

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMOGRAMAN 2**

**MODUL 4
PROSEDUR**



Oleh:

HAIKAL SATRIATAMA

2311102066

IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Dasar teori mengenai materi prosedur dalam bahasa Golang didasarkan pada konsep fungsi (function) sebagai unit dasar untuk menjalankan serangkaian instruksi. Dalam Golang, fungsi digunakan untuk memecah program menjadi blok-blok kode yang terorganisir dan modular, memudahkan pemeliharaan dan pembacaan kode. Setiap fungsi di Golang memiliki nama, parameter, dan tipe pengembalian (return type).

Parameter adalah nilai-nilai yang dapat diteruskan ke fungsi untuk diproses, sedangkan tipe pengembalian menentukan nilai apa yang dikembalikan oleh fungsi tersebut setelah eksekusi selesai. Fungsi dapat dipanggil di mana saja dalam program untuk mengurangi pengulangan kode, meningkatkan efisiensi, dan memfasilitasi reuse (penggunaan ulang) kode. Golang juga mendukung fungsi anonim (anonymous function) dan fungsi sebagai nilai (first-class function), yang memungkinkan fleksibilitas lebih dalam pemrograman prosedural.

Prosedur atau fungsi juga mendukung konsep *defer*, *panic*, dan *recover* yang digunakan untuk menangani eksekusi program dalam kondisi tertentu, seperti error atau situasi tak terduga. *Defer* digunakan untuk menunda eksekusi sebuah fungsi hingga fungsi induk selesai dieksekusi, biasanya diterapkan untuk pembersihan sumber daya seperti file atau koneksi database. *Panic* digunakan untuk menangani kondisi fatal yang membuat program tidak dapat dilanjutkan, sedangkan *recover* memungkinkan program untuk melanjutkan eksekusi setelah sebuah *panic* terjadi, dengan cara mengembalikan kontrol dari situasi error. Pendekatan ini menjadikan Golang tidak hanya efisien dalam pengelolaan prosedur, tapi juga tangguh dalam menangani error.

II. GUIDED

1. Guided 1

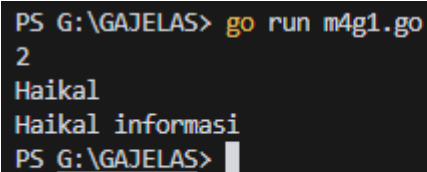
```
package main

import "fmt"

func main() {
    var bilangan int
    var pesan string
    fmt.Scan(&bilangan, &pesan)
    cetakPesan(pesan, bilangan)
}

func cetakPesan(M string, flag int) {
    var jenis string
    if flag == 0 {
        jenis = "error"
    } else if flag == 1 {
        jenis = "warning"
    } else if flag == 2 {
        jenis = "informasi"
    }
    fmt.Println(M, jenis)
}
```

Output



```
PS G:\GAJELAS> go run m4g1.go
2
Haikal
Haikal informasi
PS G:\GAJELAS>
```

Penjelasan

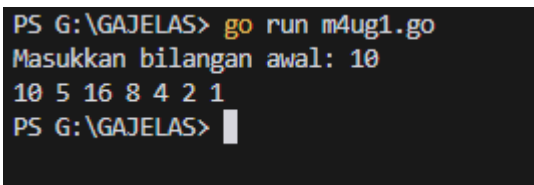
Program ini membaca sebuah bilangan dan pesan dari pengguna, lalu mencetak pesan tersebut dengan jenis "error," "warning," atau "informasi" berdasarkan nilai bilangan (0, 1, atau 2).

III. UNGUIDED

1. Unguided 1

```
package main
import "fmt"
func cetakDeret(n int) {
    for n != 1 {
        fmt.Printf("%d ", n)
        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Println(n)
}
func main() {
    var n int
    fmt.Print("Masukkan bilangan awal: ")
    fmt.Scan(&n)
    if n > 0 && n < 1000000 {
        cetakDeret(n)
    } else {
        fmt.Println("Input harus bilangan positif dan kurang dari 1000000.")
    }
}
```

Screenshoot Output



```
PS G:\GAJELAS> go run m4ug1.go
Masukkan bilangan awal: 10
10 5 16 8 4 2 1
PS G:\GAJELAS>
```

Penjelasan :

Program ini mengimplementasikan algoritma Collatz atau dikenal sebagai "Conjecture $3n + 1$ ". Program dimulai dengan meminta input dari pengguna berupa bilangan bulat positif. Jika bilangan tersebut valid (lebih besar dari 0 dan kurang dari 1.000.000), program akan memprosesnya dengan aturan: jika bilangan genap, bilangan dibagi dua, sedangkan jika ganjil, bilangan tersebut dikalikan tiga lalu ditambah satu. Proses ini berulang hingga bilangan menjadi 1, dan setiap bilangan dalam proses tersebut dicetak ke layar. Jika input tidak valid, program akan menampilkan pesan bahwa input harus berupa bilangan positif dan kurang dari 1.000.000.