

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV  
PROSEDUR**



**Disusun Oleh :  
Wisnu Rananta Raditya Putra / 2311102013  
IF-11-06**

**Dosen Pengampu:  
Abednego Dwi Septiadi, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

# I. DASAR TEORI

## 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, ...

## Paremeter

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

Berdasarkan letak penulisannya pada program, 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 jari\_jari, tinggi pada deklarasi fungsi volumeTabung adalah parameter formal (teks berwarna merah). Artinya ketika memanggil volumeTabung maka kita harus mempersiapkan dua integer (berapapun nilainya) sebagai jari\_jari 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 kode di atas (teks berwarna biru) adalah parameter aktual, 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`.

#### 1. Pass by Value

Nilai pada parameter aktual akan disalin ke variabel lokal (parameter formal) pada subprogram. Artinya parameter aktual dan formal dialokasikan di dalam memori komputer dengan alamat memori yang berbeda. Subprogram dapat menggunakan nilai pada parameter formal tersebut untuk proses apapun, tetapi tidak dapat mengembalikan informasinya ke pemanggil melalui parameter aktual karena pemanggil tidak dapat mengakses memori yang digunakan oleh subprogram. `Pass by value` bisa digunakan baik oleh fungsi ataupun prosedur.

Pada notasi pseudocode, secara semua parameter formal pada fungsi adalah `pass by value`, sedangkan pada prosedur diberi kata kunci `in` pada saat penulisan parameter formal. Sedangkan pada bahasa pemrograman Go sama seperti fungsi pada pseudocode, tidak terdapat kata kunci khusus untuk parameter formal fungsi dan prosedur.

#### 2. Pass by Reference (Pointer)

Ketika parameter didefinisikan sebagai `pass by reference`, maka pada saat pemanggilan parameter formal akan berperan sebagai pointer yang menyimpan alamat memori dari parameter aktual. Sehingga perubahan nilai yang terjadi pada parameter formal tersebut akan berdampak pada parameter aktual. Artinya nilai terakhirnya akan dapat diketahui oleh si pemanggil setelah subprogram tersebut selesai dieksekusi. `Pass by reference` sebaiknya digunakan hanya untuk prosedur.

Penulisan parameter `pass by reference` pada prosedur baik pseudocode dan Go menggunakan kata kunci atau identifier khusus. Pada pseudocode menggunakan kata kunci `in/out`, sedangkan pada bahasa Go diberi identifier asterik (\*) sebelum tipe data di parameter formal yang menjadi `pass by reference`.

## II. GUIDED

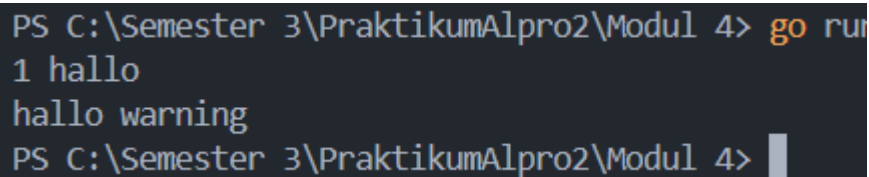
### Guided 1

```
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)
}
```

### Screenshots Output



```
PS C:\Semester 3\PraktikumAlpro2\Modul 4> go run 1 hallo
hallo warning
PS C:\Semester 3\PraktikumAlpro2\Modul 4>
```

### Deskripsi:

Program ini menerima dua input dari pengguna sebuah bilangan dan teks pesan. Bilangan digunakan untuk menentukan jenis pesan, yaitu "error" (untuk bilangan 0), "warning" (untuk bilangan 1), atau "informasi" (untuk bilangan 2). Fungsi cetakPesan kemudian mencetak pesan yang diberikan diikuti dengan jenis pesan yang sesuai.

### Guided 2

```
package main
import "fmt"
```

```

func sendEmailNotification(email string){
    fmt.Printf("Mengirim email ke %s: Pendaftaran berhasil.\n", email)
}

func main(){
    emails := []string{"user1@example.com", "user2@example.com",
"user3@example.com"}

    fmt.Println("Mengirim email ke pengguna yang baru terdaftar:")
    for _, email := range emails{
        sendEmailNotification(email)
    }
}

```

Screenshots Output:

```

PS C:\Semester 3\PraktikumAlpro2\Modul 4> go run "c:\Semest
Mengirim email ke pengguna yang baru terdaftar:
Mengirim email ke user1@example.com: Pendaftaran berhasil.
Mengirim email ke user2@example.com: Pendaftaran berhasil.
Mengirim email ke user3@example.com: Pendaftaran berhasil.
PS C:\Semester 3\PraktikumAlpro2\Modul 4> 

```

Deskripsi :

Program di atas adalah program sederhana dalam bahasa Go yang bertujuan untuk mengirim notifikasi email kepada pengguna yang baru terdaftar. Program dimulai dengan mendefinisikan daftar email pengguna dalam bentuk array (emails). Kemudian, melalui perulangan for, setiap email dalam daftar tersebut diproses satu per satu dan dikirimkan notifikasi menggunakan fungsi sendEmailNotification. Fungsi ini menampilkan pesan konfirmasi yang menunjukkan bahwa email notifikasi berhasil dikirim ke masing-masing pengguna. Outputnya berupa pesan yang memberi tahu bahwa email telah dikirim ke setiap alamat yang ada dalam daftar.

### Guided 3

```

package main
import "fmt"

func f1(x, y int) float64 {
    var hasil float64
    hasil = float64(2*x) - 0.5*float64(y) + 3.0
    return hasil
}

func f2(x, y int, hasil *float64){
    *hasil = float64(2*x) - 0.5*float64(y) + 3.0
}

```

```

}

func main(){
    var a, b int
    var c float64

    fmt.Print("enter two integers: ")
    fmt.Scan(&a, &b)

    f2(a, b, &c)

    fmt.Println("result from f2 (stored in c):", c)

    resultF1 := f1(b, a)
    fmt.Println("Result from f1:", resultF1)
}

```

Screenshots Output:

```

PS C:\Semester 3\PraktikumAlpro2\Modul 4> go run "c
enter two integers: 1 2
result from f2 (stored in c): 4
Result from f1: 6.5
PS C:\Semester 3\PraktikumAlpro2\Modul 4>

```

Deskripsi:

Program di atas merupakan program Go yang mendefinisikan dua fungsi: f1 dan f2. Fungsi f1 menerima dua parameter integer (x dan y) dan mengembalikan nilai float64, sedangkan fungsi f2 juga menerima dua parameter integer, tetapi hasil perhitungannya disimpan di variabel pointer (hasil). Program meminta input dua bilangan dari user, lalu menghitung hasil menggunakan f2 dan menyimpannya di variabel c, serta menghitung hasil lain dengan f1 yang langsung dicetak.

### III. UNGUIDED

#### Unguided 1

##### Study Case:

Minggu ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakah kalian membantu Jonas? (tidak tentunya ya :p)

Masukan terdiri dari empat buah bilangan asli  $a$ ,  $b$ ,  $c$ , dan  $d$  yang dipisahkan oleh spasi, dengan syarat  $a \geq c$  dan  $b \geq d$ .

Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi  $a$  terhadap  $c$ , sedangkan baris kedua adalah hasil permutasi dan kombinasi  $b$  terhadap  $d$ .

Catatan: permutasi (P) dan kombinasi (C) dari  $n$  terhadap  $r$  ( $n \geq r$ ) dapat dihitung dengan menggunakan persamaan berikut!

$$P(n, r) = \frac{n!}{(n-r)!}, \text{ Sedangkan } C(n, r) = \frac{n!}{r!(n-r)!}$$

```
//WISNU RANANTA RADITYA PUTRA (2311102013) IF-11-06

package main

import (
    "fmt"
)

func faktorial(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) int {
    return faktorial(n) / faktorial(n-r)
}

func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}
```

```

func main() {
    var a, b, c, d int

    fmt.Print("Masukkan a, b, c, d: ")
    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

    if a >= c && b >= d {
        p1_2311102013 := permutasi(a, c)
        c1_2311102013 := kombinasi(a, c)

        p2_2311102013 := permutasi(b, d)
        c2_2311102013 := kombinasi(b, d)

        fmt.Printf("P(a, c) = %d, K(a, c) = %d\n", p1_2311102013,
c1_2311102013)
        fmt.Printf("P(b, d) = %d, K(b, d) = %d\n", p2_2311102013,
c2_2311102013)
    } else {
        fmt.Println("Syarat a >= c dan b >= d harus dipenuhi.")
    }
}

```

#### Screenshots Output:

```

PS C:\Semester 3\PraktikumAlpro2\Modul 3> go run "c:\Semester 3\PraktikumAlpro2\Modul 3\unguided\unguided-1.go"
Masukkan input = 8 0 2 0
56, 28
1, 1
PS C:\Semester 3\PraktikumAlpro2\Modul 3> 

```

#### Deskripsi:

Program ini merupakan program sederhana yang menggunakan bahasa Go untuk menghitung permutasi dan kombinasi dua pasang angka (a, c) dan (b, d). Pertama, program meminta input dari pengguna berupa empat angka (a, b, c, d). Kemudian, program memeriksa apakah nilai a lebih besar atau sama dengan c, dan b lebih besar atau sama dengan d. Jika syarat ini terpenuhi, program menghitung dan mencetak nilai permutasi serta kombinasi dari (a, c) dan (b, d) menggunakan fungsi `faktorial`, `permutasi`, dan `kombinasi` yang didefinisikan. Namun, jika syarat tersebut tidak dipenuhi, program akan menampilkan pesan kesalahan.



## Unguided 2

### Soal Study Case:

Kompetisi pemrograman tingkat nasional berlangsung ketat. Setiap peserta diberikan 8 soal yang harus dapat diselesaikan dalam waktu 5 jam saja. Peserta yang berhasil menyelesaikan soal paling banyak dalam waktu paling singkat adalah pemenangnya.

Buat program gema yang mencari pemenang dari daftar peserta yang diberikan. Program harus dibuat modular, yaitu dengan membuat prosedur `hitungSkor` yang mengembalikan total soal dan total skor yang dikerjakan oleh seorang peserta, melalui parameter formal. Pembacaan nama peserta dilakukan di program utama, sedangkan waktu pengerjaan dibaca di dalam prosedur.

#### **prosedur `hitungSkor`(in/out soal, skor : integer)**

Setiap baris masukan dimulai dengan satu string nama peserta tersebut diikuti dengan adalah 8 integer yang menyatakan berapa lama (dalam menit) peserta tersebut menyelesaikan soal. Jika tidak berhasil atau tidak mengirimkan jawaban maka otomatis dianggap menyelesaikan dalam waktu 5 jam 1 menit (301 menit).

Satu baris keluaran berisi nama pemenang, jumlah soal yang diselesaikan, dan nilai yang diperoleh. Nilai adalah total waktu yang dibutuhkan untuk menyelesaikan soal yang berhasil diselesaikan.

```
//WISNU RANANTA RADITYA PUTRA (2311102013) IF-11-06

package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

func hitungSkor(jmlSoal *int, totalSkor *int, waktuPeserta_2311102013
[]int) {
    *jmlSoal = 0
    *totalSkor = 0
    for _, waktu := range waktuPeserta_2311102013 {
        if waktu <= 300 {
            *totalSkor += waktu
            *jmlSoal += 1
        }
    }
}

func main() {
```

```

var nama_2311102013 string
var waktuPeserta_2311102013 []int
var pemenang_2311102013 string
var maxSoal, minWaktu int
reader := bufio.NewReader(os.Stdin)
for {
    line, _ := reader.ReadString('\n')
    line = strings.TrimSpace(line)
    if line == "Selesai" {
        break
    }
    input := strings.Fields(line)
    nama_2311102013 = input[0]
    waktuPeserta_2311102013 = make([]int, 0)
    for _, w := range input[1:] {
        waktu, err := strconv.Atoi(w)
        if err == nil {
            waktuPeserta_2311102013 = append(waktuPeserta_2311102013,
waktu)
        }
    }
    var jmlSoal, totalSkor int
    hitungSkor(&jmlSoal, &totalSkor, waktuPeserta_2311102013)
    if jmlSoal > maxSoal || (jmlSoal == maxSoal &&
        totalSkor < minWaktu) {
        pemenang_2311102013 = nama_2311102013
        maxSoal = jmlSoal
        minWaktu = totalSkor
    }
}
if pemenang_2311102013 != "" {
    fmt.Printf("%s %d %d\n", pemenang_2311102013, maxSoal, minWaktu)
} else {
    fmt.Println("Tidak ada peserta.")
}
}

```

Screenshots Output:

```

PS C:\Semester 3\PraktikumAlpro2\Modul 4> go run "
astuti 20 50 301 301 61 71 75 10
bertha 25 47 301 26 50 60 65 21
Selesai
bertha 7 294
PS C:\Semester 3\PraktikumAlpro2\Modul 4>

```

Deskripsi:

Program ini merupakan program sederhana yang menggunakan bahasa Go untuk mencari siapa pemenang dari sebuah lomba pemrograman. Setiap peserta harus

menyelesaikan 8 soal, dan program ini akan menghitung berapa soal yang bisa mereka selesaikan dalam waktu kurang dari atau sama dengan 300 menit. Program membaca data peserta satu per satu, menghitung berapa banyak soal yang berhasil dikerjakan dan berapa total waktu yang dihabiskan. Pemenang ditentukan dari siapa yang bisa menyelesaikan soal paling banyak. Kalau ada yang menyelesaikan soal dengan jumlah yang sama, pemenang ditentukan dari siapa yang mengerjakannya paling cepat. Program ini terus menerima input sampai kata "Selesai" diketik, lalu akan mengumumkan pemenangnya dengan jumlah soal dan total waktu yang mereka habiskan

### Unguided 3

#### Soal Study Case:

Skiena dan Revilla dalam Programming Challenges mendefinisikan sebuah deret bilangan. Deret dimulai dengan sebuah bilangan bulat  $n$ . Jika bilangan  $n$  saat itu genap, maka suku berikutnya adalah  $\frac{1}{2}n$ , tetapi jika ganjil maka suku berikutnya bernilai  $3n+1$ . Rumus yang sama digunakan terus menerus untuk mencari suku berikutnya. Deret berakhir ketika suku terakhir bernilai 1. Sebagai contoh jika dimulai dengan  $n=22$ , maka deret bilangan yang diperoleh adalah: 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Untuk suku awal sampai dengan 1000000, diketahui deret ini selalu mencapai suku dengan nilai 1. Buat program skiena yang akan mencetak setiap suku dari deret yang dijelaskan diatas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal. prosedur cetakDeret(in  $n$  : integer )

Masukan berupa satu bilangan integer positif yang lebih kecil dari 1000000.

Keluaran terdiri dari satu baris saja. Setiap suku dari deret tersebut dicetak dalam baris yang dan dipisahkan oleh sebuah spasi.

```
//WISNU RANANTA RADITYA PUTRA (2311102013) IF-11-06

package main

import (
    "fmt"
)

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

```

    fmt.Println(1)
}

func main() {
    var n_2311102013 int
    fmt.Println("Masukkan nilai suku awal (bilangan positif <
1000000):")
    fmt.Scan(&n_2311102013)

    if n_2311102013 <= 0 || n_2311102013 >= 1000000 {
        fmt.Println("Masukan tidak valid, masukkan bilangan positif
kurang dari 1000000.")
        return
    }

    cetakDeret(n_2311102013)
}

```

#### Screenshots Output:

```

PS C:\Semester 3\PraktikumAlpro2\Modul 4> go run "c:\Semester 3\PraktikumAlpro2\Modul 4\unguided\unguided-3.go"
Masukkan nilai suku awal (bilangan positif < 1000000):
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Semester 3\PraktikumAlpro2\Modul 4>

```

#### Deskripsi:

Program ini merupakan program sederhana yang menggunakan bahasa Go untuk mencetak deret bilangan. User diminta memasukkan sebuah bilangan positif kurang dari 1.000.000. Jika bilangan yang dimasukkan valid, program akan memulai deret dari angka tersebut. Jika bilangan genap, maka akan dibagi dua, dan jika ganjil, akan dikalikan tiga lalu ditambah satu. Proses ini terus berlanjut sampai mencapai angka 1, kemudian deret selesai dan dicetak ke layar. Jika pengguna memasukkan angka yang tidak valid (bilangan negatif atau lebih dari 1.000.000), program akan menampilkan pesan kesalahan dan berhenti.