

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4
PROSEDUR**



Oleh:

HISYAM NURDIATMOKO

103112400049

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

1. Definisi Prosedur

Prosedur merupakan bagian dari program yang berisi kumpulan instruksi yang dikemas menjadi suatu unit terpisah. Prosedur digunakan untuk mengurangi kompleksitas kode dalam program yang besar, serta memungkinkan pemrograman yang lebih modular dan terstruktur.

Suatu subprogram dikatakan sebagai prosedur apabila:

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

Prosedur memiliki fungsi yang serupa dengan instruksi dasar seperti pernyataan assignment atau fungsi bawaan dari pustaka tertentu. Oleh karena itu, nama prosedur sebaiknya berupa kata kerja atau sesuatu yang merepresentasikan proses, misalnya: *cetak*, *hitungRerata*, *cariNilai*.

2. Deklarasi Prosedur

Deklarasi prosedur dalam algoritma atau bahasa pemrograman dilakukan di luar program utama dan dapat ditulis sebelum atau sesudah blok utama. Struktur umum dari deklarasi prosedur terdiri dari:

1. Nama prosedur.
2. Parameter (jika ada).
3. Kamus variabel lokal (jika diperlukan).
4. Badan prosedur, yang berisi langkah-langkah eksekusi.

Dalam bahasa pemrograman, prosedur dapat ditulis dengan format tertentu tergantung pada sintaks bahasa yang digunakan.

3. Cara Pemanggilan Prosedur

Prosedur hanya akan dieksekusi apabila dipanggil oleh program utama atau melalui subprogram lain. Pemanggilan prosedur dilakukan dengan cara menuliskan nama prosedur beserta argumen yang sesuai dengan parameter yang telah dideklarasikan. Terdapat dua cara pemanggilan prosedur:

1. Menggunakan variabel sebagai argumen.
2. Menggunakan nilai langsung sebagai argumen.

Kesalahan dalam pemanggilan prosedur, seperti urutan parameter yang tidak sesuai atau tipe data yang tidak cocok, dapat menyebabkan error dalam eksekusi program.

4. Parameter dalam Prosedur

Parameter merupakan elemen penting dalam prosedur karena memungkinkan komunikasi antara subprogram dengan pemanggilnya. Parameter dapat dibagi menjadi beberapa jenis berdasarkan karakteristiknya:

A. Berdasarkan Letaknya dalam Program

1. Parameter Formal

- Dideklarasikan saat mendefinisikan prosedur.
- Berfungsi sebagai placeholder untuk menerima nilai saat prosedur dipanggil.

2. Parameter Aktual

- Merupakan nilai atau variabel yang diberikan sebagai argumen saat prosedur dipanggil.
- Harus memiliki jumlah dan tipe data yang sesuai dengan parameter formal.

B. Berdasarkan Alokasi Memori

1. Pass by Value

- Nilai parameter aktual disalin ke parameter formal.
- Perubahan nilai parameter formal tidak akan mempengaruhi parameter aktual.
- Cocok untuk fungsi yang tidak perlu mengubah nilai argumen.

2. Pass by Reference

- Parameter formal berisi alamat memori dari parameter aktual.
- Perubahan pada parameter formal juga akan berdampak pada parameter aktual.
- Umumnya digunakan dalam prosedur untuk memungkinkan modifikasi langsung terhadap data.

Pass by reference dalam beberapa bahasa pemrograman menggunakan simbol khusus, seperti * dalam Go untuk menunjukkan pointer.

5. Manfaat dan Keuntungan Menggunakan Prosedur

Menggunakan prosedur dalam pemrograman memberikan berbagai manfaat, di antaranya:

1. Modularitas – Memecah kode menjadi bagian-bagian yang lebih kecil dan mudah dipahami.
2. Reusability – Prosedur yang sudah dibuat dapat digunakan kembali di bagian lain dalam program.
3. Pemeliharaan yang Lebih Mudah – Perubahan dalam satu prosedur tidak akan berdampak pada seluruh kode, sehingga lebih mudah untuk diperbaiki atau ditingkatkan.
4. Peningkatan Efisiensi – Dengan mengurangi pengulangan kode, program menjadi lebih ringkas dan efisien.

II. GUIDED

Source code Guided 1:

```
103112400049_MODUL4 > 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      fmt.Println("\n=== Slip Gaji ===")
9      fmt.Println("Nama Karyawan:", nama)
10     fmt.Printf("Gaji Pokok : Rp%.2f\n", gajiPokok)
11     fmt.Printf("Bonus Lembur: Rp%.2f (%d jam x Rp50,000)\n", bonusLembur, jamLembur)
12     fmt.Printf("Total Gaji : Rp%.2f\n", totalGaji)
13 }
14
15 func main() {
16     var nama string
17     var gajiPokok float64
18     var jamLembur int
19     fmt.Print("Masukkan Nama Karyawan: ")
20     fmt.Scanln(&nama)
21     fmt.Print("Masukkan Gaji Pokok: ")
22     fmt.Scanln(&gajiPokok)
23     fmt.Print("Masukan Jumlah Jam Lembur: ")
24     fmt.Scanln(&jamLembur)
25     hitungGaji(nama, gajiPokok, jamLembur)
26 }
```

Output :

```
PS D:\1_Matkul\Alpro2\week4> go run "d:\1_Matkul\Alpro2\week4\103112400049_MODUL4\Guided1\Guided1.go"
Masukkan Nama Karyawan: Hynrx
Masukkan Gaji Pokok: 27000000
Masukan Jumlah Jam Lembur: 7

=== Slip Gaji ===
Nama Karyawan: Hynrx
Gaji Pokok : Rp27000000.00
Bonus Lembur: Rp350000.00 (7 jam x Rp50,000)
Total Gaji : Rp27350000.00
PS D:\1_Matkul\Alpro2\week4> |
```

Deskripsi :

Program Guided 1 ini adalah aplikasi sederhana untuk menghitung gaji karyawan berdasarkan gaji pokok dan jam lembur yang dimasukkan oleh pengguna. Pengguna diminta untuk memasukkan nama karyawan, gaji pokok, dan jumlah jam lembur yang telah dilakukan. Program kemudian menghitung bonus lembur dengan mengalikan jumlah jam lembur dengan tarif Rp50.000 per jam, lalu menambahkan bonus tersebut ke gaji pokok untuk mendapatkan total gaji. Hasil perhitungan gaji, termasuk gaji pokok, bonus lembur, dan total gaji, ditampilkan dalam format slip gaji

Source code Guided 2 :

```
103112400049_MODUL4 > Guided2 > go Guided2.go > main
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 :

```
PS D:\1_Matkul\Alpro2\week4> go run "d:\1_Matkul\Alpro2\week4\103112400049_MODUL4\Guided2\Guided2.go"
Masukkan Nama Mahasiswa: Hynrx
Masukkan Nilai 1: 90
Masukkan Nilai 2: 92
Masukkan Nilai 3: 97

=== Hasil Akademik ===
Nama Mahasiswa : Hynrx
Nilai 1      : 90.00
Nilai 2      : 92.00
Nilai 3      : 97.00
Rata-rata    : 93.00
Status       : Lulus
PS D:\1_Matkul\Alpro2\week4> go run "d:\1_Matkul\Alpro2\week4\103112400049_MODUL4\Guided2\Guided2.go"
Masukkan Nama Mahasiswa: Panjul
Masukkan Nilai 1: 30
Masukkan Nilai 2: 50
Masukkan Nilai 3: 10

=== Hasil Akademik ===
Nama Mahasiswa : Panjul
Nilai 1      : 30.00
Nilai 2      : 50.00
Nilai 3      : 10.00
Rata-rata    : 30.00
Status       : Tidak Lulus
```

Deskripsi :

Program Guided 2 ini digunakan untuk menghitung rata-rata nilai mahasiswa dan menentukan apakah mahasiswa tersebut lulus atau tidak. Program akan meminta untuk memasukkan nama mahasiswa dan tiga nilai yang ingin dihitung rata-ratanya. Kemudian program akan menghitung rata-rata dari ketiga nilai tersebut. Jika rata-rata nilai lebih dari atau sama dengan 60, status mahasiswa akan ditampilkan sebagai "Lulus", sedangkan jika kurang dari 60, statusnya adalah "Tidak Lulus". Hasilnya, termasuk nama mahasiswa, nilai-nilai yang dimasukkan, rata-rata, dan status kelulusan, akan ditampilkan dalam format slip hasil akademik.

III. UNGUIDED

Source code Unguided 1 :

```
103112400049_MODUL4 > Unguided1 > 103112400049_Unguided1.go > factorial
1 package main
2 //HISYAM NURDIATMOKO 103112400049
3 import "fmt"
4
5 func factorial(n int) int {
6     result := 1
7     for i := 1; i <= n; i++ {
8         result *= i
9     }
10    return result
11}
12
13 func permutation(n, r int) int {
14     return factorial(n) / factorial(n-r)
15}
16
17 func combination(n, r int) int {
18     return factorial(n) / (factorial(r) * factorial(n-r))
19}
20
21 func main() {
22     var a, b, c, d int
23     fmt.Scan(&a, &b, &c, &d)
24     fmt.Println(permutation(a, c), combination(a, c))
25     fmt.Println(permutation(b, d), combination(b, d))
26}
```

Output:

```
PS D:\1_Matkul\Alpro2\week4> go run "d:\1_Matkul\Alpro2\week4\103112400049_MODUL4\Unguided1\103112400049_Unguided1.go"
5 10 3 10
60 10
3628800 1
PS D:\1_Matkul\Alpro2\week4> go run "d:\1_Matkul\Alpro2\week4\103112400049_MODUL4\Unguided1\103112400049_Unguided1.go"
8 0 2 0
56 28
1 1
```

Deskripsi:

Program Unguided 1 ini digunakan untuk menghitung permutasi dan kombinasi berdasarkan input yang diberikan oleh pengguna. Fungsi factorial digunakan untuk menghitung faktorial dari suatu bilangan, yang kemudian digunakan dalam perhitungan permutasi dan kombinasi. Program meminta empat angka sebagai input: dua nilai untuk perhitungan permutasi dan dua nilai untuk perhitungan kombinasi. Hasil dari perhitungan permutasi dan kombinasi akan ditampilkan secara urut.

Source code Unguided 2 :

```
103112400049_MODUL4 > Unguided2 > - 103112400049_Unguided2.go > ...
1 package main
2 //HISYAM NURDIATMOKO 103112400049
3 import "fmt"
4
5 func hitungSkor(jumlahSoal int, waktu *int, soal *int) {
6     if jumlahSoal <= 300 {
7         *soal += 1
8         *waktu += jumlahSoal
9     }
10 }
11
12 func main() {
13     var nama1, nama2 string
14     var soalInput int
15     var totalWaktu, totalSoal int
16     var maxSoal, minWaktu int
17     var pesertaWaktu, pesertaSoal int
18     fmt.Scan(&nama1)
19     if nama1 == "Selesai" {
20         return
21     }
22     nama2 = nama1
23     for i := 0; i < 8; i++ {
24         fmt.Scan(&soalInput)
25         hitungSkor(soalInput, &totalWaktu, &totalSoal)
26     }
27     maxSoal = totalSoal
28     minWaktu = totalWaktu
29     for {
30         fmt.Scan(&nama1)
31         if nama1 == "Selesai" {
32             break
33         }
34         pesertaWaktu, pesertaSoal = 0, 0
35         for i := 0; i < 8; i++ {
36             fmt.Scan(&soalInput)
37             hitungSkor(soalInput, &pesertaWaktu, &pesertaSoal)
38         }
39         if pesertaSoal > maxSoal || (pesertaSoal == maxSoal && pesertaWaktu < minWaktu) {
40             nama2 = nama1
41             maxSoal = pesertaSoal
42             minWaktu = pesertaWaktu
43         }
44     }
45     fmt.Println(nama2, maxSoal, minWaktu)
46 }
```

Output:

```
PS D:\1_Matkul\Alpro2\week4> go run "d:\1_Matkul\Alpro2\week4\103112400049_MODUL4\Unguided2\103112400049_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 D:\1_Matkul\Alpro2\week4> 
```

Deskripsi:

Program Unguided 2 ini dirancang untuk menentukan pemenang dalam sebuah kompetisi berdasarkan jumlah soal yang dijawab dan waktu yang dihabiskan. Program pertama kali meminta input nama peserta dan jumlah soal yang dijawab pada 8 sesi pertama. Total soal yang dijawab dan total waktu yang dihabiskan dihitung untuk peserta pertama. Setelah itu, program akan terus meminta input nama dan jumlah soal yang dijawab untuk peserta lainnya. Program akan menghitung jumlah soal yang dijawab dan waktu yang dihabiskan oleh setiap peserta. Pemenang ditentukan berdasarkan jumlah soal terbanyak yang dijawab, dan jika ada peserta dengan jumlah soal yang sama, pemenang akan ditentukan berdasarkan waktu yang lebih sedikit. Proses ini berlanjut hingga kata "Selesai" diinputkan sebagai tanda berakhirnya input, dan nama pemenang beserta jumlah soal yang dijawab dan waktu yang dibutuhkan akan ditampilkan.

Source code Unguided 3 :

```
103112400049_MODUL4 > Unguided3 > -o 103112400049_Unguided3.go > cetakDeret
1  package main
2  //HISYAM NURDIATMOKO 103112400049
3  import "fmt"
4
5  func cetakDeret(n int) {
6      for n != 1 {
7          fmt.Print(n, " ")
8          if n%2 == 0 {
9              n = n / 2
10         } else {
11             n = 3*n + 1
12         }
13     }
14     fmt.Print(1)
15 }
16
17 func main() {
18     var n int
19     fmt.Scan(&n)
20     cetakDeret(n)
21 }
```

Output:

```
PS D:\1_Matkul\Alpro2\week4> go run "d:\1_Matkul\Alpro2\week4\103112400049_MODUL4\Unguided3\103112400049_Unguided3.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
○ PS D:\1_Matkul\Alpro2\week4> █
```

Deskripsi:

Program Unguided 3 ini digunakan untuk mencetak deret dari suatu bilangan yang dimasukkan oleh pengguna. Deret ini bekerja dengan aturan sederhana: jika bilangan saat ini genap, maka dibagi dua; jika ganjil, maka dikalikan tiga lalu ditambah satu. Proses ini terus berlanjut hingga mencapai angka 1. Program meminta pengguna untuk memasukkan sebuah bilangan, lalu mencetak setiap angka dalam deret hingga mencapai 1.

IV. KESIMPULAN

Pada modul ini, kita mempelajari konsep prosedur dalam pemrograman, yang berfungsi untuk memecah program besar menjadi bagian-bagian lebih kecil yang mudah dikelola. Prosedur mempermudah proses pemrograman dengan meningkatkan modularitas dan reusability kode, serta mempermudah pemeliharaan program. Keuntungan menggunakan prosedur termasuk efisiensi dalam menulis kode dan pengurangan duplikasi kode yang dapat menyebabkan kesalahan.

Pada praktik yang dilakukan, terdapat beberapa contoh program, baik yang dibimbing (guided) maupun tidak dibimbing (unguided). Program yang dibimbing mencakup perhitungan gaji karyawan dan perhitungan rata-rata nilai mahasiswa, sementara yang tidak dibimbing melibatkan perhitungan permutasi dan kombinasi, penentuan pemenang kompetisi, serta pencetakan deret bilangan berdasarkan aturan tertentu. Setiap program ini mengilustrasikan penerapan prosedur dalam memecah masalah-masalah komputasi menjadi langkah-langkah yang lebih sederhana dan terstruktur.

Secara keseluruhan, modul ini memberikan pemahaman yang lebih mendalam mengenai bagaimana prosedur dapat digunakan untuk meningkatkan kualitas dan efisiensi pemrograman.

V. REFERENSI

Modul 4 – Prosedur. Praktikum Algoritma Pemrograman 2