

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4
PROSEDUR**



Oleh:

MUHAMMAD RUSDIYANTO

2311102053

S1IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Secara sederhana, prosedur merupakan fungsi yang tidak mengembalikan nilai. Prosedur umumnya digunakan untuk mengurangi kerumitan kode program yang kompleks. Hal ini bisa dilakukan dengan tujuan untuk menambah readability (kemudahan pembacaan) kode program dan tentunya supaya kode yang dijadikan prosedur bisa digunakan kembali. Prosedur umumnya tidak memiliki keyword `return`. Atas dasar itu, penamaan fungsi sebaiknya menggunakan kata kerja atau kata yang menggambarkan proses dari prosedur.

Deklarasi prosedur dalam Go, bisa dilakukan seperti ini :

```
func [nama_prosedur]([parameter] tipe_data) {  
    [badan]  
}
```

Deklarasi parameter prosedur bisa dilakukan berdasarkan nilai (pass by reference) atau alamat memori (pass by reference). Jika nilai, maka nilai variabel asli tidak akan berubah meskipun prosedur telah memproses nilai yang di-pass. Jika alamat memori, maka nilai variabel asli juga akan berubah sesuai dengan perubahan yang terjadi di prosedur.

Pemanggilan prosedur dalam Go kurang lebih sama dengan pemanggilan fungsi. Prosedur juga bisa dijalankan secara rekursif layaknya fungsi.

II. GUIDED

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)
}
```

Guided 1 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day3\4> go run guided1.go
Permutasi dari 5P3 adalah: 60
```

Guided 1 | Penjelasan

Program di atas merupakan program yang bisa menentukan hasil permutasi dari suatu bilangan. Program ini mirip dengan program pada guided 1 di modul sebelumnya (modul 3). Perbedaan program terletak pada penggunaan prosedur untuk mencari hasil dari permutasi. Karena prosedur tidak

menggunakan memiliki parameter pointer, maka hasil harus dioutputkan pada prosedur tersebut.

Cara kerja dari program di atas dimulai dari deklarasi dan assignment variabel `n` dan `r`. Setelah itu, program akan memanggil prosedur permutasi untuk menghitung n permutasi r . Di dalam prosedur, hasil permutasi dihitung dengan bantuan fungsi `factorial`. Setelah hasil didapatkan, maka prosedur akan lanjut dengan menampilkan output hasil. Setelah itu program selesai.

III. UNGUIDED

Unguided 1 | Source Code

```
package main

import "fmt"

func cetakDeret(x *int) {
    fmt.Printf("%v ", *x)
    if *x > 1 {
        if *x%2 == 0 {
            *x /= 2
        } else {
            *x = *x*3 + 1
        }
        cetakDeret(x)
    }
}

func main() {
    var input int
    fmt.Print("Masukkan angka positif yang lebih kecil dari 1,000,000 : ")
    fmt.Scanln(&input)
    if input > 1000000 {
        fmt.Println("Angka lebih dari satu juta")
    } else {
        cetakDeret(&input)
    }
}
```

Unguided 1 | Output

```
PS C:\MyFiles\Visual Studio Projects HERE\alpro2\day3\4> go run unguided1.go
Masukkan angka positif yang lebih kecil dari 1,000,000 : 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Unguided 1 | Penjelasan

Program di atas merupakan sebuah program yang dapat menampilkan deret bilangan dari bilangan yang dimasukkan. Proses dalam program ini dimulai dari input angka, dimana angka yang dimasukkan harus kurang dari 1.000.000 (satu juta). Jika lebih dari satu juta, maka program akan menampilkan pesan bahwa angka lebih dari satu juta dan program akan berhenti. Jika tidak, maka program akan menjalankan prosedur `cetakDeret`.

Prosedur `cetakDeret` dalam program ini memiliki variabel `x` untuk menyimpan reference (alamat memori) dari angka yang dimasukkan. Dalam konteks ini, alamat yang di-pass adalah alamat variabel `input`. Proses di dalam prosedur ini dimulai dengan mencetak nilai dari x. Setelah itu, program akan mengecek apakah nilai dari x lebih dari satu. Jika tidak, maka program tidak akan melakukan apapun (prosedur selesai). Jika iya, maka program akan mengecek kembali apakah x modulus 2 sama dengan 0 (x jika dibagi 2 memiliki sisa bagi 0). Hal ini dilakukan untuk mengecek apakah x merupakan angka genap atau bukan. Jika iya (angka genap), maka program akan membagi `x` dengan 2. Jika tidak (angka ganjil), maka program akan mengalikan `x` dengan 3 dan ditambah 1. Setelah percabangan, program akan memanggil prosedur kembali. Dalam kata lain, prosedur `cetakDeret` merupakan prosedur rekursif. Setelah prosedur selesai (ketika angka lebih kecil atau sama dengan 1), maka program juga akan selesai.