

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL IV**

**PROSEDUR**



Oleh:

Mansyuroh

NIM:

2311102234

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### A. Definisi Prosedur

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:

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

Kedudukannya prosedur sama seperti instruksi dasar yang sudah ada sebelumnya (assignment) dan atau instruksi 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. Contoh: cetak, hitungRerata, cariNilai, belok, mulai ...

### B. Deklarasi Prosedur

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 cetakNFibo di atas dipanggil dengan menuliskan namanya, kemudian sebuah variabel atau nilai integer tertentu sebagai argumen untuk parameter n. Contoh:

Berikut ini adalah cara penulisan deklarasi prosedur pada notasi Pseudocode dan GoLang.

|   | Notasi Algoritma                          |
|---|-------------------------------------------|
| 2 | procedure <nama procedure> («params»)     |
| 3 | kamus                                     |
| 4 | {deklarasi variabel lokal dari procedure} |
| 5 | algoritma                                 |
| 6 | {badan algoritma procedure}               |
| 7 | endprocedure                              |
| 8 |                                           |
|   | Notasi dalam bahasa Go                    |

|    |                                               |
|----|-----------------------------------------------|
| 9  | func <nama procedure> < (params)>             |
| 10 | /* deklarasi variabel lokal dari procedure */ |
| 11 | /* badan algoritma procedure */               |
| 12 |                                               |
| 13 |                                               |
| 14 |                                               |

Penulisan deklarasi ini berada di luar blok yang dari program utama atau func main() pada suatu program Go, dan bisa ditulis sebelum atau setelah dari blok program utama tersebut.

### C. Cara pemanggilan prosedur

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 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 variabel atau nilai integer tertentu sebagai argument untuk parameter n.

### D. Contoh program dengan prosedur

Berikut ini adalah contoh prosedur pada suatu program lengkap. Buatlah sebuah program beserta prosedur yang digunakan untuk menampilkan suatu pesan error, warning atau informasi berdasarkan masukan dari user.

Masukan terdiri dari sebuah bilangan bulat flag (0 s.d 2) dan sebuah string pesan M. Keluaran berupa string pesan M beserta jenis pesannya, yaitu error, warning atau informasi berdasarkan nilai flag 0, 1 dan 2 secara berturut-turut.

```
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"
    } else {
        jenis = "jenis tidak diketahui"
    }

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

Penulisan argumen pada parameter cetakPesan(pesan, bilangan) harus sesuai urutan tipe data pada func cetakPesan(M string, flag int), yaitu string kemudian integer.

#### E. 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.

- Parameter formal merupakan parameter yang ditulis pada saat deklarasi suatu subprogram, parameter ini berfungsi sebagai petunjuk bahwa argumen apa saja yang diperlukan pada saat pemanggilan subprogram. Misalnya parameter jari\_jari, tinggi pada deklarasi fungsi volumeTabung adalah parameter formal(jari\_jari, tinggi). Artinya ketika memanggil volumeTabung maka harus mempersiapkan dua integer (berapapun nilainya) sebagai jari\_jari dan tinggi.
- Parameter aktual adalah argument yang digunakan pada bagian parameter saat pemanggilan suatu subprogram. Banyaknya argument dan tipe data yang terdapat pada parameter aktua;, yang menyatakan nilai yang kita berikan sebagai jari-jari dan tinggi. Selain itu parameter juga dikelompokkan berdasarkan alokasi memorinya, yaitu pass by value dan pass by reference.

## II. GUIDED

## 1. Source code

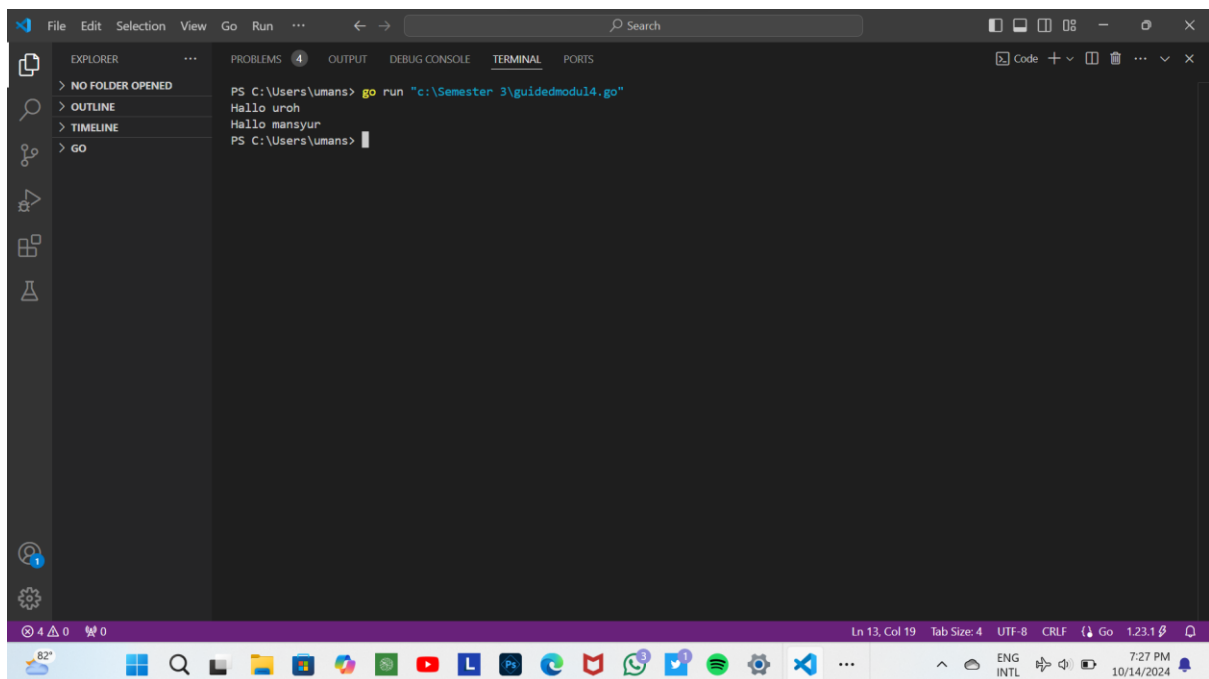
```
package main

import "fmt"

// deklarasi prosedur
func greet(name string) {
    fmt.Println("Hallo", name)
}

func main() {
    // memanggil prosedur
    greet("uroh")
    greet("mansyur")
}
```

## Screenshot program



## Deskripsi program

Program ini mendefinisikan fungsi greet yang mencetak sapaan dengan nama yang diberikan. Dalam fungsi main, fungsi greet dipanggil dua kali dengan argumen "uroh" dan "mansyur", yang menghasilkan output "Hallo uroh" dan "Hallo mansyur".

### III. UNGUIDED

#### 1. Source code

```
package main

import "fmt"

func cetakDeret(n int) {
    fmt.Print(n)

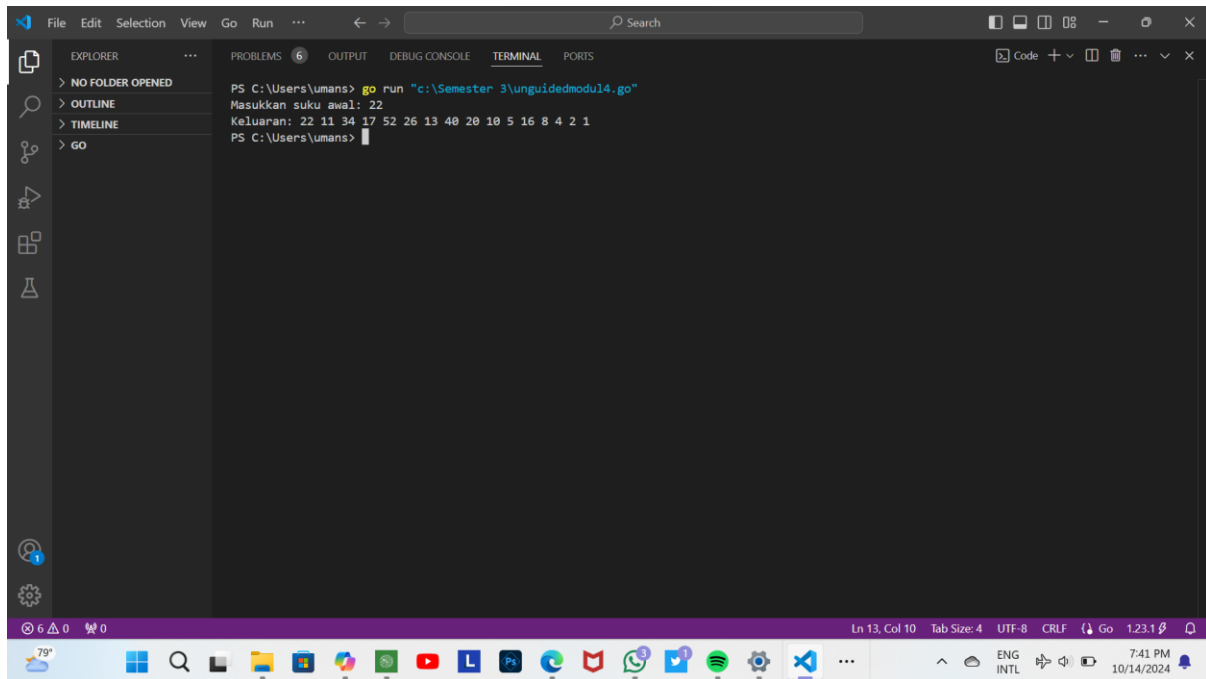
    for n != 1 {
        if n%2 == 0 {
            n /= 2
        } else {
            n = 3*n + 1
        }
        fmt.Print(" ", n)
    }
}

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

    fmt.Print("Keluaran: ")
    cetakDeret(n)

    fmt.Println()
}
```

## Screenshot code



The screenshot shows a Visual Studio Code (VS Code) interface with a terminal window open. The terminal displays the following text:

```
PS C:\Users\umans> go run "c:\Semester 3\unguidedmodul4.go"
Masukkan suku awal: 22
Keluaran: 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\umans>
```

The terminal output shows the program running successfully, taking the input '22' and producing the sequence '22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1'.

## Deskripsi program

Program ini menghitung baris deret berdasarkan suku awal yang dimasukkan pengguna. Jika bilangan genap, dibagi 2; jika ganjil, dihitung  $3*n + 1$ . Proses ini diulang hingga mencapai 1, dan seluruh deret dicetak sebagai output. Contohnya, jika suku awal 22, outputnya adalah 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1.