

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 4  
PROSEDUR**



Oleh:

Ben Waiz Pintus Widyosaputro

2311102169

IF-11-02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## **I. DASAR TEORI**

### **1. Pengertian :**

Prosedur dalam algoritma pemrograman adalah sebuah unit kode yang berfokus pada tugas tertentu dan dieksekusi secara mandiri. Prosedur ini digunakan untuk memecah-mecah program yang kompleks menjadi beberapa bagian yang lebih sederhana dan modular, sehingga memudahkan debugging, maintainability, dan scalability.

### **2. Struktur Prosedur :**

Sebuah prosedur biasanya memiliki struktur yang jelas dan terdefinisi, termasuk identificador, deklarasi parameter, block kode, dan return statement jika ada. Struktur ini membantu programmer untuk mengetahui apa saja yang dilakukan oleh prosedur tersebut dan bagaimana caranya diimplementasikan.

### **3. Keuntungan Menggunakan Prosedur :**

#### **a. Modularitas:**

Prosedur memungkinkan programmer untuk memecah program menjadi modul-modul yang lebih kecil dan terpisah, sehingga memudahkan pengembangan dan perawatan.

#### **b. Efficiency and Effectiveness:**

Dengan menggunakan prosedur, programmer dapat meregangkan logika program dan membuatnya lebih efisien dalam pengoperasian. Hal ini juga meminimalkan kesalahan (bugs) karena setiap modul dapat diuji secara terpisah.

#### **c. Redundancy Reduction:**

Prosedur membantu mengurangi redundansi kode karena hanya perlu ditulis satu kali dan dapat digunakan berkali-kali tanpa perlu repot-repot menulis lagi.

#### **d. Ease of Maintenance and Development:**

Kode yang dibangun dengan menggunakan prosedur lebih mudah dipelihara dan dikembangkan saat menambahkan fitur baru atau memperbaiki program. Programmer dapat langsung menargetkan modul yang perlu diperbaharui tanpa harus mengubah seluruh kode.

## II. GUIDED

### Guided 1

```
package main

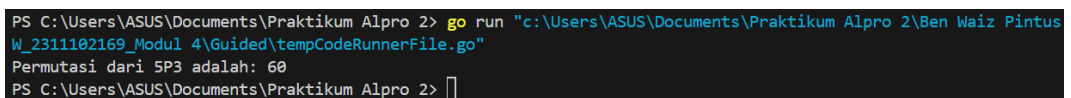
import "fmt"

// 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 output



```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintas
W_2311102169_Modul 4\Guided\tempCodeRunnerFile.go"
Permutasi dari 5P3 adalah: 60
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> 
```

### Penjelasan

Fungsi factorial menghitung faktorial dari sebuah bilangan dengan iterasi, sedangkan prosedur permutasi menghitung permutasi. Di dalam fungsi main, nilai n dan r diinisialisasi masing-masing menjadi 5 dan 3, kemudian prosedur permutasi dipanggil untuk menghitung dan menampilkan hasil permutasi dari 5P3. Hasil perhitungan akan dicetak dalam format yang telah ditentukan.

### III. UNGUIDED

#### Unguided 1

```
package main

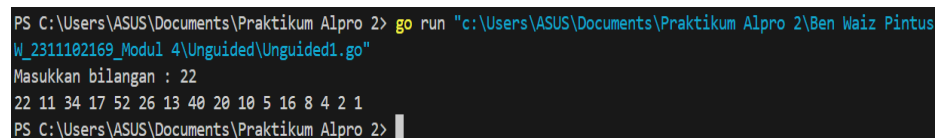
import "fmt"

func hasilkanDeret(bilangan int) {
    for bilangan != 1 {
        fmt.Printf("%d ", bilangan)

        if bilangan%2 == 0 {
            bilangan /= 2
        } else {
            bilangan = 3*bilangan + 1
        }
    }
    fmt.Println(1)
}

func main() {
    var bilangan int
    fmt.Print("Masukkan bilangan : ")
    fmt.Scan(&bilangan)
    if bilangan > 1000000 {
        fmt.Println("Bilangan harus kurang dari 1000000 !!!")
    } else {
        hasilkanDeret(bilangan)
    }
}
```

Screenshot output



```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintas
W_2311102169_Modul 4\Unguided\Unguided1.go"
Masukkan bilangan : 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> |
```

#### Penjelasan

Pengguna diminta untuk memasukkan sebuah bilangan bulat, kemudian program akan mencetak deret berdasarkan aturan: jika bilangan genap, maka dibagi dua, sedangkan jika ganjil, dikalikan tiga dan ditambah satu. Proses ini terus berlanjut hingga bilangan mencapai 1. Selain itu, terdapat validasi yang memastikan bahwa bilangan input harus kurang dari 1.000.000 jika tidak, program akan menampilkan pesan kesalahan.