

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4
PROSEDUR**



Oleh:

FAHRUR RIZQI

2311102059

S1IF-11-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

I. DASAR TEORI

1. Definisi Prosedur

Prosedur adalah sekumpulan instruksi program yang dirancang untuk mengurangi kompleksitas kode dalam program yang lebih besar. Prosedur dapat dianggap sebagai subprogram yang tidak mengembalikan nilai, dan tidak menggunakan kata kunci return dalam tubuhnya. Prosedur berfungsi untuk melakukan tugas tertentu dan dapat dipanggil dari program utama.

2. Deklarasi Prosedur

Deklarasi prosedur dapat ditulis dalam berbagai notasi, termasuk pseudocode dan bahasa pemrograman Go. Berikut adalah format umum untuk mendeklarasikan prosedur:

Notasi dalam Go:

```
func <nama_procedure>(<params>) {  
    // deklarasi variabel lokal  
  
    // badan algoritma  
  
}
```

Contoh deklarasi prosedur untuk mencetak n nilai pertama dari deret Fibonacci:

Notasi dalam Go:

```
func cetakNFibo(n int) {  
    var f1, f2, f3 int  
  
    f2 = 0  
  
    f3 = 1  
  
    for i := 1; i <= n; i++ {  
        fmt.Println(f3)  
  
        f1 = f2  
  
        f2 = f3  
  
        f3 = f1 + f2  
  
    }  
  
}
```

3. Pemanggilan Prosedur

Prosedur hanya akan dieksekusi jika dipanggil dari program utama. Pemanggilan prosedur dapat dilakukan dengan menuliskan nama prosedur dan memberikan argumen yang sesuai. Contoh pemanggilan prosedur cetakNFibo:

Notasi dalam Go:

```
func main() {  
    var x int  
  
    x = 5  
  
    cetakNFibo(x) // cara pemanggilan #1  
  
    cetakNFibo(10) // cara pemanggilan #2  
}
```

4. Parameter

Parameter adalah cara bagi subprogram untuk berkomunikasi dengan pemanggilnya. Ada dua jenis parameter:

- **Parameter Formal:** Dideklarasikan dalam subprogram dan berfungsi sebagai petunjuk untuk argumen yang diperlukan saat pemanggilan.
- **Parameter Aktual:** Argumen yang diberikan saat pemanggilan subprogram, harus sesuai dengan parameter formal dalam hal jumlah dan tipe data.

5. Pass by Value dan Pass by Reference

- **Pass by Value:** Nilai dari parameter aktual disalin ke parameter formal. Perubahan pada parameter formal tidak mempengaruhi parameter aktual.
- **Pass by Reference:** Parameter formal berfungsi sebagai pointer yang menyimpan alamat memori dari parameter aktual. Perubahan pada parameter formal akan mempengaruhi parameter aktual.

Kesimpulan

Modul 4 membahas tentang prosedur dalam pemrograman, termasuk definisi, deklarasi, pemanggilan, parameter, serta cara kerja pass by value dan pass by reference. Prosedur adalah alat penting untuk menyusun kode yang lebih terstruktur dan mudah dipahami.

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)
}
```

buat soal nomer 1

Output

```
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 4> go
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 4\Fahrur Rizqi_2311102059\Guided.go"
Permutasi dari 5P3 adalah: 60
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 4> █
```

Deskripsi Program

Program di atas menentukan hasil permutasi dari suatu bilangan. Program ini serupa dengan program pada modul sebelumnya (modul 3, guided 1), namun perbedaannya terletak pada penggunaan prosedur untuk menghitung permutasi. Karena prosedur tidak menggunakan parameter pointer, hasil permutasi harus dioutputkan langsung dalam prosedur tersebut.

Cara kerja program dimulai dengan deklarasi dan inisialisasi variabel `n` dan `r`. Setelah itu, program memanggil prosedur permutasi untuk

menghitung nilai n^r permutasi r . Dalam prosedur tersebut, perhitungan dilakukan menggunakan fungsi `factorial`. Setelah hasil permutasi diperoleh, prosedur menampilkan hasil tersebut, dan program pun selesai.

III. UNGUIDED

Unguided 1

```
package main
import "fmt"

func cetakDeret(n int) {
    for n != 1 {
        fmt.Printf("%d ", n)
        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Printf("%d\n", n) // fahrur059
}

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

    cetakDeret(n)
}
```

Output

```
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 4> go run
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 4\Fahrur Rizqi_2311102059\Unguided.go"
Masukkan nilai awal: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 4> █
```

Deskripsi Program

Program di atas adalah program yang menampilkan deret bilangan berdasarkan angka yang diinputkan. Program dimulai dengan menerima input angka, di mana angka tersebut harus kurang dari 1.000.000 (satu juta). Jika input melebihi satu juta, program akan menampilkan pesan bahwa angkanya lebih dari satu juta dan berhenti. Jika angka valid, program akan menjalankan prosedur `cetakDeret`.

Prosedur `cetakDeret` menggunakan variabel `x` yang menyimpan referensi (alamat memori) dari angka yang diinputkan, yaitu variabel `input`. Prosesnya dimulai dengan mencetak nilai `x`, lalu memeriksa apakah `x` lebih besar dari satu. Jika tidak, prosedur selesai. Jika `x` lebih dari satu, program akan memeriksa apakah `x` adalah bilangan genap (dengan `x % 2 == 0`). Jika genap, `x` dibagi dua. Jika ganjil, `x` dikalikan tiga lalu ditambah satu. Setelah itu, program memanggil kembali prosedur `cetakDeret` secara rekursif hingga nilai `x` kurang dari atau sama dengan

satu. Pada titik ini, program akan berhenti.

Prosedur ``cetakDeret`` bersifat rekursif, artinya ia akan terus memanggil dirinya sendiri sampai kondisi penghentian (nilai ``x <= 1``) tercapai.