

# **ALGORITMA PEMOGRAMAN 2**

## **MODUL 4**

### **"PROSEDUR"**



**Oleh:**

**NAMA: Davi Ilyas Renaldo**

**NIM: 103112400062**

**KELAS: 12-IF-01**

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## I.DASAR TEORI

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, ...

Prosedur banyak digunakan pada program yang terstruktur karena :

1. Merupakan penerapan konsep program Modular, yaitu memecah-mecah program yang rumit menjadi program-program bagian yang lebih sederhana dalam bentuk sebagai prosedur-prosedur.
2. Untuk hal-hal yang sering dilakukan berulang-ulang, cukup dituliskan sekali saja dalam prosedur dan dapat dipanggil atau dipergunakan sewaktu-waktu bila diperlukan.
3. Membuat kode program lebih mudah dibaca.
4. Dapat digunakan untuk menyembunyikan detail program

Prosedur memiliki ciri-ciri sebagai berikut ini :

1. Hanya memiliki satu fungsi tujuan (logical inherent). Sebuah prosedur sebaiknya hanya memiliki satu fungsi tujuan dan tidak bercampur dengan tujuan lain. Hal ini untuk membuat prosedur lebih focus.
2. Berukuran kecil (small size). Yang dimaksud ukuran disini adalah panjang algoritma atau panjang kode program pada suatu prosedur. Ukuran kecil akan mudah dibaca maupun diperbaiki.
3. Tidak tergantung pada prosedur lain (independent). Sebuah prosedur harusnya bersifat mandiri, artinya sebuah prosedur dapat berjalan dan diuji tanpa menunggu bagian lainnya selesai. Selain itu variable yang digunakan dalam prosedur tidak mempengaruhi variabel yang digunakan pada bagian lain di keseluruhan program.

Manfaat pembuatan prosedur :

1. Modularisasi : Suatu program yang besar dan kompleks dapat dibagi ke dalam beberapa prosedur sehingga setiap prosedur merupakan bagian yang mudah dikerjakan. Dengan demikian, program besar tersebut menjadi mungkin diselesaikan.
2. Simplifikasi : Dalam suatu program, sering diperlukan suatu tugas yang harus dikerjakan berulang-ulang dengan nilai-nilai variabel yang berbeda. Agar tidak merepotkan maka tugas ini cukup ditulis sekali saja dalam bentuk prosedur yang kemudian dipanggil berulang-ulang sesuai kebutuhan.

## II. GUIDED

1.

```
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.Printf("Nama Karyawan : %s\n", 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 Jam Lembur : ")
    fmt.Scanln(&jamLembur)

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

## OUTPUT:

```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\GUIDED 4\1.go"
Masukkan Nama Karyawan : davi
Masukkan Gaji Pokok : 5500000
Masukkan Jam Lembur : 8

=== Slip Gaji ===
Nama Karyawan : davi
Gaji Pokok : Rp5500000.00
Bonus Lembur : Rp400000.00 (8 jam x Rp50,000)
Total Gaji : Rp5900000.00
PS C:\Users\ACER\OneDrive\alpro2\modul1>
```

## DESKRIPSI:

Program ini adalah program untuk menghitung total gaji yang diterima oleh seorang karyawan setelah pengguna memasukkan angka nominal gaji dan jumlah lemburnya.

2.

```
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:

```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\GUIDED 4\2.go"
Masukkan Nama Mahasiswa: davi
Masukkan Nilai 1: 95
Masukkan Nilai 2: 92
Masukkan Nilai 3: 90

=== Hasil Akademik ===
Nama Mahasiswa : davi
Nilai 1       : 95.00
Nilai 2       : 92.00
Nilai 3       : 90.00
Rata-rata     : 92.33
Status        : Lulus
PS C:\Users\ACER\OneDrive\alpro2\modul1>
```

#### DESKRIPSI:

Program ini adalah program untuk menghitung keseluruhan rata-rata nilai mahasiswa serta lulus atau tidak berdasarkan nilai yang diinput oleh pengguna.

### III. UNGUIDED

1.

```
// M.DAVI ILYAS RENALDO
// 103112400062
package main

import (
    "fmt"
)

func Faktorial(n int) int {
    if n == 0 {
        return 1
    }
    result := 1
    for i := 2; i <= n; i++ {
        result *= i
    }
    return result
}

func Permutasi(n, r int) int {
    if r > n {
        return 0
    }
    return Faktorial(n) / Faktorial(n-r)
}

func Kombinasi(n, r int) int {
    if r > n {
        return 0
    }
    return Faktorial(n) / (Faktorial(r) * Faktorial(n-r))
}

func CetakHasil(n, r int) {
    fmt.Println(Permutasi(n, r), Kombinasi(n, r))
}

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

    if a >= c && b >= d {
        CetakHasil(a, c)
        CetakHasil(b, d)
    } else {
        fmt.Println("Input tidak valid: pastikan a >= c dan b >= d.")
    }
}
```

#### OUTPUT:

```
PS C:\Users\ACER\OneDrive\alproz\modul1> go run "c:\Users\ACER\OneDrive\alproz\modul1\UNGUIDED 4\1.go"
5 10 3 10
60 10
3628800 1
PS C:\Users\ACER\OneDrive\alproz\modul1> go run "c:\Users\ACER\OneDrive\alproz\modul1\UNGUIDED 4\1.go"
8 0 2 0
56 28
1 1
PS C:\Users\ACER\OneDrive\alproz\modul1> █
```

#### DESKRIPSI:

Program ini adalah program yang menghitung permutasi dan kombinasi berdasarkan dua pasang angka yang diinputkan oleh pengguna. Maka pengguna akan diminta untuk memasukkan dua pasang angka dengan syarat  $a \geq c$  dan  $b \geq d$ . Kemudian program akan bekerja menghitung hasil dari permutasi dan kombinasi dari dua pasang angka yang diinputkan oleh pengguna apakah bilangan tersebut valid atau tidak berdasarkan syarat dan rumus pada program ini.

2.

```
// M.DAVI ILYAS RENALDO
// 103112400062
package main
import "fmt"

func main() {
    var pemenang string
    var totalSoal, skorPemenang int
    totalSoal = 0
    skorPemenang = maxTime * 8

    for {
        var nama string
        var waktu [8]int
        fmt.Scan(&nama)

        if nama == "Selesai" {
            break
        }

        for i := 0; i < 8; i++ {
            fmt.Scan(&waktu[i])
        }

        var soal, skor int
        hitungSkor(waktu, &soal, &skor)

        if soal > totalSoal || (soal == totalSoal && skor < skorPemenang) {
            pemenang = nama
            totalSoal = soal
            skorPemenang = skor
        }
    }

    fmt.Printf("%s %d %d\n", pemenang, totalSoal, skorPemenang)
}
const maxTime = 301

func hitungSkor(waktu [8]int, soal *int, skor *int) {
    *soal = 0
    *skor = 0
    for _, waktuSoal := range waktu {
        if waktuSoal < maxTime {
            *soal++
            *skor += waktuSoal
        }
    }
}
```

## OUTPUT:

```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 4\2.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:\Users\ACER\OneDrive\alpro2\modul1>
```



**DESKRIPSI:**

Program ini adalah suatu program untuk menentukan siapa pemenang dalam suatu kompetisi berdasarkan jumlah soal yang berhasil diselesaikan dan total waktu pengerjaan soal. Maka pengguna diminta untuk memasukkan nama dan durasi waktu pengerjaan setiap soal, kemudian program ini akan berjalan dan menghasilkan siapa yang menjadi pemenang berdasarkan total soal dan durasi pengerjaanya.

3.

```
// M.DAVI ILYAS RENALDO
// 103112400062
package main

import "fmt"

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

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

#### OUTPUT:

```
PS C:\Users\ACER\OneDrive\alpro2\modul1> go run "c:\Users\ACER\OneDrive\alpro2\modul1\UNGUIDED 4\3.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\ACER\OneDrive\alpro2\modul1> █
```

#### DESKRIPSI:

Program ini adalah program yang bertujuan untuk menghasilkan dan mencetak deret bilangan berdasarkan aturan tertentu. Deret bilangan ini dimulai dengan sebuah bilangan bulat positif yang diinputkan oleh pengguna.

## **IV.KESIMPULAN**

Berdasarkan laporan praktikum saya diatas dapat disimpulkan bahwa:

- dengan membagi kode ke dalam prosedur atau fungsi, kode menjadi lebih terorganisir, mudah dibaca, dan dapat digunakan kembali.
- kesalahan lebih mudah ditemukan karena setiap prosedur/fungsi hanya memiliki satu tanggung jawab spesifik.
- struktur program menjadi lebih rapi.

## **V.REFERENSI**

- Modul 4 prosedur
- <https://wisnuagung.github.io/mkdocs-fix/menu/php/prosedurFungsi/>