

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 4
PROSEDUR**



Oleh:

MUHAMAD IHSAN

2311102077

IF - 11 – 02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu instruksi baru yang dibuat untuk mengurangi kerumitan dari kode 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 dikatakan prosedur apabila:

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

Kedudukannya prosedur sama seperti instruksi dasar yang sudah ada sebelumnya (assignment) dan/atau instruksi yang berasal dari paket (fmt), seperti fmt.Scan dan fmt.Print. Karena itu selalu pilih nama prosedur yang berbentuk kata kerja atau sesuatu yang merepresentasikan proses sebagai nama dari prosedur. Contoh: cetak, hitung Rerata, carINilal, belok, mulal,

- **Cara pemanggilan procedure**

Seperti yang sudah dijelaskan sebelumnya, suatu prosedur hanya akan dieksekusi apabila dipanggil baik secara langsung atau tidak langsung oleh program utama. Tidak langsung di sini maksudnya adalah prosedur dipanggil oleh program utama melalui perantara subprogram yang lain.

Pemanggilan suatu prosedur cukup mudah, yaitu dengan hanya menuliskan nama beserta parameter atau argumen yang diminta dari suatu prosedur. Sebagai contoh prosedur cetakNFibo di atas dipanggil dengan menuliskan namanya, kemudian sebuah variabel atau nilai integer tertentu sebagai argumen untuk parameter n. Contoh:

	Notasi Algoritma	
1	program contohprosedur	
2	kamus	
3	x : integer	
4	algoritma	
5	x ← 5	
6	cetakNFibo(x)	{cara pemanggilan #1}
7	cetakNFibo(100)	{cara pemanggilan #2}
8	endprogram	
	Notasi dalam bahasa Go	
9	func main() {	
10	var x int	
11	x = 5	
12	cetakNFibo(x)	{cara pemanggilan #1}
13	cetakNFibo(100)	{cara pemanggilan #2}
14	}	

- Contoh Program dengan Procedure

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, yaitu error, warning atau informasi berdasarkan nilai flag 0, 1 dan 2 secara berturut-turut.

```
1 package main
2 import "fmt"
3
4 func main(){
5     var bilangan int
6     var pesan string
7     fmt.Scan(&bilangan, &pesan)
8     cetakPesan(pesan, bilangan)
9 }
10
11 func cetakPesan(M string, flag int){
12     var jenis string = ""
13     if flag == 0 {
14         jenis = "error"
15     }else if flag == 1 {
16         jenis = "warning"
17     }else if flag == 2 {
18         jenis = "informasi"
19     }
20     fmt.Println(M, jenis)
21 }
```

- Parameter

Suatu subprogram yang dipanggil dapat berkomunikasi dengan pemanggilnya melalui argumen yang diberikan melalui parameter yang dideklarasikan pada subprogramnya. Berikut ini jenis atau pembagian dari parameter. Berdasarkan letak penulisannya pada program, maka parameter dapat dikelompokkan menjadi dua, yaitu parameter formal dan parameter aktual.

```
1 func volumeTabung(jari_jari, tinggi int) float64 {
2     var luasAlas, volume float64
3
4     luasAlas = 3.14 * float64(jari_jari * jari_jari)
5     volume = luasAlas * tinggi
6     return volume
7 }
8
9 func main() {
10     var r, t int
11     var v1, v2 float64
12     r = 5; t = 10
13     v1 = volumeTabung(r, t)
14     v2 = volumeTabung(r, t) + volumeTabung(15, t)
15     fmt.Println(volumeTabung(14, 100))
16 }
```

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 (teks berwarna merah). Artinya ketika memanggil `volumeTabung` 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

Guided 1

```
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)

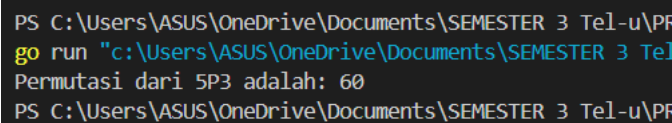
    fmt.Printf("Permutasi dari %dP%d adalah: %d\n", n, r,
        hasilPermutasi)
}

func main() {
    // Memanggil prosedur untuk menghitung dan menampilkan
    permutasi

    n, r := 5, 3
}
```

```
    permutasi(n, r)
}
```

Screenshot output



```
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PR
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PR"
Permutasi dari 5P3 adalah: 60
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PR
```

Deskripsi program

Program ini dirancang untuk menghitung dan menampilkan permutasi dari dua bilangan bulat (n) dan (r). Program terdiri dari beberapa komponen utama, termasuk fungsi factorial yang menghitung faktorial dari (n), dan fungsi permutasi yang menggunakan nilai faktorial tersebut untuk menghitung permutasi serta menampilkan hasilnya. Dalam fungsi main, nilai (n) dan (r) diinisialisasi dengan 5 dan 3, kemudian fungsi permutasi dipanggil untuk menghitung dan menampilkan hasilnya. Sebagai contoh, jika ($n = 5$) dan ($r = 3$), output program akan menunjukkan "Permutasi dari 5P3 adalah: 60". Program ini memberikan cara yang efisien untuk menghitung permutasi dalam konteks kombinatorial.

III. UNGUIDED

Soal Latihan Modul 4

1. Buat program skiena yang akan mencetak setiap suku dari deret yang dijelaskan di atas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal.

procedure cetakDeret(in n : integer)

Masukan berupa satu bilangan integer positif yang lebih kecil dari 1000000.

Keluaran terdiri dari satu baris saja. Setiap suku dari deret tersebut dicetak dalam baris yang dan dipisahkan oleh sebuah spasi.

No	Masukan	Keluaran
1	22	22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Source code

```
package main

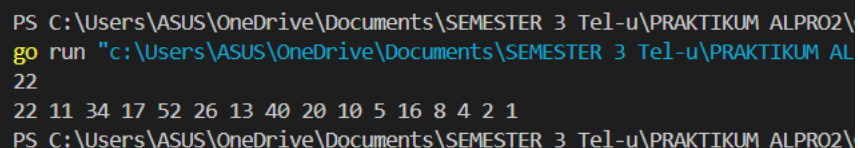
import "fmt"

func deret(n int) {
    var hasil int = n
    fmt.Print(hasil, " ")
    for {
        if hasil%2 == 0 {
            hasil = hasil / 2
            fmt.Print(hasil, " ")
        } else if hasil%2 == 1 {
            hasil = 3*hasil + 1
            fmt.Print(hasil, " ")
        }
        if hasil == 1 {
            break
        }
    }
}

func main() {
    var n int

    fmt.Scan(&n)
    deret(n)
}
```

Screenshot output



```
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\
go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM AL
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\
```

Deskripsi

Program ini menghasilkan deret berdasarkan nilai yang diberikan menggunakan algoritma Collatz. Fungsi deret(n int) menerima parameter

nnn dan mencetak nilai awal serta menghitung nilai baru hingga mencapai 1. Jika nilai genap, dibagi dua; jika ganjil, dihitung dengan rumus $3 \times \text{hasil} + 1$. Fungsi `main()` meminta pengguna memasukkan nilai integer nnn, yang kemudian diproses untuk menghasilkan deret. Sebagai contoh, jika pengguna memasukkan $n=6$, output yang dihasilkan adalah 6 3 10 5 16 8 4 2 1, mengikuti aturan algoritma Collatz.