

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 12**

**WHILE LOOP**



**Disusun oleh:**

**FERDINAND AXEL VALERIAN**

**109082500154**

**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 num int

    fmt.Println("Masukkan bilangan bulat non-negatif: ")
    fmt.Scan(&num)

    if num < 0 {
        fmt.Println("Input harus bilangan bulat non-
negatif.")
        return
    }

    if num == 0 || num == 1 {
        fmt.Println(num, " -> 1")
        return
    }

    fmt.Printf("%d -> ", num)

    for i := num; i > 1; i-- {
        fmt.Printf("%d x ", i)
    }

    fmt.Println("1")
}
```

## Screenshot program

The screenshot shows a code editor with a dark theme displaying a Go file named g1.go. The code prompts the user to enter a non-negative integer, checks if it's negative or zero, and then prints its factorial. The terminal window shows the execution of the program, inputting 109082500154, which outputs 109082500154, S1F-13-07, and Ferdinand Axel Valerian.

```
∞ g1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var num int
7
8     fmt.Print("Masukkan bilangan bulat non-negatif: ")
9     fmt.Scan(&num)
10
11    if num < 0 {
12        fmt.Println("Input harus bilangan bulat non-negatif.")
13        return
14    }
15
16    if num == 0 || num == 1 {
17        fmt.Println(num, " -> 1")
18        return
19    }
20
21    fmt.Printf("%d -> ", num)
22
23    for i := num; i > 1; i-- {
24        fmt.Printf("%d x ", i)
25    }
26
27    fmt.Println("1")
28 }
```

- PS D:\vscode\College\Modul 12> go run g1.go  
Masukkan bilangan bulat non-negatif: 0  
0 -> 1
- PS D:\vscode\College\Modul 12> go run g1.go  
Masukkan bilangan bulat non-negatif: 5  
5 -> 5 x 4 x 3 x 2 x 1
- PS D:\vscode\College\Modul 12> go run g1.go  
Masukkan bilangan bulat non-negatif: 10  
10 -> 10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
- PS D:\vscode\College\Modul 12> go run g1.go  
Masukkan bilangan bulat non-negatif: 1  
1 -> 1

The screenshot shows a terminal window with a dark theme. It displays the same Go program execution as the previous screenshot, with the output 109082500154, S1F-13-07, and Ferdinand Axel Valerian.

## Deskripsi program

Program Go ini dirancang untuk menerima input berupa bilangan bulat non-negatif n dan kemudian menampilkan deret perkalian yang merepresentasikan perhitungan faktorial dari bilangan tersebut. Program dimulai dengan memvalidasi input, memastikan bahwa n tidak negatif. Jika n adalah 0 atau 1, program langsung menampilkan 1 sebagai hasilnya. Untuk nilai n lebih besar dari 1, program menggunakan perulangan yang berulang dari n ke bawah hingga 2, mencetak setiap angka diikuti dengan " x ", dan diakhiri dengan angka 1 untuk melengkapi deret perkalian faktorial n dikali (n kurang 1) dikali seterusnya hingga dikali 2 dikali 1.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
    const validToken = "12345abcde"
    var inputToken string

    fmt.Println("Program Login Aplikasi")

    for inputToken != validToken {
        fmt.Print("Masukkan token: ")
        fmt.Scan(&inputToken)

        if inputToken == validToken {
            fmt.Println("Selamat Anda berhasil login")
        } else {
            fmt.Println("Token salah. Silakan coba
lagi.")
        }
    }
}
```

### Screenshot program

```
g2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     const validToken = "12345abcde"
7     var inputToken string
8
9     fmt.Println("Program Login Aplikasi")
10
11    for inputToken != validToken {
12        fmt.Print("Masukkan token: ")
13        fmt.Scan(&inputToken)
14
15        if inputToken == validToken {
16            fmt.Println("Selamat Anda berhasil login")
17        } else {
18            fmt.Println("Token salah. Silakan coba lagi.")
19        }
20    }
21 }
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\vscode\College\Modul 12> go run g2.go

Program Login Aplikasi

Masukkan token: Qwe12312

Token salah. Silakan coba lagi.

Masukkan token: 231234

Token salah. Silakan coba lagi.

Masukkan token: 13213

Token salah. Silakan coba lagi.

Masukkan token: 123lijwe

Token salah. Silakan coba lagi.

Masukkan token: 12345abcde

Selamat Anda berhasil login

109082500154

S1F-13-07

Ferdinand Axel Valerian

Ln 3, Col 24 | 46 character | Plain text | 100% | Wind | UTF-8

## Deskripsi program

Program Go ini berfungsi sebagai simulasi proses login yang memerlukan validasi token rahasia. Program menetapkan token yang benar sebagai konstanta (`validToken = "12345abcde"`). Program menggunakan perulangan tak terbatas (`for`) yang akan terus meminta pengguna untuk memasukkan token (`fmt.Scan`) dan membandingkannya dengan token yang valid. Jika token yang dimasukkan salah, program mencetak pesan kesalahan dan perulangan berlanjut; jika token yang dimasukkan sama, program mencetak pesan keberhasilan: "Selamat Anda berhasil login", dan perulangan secara otomatis berhenti karena kondisi `inputToken != validToken` sudah terpenuhi.

### 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
}

}

```

## Screenshoot program

The screenshot shows the Visual Studio Code interface with a Go file named g3.go open. The code implements a simple Fibonacci sequence generator. The terminal below shows the execution of the program and its output.

```

g3.go > main
1 package main
2 import "fmt"
3 func main() {
4     var N, s1, s2, j, temp int
5
6     fmt.Scan(&N)
7     s1 = 0
8     s2 = 1
9     j = 0
10
11    for j < N {
12        fmt.Println(s1, " ")
13
14        temp = s1 + s2
15        s1 = s2
16        s2 = temp
17        j = j + 1
18    }
19 }

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS D:\vscode\College\Modul 12> go run g3.go
5
0 1 1 2 3
● PS D:\vscode\College\Modul 12> go run g3.go
2
0 1
● PS D:\vscode\College\Modul 12> go run g3.go
10
0 1 1 2 3 5 8 13 21 34
PS D:\vscode\College\Modul 12>

```

## Deskripsi program

Program Go ini berfungsi untuk menghasilkan dan mencetak N bilangan pertama dalam deret Fibonacci. Program meminta input bilangan bulat positif N (asumsi N lebih besar atau sama dengan 2) dari pengguna untuk menentukan berapa banyak suku yang akan ditampilkan. Deret Fibonacci dimulai dengan suku awal 0 dan 1. Program menggunakan perulangan (for) sebanyak N kali, di mana dalam setiap iterasi, ia mencetak suku saat ini (s1), kemudian menghitung suku berikutnya sebagai jumlah

dari dua suku sebelumnya ( $s_1$  ditambah  $s_2$ ), dan memperbarui nilai  $s_1$  dan  $s_2$  untuk iterasi selanjutnya, sehingga deret yang dicetak adalah 0, 1, 1, 2, 3, 5, dan seterusnya.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var gagal int
    var username string
    var password string

    gagal = 0

    for {
        fmt.Print("Masukkan username dan password: ")
        fmt.Scan(&username, &password)

        if username == "Admin" && password == "Admin" {
            break
        } else {
            gagal += 1
            fmt.Println("Username atau Password salah. Silakan coba lagi.")
        }
    }

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

#### Screenshoot program

The screenshot shows the VS Code interface with the following details:

- Code Editor:** The file `u1.go` is open, displaying a Go program. The code defines a `main` package with a `main` function. It prompts the user for a username and password, attempting to log them in with the credentials "Admin" and "Admin". If the credentials are incorrect, it increments a counter `gagal` and prints an error message. The terminal window shows the execution of the program and its output.
- Terminal Output:**

```
109082500154
S1F-13-07
Ferdinand Axel Valerian

Ln 3, Col 24 | 46 characters | Plain text | 100% | Wind | UTF-8
```
- Terminal History:**

```
PS D:\vscode\College\Modul 12> go run u1.go
Masukkan username dan password: User123 user123
Username atau Password salah. Silakan coba lagi.
Masukkan username dan password: User admin
Username atau Password salah. Silakan coba lagi.
Masukkan username dan password: Admin admin
Username atau Password salah. Silakan coba lagi.
Masukkan username dan password:
Admin Admin123
Username atau Password salah. Silakan coba lagi.
Masukkan username dan password: Admin Admin
4 percobaan gagal login
PS D:\vscode\College\Modul 12>
```

### Deskripsi program

Program Go ini berfungsi untuk mensimulasikan proses login yang menghitung dan menampilkan jumlah percobaan gagal yang dilakukan pengguna. Program menetapkan pasangan Username dan Password yang benar ("Admin" dan "Admin"). Program menggunakan perulangan tak terbatas (`for {}`) untuk terus meminta input Username dan Password dari pengguna dalam satu baris yang dipisahkan oleh spasi (menggunakan `fmt.Scan`). Jika kredensial yang dimasukkan salah, penghitung gagal akan bertambah dan pesan kesalahan akan ditampilkan. Jika kredensial yang dimasukkan benar, perulangan akan dihentikan (`break`). Setelah perulangan berhenti, program akan mencetak total angka yang tersimpan di variabel `gagal` sebagai jumlah kegagalan sebelum berhasil login.

## 2. Tugas 2

### Source code

```
package main

import "fmt"
```

```

func main() {
    var num int

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

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

    for num > 0 {
        digit := num % 10
        fmt.Println(digit)

        num /= 10
    }
}

```

## Screenshot program

The screenshot shows a terminal window with the following content:

```

~$ u2.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var num int
7
8      fmt.Print("Masukkan bilangan bulat positif: ")
9      fmt.Scan(&num)
10
11     if num <= 0 {
12         fmt.Println("Input harus bilangan bulat positif.")
13         return
14     }
15
16     for num > 0 {
17         digit := num % 10
18         fmt.Println(digit)
19
20         num /= 10
21     }
22 }

```

To the right of the terminal, a small window displays the output of the program. The window has a title bar with icons and a menu bar with "File", "Edit", "View", "Aa", "Color", "Font", "Size", and "Zoom". The main area of the window shows the following text:

```

109082500154
S1F-13-07
Ferdinand Axel Valerian

```

At the bottom of the window, there is a status bar with the text "Ln 3, Col 24 | 46 characters | Plain text | 100% | Wind | UTF-8".

The screenshot shows a terminal window in VS Code. The terminal tab is active, displaying the following Go code and its execution:

```
PS D:\vscode\College\Modul 12> go run u2.go
Masukkan bilangan bulat positif: 2
2
PS D:\vscode\College\Modul 12> go run u2.go
Masukkan bilangan bulat positif: 2544
4
4
5
5
6
4
5
5
3
2
4
3
PS D:\vscode\College\Modul 12>
```

Below the terminal, a code editor window is visible, showing a file named "u2.go" with the following content:

```
package main

import "fmt"

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

    fmt.Print("Masukkan bilangan x dan y : ")
    fmt.Scan(&x, &y)
    hasil = 0

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

    fmt.Println("Hasil integer division adalah:", hasil)
}
```

### Deskripsi program

Program Go ini berfungsi untuk mencacah dan mencetak setiap digit yang terdapat dalam bilangan bulat positif yang dimasukkan oleh pengguna. Pencetakan digit dilakukan secara terbalik, dimulai dari digit terakhir (paling kanan) hingga digit pertama (paling kiri). Program menggunakan perulangan (for num > 0) yang terus berjalan selama nilai bilangan (variabel num) lebih besar dari nol. Di setiap iterasi, operator modulo 10 (num % 10) digunakan untuk mendapatkan digit terakhir, yang kemudian dicetak. Selanjutnya, operator pembagian integer (num /= 10) digunakan untuk menghilangkan digit terakhir tersebut sebelum perulangan berlanjut ke digit berikutnya.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

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

    fmt.Print("Masukkan bilangan x dan y : ")
    fmt.Scan(&x, &y)
    hasil = 0

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

    fmt.Println("Hasil integer division adalah:", hasil)
}
```

### Screenshot program

The screenshot shows a Visual Studio Code interface. On the left, there is a code editor with a Go file named u3.go. The code defines a package main with a function main() that prompts the user for two integers x and y, then calculates their integer division using a for loop. On the right, a terminal window is open, showing the execution of the program and its output. The terminal output includes three runs of the program with different inputs (x=5, y=2; x=10, y=7; x=120, y=4) and the resulting integer division outputs (2, 1, 30).

```
∞ u3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x, y, hasil int
7
8     fmt.Println("Masukkan bilangan x dan y : ")
9     fmt.Scan(&x, &y)
10    hasil = 0
11
12    for x >= y {
13        x = x - y
14        hasil++
15    }
16
17    fmt.Println("Hasil integer division adalah:", hasil)
18 }
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\vscode\College\Modul 12> go run u3.go  
Masukkan bilangan x dan y : 5 2  
Hasil integer division adalah: 2  
PS D:\vscode\College\Modul 12> go run u3.go  
↳ Masukkan bilangan x dan y : 10 7  
Hasil integer division adalah: 1  
PS D:\vscode\College\Modul 12> go run u3.go  
Masukkan bilangan x dan y : 120  
4  
Hasil integer division adalah: 30  
PS D:\vscode\College\Modul 12> []

## Deskripsi program

Program Go ini berfungsi untuk menghitung hasil integer division (pembagian bulat) dari dua bilangan bulat positif, x (pembilang) dibagi y (pembagi), tanpa menggunakan operator pembagian. Program meminta input kedua bilangan tersebut dalam satu baris. Asumsi yang digunakan adalah x lebih besar atau sama dengan y dan y adalah bilangan positif. Program menggunakan perulangan (for x >= y) yang secara berulang mengurangi nilai x dengan y dan menaikkan nilai penghitung hasil sebanyak satu setiap kali pengurangan berhasil. Nilai akhir dari hasil menunjukkan berapa kali y dapat dikurangkan dari x, yang merupakan hasil pembagian bulat x div y.