

**LAPORAN PRAKTIKUM  
ALGORITMA PEMOGRAMAN 2**

**MODUL 4  
MATERI  
PROSEDUR**



Oleh:

**FEBRIAN FALIH ALWAFI**

**2311102181**

**S1F-11-02**

**S1 TEKNIK INFORMATIKA  
UNIVERSITAS TELKOM PURWOKERTO**

**2024**

## I. DASAR TEORI

Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu instruksi baru yang dibuat untuk mengurangi kerumitan dari brode program yang kompleks pada suatu program yang besar. Prosedur akan menghasilkan suatu akibat atau efek langsung pada program ketika dipanggil pada program utama. Suatu subprogram dihatakan prosedur apabila:

1. Tidak ada deblarasi tipe nilai yang dikembalikan, dan
2. Tidak terdapat hata kunci return dalam badan subprogram

Kedudukannya prosedur sama seperti instruksi dasar yang sudah ada sebelumnya (assignment) danlatau instruksi yang berasal dari paket (fmt), seperti fmt.Scan dan fmt.Print. Karena itu selalu pilih nama prosedur yang berbentuli kata herja atau sesuatu yang merepresentasikan proses sebagai nama dan prosedur Contoh cetak, hitung Rerata, cariNilal, belok, mulai,...

### ➤ Contoh Program dengan Prosedur

Berikut ini adalah contoh penulisan prosedur pada suatu program lengkap.

Buatlah sebuah program beserta prosedur yang digunakan untuk menampilkan suatu pesan error, warning atau informasi berdasarkan masukan dari user.

**Masukan** terdiri dari sebuah bilangan bulat flag (0 s.d. 2) dan sebuah string pesan M.

**Keluaran** berupa string pesan M beserta jenis pesannya, yattu error, warning atau informest berdasarkan nilai flag 0, 1 dan 2 secara berturut-turut.

```
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)
}
```

## ➤ **Parameter**

Suatu subprogram yang dipanggil dapat berkomunikasi dengan pemangglinya melalui argumen yang diberikan melalui parameter yang dideklarasikan pada subprogramnya.

Berdasarkan letak penulisannya pada program, maka parameter dapat dikelompokkan menjadi dua, yaitu parameter formal dan parameter aktual

### **1. Parameter Formal**

Parameter formal adalah parameter yang ditulis pada saat deklarasi suatu subprogram, parameter ini berfungsi sebagai petunjuk bahwa argumen apa saja yang diperlukan pada saat pemanggilan subprogram. Sebagai contoh parameter `jari jari`, `tinggi` pada deklarasi fungsi `volumeTabung` adalah parameter formal (telis berwarna merah). Artinya ketika memanggil `volume Tabung` maka kita harus mempersiapkan dua Integer (berapapun nilainya) sebagai jari jari dan tinggi.

### **2. Parameter Aktual**

Sedangkan parameter aktual adalah argumen yang digunakan pada bagian parameter saat pemanggilan suatu subprogram. Banyaknya argumen dan tipe data yang terdapat pada parameter aktual harus mengikuti parameter formal. Sebagai contoh argumen `r`, `t`, `15`, `14` dan `100` pada contoh kode di atas (teks berwarna biru) adalah parameter aktual, yang menyatakan nilai yang kita berikan sebagai jari-jari dan tinggi.

## II. GUIDED

### 1. Source Code :

```
package main

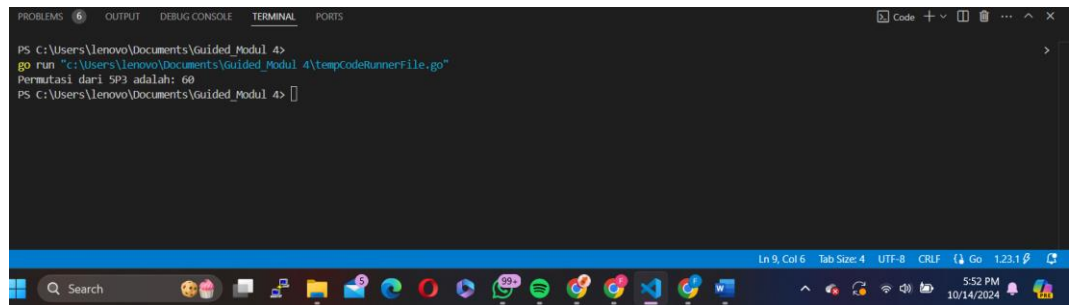
import "fmt"

// Fungsi untuk menghitung faktorial
func factorial(n int) int {
    if n == 0 {
        return 1
    }
    result := 1
    for i := 1; i <= n; i++ {
        result *= i
    }
    return result
}

// Prosedur untuk menghitung dan menampilkan permutasi
func permutasi(n, r int) {
    hasilPermutasi := factorial(n) / factorial(n-r)
    fmt.Printf("Permutasi dari %dP%d adalah: %d\n", n, r,
    hasilPermutasi)
}

func main() {
    // Memanggil prosedur untuk menghitung dan
    menampilkan permutasi
    n, r := 5, 3
    permutasi(n, r)
}
```

Output :



```
PS C:\Users\lenovo\Documents\Guided_Modul_4>
go run "c:\Users\lenovo\Documents\Guided_Modul_4\tempCodeRunnerFile.go"
Permutasi dari 5P3 adalah: 60
PS C:\Users\lenovo\Documents\Guided_Modul_4>
```

### Deskripsi :

Program ini bertujuan untuk menghitung dan menampilkan nilai permutasi dari dua bilangan,  $n$  dan  $r$ . Pertama, program mengimpor package `fmt`, yang digunakan untuk input dan output. Terdapat fungsi `factorial` yang bertugas menghitung faktorial dari bilangan bulat  $n$ . Fungsi ini mengembalikan 1 jika  $n$  adalah 0, dan jika tidak, ia menghitung faktorial dengan mengalikan semua angka dari 1 hingga  $n$ . Selanjutnya, terdapat prosedur permutasi yang menerima dua parameter,  $n$  dan  $r$ , dan menghitung permutasi dengan membagi faktorial dari  $n$  dengan faktorial dari  $(n-r)$ . Hasil perhitungan ini kemudian ditampilkan dengan format yang jelas. Di dalam fungsi `main`, variabel  $n$  dan  $r$  diatur dengan nilai 5 dan 3, lalu prosedur permutasi dipanggil untuk menghitung dan mencetak hasilnya. Program ini sederhana dan efisien, serta dapat dimodifikasi untuk menerima input dari pengguna.

## III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

### 1. Source Code :

```
package main

import "fmt"

// Fungsi untuk mencetak deret berdasarkan aturan tertentu
func cetakDeret(n int) {
    for {
        // Mencetak nilai n
        fmt.Printf("%d ", n)

        // Jika n mencapai 1, keluar dari loop
        if n == 1 {
```

```

        break
    }

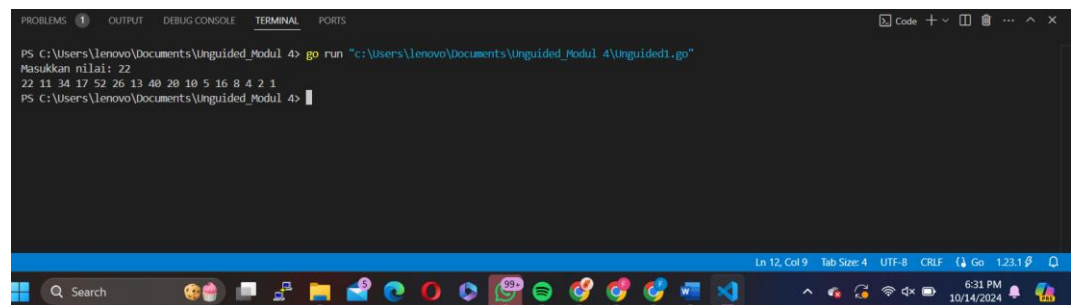
    // Aturan deret: jika genap, bagi dua; jika ganjil, kalikan 3 dan
    // tambahkan 1
    if n%2 == 0 {
        n /= 2
    } else {
        n = 3*n + 1
    }
}
fmt.Println() // Menambahkan baris baru setelah deret
} //Febrian Falih Alwafi
// 2311102181

func main() {
    var n int
    fmt.Print("Masukkan nilai: ")
    fmt.Scan(&n)

    cetakDeret(n) // Memanggil fungsi untuk mencetak deret
}

```

Output :



```

PS C:\Users\lenovo\Documents\Unguided_Modul 4> go run "c:\Users\lenovo\Documents\Unguided_Modul 4\Unguided1.go"
Masukkan nilai: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\lenovo\Documents\Unguided_Modul 4>

```

Deskripsi :

Program ini bertujuan untuk mencetak deret angka berdasarkan “aturan angka”. Terdapat dua fungsi utama: cetakDeret, yang menerima angka bulat positif dan mencetaknya hingga mencapai 1, serta main, yang meminta input pengguna. Dalam fungsi cetakDeret, jika angka genap, dibagi dua; jika ganjil, dikalikan tiga dan ditambah satu. Program ini menunjukkan transisi

angka melalui operasi matematika sederhana, meskipun tidak memvalidasi input untuk memastikan bahwa angka yang dimasukkan positif.