

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 4
MATERI**



Oleh:

DAFFA TSAQIFNA FAUZTSANY

103112400032

S1 IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. DASAR TEORI

Prosedur

1. Pengertian Prosedur

Prosedur adalah bagian dari program yang berisi perintah untuk melakukan tugas tertentu, tidak mengembalikan nilai (tidak ada return). Prosedur membantu menyederhanakan program yang kompleks.

2. Ciri Fungsi:

- Tidak memiliki nilai balik.
- Menggunakan func seperti fungsi, tapi tidak pakai return.
- Namanya biasanya berupa kata kerja, seperti cetak, hitung, proses.

Contoh Prosedur di Go:

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

Pemanggilan Prosedur:

```
cetakPesan("Data tidak valid", 0)
```

3. Parameter:

- Formal: ditulis saat deklarasi prosedur.
- Aktual: ditulis saat pemanggilan prosedur.
- Pass by value: hanya menyalin nilai.
- Pass by reference: mengirim alamat memori, ditulis dengan * dan &.

Contoh Pass by Reference:

```
func ubahNilai(x *int) {  
    *x = *x + 10  
}  
  
func main() {  
    var a = 5
```

```
ubahNilai(&a)  
fmt.Println(a) // Output: 15
```

```
}
```

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

1. GUIDED 1

Source Code:

```
package main

import "fmt"

func hitungGaji(nama string, gajiPokok float64, jamLembur int) {
    bonus := float64(jamLembur) * 50000
    totalGaji := gajiPokok + bonus

    fmt.Println("\n=== Slip Gaji ===")
    fmt.Println("Nama Karyawan:", nama)
    fmt.Printf("Gaji Pokok: Rp%.2f\n", gajiPokok)
    fmt.Printf("bonus: Rp%.2f(%d jam x Rp50000)\n", bonus, jamLembur)
    fmt.Printf("Total Gaji: Rp%.2f\n", totalGaji)
}

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

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

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

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

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

Output:



```
PS D:\test bs> go run 'd:\test bs\lab shift\smst 2\103112400032_MODUL 4\guided 4\guided-4-1.go'
Nama Karyawan: udin
Gaji Pokok: Rp500000
Jam Lembur: 3

=== Slip Gaji ===
Nama Karyawan: udin
Gaji Pokok: Rp500000.00
bonus: Rp150000.00(3 jam x Rp50000)
Total Gaji: Rp650000.00
```

Deskripsi Program:

Program ini digunakan untuk menghitung dan menampilkan **slip gaji karyawan** berdasarkan nama, gaji pokok, dan jumlah jam lembur. Bonus lembur dihitung sebesar **Rp50.000 per**

jam, kemudian dijumlahkan dengan gaji pokok untuk mendapatkan total gaji. Program menerima input dari pengguna dan mencetak hasil dalam format slip gaji yang rapi.

2. GUIDED 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("Nama mahasiswa: ")
    fmt.Scanln(&nama)

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

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

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

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

Output:

```
P S D:\test bs> go run 'd:\test bs\lab shit\smsr 2\103112400032_MODUL 4\guided 4\guide
d-4-2.go'
Nama mahasiswa: daffa
Nilai 1: 90
Nilai 2: 80
Nilai 3: 85

=== hasil akademik ===
Nama mahasiswa: daffa
Nilai 1: 90.00
Nilai 2: 80.00
Nilai 3: 85.00
Rata-rata: 85.00
Status: Lulus
```

Deskripsi Program:

Program ini digunakan untuk menghitung **rata-rata nilai akademik mahasiswa** berdasarkan tiga nilai yang diinputkan. Setelah menghitung rata-rata, program menentukan status kelulusan: jika rata-rata ≥ 60 maka mahasiswa dinyatakan "**Lulus**", jika tidak maka "**Tidak Lulus**". Hasil akhir ditampilkan dalam format laporan akademik yang mencantumkan nama mahasiswa, nilai-nilai, rata-rata, dan status.

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

1. UNGUIDED 1

Source Code:

```
package main

import "fmt"

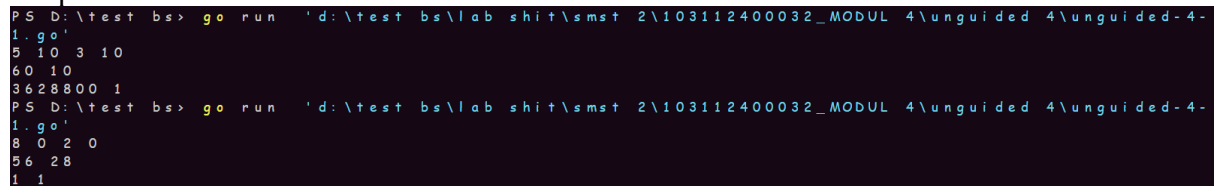
func faktorial(x int) int {
    var out int
    out = 1
    for i := 1; i <= x; i++ {
        out *= i
    }
    return out
}

func permutasi(x, y int) int {
    return faktorial(x) / faktorial(x-y)
}

func kombinasi(x, y int) int {
    return faktorial(x) / (faktorial(y) * faktorial(x-y))
}

func main() {
    var a, b, c, d int
    fmt.Scan(&a, &b, &c, &d)
    fmt.Println(permutasi(a, c), kombinasi(a, c))
    fmt.Print(permutasi(b, d), kombinasi(b, d))
}
```

Output:



```
P S D:\test bs> go run 'd:\test bs\lab shit\smsr 2\103112400032_MODUL 4\unguided 4\unguided-4-1.go'
5 10 3 10
3628800 1
P S D:\test bs> go run 'd:\test bs\lab shit\smsr 2\103112400032_MODUL 4\unguided 4\unguided-4-1.go'
8 0 2 0
56 28
1 1
```

Deskripsi Program:

Program ini digunakan untuk menghitung nilai **permutasi** dan **kombinasi** dari dua pasang bilangan yang dimasukkan oleh pengguna. Permutasi dihitung dengan rumus $P(n, r) = n! / (n - r)!$ dan kombinasi dengan rumus $C(n, r) = n! / (r!(n - r)!)$. Program menggunakan fungsi faktorial sebagai dasar perhitungan, lalu mencetak hasil permutasi dan kombinasi dari masing-masing pasangan.

2. UNGUIDED 2

Source Code:

```

package main

import (
    "fmt"
    "sort"
)

func sortandcount(x [8]int) (int, int) {
    var soal, waktu int
    numSlice := x[:]
    sort.Ints(numSlice)
    for _, num := range numSlice {
        if waktu+num > 301 {
            break
        }
        waktu += num
        soal++
    }
    return waktu, soal
}

func main() {
    var nama string
    var Wname []string
    var pengerjaan [8]int
    var Wtime, Wsoal int
    fmt.Scan(&nama)
    for i := 0; i < 8; i++ {
        fmt.Scan(&pengerjaan[i])
    }
    Wtime, Wsoal = sortandcount(pengerjaan)
    Wname = append(Wname, nama)
    for {
        fmt.Scan(&nama)
        if nama == "selesai" {
            break
        }
        for i := 0; i < 8; i++ {
            fmt.Scan(&pengerjaan[i])
        }
        Ntime, Nsoal := sortandcount(pengerjaan)
        if Nsoal > Wsoal || (Nsoal == Wsoal && Ntime < Wtime) {
            Wsoal = Nsoal
            Wtime = Ntime
            Wname = []string{nama}
        } else if Nsoal == Wsoal && Ntime == Wtime {
            Wname = append(Wname, nama)
        }
    }
    fmt.Printf("%s %d %d", Wname[0], Wsoal, Wtime)
}

```



```
}
```

Output:

```
P5 D:\test bs> go run 'd:\test bs\lab shit\smsr 2\103112400032_MODUL 4\unguided 4\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
```

Deskripsi Program:

Program ini digunakan untuk menentukan peserta dengan **jumlah soal terbanyak** yang dapat diselesaikan dalam **total waktu maksimal 301 menit**, dari sekumpulan peserta yang masing-masing memiliki waktu pengerjaan 8 soal. Program akan:

1. Mengurutkan waktu pengerjaan tiap peserta secara menaik.
2. Menjumlahkan waktu dari soal tercepat hingga waktu kumulatif melebihi 301 menit.
3. Mencatat peserta dengan jumlah soal terbanyak, dan jika jumlahnya sama, yang waktu totalnya paling sedikit.
4. Jika ada lebih dari satu peserta dengan hasil terbaik yang identik, semua nama dicatat. Setelah pengguna mengetik "selesai", program mencetak nama peserta dengan performa terbaik, beserta jumlah soal yang diselesaikan dan total waktunya.

3. UNGUIDED 3

Source Code:

```
package main

import "fmt"

func algorithm(x int) {
    fmt.Print(x, " ")
    if x == 1 {
        return
    }
    if x%2 == 0 {
        algorithm(x / 2)
    } else {
        algorithm(x*3 + 1)
    }
}

func main() {
    var x int
    fmt.Scan(&x)
    algorithm(x)
}
```

Output:

```
P S b:\test bs> go run 'd:\test bs\lab shit\smsf 2\103112400032_MODUL 4\unguided 4\unguided-4-3.go'
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Deskripsi Program:

Program ini digunakan untuk mencetak deret **Collatz** (atau dikenal juga sebagai $3n + 1$) dari suatu bilangan bulat positif x . Proses dilakukan secara rekursif:

- Jika x adalah 1, maka proses berhenti.
- Jika x genap, maka dilanjutkan dengan $x / 2$.
- Jika x ganjil, maka dilanjutkan dengan $3x + 1$.

Program mencetak seluruh langkah transformasi hingga mencapai angka 1.