

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV  
PROSEDUR**



**Disusun Oleh :**

**Hamzah Ziyad Ibadurrohman / 2311102254**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### Dasar Teori

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`, `mulal`, ...

#### 4.2 Deklarasi Procedure

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

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

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.

Contoh deklarasi prosedur mencetak  $n$  nilai pertama dari deret Fibonacci.

Notasi Algoritma	
1	procedure cetakNFibo(in n : integer)
2	kamus
3	f1, f2, f3, i : integer
4	algoritma
5	f2 $\leftarrow$ 0
6	f3 $\leftarrow$ 1
7	for i $\leftarrow$ 1 to n do
8	output(f3)
9	f1 $\leftarrow$ f2
10	f2 $\leftarrow$ f3
11	f3 $\leftarrow$ 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	}

Catatan: Kata kunci **in** pada contoh di atas akan dijelaskan pada materi parameter di modul 5 ini.

## 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 cetakNFibo 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 $\leftarrow$ 5
6	cetakNFibo(x)                    {cara pemanggilan #1}
7	cetakNFibo(100)                {cara pemanggilan #2}

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

## II. GUIDED

### 1. Soal Studi Case

Buatlah sebuah program beserta prosedur yang digunakan untuk menampilkan suatu pesan error, warning atau informasi berdasarkan masukan dari user.

#### Sourcecode

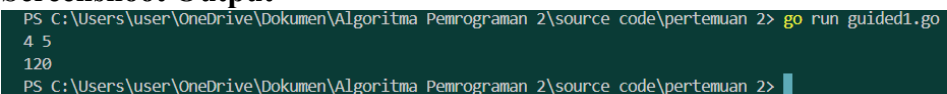
```
package main

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)
}
```

#### Screenshoot Output



```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 2> go run guided1.go
4 5
120
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 2>
```

#### Deskripsi Program

Program ini dibuat untuk menampilkan pesan error, warning dan informasi dari input pengguna. Dalam program terdapat prosedur cetakpesan. Program menggunakan if apabila flag =0, maka disebut error, jika flag = 1, pesan itu adalah warning. Jika flag = 3, maka akan disebut informasi.

## 2. Soal Studi Case

Buatlah program beserta prosedur yang digunakan untuk mengirim notifikasi email

### Sourcecode

```
package main

import "fmt"

func sendEmailNotification(email string) {

    fmt.Printf("Mengirim email ke %s: Pendaftaran berhasil.\n", email)

}

func main() {

    //daftar email pengguna baru

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

    // Mengirim email ke pemberitahuan ke setiap pengguna

    fmt.Println("Mengirim email ke pengguna yang baru terdaftar:")

    for _, email := range emails {

        sendEmailNotification(email)

    }

}
```

### Screenshot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 2> go run guided2.go
masukkan panjang, lebar, dan tinggi balok:9 14 8
luas permukaan & volume adalah : 436
1008
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 2> █
```

## Deskripsi Program

Program ini dibuat untuk mengirim notifikasi ke email pengguna. Dalam program ini terdapat prosedur `sendEmailNotification`, yang menerima satu parameter berupa alamat email pengguna dan mencetak pesan notifikasi bahwa pendaftaran berhasil. Di fungsi main terdapat array `emails` yang berisi 3 alamat email yang akan dikirim notifikasi. Program ini memakai loop untuk mengulang/ iterasi setiap email dan memanggil fungsi `sendEmailNotification`. Dengan ini, program mencetak pesan ke layar yang menunjukkan bahwa email notifikasi telah dikirim ke setiap pengguna.

## 3. Soal Studi Case

buatlah program yang digunakan untuk menghitung persamaan berikut ini:

$$f(x,y) = 2x - y/2 + 3$$

### Sourcecode

```
package main

import "fmt"

func f1(x, y int) float64 { //pass by value
    var hasil float64

    hasil = float64(2*x) - 0.5*float64(y) + 3.0

    return hasil
}
```

```

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

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

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

    f2(a, b, &c)

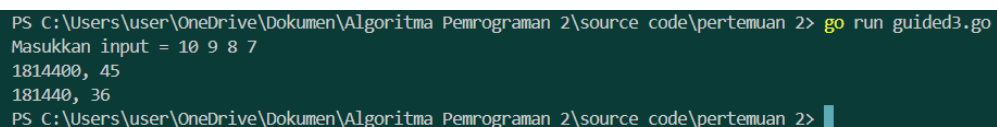
    fmt.Println("Result from f2 (stored inc):", c)

    resultF1 := f1(b, a)

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

```

## Screenshot Output



```

PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 2> go run guided3.go
Masukkan input = 10 9 8 7
1814400, 45
181440, 36
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 2>

```

## Deskripsi Program

Kode Golang ini dirancang untuk menghitung persamaan dari  $f(x) = 2x - y/2 + 3$ . Dalam program ini terdapat prosedur f1 dan f2 sebagai penghitung persamaan  $f(x, y)$ . Dalam fungsi main, pengguna diminta memasukkan dua bilangan bulat,



dan program akan memanggil prosedur f2 dan f1 . f1(x, y int) menggunakan pass by value, sehingga salinan nilai dari x dan y digunakan dalam prosedur, dan perubahan pada nilai-nilai ini tidak akan memengaruhi nilai asli di luar prosedur.

f2(x, y int, hasil \*float64) menggunakan pass by reference, di mana pointer ke variabel hasil diberikan ke prosedur, memungkinkan prosedur untuk mengubah langsung nilai hasil yang ada di luar prosedur.

### III. UNGUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

#### Soal Studi 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!

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

#### Sourcecode

```
package main

import "fmt"

var a, b, c, d int
var hasilPermutasi, hasilKombinasi int

func faktorial(n int, hasil *int) {
```

```

        *hasil = 1

        for i := 1; i <= n; i++ {
            *hasil = *hasil * i
        }
    }

    func permutasi(n, r int, hasil *int) {
        var faktN, faktNMinusR int
        faktorial(n, &faktN)
        faktorial(n-r, &faktNMinusR)
        *hasil = faktN / faktNMinusR
    }

    func kombinasi(n, r int, hasil *int) {
        var faktN, faktR, faktNMinusR int
        faktorial(n, &faktN)
        faktorial(r, &faktR)
        faktorial(n-r, &faktNMinusR)
        *hasil = faktN / (faktR * faktNMinusR)
    }

    func main() {

        fmt.Print("Masukkan input = ")
        fmt.Scan(&a, &b, &c, &d)

        if a >= c && b >= d {

            permutasi(a, c, &hasilPermutasi)
            kombinasi(a, c, &hasilKombinasi)
            fmt.Printf("%d, %d\n", hasilPermutasi,
hasilKombinasi)

            permutasi(b, d, &hasilPermutasi)
            kombinasi(b, d, &hasilKombinasi)
            fmt.Printf("%d, %d\n", hasilPermutasi,
hasilKombinasi)
        } else {
            fmt.Println("Syarat tidak terpenuhi: a harus
>= c dan b harus >= d")
        }
    }
}

```

## Screenshot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 3> go run unguided1.go
Masukkan input = 9 8 7 6
181440, 36
20160, 28
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 3> █
```

## Deskripsi Program

Kode Golang ini dirancang untuk menghitung permutasi dan kombinasi dari empat bilangan bulat yang dimasukkan oleh pengguna. Dan program ini memakai prosedur untuk menghitung faktorial, permutasi dan kombinasi. Program mendeklarasikan variabel a, b, c, dan d, dan memiliki fungsi untuk menghitung faktorial, permutasi, dan kombinasi. Dalam fungsi main, pengguna diminta memasukkan empat angka, dan program memeriksa apakah  $a \geq c$  dan  $b \geq d$ . Jika syarat terpenuhi, program mencetak nilai permutasi dan kombinasi untuk pasangan (a, c) dan (b, d). Jika tidak, program menampilkan pesan kesalahan.

## 2. Soal Studi 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.

## Sourcecode

```
package main

import "fmt"

var nama string
var soal, skor int
```

```
func HitungSkor(jawab [8]int, soal *int, skor *int) {  
    *soal = 0  
    *skor = 0  
  
    for i := 0; i < 8; i++ {  
        if jawab[i] != 301 {  
            (*soal)++  
            *skor += jawab[i]  
        }  
    }  
}  
  
func main() {  
    var jawab [8]int  
    var nama, pemenang string  
    var skormin, soalmaks int  
  
    soalmaks = 0  
    skormin = 999999  
  
    for {  
        fmt.Print("Nama peserta: ")  
        fmt.Scan(&nama)
```

```

        if nama == "Selesai" {

            break

        }

        fmt.Println("Waktu pengerjaan 8 soal: ")

        for i := 0; i < 8; i++ {

            fmt.Scan(&jawab[i])

        }

        var soalSelesai, totalWaktu int

        HitungSkor(jawab, &soalSelesai, &totalWaktu)

        if soalSelesai > soalmaks || (soalSelesai ==
soalmaks && totalWaktu < skormin) {

            pemenang = nama

            soalmaks = soalSelesai

            skormin = totalWaktu

        }

    }

    fmt.Printf("%s %d %d\n", pemenang, soalmaks, skormin)

}

```

**Screenshot output**

```

PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 3> go run unguided2.go
Nama peserta: hamzah
Waktu pengerjaan 8 soal:
98 27 85 98 76 299 85 93
Nama peserta: anto
Waktu pengerjaan 8 soal:
87 30 90 100 90 301 89 100
Nama peserta: Selesai
hamzah 8 861
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 3>

```

## Deskripsi Output

Program ini dibuat untuk menentukan pemenang dari sebuah kompetisi pemrograman berdasarkan jumlah soal yang berhasil diselesaikan dan waktu pengerjaannya dengan menggunakan prosedur. Prosedur HitungSkor dibuat untuk menghitung jumlah soal yang diselesaikan serta total waktu pengerjaannya. Dalam prosedur ini, jika sebuah soal tidak berhasil diselesaikan (waktu pengerjaan 301), maka soal tersebut tidak dihitung, namun jika soal berhasil dikerjakan, jumlah soal akan ditambahkan dan waktu pengerjaan soal tersebut juga ditambahkan ke skor total.

Di dalam fungsi main, input dari peserta berupa nama dan waktu pengerjaan 8 soal. Program akan terus meminta input hingga nama "Selesai" dimasukkan. Setiap kali waktu pengerjaan soal dimasukkan, program akan menggunakan HitungSkor untuk menghitung total soal dan waktu pengerjaan. Peserta dengan jumlah soal yang diselesaikan terbanyak akan menjadi kandidat pemenang. Jika jumlah soal yang diselesaikan sama, maka pemenang ditentukan berdasarkan waktu pengerjaan terkecil. Pada akhir program, nama pemenang, jumlah soal yang diselesaikan, dan total waktu pengerjaan akan dicetak.

### 3. Soal Studi 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  $1/2n$ , 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 selalu mencapai suku dengan nilai 1.

Buat program skiena yang akan mencetak setiap suku dari deret yang dijelaskan di atas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal.

prosedure 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.

### Sourcecode

```
package main

import "fmt"

func Cetakderet(n int) {
    for n != 1 {
        fmt.Printf("%d ", n)

        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }

    fmt.Println(n)
}

func main() {
    var n int

    fmt.Print("Masukkan nilai awal deret: ")
}
```

```
        fmt.Scan(&n)

        Cetakderet(n)

    }
```

## Screenshot Output

```
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 3> go run unguided3.go
Masukkan nilai awal deret: 16
16 8 4 2 1
PS C:\Users\user\OneDrive\Dokumen\Algoritma Pemrograman 2\source code\pertemuan 3> |
```

## Deskripsi Program

Program ini dibuat untuk mencetak setiap suku dari deret. Ketika  $n$  adalah ganjil maka  $n = 3 \times n + 1$ . dan jika  $n$  ganjil,  $n = n / 2$ . program ini memakai prosedur untuk cetak deret untuk mencetak nilai deret tersebut. Di dalam fungsi main, program meminta nilai suku untuk diproses. Ketika sudah diinput, program memanggil prosedur untuk mencetak deret. Hasil output ialah nilai cetak yang dibuat oleh prosedur cetakderet.