

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**  
**MODUL IV**  
**PROSEDUR**



Oleh:

Ilhan Sahal Mansiz

2311102029

IF-11-02

**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## I. DASAR TEORI

### 1. Definisi Prosedur

Prosedur dalam pemrograman merupakan suatu blok kode yang dapat dipanggil dan dijalankan secara berulang-ulang tanpa harus menuliskannya berkali-kali. Pada bahasa Go, prosedur disebut sebagai fungsi (function). Fungsi ini dapat menerima masukan berupa parameter, menjalankan serangkaian instruksi, dan mengembalikan nilai (return value) atau tidak mengembalikan nilai sama sekali.

Prosedur memfasilitasi pemrograman yang lebih modular dan efisien, memungkinkan kode yang lebih terstruktur, mudah dibaca, dan dikelola. Di dalam bahasa Go, fungsi bisa berupa fungsi yang memiliki return atau fungsi void (tanpa return) yang disebut dengan prosedur murni.

### 2. Deklarasi Prosedur

Untuk mendeklarasikan prosedur atau fungsi di dalam bahasa Go, kita menggunakan kata kunci `func`. Deklarasi fungsi mengikuti pola sebagai berikut:

```
func NamaFungsi(parameter1 tipeData1, parameter2 tipeData2)
tipeDataKembalian {
    // Blok kode fungsi
}
```

- `NamaFungsi`: Nama dari prosedur yang akan dipanggil untuk menjalankan blok kode.
- `parameter`: Nilai yang dapat dikirim ke dalam prosedur sebagai input (jika ada). Jika prosedur tidak membutuhkan input, parameter bisa dikosongkan.
- `tipeDataKembalian`: Tipe data yang akan dikembalikan oleh prosedur (jika ada).

### 3. Cara Pemanggilan Prosedur

Untuk memanggil prosedur yang sudah dideklarasikan, cukup tuliskan nama prosedur diikuti tanda kurung buka dan tutup. Jika prosedur menerima parameter, masukkan nilai sesuai dengan tipe data parameter tersebut.

Pemanggilan Prosedur Tanpa Parameter:

```
CetakPesan()
```

Pemanggilan Prosedur dengan Parameter:

```
hasil := Tambah(3, 5)  
fmt.Println(hasil)
```

Pada contoh di atas, nilai 3 dan 5 dikirim ke fungsi Tambah, dan hasil penjumlahannya disimpan di variabel hasil.

## II. GUIDED

```
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)
```

```

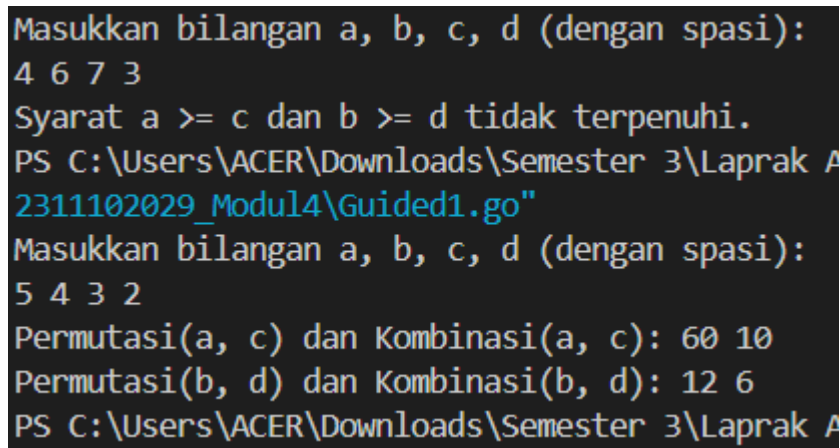
kombinasiAC := combination(a, c)

// 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 Program :



```

Masukkan bilangan a, b, c, d (dengan spasi):
4 6 7 3
Syarat a >= c dan b >= d tidak terpenuhi.
PS C:\Users\ACER\Downloads\Semester 3\Laprak A
2311102029_Modul4\Guided1.go"
Masukkan bilangan a, b, c, d (dengan spasi):
5 4 3 2
Permutasi(a, c) dan Kombinasi(a, c): 60 10
Permutasi(b, d) dan Kombinasi(b, d): 12 6
PS C:\Users\ACER\Downloads\Semester 3\Laprak A

```

Deskripsi Program :

Program ini adalah implementasi dalam bahasa Go yang menghitung permutasi dan kombinasi dari dua pasang bilangan (a, c) dan (b, d) dengan syarat bahwa nilai a harus lebih besar atau sama dengan c, dan nilai b harus lebih besar atau sama dengan d. Program dimulai dengan meminta input empat bilangan (a, b, c, d)

dari pengguna, kemudian mengecek apakah syarat tersebut terpenuhi. Jika ya, program akan menghitung dan menampilkan nilai permutasi dan kombinasi dari (a, c) dan (b, d) menggunakan fungsi `permutation` dan `combination` yang memanfaatkan fungsi `factorial` untuk menghitung nilai faktorial. Apabila syarat tidak terpenuhi, program akan menampilkan pesan bahwa syarat tidak terpenuhi.

### III. UNGUIDED

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**

Source Code :

```
package main

import "fmt"

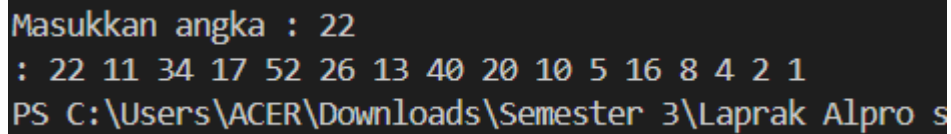
func cetakDeret(n int) {
    for n != 1 {
        fmt.Print(n, " ")
        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Println(n)
}

func main() {
    var n int
    fmt.Print("Masukkan angka : ")
    fmt.Scan(&n)

    if n < 1 || n >= 1000000 {
        fmt.Println("Angka harus positif dan kurang dari 1.000.000")
        return
    }
}
```

```
}  
  
    fmt.Print(": ")  
    cetakDeret(n)  
}
```

Screenshot Program :



```
Masukkan angka : 22  
: 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1  
PS C:\Users\ACER\Downloads\Semester 3\LapraK Alpro s
```

Deskripsi Program :

Program di atas merupakan implementasi dari deret Collatz dalam bahasa Go. Program ini meminta input dari pengguna berupa sebuah bilangan bulat positif kurang dari 1.000.000. Setelah menerima input, program akan memproses bilangan tersebut menggunakan aturan Collatz: jika bilangan genap, akan dibagi dua, sedangkan jika ganjil, akan dikalikan dengan 3 dan ditambah 1. Proses ini berlanjut sampai bilangan mencapai nilai 1, dan semua bilangan yang dihasilkan selama proses tersebut dicetak dalam satu baris dengan format output yang diawali dengan tanda ":". Jika input tidak valid (misalnya kurang dari 1 atau lebih dari 1.000.000), program akan menampilkan pesan kesalahan.