

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 12

WHILE-LOOP



**Telkom
University
PURWOKERTO**

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")
}
```

Screenshot program

The screenshot shows a code editor with a Go file named `guided1.go`. The code prints a diamond shape of asterisks based on user input. A terminal window is open below the editor, showing the execution of the program and its output.

`guided1.go` content:

```
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")
}
```

Terminal Output:

```
PS C:\coding\didid\golang\Laprak 12> go run "c:\coding\didid\golang\Laprak 12\guided1.go"
PS C:\coding\didid\golang\Laprak 12> go run "c:\coding\didid\golang\Laprak 12\guided1.go"
0
1
● PS C:\coding\didid\golang\Laprak 12> go run "c:\coding\didid\golang\Laprak 12\guided1.go"
5
5 x 4 x 3 x 2 x 1
● PS C:\coding\didid\golang\Laprak 12> go run "c:\coding\didid\golang\Laprak 12\guided1.go"
10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
● PS C:\coding\didid\golang\Laprak 12> go run "c:\coding\didid\golang\Laprak 12\guided1.go"
```

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.Println(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")
}
```

Screenshot program

```

:go guided2.go 1 X
:go guided2.go > main
1 package main
2 import "fmt"
3
Windsurf: Refactor | Explain | Generate GoDoc | X
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> []

Ln 6, Col 21 Tab Size: 4 UTF-8 CRLF { } Go 1.25.1 ⚡ (↻) Go Live Windsurf: Login ⚡ Prettier 🔍

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.Println(s1, " ")
        next = s1 + s2
        s1 = s2
        s2 = next
    }
}

```

Screenshot program

The screenshot shows a Go code editor interface with the following details:

- Code Editor Area:**

```

go guided3.go 1 ×
go guided3.go > main
import "fmt"
Windsurf: Refactor | Explain | Generate GoDoc | X
func main() {
    var N, s1, s2, next int
    fmt.Scan(&N)

    s1 = 0
    s2 = 1

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

```
- Output Terminal:**

```

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

```
- Code Snippet Preview:**

```

File Edit View Aa ⚡ 🌐 ⚙️
NIM :109082500088
KELAS :S1IF-13-07
NAMA :Didi Hermawanto
Ln 3, Col 23 | 60 character Plain t 100% Wind UTF-8

```
- Bottom Status Bar:**

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + ⚡ 🌐 ⚙️ ... | ⚙️ X

PS C:\coding\didid\golang\Laprak 12> go run "c:\coding\didid\golang\Laprak 12\guided3.go"
5
0 1 1 2 3
PS C:\coding\didid\golang\Laprak 12> go run "c:\coding\didid\golang\Laprak 12\guided3.go"
2
0 1
PS C:\coding\didid\golang\Laprak 12>
Ln 16, Col 6 Spaces: 4 UTF-8 CRLF { Go ⚡ 1.25.1 ⚡ ⚡ Go Live Windsurf: Login Prettier ⚡

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

The screenshot shows a Go development environment. On the left, the code editor displays a file named 'soal1.go' with the following content:

```
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)
```

On the right, a terminal window shows the execution of the program:

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

A modal window is open in the top right corner, displaying student information:

NIM	: 109082500088
KELAS	: S1IF-13-07
NAMA	: Didi Hermawanto

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
    }
}

```

Screenshot program

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

- Code Editor:** The file `soal2.go` is open, showing the provided Go code. A floating window titled "Windsurf" displays student information: NIM :109082500088, KELAS :S1IF-13-07, and NAMA :Didi Hermawanto.
- Terminal:** The terminal shows the command `go run "c:\coding\didil\golang\Laprak 12\soal2.go"` being run twice, resulting in the output 2544 and 2.
- Status Bar:** The status bar at the bottom shows "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)
}
```

Screenshot program

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

- Editor:** The main editor window displays a Go program named `soal3.go`. The code implements the Euclidean algorithm to find the greatest common divisor (GCD) of two numbers, `x` and `y`.
- Terminal:** Below the editor is a terminal window showing the execution of the program. It prompts the user to input values for `x` and `y`, and then prints the result "Hasil:" followed by the calculated GCD.
- Output:** A floating terminal window titled "Windsurf: Login" shows the program's output: NIM :109082500088, KELAS :S1IF-13-07, and NAMA :Didi Hermawanto.
- Bottom Bar:** The bottom bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS, along with icons for Code, +, and other tools like Prettier and Go Live.

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.