

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 12**

**WHILE LOOP**



**Disusun oleh:**

**NADIFA AZKHIA SYARIF**

**109082530002**

**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 n int
    fmt.Print("Masukkan angka: ")
    fmt.Scan(&n)

    if n == 0 || n == 1 {
        fmt.Println("1")
        return
    }

    i := n
    for {
        if i == 1 {
            fmt.Print("1")
            break
        }
        fmt.Printf("%d x ", i)
        i--
    }
}
```

```
}
```

## Screenshoot program

The screenshot shows the Visual Studio Code interface with a dark theme. On the left is the Explorer sidebar showing a project folder 'MINGGU12' containing three subfolders 'guide 1', 'guide 2', and 'guide 3', each with a corresponding .go file. The 'guide 1' folder is expanded, and its file 'guide1.go' is selected. The main editor area displays the following Go code:

```
func main() {
    var n int
    fmt.Print("Masukkan angka: ")
    fmt.Scan(&n)

    if n == 0 || n == 1 {
        fmt.Println("1")
        return
    }

    i := n
    for {
        if i == 1 {
            fmt.Print("1")
            break
        }
        fmt.Printf("%d x ", i)
        i--
    }
}
```

To the right of the editor is a terminal window showing the command line output:

```
PS C:\Users\hp\OneDrive\Documents\minggu12> cd "c:\Users\hp\OneDrive\Documents\minggu12\guide 1"
PS C:\Users\hp\OneDrive\Documents\minggu12\guide 1> go run "c:\Users\hp\OneDrive\Documents\minggu12\guide 1\guide1.go"
Masukkan angka: 5
5 x 4 x 3 x 2 x 1
PS C:\Users\hp\OneDrive\Documents\minggu12\guide 1>
```

A floating terminal window titled 'Nama Nad' shows the user's name and NIM.

## Deskripsi program

Program di atas dibuat untuk menampilkan urutan perkalian turun dari suatu angka sampai angka 1. Pertama, program menyiapkan variabel n sebagai tempat menyimpan input. Setelah itu program menampilkan teks "Masukkan angka:" dan membaca nilai yang dimasukkan melalui `fmt.Scan(&n)`. Jika nilai yang diterima adalah 0 atau 1, program langsung mencetak 1 dan berhenti, karena faktorial dari 0 maupun 1 bernilai 1. Jika angkanya lebih besar, program membuat variabel i dengan nilai awal yang sama seperti input tersebut. Selanjutnya program masuk ke dalam loop `for {}` tanpa batas. Di dalam loop, program mengecek apakah i sudah mencapai angka 1. Jika iya, program mencetak "1" tanpa tambahan "x", lalu menghentikan loop dengan `break`. Jika belum, program mencetak nilai i diikuti " x ", kemudian menurunkan nilai i satu angka (`i--`). Dengan cara ini, output akhirnya berupa deretan seperti "7 x 6 x 5 x ... x 1" sesuai angka yang dimasukkan.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

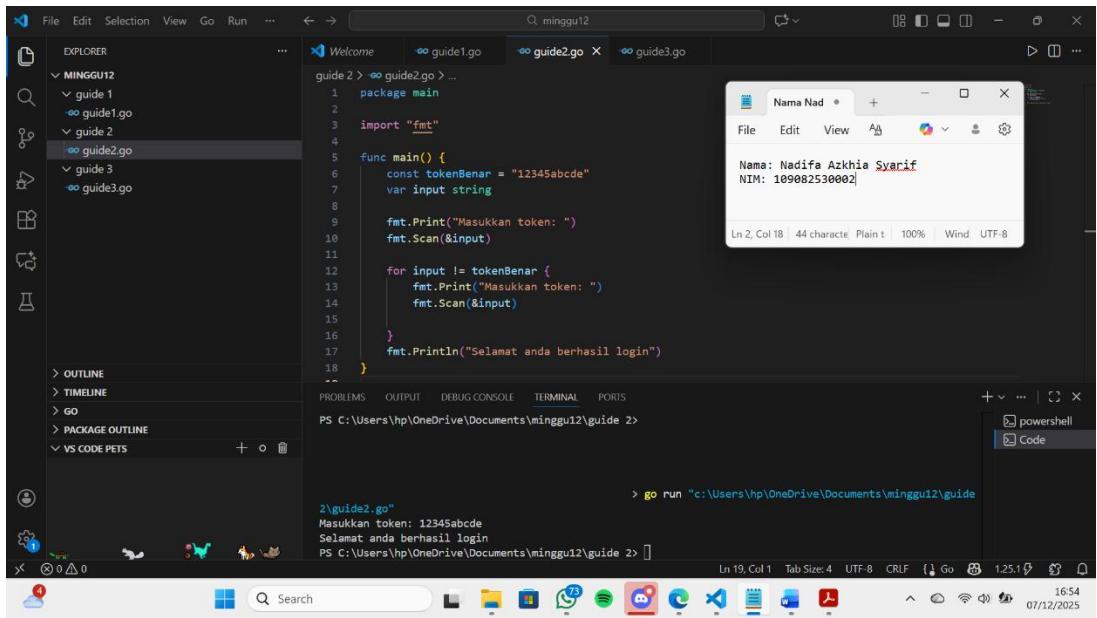
func main() {
    const tokenBenar = "12345abcde"
    var input string

    fmt.Print("Masukkan token: ")
    fmt.Scan(&input)

    for input != tokenBenar {
        fmt.Print("Masukkan token: ")
        fmt.Scan(&input)
    }

    fmt.Println("Selamat anda berhasil login")
}
```

**Screenshot program**



### Deskripsi program

Program di atas berfungsi sebagai pengecekan token sederhana. Di awal, program menentukan sebuah konstanta bernama tokenBenar yang berisi nilai “12345abcde”. Setelah itu disiapkan variabel input untuk menampung token yang dimasukkan. Program menampilkan pesan “*Masukkan token:*” lalu membaca nilai yang dimasukkan melalui `fmt.Scan(&input)`. Setelah input pertama diterima, program masuk ke dalam loop for yang akan terus berjalan selama nilai input tidak sama dengan tokenBenar. Selama kondisi belum terpenuhi, program akan terus meminta token kembali dengan mencetak “*Masukkan token:*” dan membaca input ulang. Jika akhirnya token yang dimasukkan cocok dengan tokenBenar, loop berhenti dan program menampilkan pesan “*Selamat anda berhasil login*”. Dengan alur ini, program memastikan bahwa proses login hanya berhasil jika token yang dimasukkan tepat.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan N: ")
    fmt.Scan(&n)
```

```
//  
a := 0  
b := 1  
  
fmt.Println(a, " ", b)  
  
i := 2  
  
for i < n {  
    c := a + b  
    fmt.Println(" ", c)  
  
    a = b  
    b = c  
    i++  
}  
}
```

### Screenshoot program

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

- EXPLORER**: Shows a folder named "MINGGU12" containing three sub-folders: "guide 1", "guide 2", and "guide 3", each with a corresponding ".go" file.
- EDITOR**: Displays the content of "guide3.go". The code is as follows:

```
guide 3 > guide3.go > main
func main() {
    n := Scan(&n)
    if n <= 0 {
        Print("Masukkan N: ")
        ReadLine()
        Exit(1)
    }
    a := 0
    b := 1
    fmt.Println(a, " ", b)

    i := 2
    for i < n {
        c := a + b
        fmt.Println(" ", c)
        a = b
        b = c
        i++
    }
}
```

- TERMINAL**: Shows the command line output of running the program:

```
PS C:\Users\hp\OneDrive\Documents\minggu12\guide 3> go run "c:\Users\hp\OneDrive\Documents\minggu12\guide 3\guide3.go"
Masukkan N: 5
0 1 1 2 3
PS C:\Users\hp\OneDrive\Documents\minggu12\guide 3>
```

- OUTPUT**: Shows the output of the program: "0 1 1 2 3".
- STATUS BAR**: Shows the current file is "guide3.go", line 2, column 18, 44 characters, Plain text, 100% zoom, and UTF-8 encoding.

## Deskripsi program

Program di atas berfungsi untuk menampilkan deret Fibonacci sebanyak N angka. Setelah variabel n disiapkan, program meminta input melalui “Masukkan N:” dan membaca nilainya dengan `fmt.Scan(&n)`. Dua nilai awal deret Fibonacci, yaitu  $a = 0$  dan  $b = 1$ , langsung dicetak karena keduanya selalu menjadi dua angka pertama dalam deret. Selanjutnya program menyiapkan variabel i dengan nilai awal 2 sebagai penanda posisi selanjutnya. Program kemudian masuk ke dalam loop for yang berjalan selama  $i < n$ . Di dalam loop, program menghitung angka Fibonacci berikutnya dengan rumus  $c = a + b$ , lalu mencetak nilai c. Setelah itu, nilai a digeser menjadi b, dan b digeser menjadi c supaya perhitungan berikutnya mengikuti pola deret Fibonacci. Nilai i ditambah satu setiap iterasi. Dengan alur seperti ini, program menghasilkan deret Fibonacci mulai dari angka pertama sampai jumlah yang ditentukan oleh input.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    const userBenar = "Admin"
```

```

const passBenar = "Admin"

var u, p string
gagal := 0

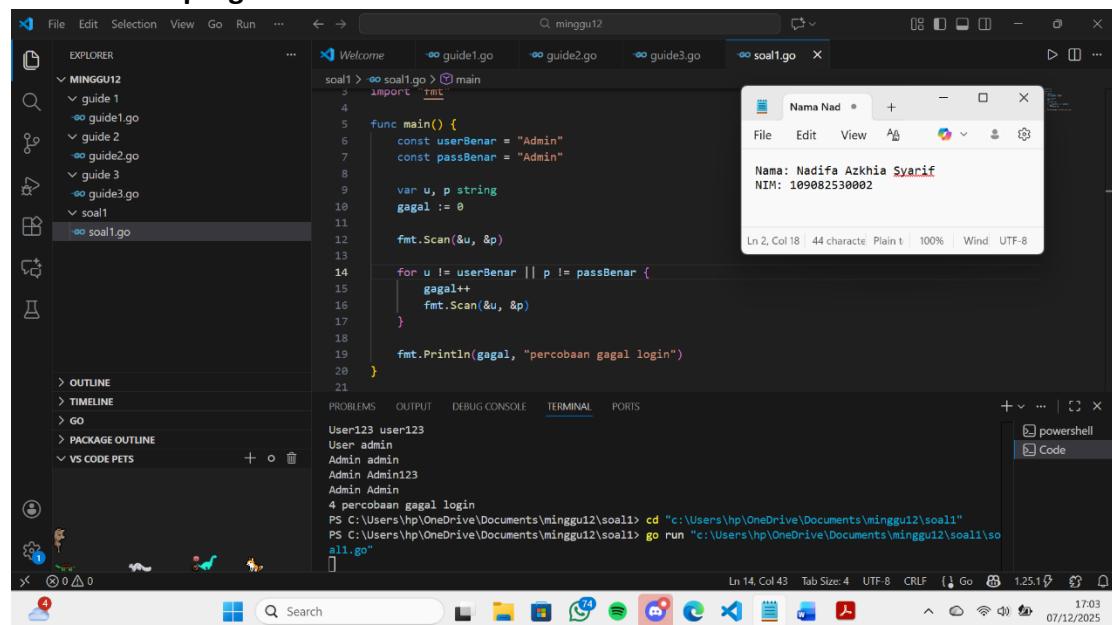
fmt.Scan(&u, &p)

for u != userBenar || p != passBenar {
    gagal++
    fmt.Scan(&u, &p)
}

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

```

## Screenshoot program



## Deskripsi program

Program ini bekerja dengan cara membaca dua input berupa username dan password. Setelah input pertama diterima, program memeriksa apakah keduanya sama dengan nilai yang benar, yaitu "Admin". Jika tidak sama, program masuk ke dalam perulangan yang terus berjalan selama kondisi masih salah. Setiap kali perulangan terjadi, nilai

gagal ditambah satu sebagai penanda bahwa percobaan login tidak berhasil. Setelah itu program membaca input baru lagi untuk dicek ulang. Perulangan berhenti ketika kombinasi username dan password sudah benar. Pada akhir program, jumlah percobaan gagal dicetak dalam format “x percobaan gagal login”.

## 2. Tugas 2

### Source code

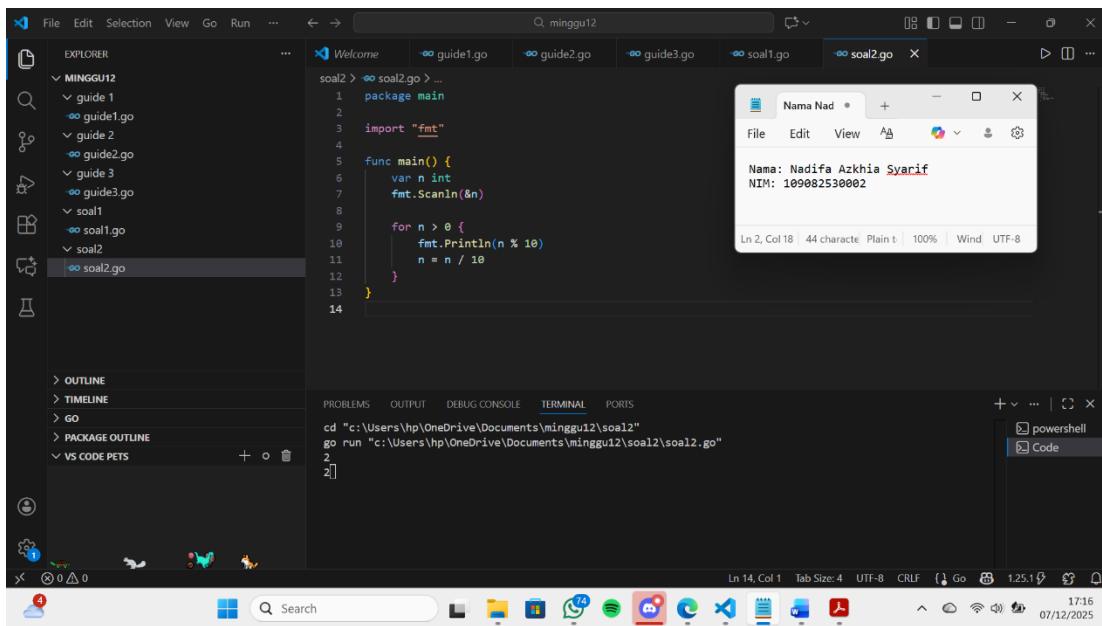
```
package main

import "fmt"

func main() {
    var n int
    fmt.Scanln(&n)

    for n > 0 {
        fmt.Println(n % 10)
        n = n / 10
    }
}
```

### Screenshoot program



### Deskripsi program

Program membaca sebuah bilangan menggunakan Scanln agar input satu angka dapat terbaca dengan benar. Setelah bilangan tersimpan di dalam variabel n, program menjalankan perulangan yang terus berjalan selama nilai n masih lebih besar dari nol. Pada setiap langkah, digit paling kanan diambil dengan operasi n % 10 dan langsung ditampilkan. Setelah itu nilai n diperkecil dengan membaginya dengan 10 sehingga digit berikutnya dapat diproses pada iterasi selanjutnya. Perulangan berhenti ketika seluruh digit sudah diambil dan nilai n menjadi nol.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

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

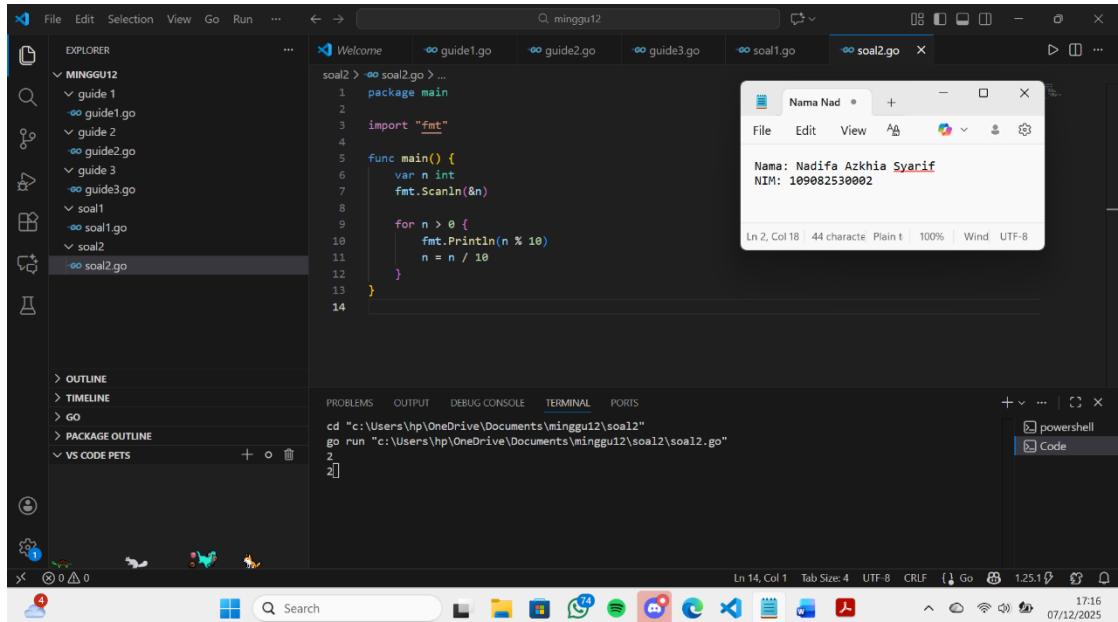
    hasil := 0

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

```
fmt.Println(hasil)
```

```
}
```

## Screenshot program



### Deskripsi program

Program ini membaca dua bilangan bulat positif  $x$  dan  $y$  sebagai masukan. Untuk menghitung hasil pembagian integer tanpa menggunakan operator pembagian, program menggunakan perulangan yang terus berjalan selama nilai  $x$  masih lebih besar atau sama dengan  $y$ . Pada setiap iterasi, nilai  $x$  dikurangi dengan  $y$ , dan variabel hasil ditambah satu. Setiap pengurangan mewakili satu kali pembagian  $y$  dari  $x$ . Ketika nilai  $x$  sudah lebih kecil dari  $y$ , perulangan berhenti karena tidak dapat dilakukan pembagian lagi. Nilai pada variabel hasil kemudian ditampilkan sebagai hasil pembagian integer dari  $x$  oleh  $y$ .