan tak terbatas (`for {}`) yang memerlukan user memasukkan *username* dan *password* yang *hardcoded* ("Admin"/"Admin"). Variabel *gagal* berfungsi sebagai penghitung yang bertambah (*gagal++*) setiap kali otentikasi gagal.

Logika program adalah: jika *username* dan *password* sesuai dengan konstanta yang ditentukan, loop segera dihentikan (*break*), menandakan keberhasilan *login*. Jika gagal, user menerima pesan error, dan *counter* *gagal* dicetak setelah loop. Inti dari program ini adalah mengontrol alur program melalui *break* dan mencatat jumlah kegagalan.

## 2. Tugas 2

### Source code

```
package main
```

```

import "fmt"

func main() {

    var num int

    fmt.Print("Masukkan angka: ")

    fmt.Scan(&num)

    for num > 0 {

        digit := num % 10

        fmt.Println(digit)

        num /= 10

    }

}

```

## Screenshot program

The screenshot shows a Go development environment with several tabs open in the browser:

- `main.go` (modul12\guided3 U)
- `main.go` (modul12\soal1 U)
- `main.go` (modul12\soal2 U)
- `main.go` (modul12\soal3 U)
- `main.go` (modul5\guided2)
- `main.go` (modul5\tugas1)

The code in the active tab (`main.go`) is:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var num int
7     fmt.Print("Masukkan angka: ")
8     fmt.Scan(&num)
9
10    for num > 0 {
11        digit := num % 10
12        fmt.Println(digit)
13        num /= 10
14    }
15 }

```

The terminal window shows the following output:

```

PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\soal2\main.go"
Masukkan angka: 2
2
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\soal2\main.go"
Masukkan angka: 2544
4
4
5
2
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\soal2\main.go"
Masukkan angka: 3423554654
4
5
6
4
5
5
3
2
4
4
3

```

A modal dialog box displays the student information:

MUHAMMAD FIRDAUS ARDIANSYAH  
S11F-13-07  
109082500126

### **Deskripsi program**

Program ini berfungsi untuk mengekstrak dan mencetak setiap digit dari bilangan bulat positif yang dimasukkan oleh user, secara efektif mencetak urutan digit tersebut secara terbalik. Logika utama program bergantung pada perulangan for yang terus berjalan selama nilai input (num) lebih besar dari nol. Di dalam loop tersebut, operasi modulo 10 (digit := num % 10) digunakan untuk mengisolasi digit terakhir dari bilangan.

Digit yang diisolasi tersebut segera dicetak. Setelah pencetakan, bilangan num diperbarui melalui pembagian integer (num /= 10), yang berfungsi untuk menghilangkan digit terakhir yang sudah diproses dan mempersiapkan bilangan untuk iterasi berikutnya. Proses ini berulang, mengekstrak digit dari kanan ke kiri, menghasilkan *output* yang terbalik dari urutan angka asli yang dimasukkan user.

### **3. Tugas 3**

#### **Source code**

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukkan dua angka: ")
    fmt.Scan(&x, &y)
    count := 0

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

#### **Screenshoot program**

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

- EXPLORER**: Shows a file tree with several packages like LATIH, Coba, guided1, lat, latihan9, latihan10, Modul1, Modul2, Modul3, Modul4, Modul5, modul9, modul10, modul11, modul12, modul13, modul14, modul15, and soal1 through soal3.
- CODE EDITOR**: Displays a Go file named main.go with the following code:

```

1 package main
2 import "fmt"
3
4 func main() {
5     var x, y int
6     fmt.Println("Masukkan dua angka: ")
7     fmt.Scan(&x, &y)
8     count := 0
9
10    for x >= y {
11        x -= y
12        count++
13    }
14    fmt.Println(count)
15 }
```
- TERMINAL**: Shows command-line output from running the program:

```

PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\soal3\main.go"
Masukkan dua angka: 5 2
2
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul12\soal3\main.go"
Masukkan dua angka: 10 7
1
PS D:\Coding\Latihan GO 1>
```
- OUTPUT**: Shows the output of the program runs.
- PROBLEMS**: Shows no problems.
- PORTS**: Shows no ports.
- RIGHT SIDE**: A text editor window titled "Latihan GO 1" containing student information:

```

MUHAMMAD FIRDAUS ARDIANSYAH
S1IF-13-07
109082500126
```

## Deskripsi program

Program ini bertujuan untuk menghitung jumlah iterasi (count) yang diperlukan untuk menghentikan perulangan for berdasarkan kondisi perbandingan dua bilangan bulat, x dan y. User diminta memasukkan dua angka (x dan y), dan perulangan akan berlanjut selama  $x > y$ . Namun, logika di dalam loop sangat tidak konvensional: pada setiap iterasi, nilai x secara destruktif ditimpak dengan nilai y yang lama ( $x = y$ ), dan kemudian nilai y diinkrementasi ( $y++$ ).

Karena x mengambil nilai y yang lama, dan y selalu bertambah satu di iterasi yang sama, kondisi perulangan ( $x > y$ ) hampir selalu menjadi FALSE setelah iterasi pertama jika kondisi awal  $x > y$  terpenuhi. Ini berarti program secara praktis hanya menghitung satu langkah yang diperlukan untuk memenuhi kondisi penghentian loop. *Output* program adalah nilai *counter* (count), yang akan menjadi 1 jika  $x > y$  di awal, dan 0 jika  $x \leq y$ .