

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4
PROSEDUR**



Oleh:

SAVILA NUR FADILLA

103112400031

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

4.1 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
2. Tidak terdapat kata kunci return dalam badan subprogram

4.2 Deklarasi Procedure

```
func <nama procedure> <(params)> {  
    /* deklarasi variabel lokal dari procedure */  
    ...  
    /* badan algoritma procedure */  
    ...  
}
```

4.3 Cara Pemanggilan Procedure

```
func main() {  
    var x int  
    x = 5  
    cetakNFibo(x) {cara pemanggilan #1}  
    cetakNFibo(100) {cara pemanggilan #2}  
}
```

II. GUIDED

1.) Source Code

```
package main

import "fmt"

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 jam 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

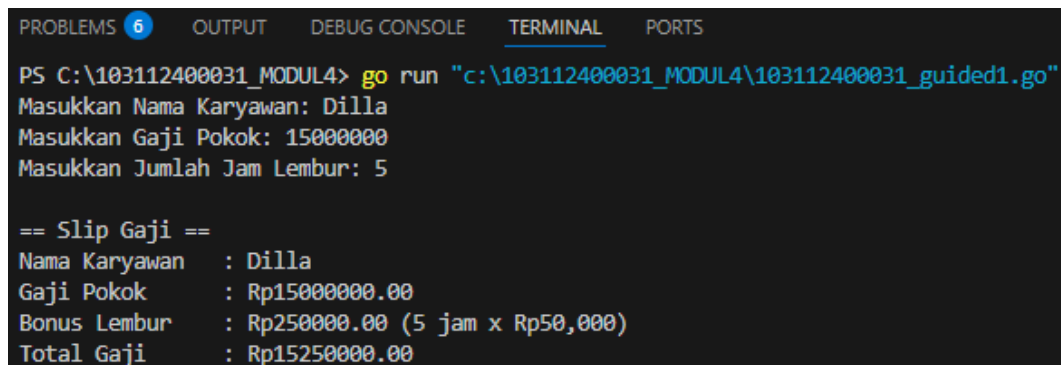
    fmt.Print("Masukkan Nama Karyawan: ")
    fmt.Scanln(&nama)

    fmt.Print("Masukkan Gaji Pokok: ")
    fmt.Scanln(&gajiPokok)

    fmt.Print("Masukkan Jumlah Jam Lembur: ")
    fmt.Scanln(&jamLembur)

    hitungGaji(nama, gajiPokok, jamLembur)
}
```

Output :



```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL4> go run "c:\103112400031_MODUL4\103112400031_guided1.go"
Masukkan Nama Karyawan: Dilla
Masukkan Gaji Pokok: 15000000
Masukkan Jumlah Jam Lembur: 5

== Slip Gaji ==
Nama Karyawan   : Dilla
Gaji Pokok      : Rp15000000.00
Bonus Lembur    : Rp250000.00 (5 jam x Rp50,000)
Total Gaji      : Rp15250000.00
```

Penjelasan : Program ini bertujuan untuk menghitung total gaji karyawan berdasarkan gaji pokok dan jumlah jam lembur. Program ini akan meminta input berupa nama karyawan, gaji pokok, dan jumlah jam lembur. Kemudian program akan menghitung bonus lemburnya dengan cara mengalikan jumlah jam lembur dengan Rp50.000 lalu setelah itu bonus lembur akan ditambah dengan gaji pokok yang sudah diinput dan program akan menghitungnya sebagai total gaji. Output berupa nama karyawan, gaji pokok, bonus lembur, dan total gaji.

2.) Source Code

```
package main

import "fmt"

func hitungRataRata(nama string, nilai1, nilai2, nilai3 float64) {
    rataRata := (nilai1 + nilai2 + nilai3 / 3)
    status := "Tidak Lulus"
    if rataRata >= 60 {
        status = "Lulus"
    }

    fmt.Println("\n=== Hasil Akademik ===")
    fmt.Println("Nama Mahasiswa   :", nama)
    fmt.Printf("Nilai 1           : %.2f\n", nilai1)
    fmt.Printf("Nilai 2           : %.2f\n", nilai2)
    fmt.Printf("Nilai 3           : %.2f\n", nilai3)
    fmt.Printf("Rata-rata        : %.2f\n", rataRata)
    fmt.Println("Status            :", status)
}

func main() {
    var nama string
    var nilai1, nilai2, nilai3 float64

    fmt.Print("Masukkan Nama Mahasiswa: ")
    fmt.Scanln(&nama)

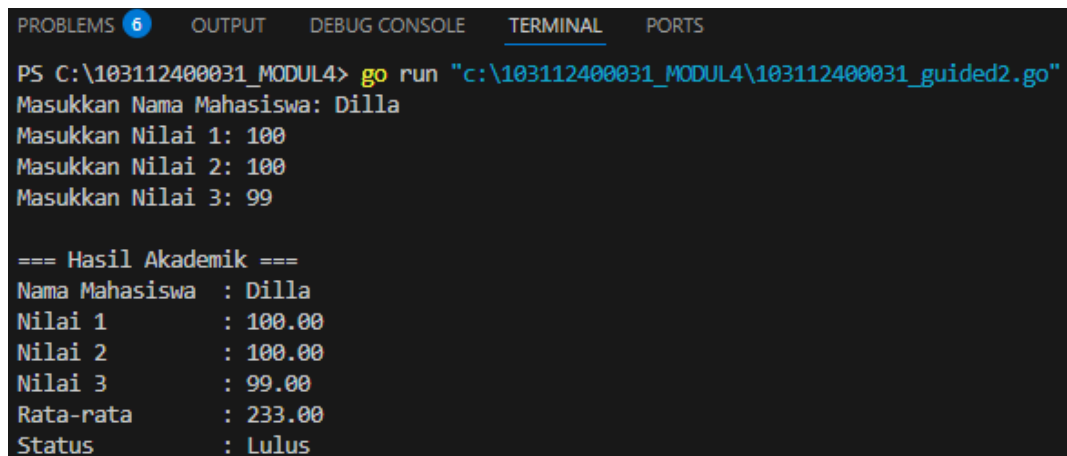
    fmt.Print("Masukkan Nilai 1: ")
    fmt.Scanln(&nilai1)

    fmt.Print("Masukkan Nilai 2: ")
    fmt.Scanln(&nilai2)

    fmt.Print("Masukkan Nilai 3: ")
    fmt.Scanln(&nilai3)

    hitungRataRata(nama, nilai1, nilai2, nilai3)
}
```

Output :

A screenshot of a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The prompt is 'PS C:\103112400031_MODUL4>'. The user enters 'go run "c:\103112400031_MODUL4\103112400031_guided2.go"'. The program prompts for 'Masukkan Nama Mahasiswa: Dilla', 'Masukkan Nilai 1: 100', 'Masukkan Nilai 2: 100', and 'Masukkan Nilai 3: 99'. It then displays '=== Hasil Akademik ===' followed by a list of values: 'Nama Mahasiswa : Dilla', 'Nilai 1 : 100.00', 'Nilai 2 : 100.00', 'Nilai 3 : 99.00', 'Rata-rata : 233.00', and 'Status : Lulus'.

```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL4> go run "c:\103112400031_MODUL4\103112400031_guided2.go"
Masukkan Nama Mahasiswa: Dilla
Masukkan Nilai 1: 100
Masukkan Nilai 2: 100
Masukkan Nilai 3: 99

=== Hasil Akademik ===
Nama Mahasiswa : Dilla
Nilai 1 : 100.00
Nilai 2 : 100.00
Nilai 3 : 99.00
Rata-rata : 233.00
Status : Lulus
```

Penjelasan : Program ini bertujuan untuk menghitung rata rata nilai mahasiswa dan menyatakan statusnya (lulus atau tidak). Program ini akan meminta input berupa nama mahasiswa dan tiga nilai (nilai1, nilai2, nilai3). Kemudian program akan menghitung rata rata nilai mahasiswa tersebut dengan menjumlahkan semua nilai lalu hasil penjumlahan ketiga nilai tersebut dibagi dengan tiga. Jika rata rata lebih dari atau sama dengan 60, maka statusnya adalah lulus, jika kurang dari 60 maka statusnya tidak lulus. Output berupa nama mahasiswa, nilai1, nilai2, nilai3, rata rata, dan status.

III. UNGUIDED

1.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import "fmt"

func faktorial(n int) int {
    hasil := 1
    for i := 1; i <= n; i++ {
        hasil *= i
    }
    return hasil
}

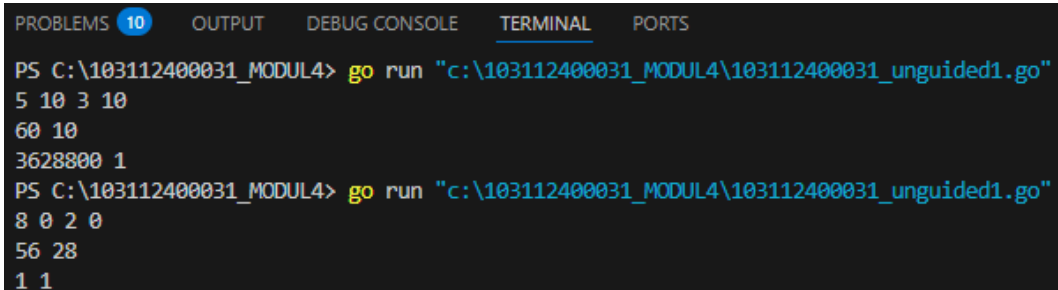
func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}

func main() {
    var a, b, c, d int
    fmt.Scan(&a, &b, &c, &d)

    fmt.Println(permutasi(a, c), kombinasi(a, c))
    fmt.Println(permutasi(b, d), kombinasi(b, d))
}
```

Output :



```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL4> go run "c:\103112400031_MODUL4\103112400031_unguided1.go"
5 10 3 10
60 10
3628800 1
PS C:\103112400031_MODUL4> go run "c:\103112400031_MODUL4\103112400031_unguided1.go"
8 0 2 0
56 28
1 1
```

Penjelasan : Program ini bertujuan untuk menghitung hasil permutasi dan kombinasi. Program meminta kita menginput 4 bilangan asli yang dipisahkan spasi, dengan syarat $a \geq c$ dan $b \geq d$. Kemudian program akan menghitung hasil dari permutasi dan kombinasinya dengan rumus permutasi dan kombinasi. Ada fungsi faktorial(n) untuk menghitung faktorial bilangan n, fungsi permutasi(n, r) menghitung cara menyusun r elemen dari n, sedangkan fungsi kombinasi(n, r) menghitung jumlah cara memilih r elemen dari n tanpa memperhatikan urutan. Output terdiri dari 2 baris. Baris pertama berupa permutasi dan kombinasi dari a terhadap c, sedangkan baris kedua berupa permutasi dan kombinasi dari b terhadap d.

2.) Source code

```
// Savila Nur Fadilla
// 103112400031

package main

import "fmt"

func hitungSkor(waktuPengerjaan []int) (int, int) {
    soalSelesai, totalDurasi := 0, 0

    for _, durasi := range waktuPengerjaan {
        if durasi <= 300 {
            soalSelesai++
            totalDurasi += durasi
        }
    }
    return soalSelesai, totalDurasi
}

func main() {
    var namaPemenang string
    var soalTerbanyak, waktuTercepat int = 0, 1000000

    for {
        var namaPeserta string
        var waktuPengerjaan [8]int

        _, err := fmt.Scan(&namaPeserta)
        if err != nil || namaPeserta == "Selesai" {
            break
        }

        for i := 0; i < 8; i++ {
            _, err = fmt.Scan(&waktuPengerjaan[i])
            if err != nil {
                break
            }
        }

        soalSelesai, totalDurasi := hitungSkor(waktuPengerjaan[:])

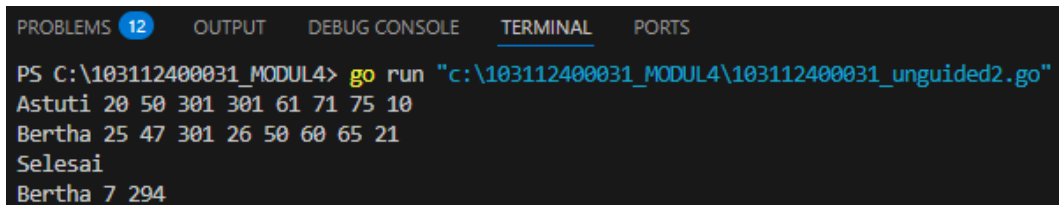
        if soalSelesai > soalTerbanyak || (soalSelesai == soalTerbanyak
        && totalDurasi < waktuTercepat) {
```

```

        namaPemenang = namaPeserta
        soalTerbanyak = soalSelesai
        waktuTercepat = totalDurasi
    }
}
fmt.Println(namaPemenang, soalTerbanyak, waktuTercepat)
}

```

Output :



```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL4> go run "c:\103112400031_MODUL4\103112400031_unguided2.go"
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294

```

Penjelasan : Program ini bertujuan untuk menentukan pemenang berdasarkan jumlah soal dan total waktu pengerjaan paling cepat. Program ini akan meminta input nama peserta dan 8 integer berupa waktu yang dibutuhkan untuk mengerjakan tiap soal (dalam detik). Kemudian akan dihitung soal benar dimana soal dihitung benar jika waktu pengerjaan tiap soalnya kurang dari atau sama dengan 301, dan akan dihitung juga total durasi. Output berupa nama pemenang, jumlah soal yang diselesaikan, dan nilai yang diperoleh. Nilai adalah total waktu yang dibutuhkan untuk menyelesaikan soal yang berhasil diselesaikan.

3.) Source code

```
// Savila Nur Fadilla
// 103112400031

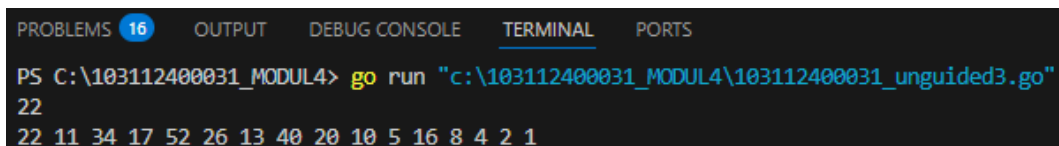
package main

import "fmt"

func cetakDeret(n int) {
    for n != 1 {
        fmt.Print(n, " ")
        if n % 2 == 0 {
            n /= 2
        } else {
            n = (3 * n) + 1
        }
    }
    fmt.Print(n)
}

func main() {
    var n int
    fmt.Scan(&n)
    cetakDeret(n)
}
```

Output :



```
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400031_MODUL4> go run "c:\103112400031_MODUL4\103112400031_unguided3.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Penjelasan : Program ini bertujuan untuk mencetak suatu deret dengan suku awal bilangan n dan suku terakhir 1. Program ini akan meminta input berupa suatu bilangan bulat positif yang kurang dari 1000000 yang nantinya akan menjadi suku pertama. Kemudian program akan menghitung suku setelahnya. Jika bilangan n saat itu genap, maka suku berikutnya adalah $1/2 n$, 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. Output berupa deret yang terdiri dari satu baris saja dan dipisahkan oleh sebuah spasi.

IV. KESIMPULAN

Berdasarkan laporan ini, dapat disimpulkan jika soal soal tersebut menggunakan struktur prosedur (fungsi tanpa return). Setiap program memiliki struktur modular yang terpisah dari fungsi utama. Struktur prosedur digunakan agar program lebih terstruktur, mudah dibaca, dan mudah dipahami. Setiap prosedur juga menangani tugasnya secara spesifik.

- a.) Program slip gaji menggunakan prosedur untuk total gaji karyawan berdasarkan gaji pokok dan jumlah jam lembur
- b.) Program hasil akademik menggunakan prosedur untuk menghitung rata rata nilai mahasiswa dan menyatakan statusnya (lulus atau tidak)
- c.) Program permutasi dan kombinasi menggunakan prosedur untuk menghitung factorial, permutasi, dan kombinasi
- d.) Program pemenang menggunakan prosedur untuk menghitung skor
- e.) Program cetak deret menggunakan prosedur untuk mencetak suatu deret dengan suku awal bilangan n dan suku terakhir 1

REFERENSI

Telkom University. (2025). *Modul Praktikum Algoritma dan Pemrograman 2*.