

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL IV
PROSEDUR**



Oleh:

OKTAVANIA AYU RAHMADANTY

2311102240

S1IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

A. Definisi Prosedur

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, hitungRerata, cariNilal, belok, mulai, ...

B. Deklarasi Prosedur

Berikut ini adalah cara penulisan deklarasi prosedur pada notasi Pseudocode dan Golang.

Notasi Algoritma	
1	procedure <nama procedure> (<params>)
2	kamus
3	{deklarasi variabel lokal dari procedure}
4	...
5	algoritma
6	{badan algoritma procedure}
7	...
8	endprocedure
Notasi dalam bahasa Go	
9	func <nama procedure> (<params>) {
10	/* deklarasi variabel lokal dari procedure */
11	...
12	/* badan algoritma procedure */
13	...
14	}

Penulisan deklarasi ini berada diluar blok yang dari program utama atau func main () pada suatu program Go, dan bisa ditulis sebelum atau setelah dari blok program utama tersebut.

Contoh deklarasi prosedur cetak n nilai pertama dari deret Fibonacci.

	Notasi Algoritma
1	procedure cetakNFibo(in n : integer)
2	kamus
3	f1, f2, f3, i : integer
4	algoritma
5	f2 ← 0
6	f3 ← 1
7	for i ← 1 to n do
8	output(f3)
9	f1 ← f2
10	f2 ← f3
11	f3 ← f1 + f2
12	endfor
13	endprocedure
	Notasi dalam bahasa Go
14	func cetakNFibo(n int) {
15	var f1, f2, f3 int
16	f2 = 0
17	f3 = 1
18	for i := 1; i <= n; i++ {
19	fmt.Println(f3)
20	f1 = f2
21	f2 = f3
22	f3 = f1 + f2
23	}
24	}

Catatan : Kata kunci in pada contoh diatas akan dijelaskan pada materi parameter di modul 5 ini.

C. Cara Pemanggilan Prosedur

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 paramter 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	}

Dari contoh di atas terlihat bahwa cara pemanggilan dengan notasi pseudocode dan Golang adalah sama. Argumen yang digunakan untuk parameter n berupa integer (sesuai deklarasi) yang terdapat pada suatu variable (cara pemanggilan #1) atau nilainya secara langsung (cara pemanggilan #2).

D. Contoh Program Dengan Prosedur

Berikut ini adalah contoh penulisan prosedur pada suatu program lengkap.

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

```

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)
}

```

Penulisan argument parameter cetakPesan(pesan,bilangan) harus sesuai urutan tipe data pada func cetakPesan(M string, flag int), yaitu string kemudian integer.

II. GUIDED

Source Code

```
package main

import "fmt"

// Fungsi buat menghitung faktorial
func factorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    result := 1
    for i := 2; i <= n; i++ {
        result *= i
    }
    return result
}

// Fungsi buat menghitung permutasi
func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

// Fungsi buat menghitung kombinasi
func combination(n, r int) int {
    return factorial(n) / (factorial(r) * factorial(n-r))
}

func main() {
    // Input 4 bilangan
    var a, b, c, d int
    fmt.Println("Masukkan bilangan a, b, c, d (dengan spasi): ")
    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

    // Cek syarat a >= c dan b >= d
    if a >= c && b >= d {
        // Menghitung permutasi dan kombinasi a dan c
        permutasiAC := permutation(a, c)
        kombinasiAC := combination(a, c)

        // Menghitung permutasi dan kombinasi b dan d
        permutasiBD := permutation(b, d)
        kombinasiBD := combination(b, d)

        // Output hasil
        fmt.Println("Permutasi(a, c) dan Kombinasi(a, c):",
            permutasiAC, kombinasiAC)
```

```

        fmt.Println("Permutasi(b, d) dan Kombinasi(b, d):",
permutasiBD, kombinasiBD)
    } else {
        fmt.Println("Syarat a >= c dan b >= d tidak
terpenuhi.")
    }
}

```

Screenshot hasil program

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\SEMESTER 3\LapraK Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 4> go run "d
311102240_Modul 4\Guided\guided1.go"
Masukkan bilangan a, b, c, d (dengan spasi):
5 10 3 10
Permutasi(a, c) dan Kombinasi(a, c): 60 10
Permutasi(b, d) dan Kombinasi(b, d): 3628800 1
PS D:\SEMESTER 3\LapraK Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 4>

```

Penjelasan

Program di atas ditulis dalam bahasa Go untuk menghitung permutasi dan kombinasi dari dua pasang bilangan yang dimasukkan oleh pengguna. Pertama, program meminta pengguna untuk memasukkan empat angka a, b, c, dan d. Kemudian, dilakukan pengecekan apakah $a \geq c$ dan $b \geq d$ sebagai syarat untuk melanjutkan perhitungan. Jika syarat terpenuhi, program akan menghitung permutasi dan kombinasi dari pasangan angka (a, c) dan (b, d). Jika syarat terpenuhi, program akan menghitung permutasi dan kombinasi untuk kedua pasangan bilangan tersebut menggunakan rumus-rumus matematika yang sesuai, kemudian menampilkan hasilnya. Fungsi faktorial yang digunakan dalam perhitungan permutasi dan kombinasi diimplementasikan secara rekursif.

III. UNGUIDED

Source Code

```
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.Printf("%d\n", n)
}

func main() {

    var n int

    fmt.Print("Masukkan nilai awal: ")

    fmt.Scan(&n)

    if n > 0 && n < 1000000 {

        cetakDeret(n)

    } else {

        fmt.Println("Nilai harus lebih besar dari 0 dan kurang dari 1000000.")

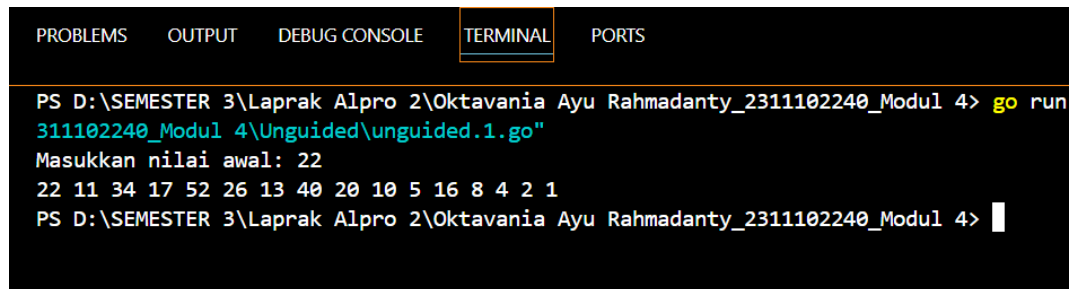
    }

}
```



```
}
```

Screenshot hasil program



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 4> go run
311102240_Modul_4\Unguided\unguided.1.go
Masukkan nilai awal: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\SEMESTER 3\Laparak Alpro 2\Oktavania Ayu Rahmadanty_2311102240_Modul 4> |
```

Penjelasan

Program ini ditulis dalam bahasa Go untuk mencetak deret bilangan berdasarkan aturan Collatz. Dimulai dari bilangan bulat `n`, jika `n` genap, program akan membaginya dengan 2, dan jika `n` ganjil, program akan mengalikan `n` dengan 3 lalu menambahkan 1. Proses ini terus berulang hingga `n` menjadi 1. Fungsi utama, ****cetakDeret****, bertanggung jawab untuk menghasilkan deret tersebut dan mencetak setiap elemen bilangan dalam satu baris, dipisahkan oleh spasi. Program juga menerima input dari pengguna untuk nilai awal `n` dan memastikan bahwa nilainya lebih besar dari 0 dan kurang dari 1 juta, sesuai batasan yang diberikan.