

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4  
PROSEDUR**



Oleh:

ARIEL AHNAF KUSUMA

103112400050

IF-12-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## **I. 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:

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.

### **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 argument untuk parameter n.

### **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.

### **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.

### **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.

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.

### **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.**

- **Parameter pada fungsi sebaiknya adalah pass by value, hal ini dikarenakan fungsi bisa mengembalikan (return) nilai ke pemanggil dan tidak memberikan efek langsung pada program, walaupun tidak menutup kemungkinan menggunakan pass by reference.**
- **Penggunaan pass by reference sebaiknya pada prosedur karena prosedur tidak bisa mengembalikan nilai ke pemanggil. Dengan memanfaatkan pass by reference maka prosedur seolah-olah bisa mengirimkan nilai kepada si pemanggil.**

## II. GUIDED

### GUIDED 1

#### SOURCE CODE

```
1 package main
2
3 import "fmt"
4
5 //Prosedur untuk menghitung gaji karyawan
6 func hitungGaji(nama string, gajiPokok float64, jamLembur int) {
7     bonusLembur := float64(jamLembur) * 50000
8     totalGaji := gajiPokok + bonusLembur
9
10    fmt.Println("\n=== Slip Gaji ===")
11    fmt.Println("Nama Karyawan :", nama)
12    fmt.Printf("Gaji Pokok: Rp %.2f\n", gajiPokok)
13    fmt.Printf("Bonus Lembur: Rp%.2f(%d jam x Rp50,000)\n", bonusLembur, jamLembur)
14    fmt.Printf("Bonus Lembur : Rp%.2f (%d x Rp50.000)\n", bonusLembur, jamLembur)
15    fmt.Printf("Total Gaji : Rp%.2f\n", totalGaji)
16 }
17
18 func main() {
19     var nama string
20     var gajiPokok float64
21     var jamLembur int
22
23     //Input dari pengguna
24     fmt.Print("Masukkan Nama Karyawan:")
25     fmt.Scanln(&nama)
26
27     fmt.Print("Masukkan Gaji Pokok:")
28     fmt.Scanln(&gajiPokok)
29
30     fmt.Print("Masukan Jam Lembur:")
31     fmt.Scanln(&jamLembur)
32
33     //memanggil prosedur dengan data dari pengguna
34     hitungGaji(nama, gajiPokok, jamLembur)
35
36 }
37
```

## OUTPUT CODE

```
PS C:\ARIEL AHNAF KUSUMA\MODUL 4>  
go run "c:\ARIEL AHNAF KUSUMA\MODUL 4\guided1.go"  
Masukkan Nama Karyawan:Leclerc  
Masukkan Gaji Pokok:45000000  
Masukan Jam Lembur:6  
  
=== Slip Gaji ===  
Nama Karyawan : Leclerc  
Gaji Pokok: Rp 45000000.00  
Bonus Lembur: Rp300000.00(6 jam x Rp50,000)  
Bonus Lembur : Rp300000.00 (6 x Rp50.000)  
Total Gaji : Rp45300000.00  
PS C:\ARIEL AHNAF KUSUMA\MODUL 4> █
```

## DESKRIPSI CODE

Program di atas digunakan untuk menghitung total gaji karyawan berdasarkan gaji pokok dan jam lembur

## GUIDED 2

## SOURCE CODE

```

1  package main
2
3  import "fmt"
4
5  //Prosedur untuk menghitung rata-rata dan menentukan kelulusan
6  func hitungRataRata(nama string, nilai1, nilai2, nilai3 float64) {
7      rataRata := (nilai1 + nilai2 + nilai3) / 3
8      status := "Tidak Lulus"
9      if rataRata >= 60 {
10         status = "Lulus"
11     }
12
13     fmt.Println("\n=== Hasil Akademik ===")
14     fmt.Println("Nama Mahasiswa:", nama)
15     fmt.Printf("Nilai 1 : %.2f\n", nilai1)
16     fmt.Printf("Nilai 2 : %.2f\n", nilai2)
17     fmt.Printf("Nilai 3 : %.2f\n", nilai3)
18     fmt.Printf("Rata-rata : %.2f\n", rataRata)
19     fmt.Println("Status :", status)
20 }
21
22 func main() {
23     var nama string
24     var nilai1, nilai2, nilai3 float64
25
26     //Input dari pengguna
27     fmt.Print("Masukkan Nama Mahasiswa: ")
28     fmt.Scanln(&nama)
29
30     fmt.Print("Masukkan Nilai 1: ")
31     fmt.Scanln(&nilai1)
32
33     fmt.Print("Masukkan Nilai 2: ")
34     fmt.Scanln(&nilai2)
35
36     fmt.Print("Masukkan Nilai 3: ")
37     fmt.Scanln(&nilai3)
38
39     // Memanggil prosedur dengan data dari pengguna
40     hitungRataRata(nama, nilai1, nilai2, nilai3)
41 }

```

## OUTPUT CODE

```

PS C:\ARIEL AHNAF KUSUMA\MODUL 4>
go run "c:\ARIEL AHNAF KUSUMA\MODUL 4\guided2.go"
Masukkan Nama Mahasiswa: LanceStroll
Masukkan Nilai 1: 90
Masukkan Nilai 2: 85
Masukkan Nilai 3: 93

=== Hasil Akademik ===
Nama Mahasiswa: LanceStroll
Nilai 1 : 90.00
Nilai 2 : 85.00
Nilai 3 : 93.00
Rata-rata : 89.33
Status : Lulus
PS C:\ARIEL AHNAF KUSUMA\MODUL 4>

```

## DESKRIPSI CODE

**Program ini menentukan apakah seorang siswa lulus atau tidak berdasarkan rata-rata tiga nilai yang dimasukkan.**

**Nilai rata-rata dihitung dengan menjumlahkan ketiga angka, lalu membaginya dengan tiga. Jika hasilnya 60 atau lebih, siswa dinyatakan lulus. Jika kurang dari 60, siswa tidak lulus.**



### III. UNGUIDED

#### UNGUIDED 1

#### SOURCE CODE

```
1  package main
2
3  import "fmt"
4
5  //ARIEL AHNAF KUSUMA 103112400050
6  func fact(n int) int {
7      result := 1
8      for i := 2; i <= n; i++ {
9          result *= i
10     }
11     return result
12 }
13
14 func perm(n, r int) int {
15     return fact(n) / fact(n-r)
16 }
17
18 func comb(n, r int) int {
19     return fact(n) / (fact(r) * fact(n-r))
20 }
21
22 func main() {
23     var a, b, c, d int
24     fmt.Scan(&a, &b, &c, &d)
25     fmt.Println(perm(a, c), comb(a, c))
26     fmt.Println(perm(b, d), comb(b, d))
27 }
28
```

#### OUTPUT CODE

```
PS C:\ARIEL AHNAF KUSUMA\MODUL 4> go run "c:\ARIEL AHNAF KUSUMA\MODUL 4\unguided1.go"
5 10 3 10
60 10
3628800 1
PS C:\ARIEL AHNAF KUSUMA\MODUL 4> go run "c:\ARIEL AHNAF KUSUMA\MODUL 4\unguided1.go"
8 0 2 0
56 28
1 1
PS C:\ARIEL AHNAF KUSUMA\MODUL 4>
```

## DESKRIPSI CODE

Program ini menghitung permutasi dan kombinasi berdasarkan input. Bilangan faktorial dihitung dengan fungsi factorial, yang kemudian digunakan untuk perhitungan permutasi dan kombinasi. Untuk perhitungan permutasi, program meminta dua nilai, sedangkan untuk perhitungan kombinasi, program meminta empat nilai. Hasil perhitungan kombinasi dan permutasi akan ditampilkan secara berurutan.

## UNGUIDED 2

### SOURCE CODE

```
1  package main
2
3  import "fmt"
4
5  //ARIEL AHNAF KUSUMA 103112400050
6  func hitungSkor(waktu int, totalWaktu, totalSoal *int) {
7      if waktu <= 300 {
8          *totalSoal++
9          *totalWaktu += waktu
10     }
11 }
12
13 func main() {
14     var namaPemenang, peserta string
15     var waktu, maxSoal, minWaktu int
16     var totalWaktu, totalSoal int
17
18     fmt.Scan(&namaPemenang)
19     if namaPemenang == "Selesai" {
20         return
21     }
22
23     for i := 0; i < 8; i++ {
24         fmt.Scan(&waktu)
25         hitungSkor(waktu, &totalWaktu, &totalSoal)
26     }
27
28     maxSoal, minWaktu = totalSoal, totalWaktu
29 }
```

```

29     for {
30         fmt.Scan(&peserta)
31         if peserta == "Selesai" {
32             break
33         }
34
35         pesertaWaktu, pesertaSoal := 0, 0
36
37         for i := 0; i < 8; i++ {
38             fmt.Scan(&waktu)
39             hitungSkor(waktu, &pesertaWaktu, &pesertaSoal)
40         }
41
42         if pesertaSoal > maxSoal || (pesertaSoal == maxSoal && pesertaWaktu < minWaktu) {
43             namaPemenang, maxSoal, minWaktu = peserta, pesertaSoal, pesertaWaktu
44         }
45     }
46
47     fmt.Println(namaPemenang, maxSoal, minWaktu)
48 }
49

```

## OUTPUT CODE

```

PS C:\ARIEL AHNAF KUSUMA\MODUL 4>
go run "c:\ARIEL AHNAF KUSUMA\MODUL 4\unguided2.go"
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294
PS C:\ARIEL AHNAF KUSUMA\MODUL 4>

```

## DESKRIPSI CODE

Program ini menilai peserta berdasarkan jumlah soal yang harus diselesaikan dalam waktu kurang dari 300 detik. Peserta dengan jumlah soal terbanyak menang jika mereka memiliki waktu penyelesaian yang sama.

## UNGUIDED 3

## SOURCE CODE

```

1  package main
2
3  import "fmt"
4
5  //ARIEL AHNAF KUSUMA 103112400050
6  func cetakDeret(n int) {
7      fmt.Print(n)
8      for n != 1 {
9          if n%2 == 0 {
10             n = n / 2
11          } else {
12             n = 3*n + 1
13          }
14          fmt.Print(" ", n)
15      }
16      fmt.Println()
17  }
18
19  func main() {
20      var masukan int
21      fmt.Print("masukkan bilangan positif: ")
22      fmt.Scan(&masukan)
23      cetakDeret(masukan)
24  }
25

```

## OUTPUT CODE

```

PS C:\ARIEL AHNAF KUSUMA\MODUL 4>
go run "c:\ARIEL AHNAF KUSUMA\MODUL 4\unguided3.go"
masukkan bilangan positif: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\ARIEL AHNAF KUSUMA\MODUL 4>

```

## DESKRIPSI CODE

Program Bahasa Go di atas bertujuan untuk mencetak deret dengan suku awal  $n$  dan suku terakhir 1. Dengan menggunakan aturan yang ada, program menghasilkan deret. Sebuah deret dimulai dengan bilangan bulat yang positif. Suku berikutnya dari  $n$  genap adalah setengah  $n$ , dan suku berikutnya dari  $n$  ganjil adalah  $3n + 1$ .

#### **IV. KESIMPULAN**

**Pada modul ini, kita mempelajari konsep prosedur pemrograman, yang membantu memecah program besar menjadi bagian-bagian lebih kecil yang lebih mudah untuk dikelola. Prosedur ini mempermudah proses pemrograman dengan meningkatkan modularitas dan reusability kode serta mempermudah pemeliharaan program. Dengan menggunakan prosedur, Anda dapat menulis kode dengan lebih cepat dan mengurangi jumlah duplikasi kode yang dapat menyebabkan kesalahan.**

## **V. REFERENSI**

### **MODUL 4 ALGORITMA PEMROGRAMMAN 2**