

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 4  
PROSEDUR**



Oleh:

NAMA : Alberta Aurora Adiswari

NIM : 2311102220

KELAS : S1-IF-11-02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM 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:

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

Kedudukannya prosedur sama seperti instruksi dasar yang sudah ada sebelumnya (assignment) dan atau dikatakan 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, cariNilai, belok, mulai,...

Berikut 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 di luar blok yang dari program utama atau func main( ) pada suatu program Go, dan bisa ditulis sebelum atau setelah dari blok program utama tersebut.

Sutu 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 argument yang diminta dari suatu prosedur. Sebagai contoh prosedur cetak NFibo diatas dipanggil dengan menuliskan Namanya, kemudian sebuah variabel atau nilai integer tertentu sebagai argument 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	}	

Dari contoh tersebut 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 variabel (cara pemanggilan #1) atau nilainya secara langsung (cara pemanggilan #2).

Suatu subprogram yang dipanggil dapat berkomunikasi dengan pemanggilnya melalui argument 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 yang ditulis pada saat deklarasi suatu subprogram, parameter ini berfungsi sebagai petunjuk bahwa argument apa saja yang diperlukan pada saat pemanggilan subprogram.

Sebagai contoh parameter **jari\_jari, tinggi** pada deklarasi fungsi volume Tabung 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

Argumen yang digunakan pada bagian parameter saat pemanggilan suatu subprogram. Banyaknya argument dan tipe data yang terdapat pada parameter actual harus mengikuti parameter formal.

Sebagai contoh argument **r, t, 15, 14, dan 100** pada contoh kode diatas (teks berwarna biru) adalah parameter actual, yang menyatakan nilai yang kita berikan sebagai jari-jari dan tinggi. Selain itu parameter juga dikelompokkan berdasarkan alokasi memorinya, yaitu pass by value dan pass by reference.

### 1. Pass by value

Nilai pada parameter actual akan disalin ke variabel local(parameter formal) pada subprogram. Dialokasikan di dalam memori computer dengan Alamat memori yang berbeda. Subprogram dapat menggunakan nilai pada parameter formal tersebut untuk proses apapun, tetapi tidak dapat mengembalikan informasinya ke pemanggil melalui parameter actual karena pemanggil tidak dapat mengakses

memori yang digunakan oleh subprogram. Pass by value bisa digunakan baik oleh fungsi ataupun prosedur.

## 2. Pass by Reference (Pointer)

Ketika parameter didefinisikan sebagai pass by reference, maka pada saat pemanggilan parameter formal akan berperan sebagai pointer yang menyimpan Alamat memori dari parameter actual. Sehingga perubahan nilai yang terjadi pada parameter formal tersebut akan berdampak pada parameter actual. Artinya nilai terakhirnya akan dapat diketahui oleh si pemanggil setelah subprogram tersebut selesai dieksekusi. Pass by reference sebaiknya digunakan hanya untuk prosedur.

Untuk lebih jelas berikut contoh sebuah subprogram yang digunakan untuk menghitung persamaan berikut ini:

$$f(x,y) = 2x - \frac{y}{2} + 3$$

Notasi Algoritma	
<pre> function f1(x,y : integer) → real kamus     hasil : real algoritma     hasil ← 2*x - 0.5*y + 3     return hasil endfunction  procedure f2(in x,y : integer, in/out     hasil:real) algoritma     hasil ← 2*x - 0.5*y + 3 endprocedure  program Contoh kamus     a,b : integer     c : real algoritma     input(a,b)     f2(a,b,c)     output(c, f1(b,a)) endprogram         </pre>	<p>x dan y pada fungsi f1 dan prosedur f2 adalah <b>pass by value</b>,</p> <p>sedangkan variabel <b>hasil</b> pada prosedur f2 adalah <b>pass by reference</b>.</p> <p>Untuk pemanggilan dengan notasi pseudocode masih sama dengan materi yang sudah dipelajari sebelumnya</p>

#### Notasi dalam bahasa Go

```
package main
import "fmt"

func f1(x,y int) float64 {
    var hasil float64
    hasil = float64(2*x) - 0.5*float64(y) + 3.0
    return hasil
}

func f2(x,y int, hasil *float64) {
    *hasil = float64(2*x) - 0.5*float64(y) + 3.0
}

func main(){
    var a,b int; var c float64
    fmt.Scan(&a,&b)
    f2(a,b,&c)
    output(c, f1(b,a))
endprogram
```

x dan y pada fungsi f1 dan prosedur f2 adalah **pass by value**,

sedangkan variabel **hasil** pada prosedur f2 adalah **pass by reference**.

Karena variabel **hasil** adalah **pointer to float64**, maka untuk mengaksesnya menggunakan **simbol bintang (\*)** pada variabelnya.

Pada bahasa Go saat pemanggilan prosedur f2, maka **parameter aktual** untuk **pass by reference** harus diberi **ampersand "&"**, contohnya **&c**

## II. GUIDED

### Source code

```
package main

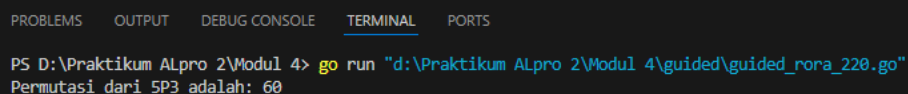
import "fmt"

// 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)
}
```

### Screenshoot program



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\Praktikum ALpro 2\Modul 4> go run "d:\Praktikum ALpro 2\Modul 4\guided\guided_rora_220.go"
Permutasi dari 5P3 adalah: 60
```

### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go untuk menghitung dan menampilkan permutasi dari dua bilangan n dan r. fungsi utama yang digunakan adalah factorial, yang menghitung factorial suatu bilangan dengan menggunakan loop untuk mengalikan semua angka dari 1 hingga n. prosedur permutasi menggunakan fungsi factorial untuk menghitung permutasi dengan rumus  $P(n, r) = \frac{n!}{(n-r)!}$ , sedangkan  $C(n, r) = \frac{n!}{r!(n-r)!}$

Dalam program tersebut nilai n dan r diinisialisasi dengan 5 dan 3, lalu program menghitung permutasi 5P3 dan menghasilkan outputan dengan nilai 60.

### III. UNGUIDED

#### Source code

```
package main

import "fmt"

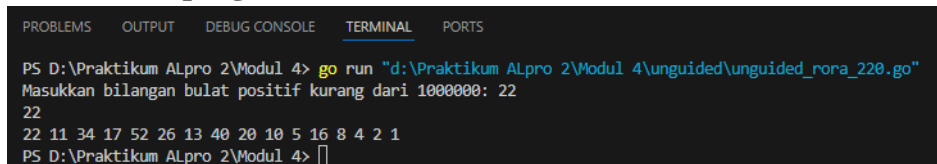
func cetakDeret(n int) {
    fmt.Printf("%d\n", n)
    fmt.Printf("%d ", n)

    for n != 1 {
        if n%2 == 0 {
            n /= 2
        } else {
            n = 3*n + 1
        }
        fmt.Printf("%d ", n)
    }
    fmt.Println()
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif kurang dari 1000000: ")
    fmt.Scan(&n)

    if n > 0 && n < 1000000 {
        cetakDeret(n)
    } else {
        fmt.Println("Masukan tidak valid. Harap masukkan bilangan antara 1 hingga 999999.")
    }
}
```

#### Screenshoot program



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Praktikum ALpro 2\Modul 4> go run "d:\Praktikum ALpro 2\Modul 4\unguided\unguided_rora_220.go"
Masukkan bilangan bulat positif kurang dari 1000000: 22
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\Praktikum ALpro 2\Modul 4> 
```

#### Deskripsi program

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program ini menghitung dan mencetak deret bilangan berdasarkan aturan  $3n + 1$  problem. Pengguna diminta untuk menginputkan bilangan bulat positif yang kurang dari 1000000. Jika inputan valid, program menghitung deret dengan aturan:

- Jika bilangan genap maka dibagi dua
- Jika bilangan ganjil maka dihitung dengan rumus  $3n + 1$

Proses berulang hingga nilai bilangan mencapai 1. Hasil perhitungan deret dicetak pada satu baris. Jika inputan tidak valid, program akan



menampilkan pesan kesalahan. Pada program diatas menghasilkan outputan dengan input 22 adalah 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1.