

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 12

DO WHILE



Disusun oleh:

Andra Dwicki Saputra

109082500206

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

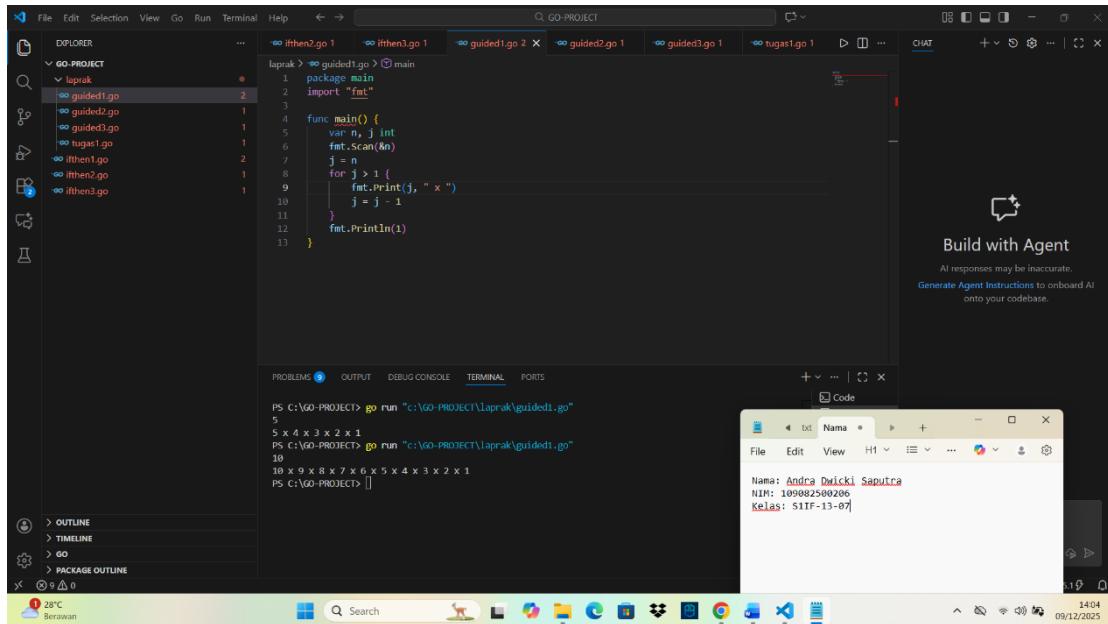
Source Code

```
package main

import "fmt"

func main() {
    var n, j int
    fmt.Scan(&n)
    j = n
    for j > 1 {
        fmt.Println(j, " x ")
        j = j - 1
    }
    fmt.Println(1)
}
```

Screenshot program



Deskripsi program

program tersebut berfungsi untuk menampilkan urutan perkalian turun dari suatu bilangan sampai angka 1, mirip seperti cara menuliskan proses faktorial secara manual. Pertama, program meminta pengguna memasukkan sebuah angka melalui `fmt.Scan(&bilangan)`. Lalu nilai tersebut disalin ke variabel `j`, yang akan digunakan sebagai penghitung dalam perulangan. Program kemudian menjalankan perulangan `for j > 1`, yang berarti selama nilai `j` masih lebih besar dari 1, program akan mencetak nilai `j` diikuti tanda “`x`”, lalu nilai `j` dikurangi satu setiap iterasi. Ketika perulangan selesai (karena `j` sudah menjadi 1), program mencetak angka 1 sebagai bagian akhir dari urutan. Hasil akhirnya adalah tampilan rangkaian seperti $5 \times 4 \times 3 \times 2 \times 1$ sesuai bilangan yang dimasukkan pengguna.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var token string
    fmt.Print("Masukkan token: ")
    fmt.Scan(&token)
    for token != "12345abcde" {
        fmt.Print("Masukkan token: ")
        fmt.Scan(&token)
    }
    fmt.Println("Selamat anda berhasil login")
}
```

Screenshot program

The screenshot shows a code editor interface with a dark theme. In the top navigation bar, the tabs are: File, Edit, Selection, View, Go, Run, Terminal, Help. The title bar says "GO-PROJECT". The left sidebar has an "EXPLORER" section showing a project structure under "GO-PROJECT": laprak (with files guided1.go, guided2.go, guided3.go, iithen1.go, iithen2.go, iithen3.go) and a file named "guided2.go" which is currently selected. The code for "guided2.go" is displayed in the main editor area:

```
laprak > go guided2.go > main
1 package main
2 import "fmt"
3
4 func main() {
5     var token string
6     fmt.Print("Masukkan token: ")
7     fmt.Scan(&token)
8     for token != "12345abcde" {
9         fmt.Print("Masukkan token: ")
10        fmt.Scan(&token)
11    }
12    fmt.Println("Selamat anda berhasil login")
13 }
```

To the right of the editor, there is a "Build with Agent" panel with the message: "Build with Agent" and "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase." Below the editor, there is a terminal window showing the command "go run" being run and some sample input and output:

```
PS C:\GO-PROJECT> go run "c:\GO-PROJECT\laprak\guided2.go"
Masukkan token: qerty123
Masukkan token: abc1234
Masukkan token: 12345abcde
Selamat anda berhasil login
PS C:\GO-PROJECT> [
```

At the bottom of the screen, there is a small floating window containing student information:

Nama: Andra Dwiciki Saputra
NIM: 10908250206
Kelas: SIIF-13-07

Deskripsi program

program tersebut digunakan untuk melakukan validasi token hingga pengguna memasukkan token yang benar. Pertama, program meminta input token melalui `fmt.Scan(&token)`. Setelah itu, program menjalankan perulangan `for` dengan kondisi `token != "12345abcde"`, artinya selama token yang dimasukkan tidak sama dengan token yang benar, program akan terus meminta input ulang dari pengguna. Setiap kali token salah, program menampilkan pesan “Masukan Token :” lalu membaca input berikutnya. Ketika akhirnya pengguna memasukkan token yang tepat, perulangan berhenti dan program menampilkan pesan “Selamat anda berhasil login”, yang menandakan validasi berhasil.

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
    }
}
```

Screenshot program

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

- Explorer:** Shows files in the "GO-PROJECT" folder, including "ifthen1.go", "ifthen2.go", "ifthen3.go", "guided1.go", "guided2.go", and "guided3.go".
- Code Editor:** Displays the content of "guided3.go". The code defines a main package with a main function that prints a Fibonacci sequence up to N.
- Terminal:** Shows the command "go run %cd%\GO-PROJECT\laprak\guided3.go" being run, followed by the output of the Fibonacci sequence: 0 1 1 2 3 5 8 13 21 34.
- File Explorer:** A separate window shows a file named "laprak" containing student information: Nama: Andra Dwiciki Saputra, NIM: 10908250206, Kelas: S1IF-13-07.

Deskripsi program

program tersebut menampilkan deret Fibonacci sebanyak N angka berdasarkan input dari pengguna. Pertama, program membaca nilai N , yaitu jumlah angka Fibonacci yang ingin ditampilkan. Dua variabel awal, $s1 = 0$ dan $s2 = 1$, digunakan sebagai dua angka pertama dalam deret Fibonacci. Selanjutnya, program menjalankan perulangan for $j < N$ untuk mencetak setiap angka Fibonacci dimulai dari $s1$. Di dalam loop, program menghitung angka berikutnya menggunakan $\text{temp} = s1 + s2$, lalu menggeser nilai: $s1$ menjadi $s2$, dan $s2$ menjadi angka baru temp . Proses ini diulang sampai jumlah angka yang dicetak mencapai N , sehingga menghasilkan deret Fibonacci sesuai permintaan

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    const Username = "Admin"
    const Password = "Admin"

    var username, password string
    var gagal int

    for {
        fmt.Print("Masukkan Username: ")
        fmt.Scan(&username)

        fmt.Print("Masukkan Password: ")
        fmt.Scan(&password)

        if username == Username && password == Password {
            break
        } else {
            gagal++
            fmt.Println("Username atau password salah! Coba lagi.\n")
        }
    }

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

Screenshot program

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface for a Go project named "GO-PROJECT". The "GO-PROJECT" folder contains several files: "lapak", "guided1.go", "guided2.go", "guided3.go", "tugas1.go", "ifthen1.go", "ifthen2.go", and "ifthen3.go". The "tugas1.go" file is currently open in the editor, displaying the following Go code:

```
func main() {
    const Username = "admin"
    const Password = "Admin"

    var username, password string
    var gagal int

    for {
        fmt.Print("Masukkan Username: ")
        fmt.Scan(&username)

        fmt.Print("Masukkan Password: ")
        fmt.Scan(&password)

        if username == Username && password == Password {
            break
        } else {
            gagal++
            fmt.Println("Username atau password salah! Coba lagi.\n")
        }
    }
}
```

The "TERMINAL" tab shows the command "go run "c:\GO-PROJECT\lapak\tugas1.go"" being run, followed by the program's output:

```
Masukkan Username: Admin
Masukkan Password: admin
Username atau password salah! Coba lagi.

Masukkan Username: admin
Masukkan Password: Admin
Username atau password salah! Coba lagi.

Masukkan Username: Admin
Masukkan Password: Admin
2 percobaan gagal login
PS C:\GO-PROJECT>
```

A separate window titled "Nama" displays the user's information:

Nama:	Andra Dwicki Saputra
NIM:	109082500206
Kelas:	SIIIF-13-07

Deskripsi program

Program tersebut digunakan untuk membuat sistem login sederhana dengan batas percobaan yang dihitung. Pertama, program menyimpan username dan password yang benar dalam konstanta Username dan Password. Lalu program meminta input username dan password dari pengguna dalam sebuah perulangan for tanpa batas. Setiap kali pengguna memasukkan data, program mengecek apakah keduanya sama dengan nilai yang benar; jika cocok, loop akan berhenti menggunakan break, namun jika salah, variabel gagal akan bertambah satu dan program menampilkan pesan “Username atau password salah!”. Setelah login berhasil, program mencetak jumlah percobaan gagal yang telah dilakukan sebelum pengguna memasukkan username dan password yang benar.

2. Tugas 2

Source code

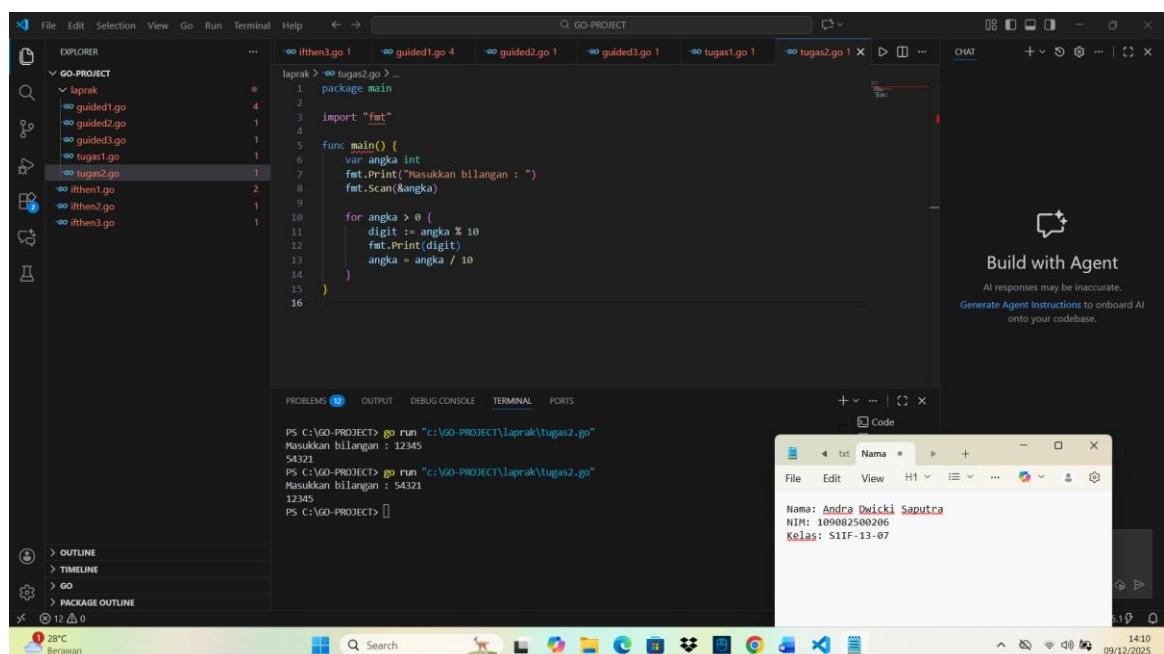
```
package main

import "fmt"

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

    for angka > 0 {
        digit := angka % 10
        fmt.Print(digit)
        angka = angka / 10
    }
}
```

Screenshot program



Deskripsi program

program tersebut digunakan untuk membalik urutan digit dari sebuah bilangan dengan cara mengambil angka dari belakang. Pertama, program meminta pengguna memasukkan sebuah bilangan melalui `fmt.Scan(&angka)`. Setelah itu, program menjalankan perulangan `for angka > 0`, yang berarti loop terus berjalan selama nilai angka masih lebih besar dari 0. Di dalam loop, program mengambil digit terakhir dengan rumus `angka % 10` dan langsung mencetaknya sehingga digit paling belakang muncul terlebih dahulu. Setelah dicetak, angka diperkecil dengan `angka / 10` untuk menghapus digit terakhir tersebut. Proses ini berulang sampai semua digit habis, sehingga hasil akhirnya adalah urutan digit bilangan yang terbalik.

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)

    count := 0

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

    fmt.Println(count)
}
```

Screenshot program

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

- EXPLORER:** Shows the project structure under "GO-PROJECT" with files like guided1.go, guided2.go, guided3.go, tugas1.go, tugas2.go, and tugas3.go.
- CODE EDITOR:** Displays the source code for `tugas3.go`.
- TERMINAL:** Shows the command `go run "c:\GO-PROJECT\laprak\tugas3.go"` being run twice, resulting in the output "1" and "10".
- OUTPUT:** Shows the output from the terminal and a code editor window showing the results of the program execution.
- STATUS BAR:** Shows system information like battery level (28%), date (09/12/2025), and time (14:11).

Deskripsi program

Program tersebut digunakan untuk menghitung berapa kali nilai y dapat dikurangkan dari x sebelum x menjadi lebih kecil dari y. Pertama, program meminta pengguna memasukkan dua bilangan, yaitu x dan y, melalui `fmt.Scan(&x, &y)`. Selanjutnya, variabel count digunakan untuk menghitung jumlah pengurangan yang berhasil dilakukan. Pada bagian perulangan `for x >= y`, program akan terus melakukan $x = x - y$ selama nilai x masih lebih besar atau sama dengan y, dan setiap pengurangan akan menambah nilai count sebesar 1. Ketika x sudah lebih kecil dari y, perulangan berhenti, dan program menampilkan nilai count, yang merupakan jumlah total pengurangan. Program ini pada dasarnya meniru proses pembagian tetapi menggunakan pengurangan berulang untuk mendapatkan hasilnya.