

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 12

DO WHILE



Disusun oleh:

MOHAMAD ERLANGGA ZEIN

109082500020

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

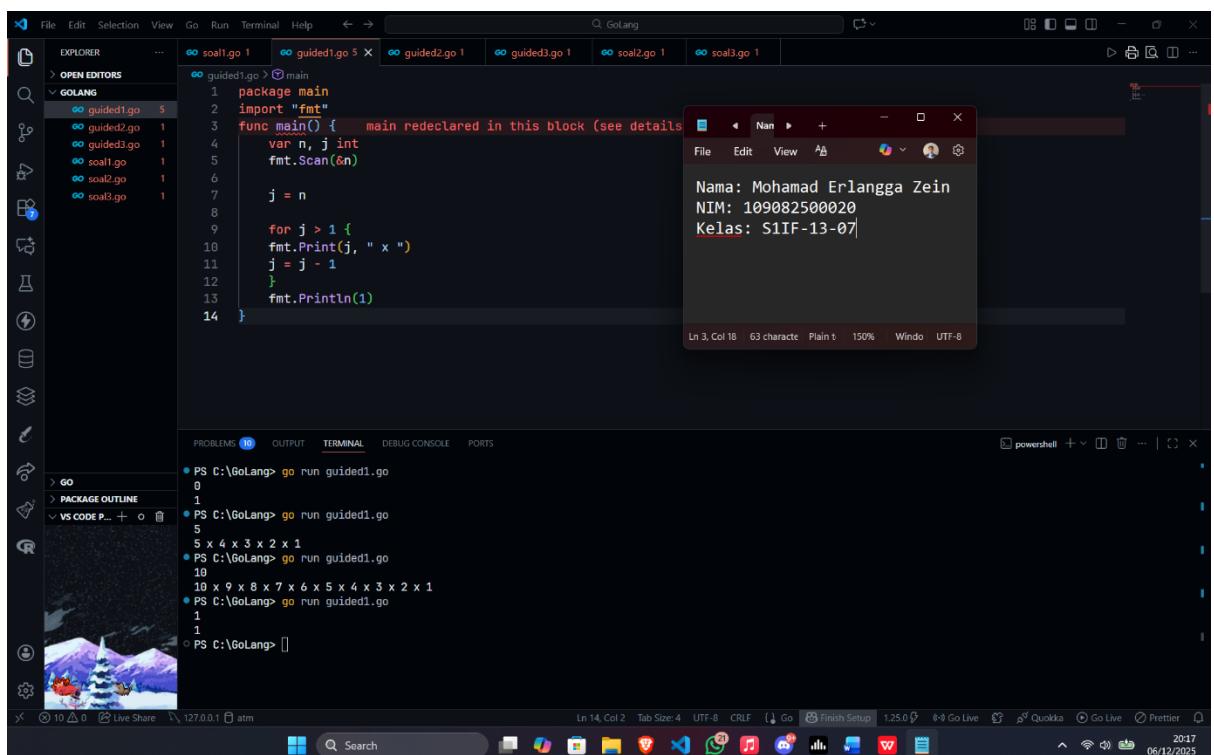
    fmt.Scan(&n)

    j = n

    for j > 1 {
        fmt.Println(j, " x ")
        j = j - 1
    }

    fmt.Println(1)
}
```

Screenshot program



Deskripsi program

Program ini adalah program untuk menghitung atau penjabaran dari suatu bilangan faktorial, contohnya adalah ketika user memasukkan 5 maka ini diartikan juga sebagai $5!$ yang dimana $5!$ ketika dijabarkan atau dihitung adalah $= 5 \times 4 \times 3 \times 2 \times 1$, dan untuk $0!$ Itu $= 1$ dan $1!$ Juga $= 1$, disini menggunakan algoritma do-while, namun dalam bahasa go, do while tidak terdapat karena alasan untuk efisiensi, sehingga hanya ada for dalam bahasa go, sehingga dalam algoritma nya menggunakan for, disini menggunakan 2 variabel yang bertipe data int, yaitu j & n, dimana nilai dari j sama dengan nilai n yang telah di input oleh user, kemudian di dalam for, kita isikan perintahnya, karena disini tugasnya untuk menampilkan output dari penjabaran faktorial, maka yang mekanismenya hampir mirip adalah dengan melakukan decrement(penurunan/--) terhadap angka yang telah di inputkan oleh user, namun dengan mendeklarasinya, dimana untuk perintah dari for nya itu ketika nilai $j > 1$, dan ketika menginputkan angka nya 0, maka faktorial tersebut akan bernilai 1, yang dimana itu ada di luar for, yang artinya ketika user menginputkan nilai yang nilainya tidak > 1 , maka akan langsung berhenti.

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

The screenshot shows a Visual Studio Code interface with several tabs open in the Explorer sidebar:

- guided1.go
- guided2.go
- guided3.go
- soal1.go
- soal2.go
- soal3.go

The guided2.go tab is active, displaying the following Go 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")
}
```

A red error highlight is shown on the line `func main() {` with the message `main redeclared in this block`. To the right, a terminal window shows the output of running the program:

```
PS C:\GoLang> go run guided2.go
Qwe12312
231234
13213
123Lijwe
12345abcde
Selamat Anda berhasil login
PS C:\GoLang> go run guided2.go
12345abcde
Selamat Anda berhasil login
PS C:\GoLang>
```

The status bar at the bottom indicates the file is 1.25.0, the terminal is in powershell mode, and the date is 06/13/2023.

Deskripsi program

Program tersebut adalah program untuk mengidentifikasi pengisian token yang benar, dimana sebelumnya tokennya sudah diberi tahu bahwa token yang benar adalah “12345abcde”, disini ada 1 variabel yang bertipe data string, kemudian untuk isi dari variabel nya itu sudah diberi tahu sebelumnya yaitu “12345abcde”, sehingga untuk algoritma do-while nya yaitu `for token != "12345abcde"`, yang artinya ketika user menginputkan tokennya selain “12345abcde” maka yang akan terjadi adalah perulangan terus menerus hingga nanti user memasukkan token yang sesuai, jadi ketika user menginputkan 231234, Qwe12312, 13213, dan sebagainya, maka perulangan akan terus berjalan(tidak berhenti), sampai ketika user menginput “12345abcde” maka perulangan akan berhenti dan program akan memberikan keluaran/output “Selamat anda berhasil login”.

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 Visual Studio Code interface for a Go project named 'GoLang'. The left sidebar lists files: 'soal1.go', 'guided1.go', 'guided2.go', 'guided3.go' (the active file), 'soal2.go', and 'soal3.go'. The code editor displays the following Go code:

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

The terminal window shows the output of running the program:

```
PS C:\GoLang> go run guided3.go
5
0 1 1 2 3
PS C:\GoLang> go run guided3.go
2
0 1
PS C:\GoLang> go run guided3.go
10
0 1 1 2 3 5 8 13 21 34
PS C:\GoLang>
```

A floating terminal window titled 'Nan' displays the student's information:

```
Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: S1IF-13-07
```

Deskripsi program

Dalam program ini, seluruh variabel (N , $s1$, $s2$, j , $temp$) dideklarasikan dengan tipe data int (Integer) karena deret Fibonacci hanya melibatkan operasi bilangan bulat tanpa pecahan. Variabel N berfungsi sebagai batas input pengguna untuk menentukan panjang deret, sedangkan j bertugas sebagai penghitung (*counter*) untuk menghentikan perulangan saat jumlah angka yang dicetak sudah terpenuhi. Inti algoritmanya terletak pada interaksi $s1$ (angka saat ini) dan $s2$ (angka berikutnya), di mana variabel $temp$ memegang peran krusial sebagai wadah penyimpanan sementara hasil penjumlahan ($s1 + s2$), tanpa $temp$, nilai penjumlahan akan hilang saat variabel diperbarui, sehingga $temp$ menjamin pergeseran nilai antar-variabel berjalan akurat untuk menghasilkan urutan deret yang benar.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var gagal int

    var username string

    var password string

    gagal = 0


    for {

        fmt.Print("Masukkan username dan password: ")

        fmt.Scan(&username, &password)

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

            break

        } else {

            gagal += 1

        }

    }

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

}
```

Screenshot program

The screenshot shows a GoLang development environment in VS Code. The code editor displays a file named `seal1.go` with the following content:

```
func main() {
    var gagal int
    var username string
    var password string
    gagal = 0

    for {
        fmt.Print("Masukkan username dan password: ")
        fmt.Scan(&username, &password)

        if username == "Admin" && password == "Admin"{
            break
        } else {
            gagal += 1
        }
    }
    fmt.Println(gagal , "percobaan gagal login")
}
```

The terminal window shows the execution of the program:

```
PS C:\GoLang> go run seal1.go
Masukkan username dan password: User123 user123
Masukkan username dan password: User admin
Masukkan username dan password: Admin admin
Masukkan username dan password: Admin Admin123
Masukkan username dan password: Admin Admin
4 percobaan gagal login
PS C:\GoLang> go run seal1.go
Masukkan username dan password: Admin Admin
0 percobaan gagal login
PS C:\GoLang>
```

A modal window titled "Nan" is open, displaying student information:

Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: S1IF-13-07

Deskripsi program

Program ini adalah program untuk menghitung seberapa banyak user mengalami gagal login karena user salah dalam menginputkan username dan password, dimana username dan password yang benar adalah `username = "Admin"` dan `password = "Admin"`, sehingga untuk mempermudah dalam perhitungannya, disini terdapat 3 variabel dengan tipe data yang berbeda, dimana untuk variabel gagal menggunakan tipe data `int`, dan untuk `username` dan `password` menggunakan tipe data `string`, dimana nilai variabel `gagal = 0`, serta dalam program ini menggunakan algoritma `do-while` atau dalam bahasa `go` adalah `for` dan pengkondisian `if else` untuk mempermudah dalam mengidentifikasi suatu nilai, untuk pengkondisian `if` yaitu ketika `if username == "Admin" && password == "Admin" { break }` else `{ gagal += 1 }`, yang artinya ketika user menginputkan `username` dan `admin` nya `"Admin"`, maka program tersebut akan `break/berhenti` dan program mengeluarkan output `"(jumlah gagal), "Percobaan gagal login"` yang artinya ketika user menginputkan `username` dan `password` nya `"Admin"` maka outputnya adalah `"0 Percobaan gagal login"` dan ketika user menginputkan `username` dan `password` nya tidak sesuai, maka akan dihitung `+ 1` pada variabel `gagal`, sehingga ketika user menginputkan 4 kali `username` dan `password` yang tidak sesuai maka output nya `"4 percobaan gagal login"`.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var n, urut int

    fmt.Print("Masukkan angka: ")

    fmt.Scan(&n)

    for n > 0 {
        urut = n % 10

        fmt.Println(urut)

        n = n / 10
    }
}
```

Screenshot program

The screenshot shows a GoLang IDE interface. On the left, the Explorer sidebar lists several files: soal1.go, guided1.go, guided2.go, guided3.go, soal2.go, soal3.go, and soal3.go. The main editor window displays the provided Go code. A terminal window on the right shows the execution of the program:

```
PS C:\GoLang> go run soal2.go
Masukkan angka: 2
2
PS C:\GoLang> go run soal2.go
Masukkan angka: 2544
4
4
5
2
PS C:\GoLang> go run soal2.go
Masukkan angka: 3423554654
4
5
6
4
5
5
3
2
4
3
```

The terminal also displays the input prompt "Masukkan angka: " and the user's responses.

Deskripsi program

Program ini adalah program untuk mengurutkan angka dari angka paling belakang, kemudian output nya diurutkan dari atas sampe bawah, disini terdapat 2 variabel dengan tipe data int, yaitu n dan urut, dimana n adalah bilangan yang diinputkan user, kemudian pada variabel urut nilainya adalah nilai dari n yang diinputkan user di modulus(%) 10, lalu ditampilkan hasil dari nilai variabel urut($n \% 10$), lalu kemudian nilai n nya di bagi dengan $10(n / 10)$, algoritma ini akan diulang terus menerus hingga hasil dari n nya itu habis(0), contohnya disini $2544, 2544$ di modulus 10 hasilnya 4 , kemudian $2544 / 10 = 254, 254 \% 10 = 4, 254 / 10 = 25, 25 \% 10 = 5, 25 / 10 = 2, 2 \% 10 = 2, 2 / 10 = 0$, sehingga hasil akhirnya adalah $4\ 4\ 5\ 2$, ini berlaku untuk semua angka ketika user tidak menginputkan n nya < 0 .

3. Tugas 3

Source code

```
package main
import "fmt"

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

    hasil = 0

    for x >= y {
        x = x - y
        hasil ++
    }
    fmt.Print(hasil)
}
```

Screenshot program

The screenshot shows a GoLang IDE interface. In the top right, there's a terminal window with the following command history:

- PS C:\GoLang> go run soal3.go
- Masukkan bilangan: 5 2
- 2
- PS C:\GoLang> go run soal3.go
- Masukkan bilangan: 10 7
- 1
- PS C:\GoLang> go run soal3.go
- Masukkan bilangan: 120 4
- 30
- PS C:\GoLang>

In the center, there's a code editor window for file `soal3.go` with the following content:

```
package main
import "fmt"

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

    hasil = 0

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

A modal window is open in the top right, displaying the results of the program execution:

Nama: Mohamad Erlangga Zein
NIM: 109082500020
Kelas: S1IF-13-07

Deskripsi program

Program ini adalah program untuk menghitung hasil dari $x \text{ div } y$. Ketika nilai dari $x \geq y$, untuk mengetahui hasilnya, maka perlu untuk tidak langsung menggunakan operator “/”, disini mengambil habis sisa dari pengurangan hingga algoritma $x \geq y$ tidak bisa lagi/false, bisa diambil contoh $x = 5$, $y = 2$, bisa dilihat disini apakah $x \geq y$? ya, $5 \geq 2$? Ya, sehingga variable hasil + 1, maka disini hasil = 1, lalu kemudian digunakanlah perhitungan $x-y$, $5-2 = 3$, Dimana hasil dari pengurangan ini termasuk dalam variable x, maka perulangannya akan berlanjut, $3 \geq 2$? Ya, sehingga nilai variable hasil bertambah dari semula 1, kini $1 + 1 = 2$, kemudian akan berlanjut lagi terhadap x yang selanjutnya, yakni $x = 1$ (hasil dari $3-2$) dan $y = 2$, maka apakah $1 \geq 2$? Tidak, sehingga perulangan akan berhenti, dan hasil akhirnya Adalah hasil = 2. Ini juga berlaku terhadap bilangan yang diinputkan oleh user seterusnya Ketika $x \geq y$.