

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL IV
PENGENALAN PROSEDUR**



Oleh:

NAMA : KARTIKA PRINGGO HUTOMO

NIM : 2311102196

KELAS ; IF-11-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

I. DASAR TEORI

Prosedur dalam bahasa Go adalah blok kode yang dirancang untuk melakukan tugas tertentu. Hal ini memungkinkan masalah yang kompleks dipecah menjadi bagian-bagian yang lebih kecil, sehingga menghasilkan kode yang lebih terstruktur, mudah dibaca, dan dipelihara. Setiap prosedur memiliki nama unik, dapat menerima parameter sebagai masukan, dan prosedur tidak harus mengembalikan nilai seperti fungsi. Konsep seperti rekursi, fungsi anonim, dan penutupan memungkinkan Anda menulis kode yang lebih fleksibel dan ekspresif. Selain itu, Go juga mendukung konsep metode, atau fungsi yang terkait dengan tipe data. Dengan menggunakan teknik secara efektif, Anda dapat membuat aplikasi Go yang lebih modular dan efisien.

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 D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 4> go run "d:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 4\Guided1.go"
Permutasi dari 5P3 adalah: 60
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 4>
```

Penjelasan

Program ini digunakan untuk menghitung dan menampilkan permutasi dua bilangan.

Permutasi dihitung menggunakan rumus $P(n, r) = \frac{n!}{(n-r)!}$.

Pertama, kita mempunyai fungsi faktorial yang menghitung faktorial dari bilangan bulat 'n' dan menangani kasus dasar dimana faktorial dari 0 adalah 1.

Fungsi ini menghitung faktorial dengan mengalikan semua bilangan bulat dari 1 hingga 'n' dalam satu lingkaran.

Prosedur Permutasi kemudian menghitung permutasi bilangan 'n' dan 'r' menggunakan fungsi faktorial dan mencetak hasilnya dalam format yang ringkas.

Pada fungsi Utama, program memanggil prosedur Permutasi dengan n sebagai 5 dan r sebagai 3.

Ini menghasilkan keluaran berupa nilai permutasi 5P3.

Secara keseluruhan, program ini memberikan cara mudah untuk memahami dan menghitung permutasi menggunakan konsep faktorial.

III. UNGUIDED

Source cod

```
package main
```

```
import "fmt"
```

```
func cetakDeret(n int) {  
    hasil := n  
    fmt.Print(hasil)
```

```
    for hasil != 1 {  
        if hasil%2 == 0 {  
            hasil = hasil / 2  
        } else {  
            hasil = 3*hasil + 1  
        }  
        fmt.Print(" ", hasil) // Tambahkan spasi sebelum angka berikutnya  
    }  
    fmt.Println() // Tambahkan baris baru setelah selesai mencetak deret  
}
```

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

```
    cetakDeret(n)  
}
```

Output :

```
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 3> go run "d:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 3\Unguided.go"  
Masukkan pusat dan radius lingkaran 1 (cx1, cy1, r1): 1 2 3  
Masukkan pusat dan radius lingkaran 2 (cx2, cy2, r2): 1 2 3  
Masukkan koordinat titik (x, y): 5 2  
Titik di luar lingkaran 1 dan 2  
PS D:\Alpro 2\Kartika Pringgo Hutomo_2311102196_Modul 3> |
```

Penjelasan

Tujuan dari program ini adalah untuk mengeluarkan serangkaian angka berdasarkan angka yang dimasukkan oleh pengguna menggunakan aturan yang dikenal sebagai "dugaan Colatz" atau "masalah $3n + 1$ ".

Di dalam fungsi "printSeries", program pertama-tama menerima masukan bilangan bulat "n" dari pengguna dan mencetak nomor tersebut.

Program kemudian mengeksekusi perulangan hingga nilai "hasil" sama dengan 1.

Jika "hasil" adalah bilangan genap di setiap iterasi perulangan, program membagi "hasil" dengan 2.

Jika "hasil" adalah bilangan ganjil, program mengalikan "hasil" dengan 3 dan menambahkan 1.

Hasil setiap perhitungan dicetak dengan dipisahkan spasi agar keluarannya lebih mudah dibaca.

Setelah seri dicetak, program menambahkan baris baru untuk membersihkan tampilan.

Oleh karena itu, program ini menyediakan cara untuk melihat secara visual bagaimana angka berkurang atau berubah menggunakan aturan yang telah ditetapkan.