

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 12  
WHILE-LOOP**



**Disusun oleh:**

**Didi Hermawanto**

**109082500088**

**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.Print("1")
}
```

#### Screenshoot program

The screenshot displays a Go IDE with the source code for a program that prints a descending multiplication sequence. The code is as follows:

```
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.Print("1")
}
```

The terminal output shows the program being run multiple times with different inputs (0, 1, 5, 10). The output for input 5 is "5 x 4 x 3 x 2 x 1" and for input 10 is "10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1".

Below the code editor, there is a small window titled "NIP" containing the following information:

NIM	:10908250088
KELAS	:S1IF-13-07
NAMA	:Didi Hermawanto

The status bar at the bottom indicates the current file is "guided1.go" at line 15, column 2, with a tab size of 4, UTF-8 encoding, and CRLF line endings. It also shows the Go version as 1.25.1 and the Go Live extension.

### Deskripsi program

Program ini digunakan untuk mencetak deret angka menurun dalam format perkalian. Setelah pengguna memasukkan sebuah angka, nilainya disimpan ke variabel `n` lalu disalin ke `j` sebagai penghitung. Selama `j` lebih besar dari 1, program mencetak nilai `j` dan tanda " x ", kemudian mengurangi nilai `j` satu per satu. Ketika perulangan selesai, program mencetak angka "1" sebagai bagian akhir deret. Kode `fmt.Scan(&n)` membaca input, `j = n` mengatur nilai awal penghitung, perintah `for j > 1` mengatur proses pengulangan, `fmt.Print(j, " x ")` menampilkan angka, dan `j = j - 1` menurunkan nilai hingga mencapai akhir. Hasil akhirnya berupa tampilan seperti "5 x 4 x 3 x 2 x 1". Alur algoritmanya sederhana: program membaca input angka, menginisialisasi variabel penghitung, menjalankan proses pengulangan untuk mencetak angka dari yang terbesar hingga mendekati batas akhir, lalu menutup keluaran dengan angka 1 sebagai elemen terakhir.

## 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 IDE with a file named `guided2.go`. The code defines a `main` function that prompts the user for a token and checks if it matches `"12345abcde"`. If it matches, it prints `"Selamat Anda Berhasil Login"`. The terminal shows the program being run, with input tokens `Qwe12312`, `231234`, `123lijwe`, and `12345abcde`. The output shows the success message after the correct token is entered.

```
guided2.go 1 X
guided2.go > main
1 package main
2 import "fmt"
3
4 func main() {
5     var token string
6     fmt.Scan(&token)
7
8     for token != "12345abcde" {
9         fmt.Scan(&token)
10    }
11
12    fmt.Println("Selamat Anda Berhasil Login")
13 }
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\coding didi\golang\Laprak 12> go run "c:\coding didi\golang\Laprak 12\tempCodeRunnerFile.go"

Qwe12312  
231234  
123lijwe  
12345abcde  
Selamat Anda Berhasil Login  
PS C:\coding didi\golang\Laprak 12>

### Deskripsi program

Program ini digunakan untuk melakukan pengecekan token sederhana sebelum pengguna dinyatakan berhasil masuk. Setelah program dijalankan, pengguna diminta memasukkan sebuah token yang disimpan ke dalam variabel token. Program kemudian memeriksa apakah token tersebut sama dengan nilai yang dianggap benar, yaitu `"12345abcde"`. Selama token yang dimasukkan tidak cocok, perulangan `for` akan terus meminta pengguna menginput token baru. Jika akhirnya token yang dimasukkan sesuai, perulangan berhenti dan program menampilkan pesan `"Selamat Anda Berhasil Login"`. Alur algoritmanya dimulai dari membaca input token, mengecek kecocokannya, mengulangi permintaan input jika salah, lalu mengakhiri proses dengan mencetak pesan keberhasilan saat token memenuhi syarat.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var N, s1, s2, next int
    fmt.Scan(&N)
```

```
s1 = 0
s2 = 1

for i := 0; i < N; i++ {
    fmt.Print(s1, " ")
    next = s1 + s2
    s1 = s2
    s2 = next
}
```

## Screenshoot program

The screenshot shows a Go program in VS Code. The code defines a `main` function that reads an integer `N` and a string `s1`, then prints a sequence of numbers and the string `s1` for `i` from 0 to `N-1`. The terminal output shows the program was run with `N=5` and `s1="0 1 1 2 3"`, resulting in the output `0 1 1 2 3`. A variable viewer window is open, showing the values of `NIM`, `KELAS`, and `NAMA`.

```
1 package main
2 import "fmt"
3
4 func main() {
5     var N, s1, s2, next int
6     fmt.Scan(&N)
7
8     s1 = "0 1 1 2 3"
9     s2 = "0 1 1 2 3"
10
11     for i := 0; i < N; i++ {
12         fmt.Print(s1, " ")
13         next = s1 + s2
14         s1 = s2
15         s2 = next
16     }
17 }
```

Terminal Output:

```
PS C:\coding didi\golang\Laprak 12> go run "c:\coding didi\golang\Laprak 12\guided3.go"
5
0 1 1 2 3
PS C:\coding didi\golang\Laprak 12> go run "c:\coding didi\golang\Laprak 12\guided3.go"
2
0 1
PS C:\coding didi\golang\Laprak 12>
```

Variable Viewer:

Variable	Value
NIM	109082500088
KELAS	S1IF-13-07
NAMA	Didi Hermawanto

### Deskripsi program

Program ini digunakan untuk menampilkan deret Fibonacci sebanyak N angka sesuai input pengguna. Setelah nilai N dimasukkan, program mengatur dua nilai awal Fibonacci yaitu  $s1 = 0$  dan  $s2 = 1$ . Dalam perulangan, program mencetak nilai  $s1$  terlebih dahulu, lalu menghitung angka berikutnya dengan menjumlahkan  $s1$  dan  $s2$  dan menyimpannya ke variabel  $next$ . Setelah itu, nilai  $s1$  digeser menjadi  $s2$  dan  $s2$  diganti menjadi  $next$ , sehingga perhitungan berlanjut ke angka Fibonacci berikutnya. Proses ini berlangsung sampai perulangan mencapai N kali. Alur algoritmanya dimulai dari membaca nilai N, menyiapkan dua angka awal deret, menampilkan angka dari nilai terkecil, menghitung angka berikutnya, lalu menggeser nilai untuk iterasi selanjutnya hingga semua angka Fibonacci yang dibutuhkan tercetak.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var user, pass string
    gagal := 0

    for {
        fmt.Scan(&user, &pass)

        if user == "Admin" && pass == "Admin" {
            break
        } else {
            gagal++
        }
    }

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

## Screenshoot program

```
soal1.go 1 X
soal1.go > main

6  gagal := 0
7
8  for {
9      fmt.Scan(&user, &pass)
10
11      if user == "Admin" && pass == "Admin" {
12          break
13      } else {
14          gagal++
15      }
16  }
17  fmt.Printf("%d percobaan gagal login\n", gagal)
```

```
PS C:\coding didi\golang\Laprak 12> go run "c:\coding didi\golang\Laprak 12\soal1.go"
User123 user123
User admin
Admin admin
Admin Admin123
Admin Admin
4 percobaan gagal login
PS C:\coding didi\golang\Laprak 12> go run "c:\coding didi\golang\Laprak 12\soal1.go"
Admin Admin
0 percobaan gagal login
```

## Deskripsi program

Program ini digunakan untuk menghitung berapa kali seorang pengguna gagal melakukan login akibat salah memasukkan username atau password. Ketika program dijalankan, pengguna diminta memasukkan dua input berupa username dan password. Perulangan terus berjalan karena menggunakan for {} tanpa syarat, sehingga program akan selalu meminta input sampai kondisi login benar terpenuhi. Di dalam perulangan, program memeriksa apakah username dan password sama-sama bernilai "Admin". Jika benar, perulangan dihentikan dengan break, menandakan login berhasil. Jika salah, variabel gagal ditambah satu untuk mencatat jumlah kegagalan login. Setelah keluar dari perulangan, program menampilkan jumlah total percobaan yang salah melalui fungsi fmt.Printf. Alur algoritmanya dimulai dari membaca input, mengecek kecocokan data, mencatat jumlah kesalahan setiap kali input tidak sesuai, lalu menampilkan total kegagalan setelah pengguna berhasil login.

## 2. Tugas 2

### Source code

```
package main
```

```

import "fmt"

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

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

```

### Screenshoot program

```

soal2.go 1 X
soal2.go > main
1 package main
2 import "fmt"
3
4 func main() {
5     var n int
6     fmt.Scan(&n)
7
8     for n > 0 {
9         digit := n % 10
10        fmt.Println(digit)
11        n = n / 10
12    }
}

```

Windsurf: Refactor | Explain | Generate GoDoc | X

Debug Console (Ctrl+Shift+Y)

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\coding didi\golang\Laprak 12> go run "c:\coding didi\golang\Laprak 12\soal2.go"
2544
4
4
5
2
2
PS C:\coding didi\golang\Laprak 12> go run "c:\coding didi\golang\Laprak 12\soal2.go"
2
2
2

```

Ln 9, Col 29 Tab Size: 4 UTF-8 CRLF Go 1.25.1 Go Live Windsurf: Login Prettier

### Deskripsi program

Program ini digunakan untuk memecah dan menampilkan setiap digit dari sebuah bilangan bulat positif dimulai dari digit paling kanan. Setelah angka dimasukkan oleh pengguna, program menjalankan perulangan yang terus berlangsung selama nilai  $n$  masih lebih besar dari nol. Pada setiap langkah, program mengambil digit terakhir dari



bilangan dengan operasi  $n \% 10$  dan langsung menampilkannya ke layar. Setelah digit tersebut dicetak, bilangan dibagi sepuluh menggunakan pembagian bulat  $n = n / 10$  untuk menghilangkan digit yang sudah diproses. Proses ini berulang hingga seluruh digit habis dan nilai  $n$  menjadi nol.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

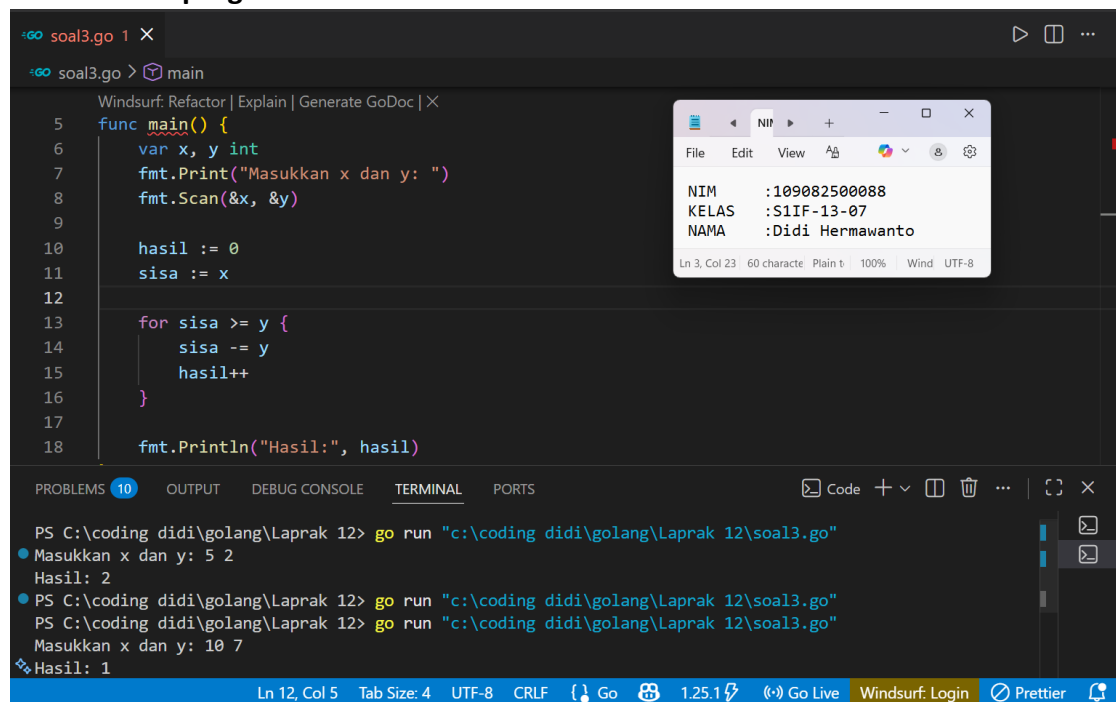
func main() {
    var x, y int
    fmt.Print("Masukkan x dan y: ")
    fmt.Scan(&x, &y)

    hasil := 0
    sisa := x

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

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

#### Screenshoot program



The screenshot shows a code editor with the following Go source code:

```
5 func main() {
6     var x, y int
7     fmt.Print("Masukkan x dan y: ")
8     fmt.Scan(&x, &y)
9
10    hasil := 0
11    sisa := x
12
13    for sisa >= y {
14        sisa -= y
15        hasil++
16    }
17
18    fmt.Println("Hasil:", hasil)
19 }
```

The terminal output shows the program being run twice:

```
PS C:\coding didi\golang\Laprak 12> go run "c:\coding didi\golang\Laprak 12\soal3.go"
Masukkan x dan y: 5 2
Hasil: 2
PS C:\coding didi\golang\Laprak 12> go run "c:\coding didi\golang\Laprak 12\soal3.go"
Masukkan x dan y: 10 7
Hasil: 1
```

## Deskripsi program

Program ini dibuat untuk menghitung hasil pembagian bilangan bulat secara integer tanpa memakai operator pembagian. Pengguna memasukkan dua bilangan, yaitu  $x$  sebagai nilai yang akan dibagi dan  $y$  sebagai pembaginya. Program kemudian memakai perulangan yang bekerja selama nilai *sisa* masih lebih besar atau sama dengan  $y$ . Pada setiap iterasi, nilai *sisa* dikurangi dengan  $y$ , dan setiap pengurangan dihitung sebagai satu langkah pembagian. Setelah *sisa* tidak lagi cukup besar untuk dikurangi, loop berhenti dan nilai *hasil* yang mewakili berapa kali  $y$  dapat masuk ke dalam  $x$  ditampilkan. Secara alurnya, program dimulai dengan membaca input, menginisialisasi variabel *hasil* dan *sisa*, menjalankan proses pengurangan berulang, menghitung jumlah pengurangan sebagai hasil pembagian, lalu menampilkan nilai akhir kepada pengguna. Program ini pada dasarnya meniru cara kerja pembagian manual menggunakan konsep pengurangan berulang.