LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2

MODUL 4

PROSEDUR

Oleh:

ANANDA BASKORO PUTRA

2311102187

IF 11 02

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO 2024

I. DASAR TEORI

1. Definisi Prosedur

Prosedur adalah bagian dari program yang dirancang untuk menjalankan tugas atau operasi tertentu secara mandiri, tanpa mengembalikan nilai (berbeda dengan fungsi yang biasanya mengembalikan nilai). Dalam pemrograman, prosedur digunakan untuk memecah kode program menjadi beberapa bagian yang lebih terstruktur, sehingga meningkatkan modularitas, keterbacaan, dan efisiensi. Prosedur membantu menghindari pengulangan kode, karena dapat dipanggil berkali-kali di berbagai bagian program.

2. Deklarasi Prosedur

Deklarasi prosedur melibatkan penulisan nama prosedur, parameter yang digunakan (jika ada), dan serangkaian instruksi yang akan dijalankan oleh prosedur tersebut. Sintaksnya tergantung pada bahasa pemrograman yang digunakan, namun secara umum deklarasi prosedur dapat dilihat sebagai berikut:

3. Pemanggilan Prosedur

Pemanggilan prosedur adalah cara untuk menjalankan prosedur yang telah dideklarasikan sebelumnya. Pemanggilan dilakukan dengan menyebutkan nama prosedur dan, jika diperlukan, memberikan argumen yang sesuai dengan parameter yang dideklarasikan dalam prosedur. Saat prosedur dipanggil, instruksi yang ada di dalamnya akan dieksekusi.

3. Pemanggilan Prosedur

Pemanggilan prosedur adalah cara untuk menjalankan prosedur yang telah dideklarasikan sebelumnya. Pemanggilan dilakukan dengan menyebutkan nama prosedur dan, jika diperlukan, memberikan argumen yang sesuai dengan parameter yang dideklarasikan dalam prosedur. Saat prosedur dipanggil, instruksi yang ada di dalamnya akan dieksekusi.Pemanggilan prosedur dalam pemrograman

berguna untuk mengorganisir aliran kontrol, menjalankan tugas-tugas berulang dengan mudah, dan memfasilitasi pemeliharaan kode.

4. Manfaat Penggunaan Prosedur

- **Memudahkan pemeliharaan kode**: Kode lebih terstruktur dan modular.
- **Mengurangi pengulangan kode**: Prosedur memungkinkan kode yang sering digunakan hanya ditulis sekali.
- **Meningkatkan keterbacaan**: Prosedur membantu memecah masalah besar menjadi bagian kecil yang lebih mudah dipahami.
- **Efisiensi eksekusi**: Penggunaan prosedur memungkinkan penghematan memori dan waktu eksekusi program, karena hanya bagian tertentu yang perlu dijalankan saat dibutuhkan.

I. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

Guided 1

```
package main

import "fmt"

func factorial(n int) int {
	if n == 0 {
		return 1
	}
	result := 1
	for i := 1; i <= n; i++ {
		result *= i
	}
	return result
}

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() {
		n, r := 6, 2
	permutasi(n, r)
}
```

Screenshoot Output

```
PS D:\Bedor\KULIAH\laprak modul 4> go run "d:\Bedor\KULIAH\laprak modul 4\guided\guided1.go"
Permutasi dari 6P2 adalah: 30
PS D:\Bedor\KULIAH\laprak modul 4>
```

Deskripsi Program

Program Go di atas menghitung permutasi dengan memanfaatkan fungsi faktorial. Fungsi factorial digunakan untuk menghitung faktorial dari suatu bilangan, dan prosedur permutasi menghitung permutasi dari dua bilangan n dan r menggunakan rumus permutasi . Di dalam main, prosedur permutasi dipanggil dengan nilai n = 6 dan r = 2, sehingga hasil permutasi 6P2 ditampilkan sebagai 30.

II. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

Unguided 2b_1

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

```
        fmt.Print(": ")

        cetakDeret(n)

}
```

Screenshoot Output

Deskripsi Program

Program di atas mencetak deret angka berdasarkan aturan Collatz. Fungsi cetakDeret(n int) memulai dari bilangan n, kemudian jika n genap, dibagi dua, dan jika ganjil, dikalikan tiga lalu ditambah satu. Proses ini berulang hingga nilai n menjadi 1. Pada fungsi main, program meminta input dari pengguna untuk nilai n yang harus positif dan kurang dari 1.000.000, lalu memanggil fungsi cetakDeret untuk mencetak deret tersebut. Program juga memeriksa apakah input valid sebelum menjalankan perhitungan.