

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 4

MATERI



Oleh:

NAMA: Muhammad Fahruli Ma'ruf

NIM: 103112400057

KELAS: 12-IF-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Prosedur dapat dianggap sebagai bagian dari program yang berisi sekumpulan instruksi baru. Prosedur dirancang untuk mengurangi kerumitan kode program yang kompleks pada program-program besar. Prosedur tidak menghasilkan atau mengembalikan suatu nilai secara langsung pada program. Sebuah subprogram dikatakan sebagai prosedur jika:

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

Kedudukan suatu prosedur mirip dengan instruksi yang sudah ada sebelumnya, seperti instruksi dari paket fmt (misalnya fmt.Scan atau fmt.Print). Oleh karena itu, sebaiknya pilih nama prosedur yang berbentuk kata kerja atau frasa yang merepresentasikan proses yang dilakukan prosedur tersebut. Peran dasar dari suatu prosedur adalah untuk menyediakan titik referensi yang lebih kecil untuk tujuan pengembangan atau bagi programmer untuk memanggil prosedur itu sendiri. Prosedur juga dikenal sebagai fungsi kosong (void function), subrutin, rutin, metode, atau subprogram.

Dengan menggunakan prosedur, seorang programmer dapat membuat program yang melakukan satu hal dengan berbagai cara, misalnya dengan menggunakan parameter dan set data yang berbeda. Prosedur hanya akan dieksekusi apabila dipanggil secara tidak langsung oleh program utama. Artinya, prosedur dipanggil melalui perantara (subprogram lain).

II. GUIDED

```
package main

import "fmt"

//prosedur untuk menghitung total gaji karyawan
func hitungGaji(nama string, gajiPokok float64, jamLembur int) {
    bonusLembur := float64(jamLembur) * 50000
    totalGaji := gajiPokok * bonusLembur

    fmt.Println("\n=== Slip Gaji ===")
    fmt.Println("Nama Karyawan :", nama)
    fmt.Printf("Gaji Pokok : Rp%.2f\n", gajiPokok)
    fmt.Printf("Bonus Lembur : Rp%.2f (%d x Rp50.000)\n", bonusLembur, jamLembur)
    fmt.Printf("Total Gaji : Rp%.2f\n", totalGaji)
}

func main() {
    var nama string
    var gajiPokok float64
    var jamLembur int

    //input dari pengguna
    fmt.Print("masukkan nama karyawan: ")
    fmt.Scanln(&nama)

    fmt.Print("masukkan gaji pokok: ")
    fmt.Scanln(&gajiPokok)

    fmt.Print("masukkan jumlah jam lembur: ")
    fmt.Scanln(&jamLembur)

    //memanggil prosedur dengan data dari pengguna
    hitungGaji(nama, gajiPokok, jamLembur)
}
```

Penjelasan: Program Bahasa Go ini dirancang untuk menghitung total gaji berdasarkan input gaji pokok dan jumlah jam lembur dari pengguna. Setelah pengguna memasukkan kedua nilai tersebut, program akan melakukan perhitungan. Sebagai contoh, jika gaji pokok yang dimasukkan adalah Rp 2.000.000 dan jam lembur 2 jam, maka program akan menghitung biaya lembur dengan rumus: (jumlah jam lembur * Rp 50.000).

```

PS C:\Users\HP\OneDrive\modul4> go run "c:\Users\HP\OneDrive\modul4\contoh1\1.go"
masukkan nama karyawan: Maup
masukkan gaji pokok: 2000000
masukkan jumlah jam lembur: 2

=== Slip Gaji ===
Nama Karyawan : Maup
Gaji Pokok : Rp2000000.00
Bonus Lembur : Rp100000.00 (2 x Rp50.000)
Total Gaji : Rp200000000000.00

```

```

1  package main
2
3  import "fmt"
4
5  func hitungRataRata(nama string, nilai1, nilai2, nilai3 float64) {
6
7      ratarata := (nilai1+ nilai2 + nilai3) / 3
8      status := "Tidak Lulus"
9
10     if ratarata >= 60 {
11         status = "Lulus"
12     }
13
14     fmt.Println("\n=== Hasil Akademik ===")
15     fmt.Println("Nama Mahasiswa :", nama)
16     fmt.Printf("Nilai 1      : %.2f\n", nilai1)
17     fmt.Printf("Nilai 2      : %.2f\n", nilai2)
18     fmt.Printf("Nilai 3      : %.2f\n", nilai3)
19     fmt.Printf("Rata-rata    : %.2f\n", ratarata)
20     fmt.Println("Status      :", status)
21 }
22
23 func main() {
24     var nama string
25     var nilai1, nilai2, nilai3 float64
26
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     hitungRataRata(nama, nilai1, nilai2, nilai3)
40 }

```

Penjelasan: Program Go ini menghitung rata-rata dari tiga angka yang diinput pengguna. Setelah angka dimasukkan, program akan menghitung rata-ratanya dengan rumus $(\text{nilai1} + \text{nilai2} + \text{nilai3}) / 3$. Kemudian, program akan menentukan status: jika rata-rata ≥ 60 , statusnya "lulus", dan jika rata-rata < 60 , statusnya "tidak lulus".

```
PS C:\Users\HP\OneDrive\modul4> go run "c:\Users\HP\OneDrive\modul4\contoh2\2.go"
Masukkan Nama Mahasiswa: Maup
Masukkan Nilai 1: 90
Masukkan Nilai 2: 84
Masukkan Nilai 3: 98

=== Hasil Akademik ===
Nama Mahasiswa : Maup
Nilai 1        : 90.00
Nilai 2        : 84.00
Nilai 3        : 98.00
Rata-rata      : 90.67
Status        : Lulus
```

UNGUIDED

bari 7 / 1.go / permutasi

```
1 package main
2
3 import "fmt"
4
5 func faktorial(n int) int {
6     if n == 0 {
7         return 1
8     }
9     return n * faktorial(n-1)
10 }
11
12 func permutasi(n, r int) int {
13     return faktorial(n) / faktorial(n-r)
14 }
15
16 func kombinasi(n, r int) int {
17     return faktorial(n) / (faktorial(r) * faktorial(n-r))
18 }
19
20 func main() {
21     var a, b, c, d int
22
23     fmt.Println("Program Menghitung Permutasi dan Kombinasi")
24     fmt.Println("=====")
25     fmt.Print("Masukkan empat bilangan (a b c d): ")
26     _, err := fmt.Scan(&a, &b, &c, &d)
27     if err != nil {
28         fmt.Println("Error: Input tidak valid!")
29         return
30     }
31     if a < 0 || b < 0 || c < 0 || d < 0 {
32         fmt.Println("Error: Bilangan tidak boleh negatif!")
33         return
34     }
35
36     if c > a || d > b {
37         fmt.Println("Error: r tidak boleh lebih besar dari n!")
38         return
39     }
40     fmt.Println("\nHasil Perhitungan:")
41     fmt.Println("-----")
42     fmt.Printf("Untuk n=%d dan r=%d:\n", a, c)
43     fmt.Printf("Permutasi: %d\n", permutasi(a, c))
44     fmt.Printf("Kombinasi: %d\n", kombinasi(a, c))
45     fmt.Printf("\nUntuk n=%d dan r=%d:\n", b, d)
46     fmt.Printf("Permutasi: %d\n", permutasi(b, d))
47     fmt.Printf("Kombinasi: %d\n", kombinasi(b, d))
48 }
```

Penjelasan: Program Go ini adalah kalkulator permutasi dan kombinasi. Yang menerima empat bilangan dari pengguna, lalu menghitung permutasi dan kombinasi untuk dua pasang bilangan tersebut, serta menampilkan hasilnya. Program ini juga dilengkapi validasi untuk memastikan input sesuai dengan syarat perhitungan matematika.

```
=====
Masukkan empat bilangan (a b c d): 5 10 3 10

Hasil Perhitungan:
-----
Untuk n=5 dan r=3:
Permutasi: 60
Kombinasi: 10

Untuk n=10 dan r=10:
Permutasi: 3628800
Kombinasi: 1
PS C:\Users\HP\OneDrive\modul4> go run "c:\Users\HP\OneDrive\modul4\soal1\1.go"
Program Menghitung Permutasi dan Kombinasi
=====
Masukkan empat bilangan (a b c d): 8 0 2 0

Hasil Perhitungan:
-----
Untuk n=8 dan r=2:
Permutasi: 56
Kombinasi: 28

Untuk n=0 dan r=0:
Permutasi: 1
Kombinasi: 1
```

```

1 package main
2
3 import "fmt"
4
5 const maxTime = 301
6
7 func hitungSkor(waktu [8]int, soal *int, skor *int) {
8     *soal = 0
9     *skor = 0
10    for _, waktuSoal := range waktu {
11        if waktuSoal < maxTime {
12            *soal++
13            *skor += waktuSoal
14        }
15    }
16 }
17
18 func main() {
19     var pemenang string
20     var soalPemenang, skorPemenang int
21     soalPemenang = -1
22     skorPemenang = maxTime * 8
23     pesertaCount := 0
24
25     for {
26         var nama string
27         fmt.Print("Nama peserta (Selesai untuk berhenti): ")
28         fmt.Scan(&nama)
29
30         if nama == "Selesai" {
31             if pesertaCount == 0 {
32                 fmt.Println("Belum ada peserta!")
33                 continue
34             }
35             break
36         }
37
38         var waktu [8]int
39         fmt.Print("Waktu (8 soal): ")
40         for i := 0; i < 8; i++ {
41             fmt.Scan(&waktu[i])
42         }
43
44         var soal, skor int
45         hitungSkor(waktu, &soal, &skor)
46         pesertaCount++
47
48         if soalPemenang == -1 || soal > soalPemenang || (soal == soalPemenang && skor < skorPemenang) {
49             pemenang = nama
50             soalPemenang = soal
51             skorPemenang = skor
52         }
53     }
54
55     fmt.Printf("\nPemenang: %s\n", pemenang)
56     fmt.Printf("Soal selesai: %d\n", soalPemenang)
57     fmt.Printf("Total waktu: %d menit\n", skorPemenang)
58     fmt.Printf("Jumlah peserta: %d\n", pesertaCount)
59 }

```

Penjelasan: Program Go ini adalah sistem penilaian kompetisi yang menghitung skor untuk setiap peserta berdasarkan waktu pengerjaan 8 soal. Program akan meminta nama peserta dan waktu pengerjaan soal, lalu menentukan pemenang berdasarkan jumlah soal yang benar dan skor (lebih cepat lebih baik), kemudian menampilkan detail pemenang dan jumlah peserta.


```
PS C:\Users\HP\OneDrive\modul4> go run "c:\Users\HP\OneDrive\modul4\soal2\2.go"
Nama peserta (Selesai untuk berhenti): Astuti
PS C:\Users\HP\OneDrive\modul4> go run "c:\Users\HP\OneDrive\modul4\soal2\2.go"
Nama peserta (Selesai untuk berhenti): Astuti
Waktu (8 soal): 20 50 301 301 61 71 75 10
Nama peserta (Selesai untuk berhenti): Bertha
Waktu (8 soal): 25 47 301 26 50 60 65 21
Nama peserta (Selesai untuk berhenti): Selesai

Pemenang: Bertha
Soal selesai: 7
Total waktu: 294 menit
Jumlah peserta: 2
```

```

1 package main
2
3 import "fmt"
4
5 func cetakDeret(n int) {
6     if n <= 0 {
7         fmt.Println("Error: Bilangan harus positif!")
8         return
9     }
10    langkah := 1
11    fmt.Printf("Deret Collatz untuk n = %d:\n", n)
12    fmt.Print(n)
13    for n != 1 {
14        if n%2 == 0 {
15            n = n / 2
16        } else {
17            n = 3*n + 1
18        }
19        fmt.Print(" → ", n)
20        langkah++
21    }
22    fmt.Println()
23    fmt.Printf("Jumlah langkah: %d\n", langkah)
24 }
25
26 func main() {
27     fmt.Println("Program Deret Collatz (3n+1)")
28     fmt.Println("=====")
29
30     var masukan int
31     fmt.Print("Masukkan bilangan positif: ")
32     _, err := fmt.Scan(&masukan)
33
34     if err != nil {
35         fmt.Println("Error: Input tidak valid!")
36         return
37     }
38
39     cetakDeret(masukan)
40 }

```

Penjelasan: Program Go ini adalah implementasi dari Deret Collatz (juga dikenal sebagai masalah $3n+1$). Program akan meminta input bilangan bulat positif dari pengguna, kemudian menghasilkan dan mencetak setiap langkah dalam deret Collatz hingga mencapai angka 1, serta menampilkan total jumlah langkah yang dibutuhkan.

```

PS C:\Users\HP\OneDrive\modul4> go run "c:\Users\HP\OneDrive\modul4\soal3\3.go"
Program Deret Collatz (3n+1)
=====
Masukkan bilangan positif: 22
Deret Collatz untuk n = 22:
22 → 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1
Jumlah langkah: 16

```

III. KESIMPULAN

Secara ringkas, prosedur adalah blok instruksi program yang dirancang untuk menyederhanakan kode yang kompleks pada program besar. Fungsi utamanya adalah menghasilkan efek langsung pada program ketika dipanggil dari program utama, serta menyediakan titik referensi terpadu untuk menjalankan tugas-tugas spesifik yang dapat diakses oleh pengembang.

IV. REFERENSI