

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II
MODUL IV
“PROCEDURE”**



Oleh:

ZAHRINA ANTIKA MALAHATI

2311102109

IF 11 02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

➤ Definisi

Prosedur dapat dianggap sebagai potongan beberapa intruksi program menjadi suatu intruksi 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. Prosedur dapat dianggap sebagai pembentukan suatu **instruksi baru** yang dibuat untuk mempermudah pemahaman algoritma program yang lebih besar. Kedudukannya sama seperti instruksi dasar yang sudah ada sebelumnya (**assignment**) dan/atau instruksi yang berasal dari paket (**fmt**), seperti **fmt.Scandan** dan **fmt.Print**. Karena itu selalu pilih nama prosedur yang berbentuk kata kerja atau sesuatu yang merepresentasikan proses sebagai nama dari prosedur.

➤ Deklarasi Procedure dalam bahasa Go

```
func <nama procedure> <(params)> {  
    /* deklarasi variabel lokal dari procedure */  
    ...  
    /* badan algoritma procedure */  
    ...  
}
```

➤ Cara pemanggilan Procedure dalam bahasa Go

```
func main() {  
    var x int  
    x = 5  
    cetakNFibo(x)    {cara pemanggilan #1}  
    cetakNFibo(100) {cara pemanggilan #2}  
}
```

II. GUIDED

❖ Guided 1

Source code

```
package main

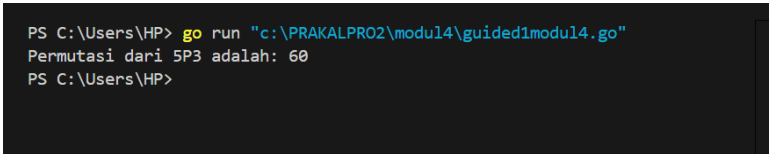
import "fmt"

// 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)
}
```

Output



```
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul4\guided1modul4.go"
Permutasi dari 5P3 adalah: 60
PS C:\Users\HP>
```

Deskripsi

Program ini adalah sebuah program dengan menggunakan bahasa Go untuk menghitung dan menampilkan hasil dari **permutasi** dari dua bilangan, n dan r, di mana permutasi adalah salah satu konsep matematika yang sering digunakan dalam kombinatorika. Program ini merupakan contoh sederhana namun efektif dari bagaimana menghitung permutasi menggunakan faktorial dalam bahasa Go. Program ini menggabungkan konsep dasar kombinatorika (permutasi) dengan penerapan fungsi dan prosedur dalam pemrograman.

III. UNGUIDED

❖ Unguided 1 (soal nomor 3)

Soal

- 3) Skiena dan Revilla dalam *Programming Challenges* mendefinisikan sebuah deret bilangan. Deret dimulai dengan sebuah bilangan bulat n . Jika bilangan n saat itu genap, maka suku berikutnya adalah $\frac{1}{2}n$, tetapi jika ganjil maka suku berikutnya bernilai $3n+1$. Rumus yang sama digunakan terus menerus untuk mencari suku berikutnya. Deret berakhir ketika suku terakhir bernilai 1. Sebagai contoh jika dimulai dengan $n=22$, maka deret bilangan yang diperoleh adalah:

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Untuk suku awal sampai dengan 1000000, diketahui deret selalu mencapai suku dengan nilai 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.

prosedure 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 cetakDeret(n int) {
    for n != 1 {
        fmt.Print(n, " ")
        if n%2 == 0 {
            n /= 2
        } else {
            n = 3*n + 1
        }
    }
}
```

```
        fmt.Println(1)
    }

    func main() {
        var n int
        fmt.Scan(&n)
        cetakDeret(n)
    }

    //Zahrina Antika Malahati_2311102109
```

Output

```
PS C:\Users\HP> go run "c:\PRAKALPRO2\modul4\unguided2modul4.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\HP>
```

Deskripsi

Program ini adalah program dengan bahasa Go yang mengimplementasikan aturan Collatz dengan cara yang efisien dan sederhana. Dengan menggunakan loop dan operasi aritmatika dasar, program dapat menghasilkan deret yang sangat menarik dari setiap bilangan bulat positif. Meski aturan Collatz belum terbukti benar untuk semua bilangan bulat, setiap bilangan yang diuji hingga sekarang selalu berakhir di angka 1, seperti yang ditunjukkan dalam program ini. Cara kerja program ini adalah

- **Deklarasi fungsi cetakDeret(n):** Fungsi ini menerima satu parameter integer n yang merupakan bilangan awal deret.
- **Looping:** Fungsi ini menggunakan for loop yang terus berjalan selama n tidak sama dengan 1.
 - **Mencetak nilai n:** Di setiap iterasi, nilai n saat itu dicetak.
 - **Kondisi genap:** Jika n habis dibagi 2 (genap), maka n dibagi 2.
 - **Kondisi ganjil:** Jika n tidak habis dibagi 2 (ganjil), maka n dikalikan 3 lalu ditambah 1.
- **Mencetak 1:** Setelah loop selesai (artinya n sudah menjadi 1), nilai 1 dicetak.
- **Fungsi main():**
 - **Meminta input:** Program meminta pengguna untuk memasukkan nilai awal n.
 - **Memanggil fungsi:** Nilai n yang diinputkan pengguna kemudian diberikan sebagai argumen ke fungsi cetakDeret.