

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 4
PROSEDUR**



Oleh:

Rifki Nur Wibowo

21102308

IF 11 07

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

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 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`, ...

II. GUIDED

```
package main

import "fmt"

func main() {
    var bilangan int
    var pesan string
    fmt.Scan(&bilangan, &pesan)
    cetakPesan(pesan, bilangan)
}

func cetakPesan(M string, flag int) {
    var jenis string = ""
    if flag == 0 {
        jenis = "error"
    } else if flag == 1 {
        jenis = "warning"
    } else if flag == 2 {
        jenis = "informasi"
    }
    fmt.Println(M, jenis)
}
```

▼ Run

📄 Ask AI

44s • 2 minutes ago ✓ ▲

```
hello word
error
```

III. UNGUIDED

1. Jonas adalah seorang mahasiswa yang sedang mempelajari kombinasi dan permutasi. Dia ingin membuat program untuk menghitung permutasi dan kombinasi dari dua bilangan

Program

```
package main

import (
    "fmt"
)

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

func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

func combination(n, r int) int {
    return factorial(n) / (factorial(r) * factorial(n-r))
}

func main() {
    var a, b, c, d int
    fmt.Print("Masukkan empat bilangan bulat positif: ")
    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)
    if a >= c && b >= d {
        // Menghitung permutasi dan kombinasi
        pAC := permutation(a, c)
        cAC := combination(a, c)
        pBD := permutation(b, d)
        cBD := combination(b, d)
        fmt.Printf("Permutasi P(%d, %d): %d\n", a, c, pAC)
```

```

        fmt.Printf("Kombinasi C(%d, %d): %d\n", a, c, cAC)
        fmt.Printf("Permutasi P(%d, %d): %d\n", b, d, pBD)
        fmt.Printf("Kombinasi C(%d, %d): %d\n", b, d, cBD)
    } else {
        fmt.Println("Syarat tidak terpenuhi: a harus lebih besar atau sama
dengan c dan b harus lebih besar atau sama dengan d.")
    }
}

```

Output

```

Run
Masukkan empat bilangan bulat positif: 5 10 3 10
Permutasi P(5, 3): 60
Kombinasi C(5, 3): 10
Permutasi P(10, 10): 3628800
Kombinasi C(10, 10): 1

```

2. Buat program gema yang mencari pemenang dari daftar peserta yang diberikan. Program harus dibuat modular, yaitu dengan membuat prosedur `hitungSkor` yang mengembalikan total soal dan total skor yang dikerjakan oleh seorang peserta, melalui parameter formal. Pembacaan nama peserta dilakukan di program utama, sedangkan waktu pengerjaan dibaca di dalam prosedur.

Program

```

package main

import "fmt"

type Peserta struct {
    Nama      string
    WaktuSoal []int
    Skor      int
    TotalWaktu int
}

func hitungSkor(soal []int) (int, int) {
    var skor int
    var totalWaktu int
    for i, waktu := range soal {
        if waktu != 301 {

```

```

        skor++
        totalWaktu += waktu
    } else {
        soal[i] = 0
    }
}
return skor, totalWaktu
}

func main() {
    var peserta []Peserta
    peserta = append(peserta, Peserta{Nama: "Astuti", WaktuSoal: []int{20,
50, 301, 301, 61, 71, 75, 10}})
    peserta = append(peserta, Peserta{Nama: "Bertha", WaktuSoal: []int{25,
47, 301, 26, 50, 60, 65, 21}})

    var pemenang Peserta
    pemenang.Skor = 0
    pemenang.TotalWaktu = 301 * 8

    for i := range peserta {
        peserta[i].Skor, peserta[i].TotalWaktu =
hitungSkor(peserta[i].WaktuSoal)

        if peserta[i].Skor > pemenang.Skor {
            pemenang = peserta[i]
        } else if peserta[i].Skor == pemenang.Skor &&
peserta[i].TotalWaktu < pemenang.TotalWaktu {
            pemenang = peserta[i]
        }
    }

    fmt.Printf("%s %d %d\n", pemenang.Nama, pemenang.Skor,
pemenang.TotalWaktu)
}

```

Output

```

Run Ask AI 141ms • 9 minutes ago ✓
Bertha 7 294

```

3. Buat program skiena yang akan mencetak setiap suku dari deret yang dijelaskan di atas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal.

prosedure cetakDeret (inn integer)

program

```
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.Println(1)
}

func main() {
    var n int
    fmt.Print("Masukkan suku awal: ")
    fmt.Scanln(&n)
    cetakDeret(n)
}
```

Output

```
Run Ask AI 2s • 4 minutes ago ✓
Masukkan suku awal: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```