

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN DUA
MODUL IV
PROSEDUR**



Oleh:

YAYANG ALYA BILQIS

2311102229

IF 11 02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

- **Definisi Prosedur**

Prosedur dapat dianggap sebagai potongan beberapa intruksi program menjadi suatu intruksi baru yang dibuat untuk mengurangi kerumitan dari kode program yang kompleks pada suatu program yang besar. Prosedur akan menghasilkan suatu akibat atau efek langsung pada program Ketika dipanggil pada program utama. Suatu subprogram dikatakan prosedur apabila:

1. Tidak ada deklarasi tipe nilai yang dikembalikan
2. Tidak terdapat kata kunci return dalam badan subprogram

Kedudukan prosedur sama seperti intruksi dasar yang sudah ada sebelumnya dan atau intruksi yang berasal dari paket (fmt), seperti fmt.Scan dan fmt.Print. karena itu selalu pilih nama prosedur yang berbentuk kata kerja atau sesuatu yang merepresentasikan proses sebagai nama dari prosedur.

- **Deklarasi Fungsi**

| Notasi Algoritma | |
|------------------------|---|
| 1 | procedure <nama procedure> (<params>) |
| 2 | kamus |
| 3 | {deklarasi variabel lokal dari procedure} |
| 4 | ... |
| 5 | algoritma |
| 6 | {badan algoritma procedure} |
| 7 | ... |
| 8 | endprocedure |
| Notasi dalam bahasa Go | |
| 9 | func <nama procedure> (<params>) { |
| 10 | /* deklarasi variabel lokal dari procedure */ |
| 11 | ... |
| 12 | /* badan algoritma procedure */ |
| 13 | ... |
| 14 | } |

| | Notasi Algoritma |
|----|--------------------------------------|
| 1 | procedure cetakNFibo(in n : integer) |
| 2 | kamus |
| 3 | f1, f2, f3, i : integer |
| 4 | algoritma |
| 5 | f2 ← 0 |
| 6 | f3 ← 1 |
| 7 | for i ← 1 to n do |
| 8 | output(f3) |
| 9 | f1 ← f2 |
| 10 | f2 ← f3 |
| 11 | f3 ← f1 + f2 |
| 12 | endfor |
| 13 | endprocedure |
| | Notasi dalam bahasa Go |
| 14 | func cetakNFibo(n int) { |
| 15 | var f1, f2, f3 int |
| 16 | f2 = 0 |
| 17 | f3 = 1 |
| 18 | for i := 1; i <= n; i++ { |
| 19 | fmt.Println(f3) |
| 20 | f1 = f2 |
| 21 | f2 = f3 |
| 22 | f3 = f1 + f2 |
| 23 | } |
| 24 | } |

- **Cara Pemanggilan Fungsi**

Pemanggilan prosedur hanya akan dieksekusi apabila dipanggil baik secara langsung atau tidak langsung oleh program utama. Tidak langsung disini maksudnya adalah prosedur dipanggil oleh program utama melalui perantara subprogram yang lain. Pemanggilan suatu prosedur cukup mudah, yaitu dengan hanya menuliskan nama beserta parameter atau argument yang diminta dari suatu prosedur. Sebagai contoh prosedur cetakNFibo di atas dipanggil dengan menuliskan Namanya, kemudian sebuah variable atau nilai integer tertentu sebagai argument untuk paramer n.

- **Parameter**

Suatu subprogram yang dipanggil dapat berkomunikasi dengan pemanggilnya melalui argumen yang diberikan melalui parameter yang dideklarasikan pada subprogramnya. Berikut ini jenis atau pembagian dari parameter:

Berdasarkan letak penulisannya pada program, maka parameter dapat dikelompokkan menjadi dua, yaitu parameter formal dan parameter aktual.

1. Parameter Formal

Parameter formal adalah parameter yang ditulis pada saat deklarasi suatu subprogram, parameter ini berfungsi sebagai petunjuk bahwa argumen apa saja yang diperlukan pada saat pemanggilan subprogram.

Sebagai contoh, parameter jariJari dan tinggi pada deklarasi fungsi volumeTabung adalah parameter formal. Artinya ketika memanggil volumeTabung maka kita harus mempersiapkan dua integer (berapapun nilainya) sebagai jariJari dan tinggi.

2. Parameter Aktual

Sedangkan parameter aktual adalah argumen yang digunakan pada bagian parameter saat pemanggilan suatu subprogram. Banyaknya argumen dan tipe data yang terdapat pada parameter aktual harus mengikuti parameter formal.

Sebagai contoh argumen r, t, 5, 14, dan 100 pada contoh kode di atas adalah parameter aktual, yang menyatakan nilai yang kita berikan sebagai jari-jari dan tinggi.

- **Parameter formal:** Variabel yang dideklarasikan dalam definisi fungsi dan bertindak sebagai placeholder untuk nilai yang akan diteruskan saat fungsi dipanggil.
- **Parameter aktual:** Nilai yang sebenarnya diteruskan ke fungsi saat dipanggil.

Contoh:

Pada fungsi volumeTabung, jariJari dan tinggi adalah parameter formal, sedangkan r, t, 5, 14, dan 100 adalah parameter aktual.

Tujuan: Membedakan antara variabel yang digunakan dalam definisi fungsi (parameter formal) dengan nilai yang sebenarnya digunakan saat fungsi dipanggil (parameter aktual). Memahami bagaimana data ditransfer antara bagian yang berbeda dari sebuah program.

II. GUIDED

1. Guided Satu

```
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 Program

```
PS C:\Users\ASUS Vivobook> go run "d:\PRAKTIKUM ALPRO 2\Modul 4\Yayang Alya Bilqis_2311102229\guided1\guided1.go"
Permutasi dari 5P3 adalah: 60
PS C:\Users\ASUS Vivobook>
```

Deskripsi Program

Program diatas adalah untuk menghitung dan menampilkan nilai dari dua bilangan bulat yang diberikan. Fungsi akan menerima satu argument berupa bilangan bulat bulat lalu prosedur ini akan menerima dua argument berupa bilangan bulat n dan r dan fungsi akan Kembali mengeksekusi program.

III. UNGUIDED UNGUIDED 1

```
package main

import "fmt"

func deretSkienaRekursif(n int) {
    fmt.Print(n, " ")
    if n != 1 {
        if n%2 == 0 {
            deretSkienaRekursif(n / 2)
        } else {
            deretSkienaRekursif(3*n + 1)
        }
    }
}
```

```

    }

    func main() {
        var n int

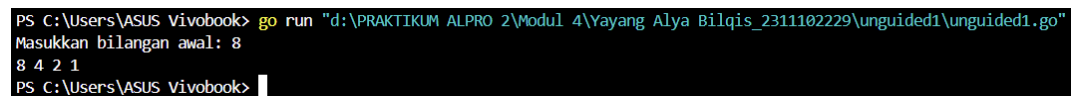
        fmt.Print("Masukkan bilangan awal: ")

        fmt.Scan(&n)

        deretSkienaRekursif(n)
    }

```

Screenshot Program



```

PS C:\Users\ASUS Vivobook> go run "d:\PRAKTIKUM ALPRO 2\Modul 4\Vayang Alya Bilqis_2311102229\unguided1\unguided1.go"
Masukkan bilangan awal: 8
8 4 2 1
PS C:\Users\ASUS Vivobook>

```

Deskripsi Program

Program ini akan menghitung dan mencetak deret bilangan yang mengikuti aturan deret skiena. Deret skiena adalah sebuah urutan bilangan yang dimulai dari suatu bilangan bulat positif dan mengikuti aturan. Parameter akan menerima sebuah bilangan bulat n sebagai titik awal deret lalu program akan meminta user memasukkan bilangan awal. Jika n genap maka fungsi akan memanggil dirinya sendiri dengan argumen $n/2$ dan jika ganjil maka fungsi akan memanggil dirinya sendiri dengan argument $3*n+1$.