

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 2**  
**MODUL IV**  
**PROSEDUR**



Oleh:

ANISSA FAUZIA ISYANTI

2311102219

S1IF-11-02

**S1 TEKNIK INFORMATIKA**  
**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**  
**2024**

## I. DASAR TEORI

### A. Definisi Prodesur

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

Contoh deklarasi prosedur mencetak 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 di atas akan dijelaskan pada materi parameter di modul 5 ini.

### C. 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	}

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

## II. GUIDED

### Source Code

```
package main

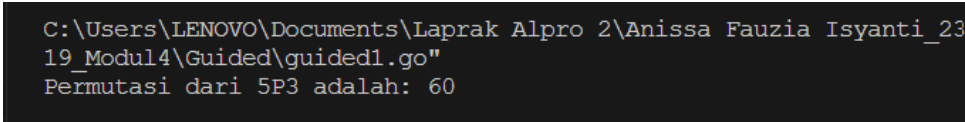
import "fmt"

// 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)
}
```

### Screenshot Program



```
C:\Users\LENOVO\Documents\Laparak Alpro 2\Anissa Fauzia Isyanti_23_19_Modul4\Guided\guided1.go"
Permutasi dari 5P3 adalah: 60
```

Program ini digunakan untuk menghitung dan menampilkan nilai permutasi dari dua bilangan yang telah ditentukan, yaitu  $n$  dan  $r$ . Di dalam program, terdapat fungsi `factorial` yang digunakan untuk menghitung nilai faktorial dari  $n$ . Fungsi ini mengembalikan hasil faktorial sesuai dengan rumus  $n!$ .

Selanjutnya, program menggunakan prosedur permutasi untuk menghitung nilai permutasi menggunakan rumus  $P(n, r) = n! / (n-r)!$ . Pada fungsi main, program menetapkan nilai n sebagai 5 dan r sebagai 3, kemudian memanggil prosedur permutasi untuk menghitung dan menampilkan hasil permutasi dari kedua bilangan tersebut.

### III. UNGUIDED

#### Source Code

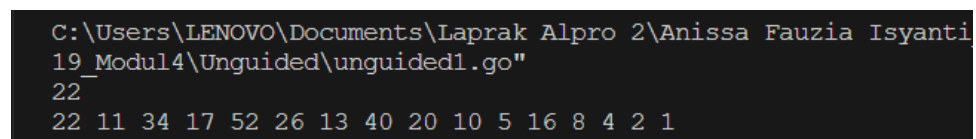
```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n)
    if n > 0 && n < 1000000 {
        PrintDeret_219(n)
    }
}

func PrintDeret_219(n int) {
    fmt.Print(n, " ")
    if n%2 == 0 { // pengecekan bilangan genap
        PrintDeret_219(n / 2)
    } else if n != 1 { // pengecekan bilangan ganjil
        PrintDeret_219(3*n + 1)
    }
}
```

#### Screenshot Program



```
C:\Users\LENOVO\Documents\Laparak Alpro 2\Anissa Fauzia Isyanti_19_Modul4\Unguided\unguided1.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Program ini adalah program untuk mencetak deret. Pengguna diminta memasukkan sebuah bilangan bulat positif (n), kemudian mencetak deret angka berdasarkan aturan tertentu. Jika n adalah bilangan genap, maka suku berikutnya program akan memanggil dirinya sendiri (rekursif) dengan nilai

$n/2$ . Jika  $n$  adalah bilangan ganjil (selain 1), maka suku berikutnya bernilai  $3*n + 1$ . Program akan berhenti sampai suku terakhir bernilai 1.