

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 4  
PROSEDUR**



Oleh:

MUHAMMAD DAFFA AL FAIZ

2311102237

S1IF-11-02

**S1 TEKNIK INFORMATIKA**  
**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**  
**2024**

**I. DASAR TEORI**

Dasar Teori

**Definisi Procedure**

Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu instruksi 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:

- Tidak ada deklarasi tipe nilai yang dikembalikan
- Tidak terdapat kata kunci return dalam badan subprogram

**Cara Pemanggilan Procedure**

Seperti yang sudah dijelaskan sebelumnya, suatu **prosedur hanya akan dieksekusi apabila dipanggil** baik secara langsung atau tidak langsung oleh program utama. Tidak langsung di sini 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 argumen yang diminta dari suatu prosedur**. Sebagai contoh prosedur cetakNthFibo di atas dipanggil dengan menuliskan namanya, kemudian sebuah variabel atau nilai integer tertentu sebagai argumen untuk parameter n. Contoh:

	Notasi Algoritma
1	program contohprosedur
2	kamus
3	x : integer
4	algoritma
5	x ← 5
6	cetakNFibo(x)           {cara pemanggilan #1}
7	cetakNFibo(100)       {cara pemanggilan #2}

laman 40 | Modul Praktikum Algoritma dan Pemrograman 2

8	endprogram
	Notasi dalam bahasa Go
9	func main() {
10	var x int
11	x = 5
12	cetakNFibo(x)       {cara pemanggilan #1}
13	cetakNFibo(100) {cara pemanggilan #2}
14	}

Dari contoh di atas terlihat bahwa cara pemanggilan dengan notasi pseudocode dan Golang adalah sama. Argumen yang digunakan untuk parameter n berupa integer (sesuai deklarasi) yang terdapat pada suatu variabel (cara pemanggilan #1) atau nilainya secara langsung (cara pemanggilan #2).

## 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.

### 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 **Jari\_jari,tinggi** pada deklarasi **fungsiI 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, 15, 14** dan **100** pada contoh Rode di atas (teks berwarna biru) adalah parameter aktual, yang menyatakan nilai yang kita berikan sebagai jari-jari dan tinggi.

## II. GUIDED

### Source Code

```
package main

import "fmt"

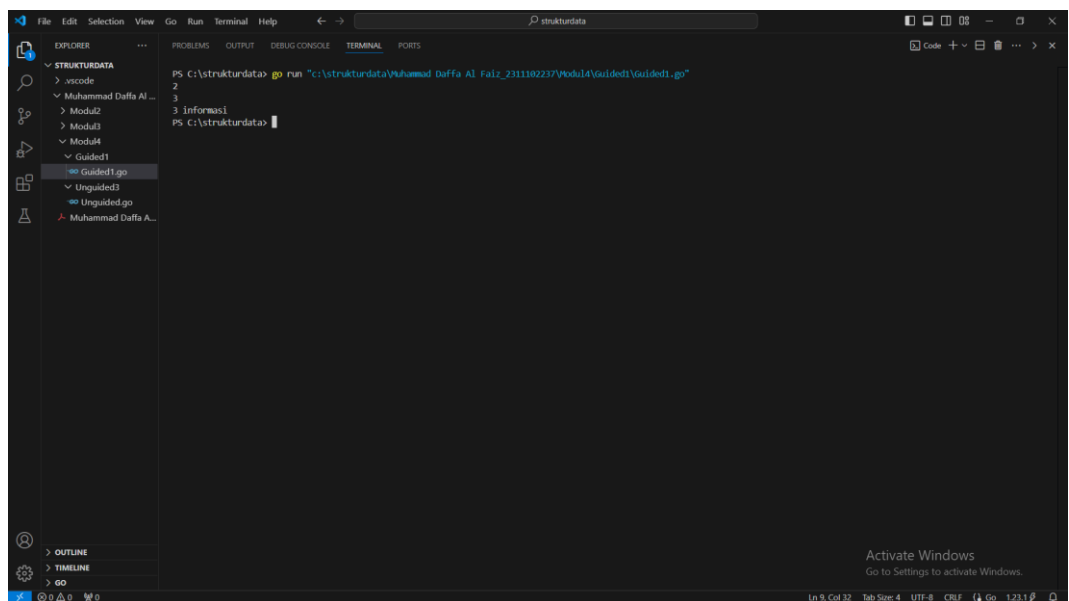
func main() {
    var bilangan int
    var pesan string
    fmt.Scan(&bilangan, &pesan)
    cetakPesan(pesan, bilangan)
}

func cetakPesan(M string, flag int) {
    var jenis string = ""

    if flag == 0 {
        jenis = "error"
    } else if flag == 1 {
        jenis = "warning"
    } else if flag == 2 {
        jenis = "informasi"
    }

    fmt.Println(M, jenis)
}
```

### Screenshot



## Deskripsi

Program ini adalah untuk melakukan klasifikasi pesan berdasarkan nilai bilangan yang diberikan

### III. UNGUIDED

#### Unguided 1

##### Source Code

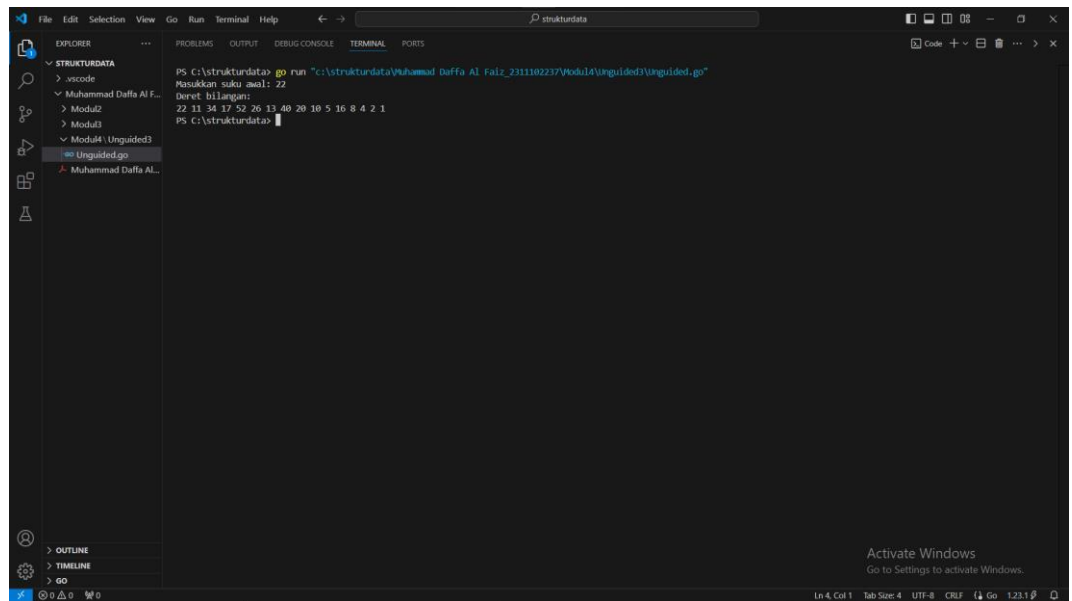
```
package main

import "fmt"

func cetakDeret(n int) {
    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 suku awal: ")
    fmt.Scan(&n)
    fmt.Println("Deret bilangan:")
    cetakDeret(n)
}
```

##### Screenshot



```
PS C:\strukturdata> go run "C:\strukturdata\Muhammad Daffa Al Faiz_2311180237\Module3\Unguided3\Unguided.go"
Masukkan suku awal: 22
Deret bilangan:
22 11 34 17 52 26 13 48 28 18 5 16 8 4 2 1
PS C:\strukturdata>
```

## Deskripsi

Program ini adalah untuk mencetak setiap suku dari deret yang dijelaskan untuk nilai suku awal yang diberikan.