

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 12  
WHILE - LOOP**



**Disusun oleh:**

**Yedija Johanan Siregar**

**109082500075**

**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, j int

    fmt.Scan(&n)

    j = n

    for j > 1 {

        fmt.Print(j, " x ")

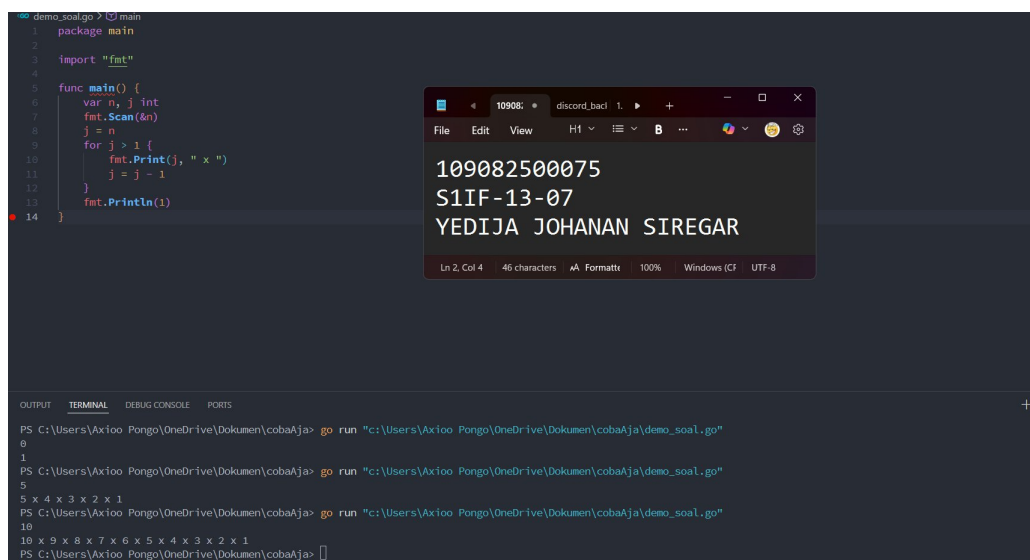
        j = j - 1

    }

    fmt.Println(1)

}
```

#### Screenshoot program



## Deskripsi program

Program di atas ditulis dalam bahasa Go dan berfungsi untuk menampilkan dekomposisi faktorial dari sebuah bilangan bulat yang dimasukkan pengguna. Pertama, program membaca bilangan tersebut dan menyimpannya dalam variabel `n`. Kemudian, program menggunakan perulangan untuk mencetak nilai dari `n` turun secara berurutan, masing-masing dipisahkan dengan tanda " x ", hingga mencapai angka 2. Terakhir, program mencetak angka 1 dan berpindah baris, sehingga menghasilkan tampilan seperti "5 x 4 x 3 x 2 x 1" untuk input bilangan 5.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var token string

    fmt.Scan(&token)

    for token != "12345abcde" {

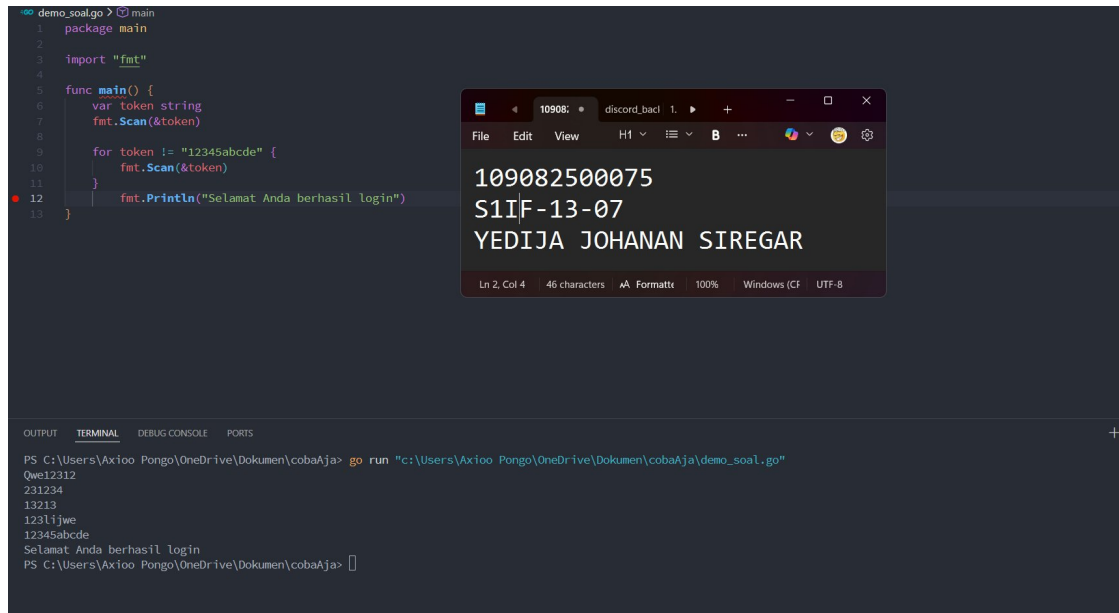
        fmt.Scan(&token)

    }

    fmt.Println("Selamat Anda berhasil login")

}
```

## Screenshoot program



The screenshot shows a Go program in a text editor and its execution in a terminal. The Go code defines a `main` function that prompts for a token and checks it against the string "12345abcde". If the input matches, it prints a success message. The terminal output shows the program being run, the input "12312 231234 13213 1231jwe 12345abcde", and the resulting message "Selamat Anda berhasil login".

```
demo_soal.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var token string
7     fmt.Scan(&token)
8
9     for token != "12345abcde" {
10        fmt.Scan(&token)
11    }
12    fmt.Println("Selamat Anda berhasil login")
13 }
```

```
109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR
```

```
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
Qwe12312
231234
13213
1231jwe
12345abcde
Selamat Anda berhasil login
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>
```

## Deskripsi program

Program di atas adalah implementasi sederhana dari sebuah sistem login berbasis teks menggunakan bahasa Go. Program akan meminta pengguna untuk memasukkan sebuah token atau kata sandi melalui input standar, yang kemudian dibaca dan disimpan dalam variabel token. Melalui sebuah perulangan for, program akan terus meminta input token dari pengguna selama nilai yang dimasukkan tidak sama dengan string "12345abcde". Jika pengguna berhasil memasukkan token yang sesuai yaitu "12345abcde", maka perulangan akan berhenti dan program akan mencetak pesan "Selamat Anda berhasil login". Dengan demikian, program ini berfungsi sebagai mekanisme otentikasi dasar yang hanya memberikan akses ketika kata sandi yang tepat diberikan.

### 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.Print(s1, " ")

        temp = s1 + s2

        s1 = s2

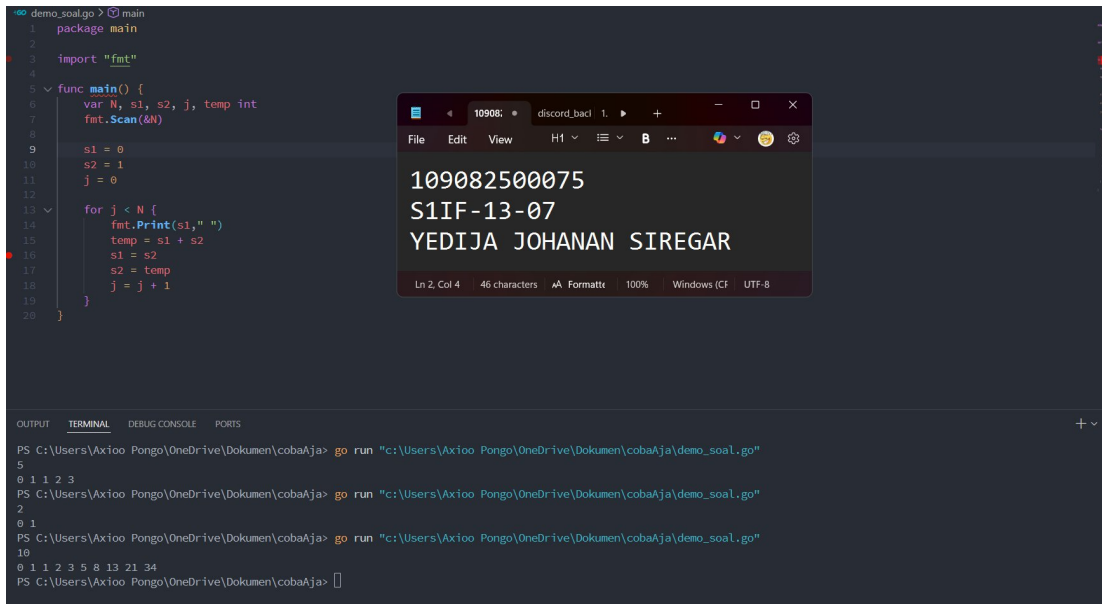
        s2 = temp

        j = j + 1

    }

}
```

## Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The program is a simple calculator for parking fees. It takes an integer for the number of hours and a string for the vehicle type. It then calculates the fee based on the vehicle type and the number of hours. The output is displayed in a browser window.

```
demo_soal.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var N, s1, s2, j, temp int
7     fmt.Scan(&N)
8
9     s1 = 0
10    s2 = 1
11    j = 0
12
13    for j < N {
14        fmt.Print(s1, " ")
15        temp = s1 + s2
16        s1 = s2
17        s2 = temp
18        j = j + 1
19    }
20 }
```

Terminal output:

```
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
5
0 1 1 2 3
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
2
0 1
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
10
0 1 1 2 3 5 8 13 21 34
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>
```

Browser window output:

```
109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR
```

## Deskripsi program

Program ini menghitung tarif parkir berdasarkan jenis kendaraan (Motor, Mobil, Truk) dan durasi parkir dalam jam. Program menerima input jenis kendaraan (string) dan durasi (integer), lalu menggunakan struktur switch tanpa kondisi untuk mengevaluasi kombinasi kasus: setiap jenis kendaraan memiliki tarif berbeda untuk durasi 1-2 jam (tarif dasar) dan durasi lebih dari 2 jam (tarif tambahan). Jika input tidak sesuai dengan kriteria yang ditentukan (misalnya jenis kendaraan salah atau durasi kurang dari 1), program akan mencetak pesan error "Jenis kendaraan atau durasi parkir tidak valid". Tarif yang sesuai kemudian ditampilkan dalam format "Rp X".

## TUGAS

### 1. Tugas 1

#### Source code

```
package main
```

```
import "fmt"

func main() {

    var username, password string

    count := 0

    for {

        fmt.Scan(&username, &password)

        if username == "Admin" && password == "Admin" {

            break

        }

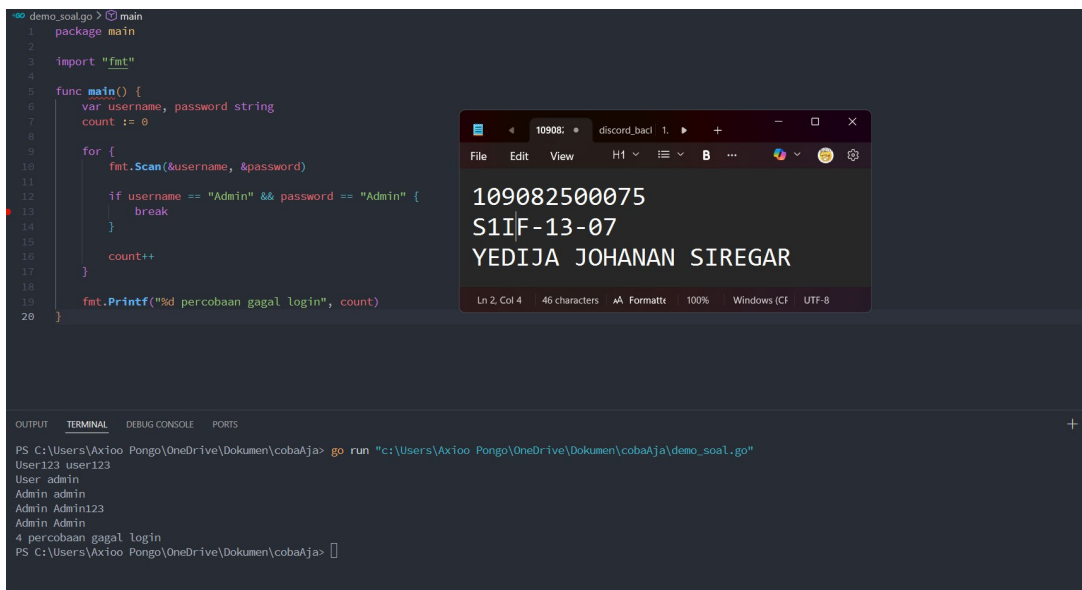
        count++

    }

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

}
```

## Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The code is a simple login program that prompts for a username and password. It checks if the credentials are "Admin" and "Admin". If they are, it breaks the loop. Otherwise, it increments a counter and prints the number of failed login attempts. The terminal output shows the program running and the user entering "Admin" and "Admin". The browser window displays the output of the program: "109082500075", "S1IF-13-07", and "YEDIJA JOHANAN SIREGAR".

```
demo_soal.go > main
package main

import "fmt"

func main() {
    var username, password string
    count := 0

    for {
        fmt.Scan(&username, &password)

        if username == "Admin" && password == "Admin" {
            break
        }

        count++
    }

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

OUTPUT TERMINAL DEBUG CONSOLE PORTS

```
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
User123 user123
User admin
Admin admin
Admin Admin123
Admin Admin
4 percobaan gagal login
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>
```

109082500075  
S1IF-13-07  
YEDIJA JOHANAN SIREGAR

## Deskripsi program

Program ini menghitung berapa kali pengguna gagal login sebelum berhasil memasukkan kredensial yang benar. Setiap kali user tersebut memasukkan username dan password yang salah (bukan "Admin" untuk keduanya), program akan mencatatnya sebagai satu percobaan gagal dan meminta input ulang. Program akan terus berjalan dalam perulangan hingga pengguna berhasil memasukkan username "Admin" dan password "Admin" secara bersamaan. Setelah login berhasil, program akan menampilkan total jumlah percobaan login yang gagal dilakukan pengguna sebelumnya.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {

    var n int
```



```

    fmt.Scan(&n)

    if n == 0 {

        fmt.Println(0)

    } else {

        for n > 0 {

            digit := n % 10

            fmt.Println(digit)

            n = n / 10

        }

    }

}

```

## Screenshoot program

The screenshot shows a Go program being executed in a terminal window. The program's output is displayed in a web browser window, showing the digits of the input number 109082500075 in reverse order, followed by a hyphen and the name YEDIJA JOHANAN SIREGAR.

```

demo_soal.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Scan(&n)
8
9     if n == 0 {
10        fmt.Println(0)
11    } else {
12        for n > 0 {
13            digit := n % 10
14            fmt.Println(digit)
15            n = n / 10
16        }
17    }
18 }

```

109082500075  
S11F-13-07  
YEDIJA JOHANAN SIREGAR

Ln 2, Col 4 46 characters AA Formatted 100% Windows (C) UTF-8

OUTPUT TERMINAL DEBUG CONSOLE PORTS

```

PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
2
2
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run "c:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja\demo_soal.go"
2544
4
4
5
2
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

```

## Deskripsi program

Program ini memecah bilangan bulat positif menjadi digit-digit penyusunnya dan menampilkannya dari digit terakhir (satuan) hingga digit pertama. Pertama, program membaca bilangan dari input. Jika bilangan adalah 0, program langsung

menampilkan 0 dan selesai. Untuk bilangan positif, program masuk ke perulangan yang terus berjalan selama bilangan lebih besar dari 0. Di setiap iterasi, program mengambil digit terakhir dengan operator % 10 (modulus 10) untuk mendapatkan sisa pembagian dengan 10, lalu mencetaknya. Kemudian bilangan dibagi 10 dengan /= 10 untuk menghilangkan digit terakhir yang sudah diproses. Proses ini berulang hingga semua digit tertampilkan dari kanan ke kiri.

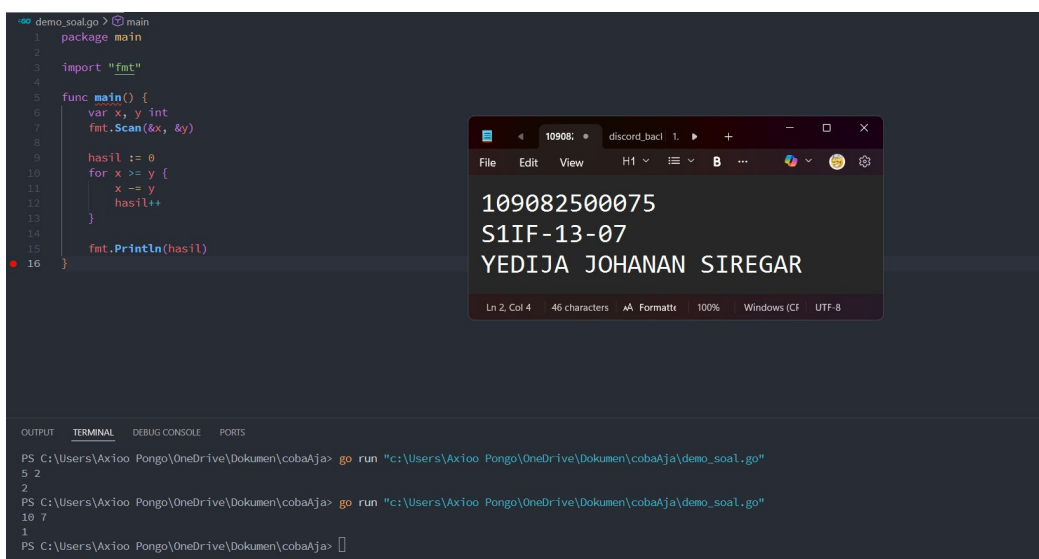
### 3. Tugas 3

#### Source code

```
package main
import "fmt"
func main() {
    var x, y int
    fmt.Scan(&x, &y)

    hasil := 0
    for x >= y {
        x -= y
        hasil++
    }
    fmt.Println(hasil)
}
```

#### Screenshoot program



**Deskripsi program**

Program ini menghitung hasil pembagian bulat ( $x$  dibagi  $y$ ) tanpa menggunakan operator pembagian. Program membaca dua bilangan positif  $x$  dan  $y$ , dengan  $x$  lebih besar atau sama dengan  $y$ . Cara kerjanya adalah dengan mengurangi  $x$  dengan  $y$  berulang kali selama  $x$  masih lebih besar atau sama dengan  $y$ . Setiap kali pengurangan berhasil dilakukan, variabel penghitung hasil ditambah satu. Proses ini terus berlanjut hingga  $x$  menjadi lebih kecil dari  $y$ . Setelah perulangan selesai, program mencetak nilai penghitung yang merupakan hasil pembagian bulat dari  $x$  oleh  $y$ .