

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4  
PROSEDUR**



**Telkom  
University**

Oleh:

Muhammad Faris Rachmadi

103112400079

IF12-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

## I. DASAR TEORI

### 4.1 Definisi Prosedure

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, ...

### 4.2 Deklarasi Prosedure

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

	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	}

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.

### **4.3 Cara Pemanggilan Prosedure**

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.

## II. GUIDED

### Guided 1

Code:

```
10311240079_Guided1 > go guided1.go > ...
1 package main
2
3 import "fmt"
4
5 func hitungGaji(nama string, gajiPokok float64, jamLembur int) {
6     bonusLembur := float64(jamLembur) * 50000
7     totalGaji := gajiPokok + bonusLembur
8
9     fmt.Println("\n=== Slip Gaji ===")
10    fmt.Println("Nama Karyawan:", nama)
11    fmt.Printf("Gaji Pokok : Rp%.2f\n", gajiPokok)
12    fmt.Printf("Bonus Lembur: Rp%.2f (%d jam x Rp50,000)\n", bonusLembur, jamLembur)
13    fmt.Printf("Total Gaji : Rp%.2f\n", totalGaji)
14 }
15
16 func main() {
17     var nama string
18     var gajiPokok float64
19     var jamLembur int
20
21     fmt.Print("Masukkan Nama Karyawan: ")
22     fmt.Scanln(&nama)
23     fmt.Print("Masukkan Gaji Pokok: ")
24     fmt.Scanln(&gajiPokok)
25     fmt.Print("Masukan Jumlah Jam Lembur: ")
26     fmt.Scanln(&jamLembur)
27     hitungGaji(nama, gajiPokok, jamLembur)
28 }
29
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\10311240079_MODUL 4> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\10311240079_Guided1.go"
Masukkan Nama Karyawan: yanto
Masukkan Gaji Pokok: 250000
Masukan Jumlah Jam Lembur: 5


=== Slip Gaji ===
Nama Karyawan: yanto
Gaji Pokok : Rp250000.00
Bonus Lembur: Rp250000.00 (5 jam x Rp50,000)
Total Gaji : Rp500000.00
```

Deskripsi:

Program ini digunakan untuk menghitung total gaji karyawan yang mencakup gaji pokok dan bonus lembur berdasarkan jumlah jam lembur yang dilakukan. Program meminta input dari pengguna berupa nama karyawan, gaji pokok, dan jumlah jam lembur. Bonus lembur dihitung dengan mengalikan jumlah jam lembur dengan Rp50.000, kemudian dijumlahkan dengan gaji pokok untuk mendapatkan total gaji. Program kemudian menampilkan slip gaji yang mencakup rincian nama karyawan, gaji pokok, bonus lembur, dan total gaji yang harus diterima.

## Guided 2

Code:

```
103112400079_Guided2 >  Guided2.go > ...
1  package main
2
3  import "fmt"
4
5  func hitungRataRata(nama string, nilai1, nilai2, nilai3 float64) {
6      ratarata := (nilai1 + nilai2 + nilai3) / 3
7      status := "Tidak Lulus"
8      if ratarata >= 60 {
9          status = "Lulus"
10     }
11     fmt.Println("\n=== Hasil Akademik ===")
12     fmt.Println("Nama Mahasiswa :", nama)
13     fmt.Printf("Nilai 1      : %.2f\n", nilai1)
14     fmt.Printf("Nilai 2      : %.2f\n", nilai2)
15     fmt.Printf("Nilai 3      : %.2f\n", nilai3)
16     fmt.Printf("Rata-rata    : %.2f\n", ratarata)
17     fmt.Println("Status      :", status)
18 }
19 func main() {
20     var nama string
21     var nilai1, nilai2, nilai3 float64
22     fmt.Print("Masukkan Nama Mahasiswa: ")
23     fmt.Scanln(&nama)
24     fmt.Print("Masukkan Nilai 1: ")
25     fmt.Scanln(&nilai1)
26     fmt.Print("Masukkan Nilai 2: ")
27     fmt.Scanln(&nilai2)
28     fmt.Print("Masukkan Nilai 3: ")
29     fmt.Scanln(&nilai3)
30     hitungRataRata(nama, nilai1, nilai2, nilai3)
31 }
```

Output:

```
Masukkan Nilai 1: 85
Masukkan Nilai 2: 90
Masukkan Nilai 3: 95

=== Hasil Akademik ===
Nama Mahasiswa : Firas
Nilai 1      : 85.00
Nilai 2      : 90.00
Nilai 3      : 95.00
Rata-rata    : 90.00
Status      : Lulus
```

Deskripsi:

Program ini menghitung rata-rata nilai mahasiswa dari tiga nilai yang dimasukkan dan menentukan status kelulusan berdasarkan rata-rata. Jika rata-rata nilai 60 atau lebih, mahasiswa dinyatakan "Lulus", jika kurang dari 60, dinyatakan "Tidak Lulus". Program menampilkan hasil berupa nama mahasiswa, nilai, rata-rata, dan status kelulusan.

### III. UNGUIDED

#### Unguided 1

Code:

```
10311240079_Unguided1 > go Unguided1.go > main
1 package main
2
3 // Muhammad Faris Rachmadi
4 // 10311240079
5 import "fmt"
6
7 func fact(n int) int {
8     if n == 0 {
9         return 1
10    }
11    result := 1
12    for i := 1; i <= n; i++ {
13        result *= i
14    }
15    return result
16 }
17
18 func permut(n, r int) int {
19     return fact(n) / fact(n-r)
20 }
21
22 func combi(n, r int) int {
23     return fact(n) / (fact(r) * fact(n-r))
24 }
25
26 func main() {
27     var a, b, c, d int
28     fmt.Println("Masukkan 4 angka:")
29     fmt.Scan(&a, &b, &c, &d)
30
31     p1 := permut(a, c)
32     c1 := combi(a, c)
33     p2 := permut(b, d)
34     c2 := combi(b, d)
35
36     fmt.Println(p1, c1, p2, c2)
37 }
38 }
```

Output:

```
PROBLEMS OUTPUT TERMINAL PORTS
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\10311240079_MODUL 4> go run "c:\Use
Masukkan 4 angka:
5 10 3 10
60 10 3628800 1
```

Deskripsi:

Program ini menghitung permutasi dan kombinasi dari dua pasangan angka yang dimasukkan oleh pengguna. Program pertama-tama meminta input berupa empat angka, kemudian menghitung permutasi dan kombinasi untuk dua set angka. Fungsi fact digunakan untuk menghitung faktorial suatu angka, yang kemudian digunakan dalam fungsi permut untuk menghitung permutasi ( $nPr$ ) dan combi untuk menghitung kombinasi ( $nCr$ ). Hasil dari perhitungan permutasi dan kombinasi ditampilkan untuk masing-masing pasangan angka yang dimasukkan.

## Unguided 2

### Code:

```
103112400079_Unguided2 > go run Unguided2.go > main
1 package main
2
3 // Muhammad Faris Rachmadi || 103112400079
4 import "fmt"
5
6 func hitungSkor(waktu []int, jumlahSoal *int, totalWaktu *int) {
7     batasWaktu := 301
8     *jumlahSoal, *totalWaktu = 0, 0
9     for _, t := range waktu {
10         if t < batasWaktu {
11             *jumlahSoal++
12             *totalWaktu += t
13         }
14     }
15 }
16
17 func main() {
18     var pemenang string
19     var maxSoal, minWaktu int = 0, 1000000
20
21     for {
22         var nama string
23         var waktu [8]int
24         _, err := fmt.Scan(&nama)
25         if err != nil || nama == "Selesai" {
26             break
27         }
28
29         for i := 0; i < 8; i++ {
30             _, err = fmt.Scan(&waktu[i])
31             if err != nil {
32                 break
33             }
34         }
35
36         var jumlahSoal, totalWaktu int
37         hitungSkor(waktu[:], &jumlahSoal, &totalWaktu)
38
39         if jumlahSoal > maxSoal || (jumlahSoal == maxSoal && totalWaktu < minWaktu) {
40             maxSoal = jumlahSoal
41             minWaktu = totalWaktu
42             pemenang = nama
43         }
44     }
45
46     fmt.Println(pemenang, maxSoal, minWaktu)
47 }
```

### Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 4> go run "c:\Users\Fari
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294
```

### Deskripsi:

Program ini menentukan pemenang kompetisi berdasarkan jumlah soal yang dijawab dalam waktu kurang dari 301 detik. Setiap peserta memasukkan nama dan waktu untuk 8 soal. Pemenang ditentukan oleh jumlah soal yang dijawab, dan jika sama, pemenang adalah yang menggunakan waktu lebih sedikit. Program akan terus menerima input hingga "Selesai" dimasukkan, lalu menampilkan nama pemenang, jumlah soal, dan total waktu.

## Unguided 3

Code:

```
103112400079_Unguided3 > -go Unguided3.go > ...
1  package main
2
3  import "fmt"
4
5  func cetakDeret(n int) {
6      for n != 1 {
7          fmt.Printf("%d ", n)
8          if n%2 == 0 {
9              n /= 2
10             } else {
11                 n = 3*n + 1
12             }
13         }
14         fmt.Println(n)
15     }
16
17 func main() {
18     var n int
19     fmt.Scan(&n)
20
21     if n < 1 || n >= 1000000 {
22         fmt.Println("Bilangan harus di antara 1 dan 999999.")
23         return
24     }
25
26     cetakDeret(n)
27 }
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\103112400079_MODUL 4> go run "c:\Users\Faris\
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Deskripsi:

Program ini mencetak deret angka berdasarkan aturan berikut: jika angka genap, bagi dengan 2; jika angka ganjil, kalikan dengan 3 dan tambahkan 1. Proses ini terus berlanjut hingga angka mencapai 1. Program meminta input berupa angka dan memeriksa apakah angka tersebut berada dalam rentang 1 hingga 999,999. Jika valid, program akan mencetak deret angka sesuai dengan aturan tersebut. Jika input tidak valid, program akan menampilkan pesan kesalahan.



#### **IV. KESIMPULAN**

Kesimpulan dari laporan praktikum ini adalah bahwa prosedur dalam pemrograman digunakan untuk menyederhanakan dan mengorganisir kode program dengan cara mengelompokkan instruksi tertentu menjadi satu unit yang dapat dipanggil pada program utama. Praktikum ini membahas berbagai contoh penerapan prosedur dalam pemrograman menggunakan bahasa Go, seperti perhitungan gaji karyawan, rata-rata nilai mahasiswa, permutasi dan kombinasi, serta algoritma penentuan pemenang kompetisi. Prosedur ini membantu memecah masalah kompleks menjadi bagian-bagian yang lebih sederhana, meningkatkan keterbacaan kode, dan memungkinkan penggunaan kembali blok kode dalam program yang lebih besar.

## **V. REFERENSI**

MODUL 4 PRAKTIKUM ALGORITMA PEMROGRAMAN 2 – PROSEDURE