

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL IV
PROSEDUR**



Oleh:

FADHLIN NAURI MAHRIJAR

2211102339

IF11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Definisi Prosedur

Prosedur adalah bagian dari suatu program yang disusun secara terpisah untuk melakukan suatu tugas khusus atau fungsi tertentu.

Pada dasarnya ada dua macam bentuk prosedur yaitu Subrutin (subprogram) dan Fungsi.

1. Subrutin (subprogram) adalah bagian dari program yang dibuat terpisah untuk melaksanakan sebagian dari tugas yang harus diselesaikan oleh suatu program. Pada umumnya yang dikenal dengan nama "Prosedur" sebenarnya adalah "Subprogram" .
2. Fungsi adalah bagian dari program yang dibuat terpisah untuk melaksanakan fungsi tertentu yang menghasilkan suatu nilai untuk dikembalikan ke program utama.

Prosedur banyak digunakan pada program yang terstruktur karena :

1. Merupakan penerapan konsep program Modular, yaitu memecah-mecah program yang rumit menjadi program-program bagian yang lebih sederhana dalam bentuk sebagai prosedur-prosedur.
2. Untuk hal-hal yang sering dilakukan berulang-ulang, cukup dituliskan sekali saja dalam prosedur dan dapat dipanggil atau dipergunakan sewaktu-waktu bila diperlukan.
3. Membuat kode program lebih mudah dibaca.
4. Dapat digunakan untuk menyembunyikan detail program

Prosedur memiliki ciri-ciri sebagai berikut ini :

1. Hanya memiliki satu fungsi tujuan (logical inherent). Sebuah prosedur sebaiknya hanya memiliki satu fungsi tujuan dan tidak bercampur dengan tujuan lain. Hal ini untuk membuat prosedur lebih focus.

2. Berukuran kecil (small size). Yang dimaksud ukuran disini adalah panjang algoritma atau panjang kode program pada suatu prosedur. Ukuran kecil akan mudah dibaca maupun diperbaiki.
3. Tidak tergantung pada prosedur lain (independent). Sebuah prosedur harusnya bersifat mandiri, artinya sebuah prosedur dapat berjalan dan diuji tanpa menunggu bagian lainnya selesai. Selain itu variable yang digunakan dalam prosedur tidak mempengaruhi variabel yang digunakan pada bagian lain di keseluruhan program.

Manfaat pembuatan prosedur :

1. Modularisasi : Suatu program yang besar dan kompleks dapat dibagi ke dalam beberapa prosedur sehingga setiap prosedur merupakan bagian yang mudah dikerjakan. Dengan demikian, program besar tersebut menjadi mungkin diselesaikan.
2. Simplifikasi : Dalam suatu program, sering diperlukan suatu tugas yang harus dikerjakan berulang-ulang dengan nilai-nilai variabel yang berbeda. Agar tidak merepotkan maka tugas ini cukup ditulis sekali saja dalam bentuk prosedur yang kemudian dipanggil berulang-ulang sesuai kebutuhan.

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

NO 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:\KULIAH\SEMESTER 5\ALPRO 2 PRAK> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 4\Guided\tempCodeRunnerFile.go"
Permutasi dari 5P3 adalah: 60
```

Penjelasan :

Kode ini memberikan contoh sederhana tentang bagaimana menghitung faktorial dan permutasi dalam bahasa Golang. Kode ini dirancang untuk menghitung dan menampilkan nilai permutasi dari dua bilangan bulat n dan r. Konsep permutasi adalah banyaknya cara untuk memilih r objek dari n objek yang berbeda tanpa memperhatikan urutan.

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

SOAL

No 3

Source code :

```
package main

import "fmt"

func cetakDeret(n int) {
    fmt.Println("Deret bilangan untuk", n, ":")
    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.Print("Masukkan nilai awal: ")
    fmt.Scan(&n)
    cetakDeret(n)
}
```

Output :

```
PS D:\KULIAH\SEMESTER 5\ALPRO 2 PRAK> go run "d:\KULIAH\SEMESTER 5\ALPRO 2 PRAK\MODUL 4\Unguided\tempCodeRunnerFile.go"
Masukkan nilai awal: 50
Deret bilangan untuk 50 :
50 25 76 38 19 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Penjelasan :

Program ini merupakan program sederhana tentang bagaimana menggunakan konsep pemrograman dasar dalam bahasa Golang untuk menyelesaikan masalah matematika seperti deret bilangan untuk-n. Dengan memahami kode ini, Anda dapat mengembangkan program yang lebih kompleks dan menarik.