

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 2**  
**MODUL IV**  
**PROSEDUR**



**Disusun Oleh :**

**Reza Alvonzo / 2311102026**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

Prosedur atau fungsi adalah blok kode yang melakukan tugas tertentu, dan Go mendukung pemrograman berbasis fungsi sebagai bagian penting dari strukturnya.

### **Pengertian Prosedur/Fungsi**

Prosedur dalam Golang adalah blok kode yang dapat dieksekusi berulang kali, biasanya dipakai untuk menghindari pengulangan kode dan meningkatkan modularitas. Dalam Go, prosedur atau fungsi ditulis dengan menggunakan kata kunci `func`. Setiap fungsi dapat menerima satu atau lebih parameter dan mengembalikan satu atau lebih nilai.

```
package main

import "fmt"

func greet(name string) {

    fmt.Println("Hello", name)

}

func main() {

    greet("Reza")

}
```

Pada contoh ini, fungsi `greet` menerima satu parameter (`name` dari tipe `string`) dan mencetak pesan ke layar. Fungsi ini dipanggil di dalam fungsi `main`, yang merupakan titik masuk (`entry point`) program Go.

## II. GUIDED I

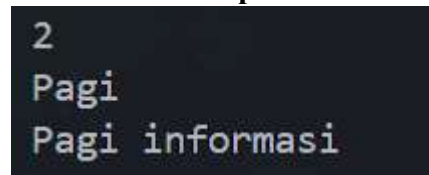
### Sourcecode

```
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)
}
```

### Screenshoot Output



```
2
Pagi
Pagi informasi
```

### Deskripsi Program

Variable bilangan: Digunakan untuk menyimpan sebuah angka bulat yang akan kita inputkan nanti. Sedangkan variable pesan: Digunakan untuk menyimpan sebuah pesan teks yang akan kita inputkan. Misalnya, kita memasukkan nilai 1 untuk bilangan dan "Perhatian!" untuk pesan. Maka program akan mencetak: "Perhatian! Warning"

## GUIDED II

### Sourcecode

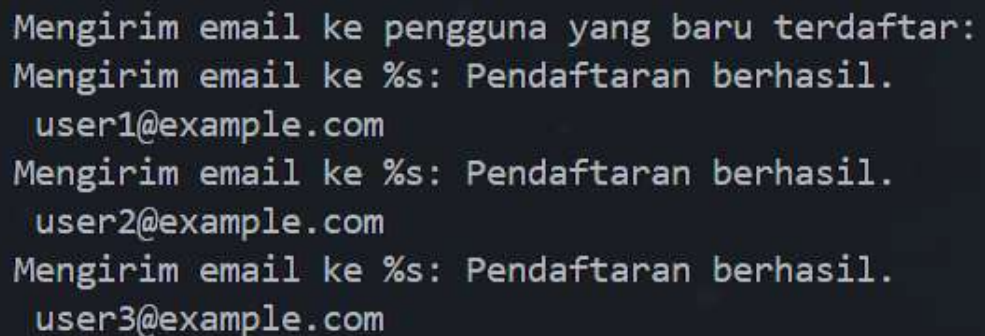
```
package main
import "fmt"

func sendEmailNotification (email string) {
    fmt.Println("Mengirim email ke %s: Pendaftaran
berhasil.\n", email)
}

func main(){
    emails := []string{"user1@example.com",
"user2@example.com", "user3@example.com"}

    fmt.Println("Mengirim email ke pengguna yang baru
terdaftar:")
    for _, email := range emails {
        sendEmailNotification(email)
    }
}
```

### Screenshoot Output



```
Mengirim email ke pengguna yang baru terdaftar:
Mengirim email ke %s: Pendaftaran berhasil.
user1@example.com
Mengirim email ke %s: Pendaftaran berhasil.
user2@example.com
Mengirim email ke %s: Pendaftaran berhasil.
user3@example.com
```

### Deskripsi Program

Fungsi sendEmailNotification Fungsi ini bertugas untuk mencetak pesan ke konsol yang mensimulasikan pengiriman email notifikasi. Untuk parameternya menerima satu parameter email yang bertipe string, yang mewakili alamat email penerima.

## GUIDED III

### Sourcecode

```
package main
import "fmt"

func f1(x, y int) float64 {
    var hasil float64
    hasil = float64(2*x) - 0.5*float64(y) + 3.0
    return hasil
}

func f2(x, y int, hasil *float64){
    *hasil = float64(2*x) - 0.5*float64(y) + 3.0
}

func main(){
    var a, b int
    var c float64

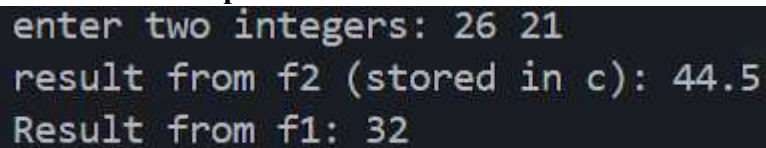
    fmt.Print("enter two integers: ")
    fmt.Scan(&a, &b)

    f2(a, b, &c)

    fmt.Println("result from f2 (stored in c):", c)

    resultF1 := f1(b, a)
    fmt.Println("Result from f1:", resultF1)
}
```

### Screenshot Output



```
enter two integers: 26 21
result from f2 (stored in c): 44.5
Result from f1: 32
```

### Deskripsi Program

Fungsi f1 Menghitung nilai dari ekspresi  $2x - 0.5y + 3$  dan mengembalikannya sebagai nilai float64. Sedangkan f2 Menghitung nilai dari ekspresi  $2x - 0.5y + 3$  dan menyimpan hasilnya dalam alamat memori yang diwakili oleh pointer hasil.

### III. UNGUIDED I

Minggu ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakah kalla membantu Jonas? (tidak tentunya ya :p) Masukan terdiri dari empat buah bilangan asli  $a$ ,  $b$ ,  $c$ , dan  $d$  yang dipisahkan oleh spasi, dengan syarat  $a \geq c$  dan  $b \geq d$  Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi  $a$  terhadap  $c$ , sedangkan baris kedua adalah hasil permutasi dan kombinasi  $b$  terhadap  $d$ . Catatan : permutasi ( $P$ ) dan kombinasi ( $C$ ) dari  $n$  terhadap  $r$  ( $n \geq r$ ) dapat dihitung dengan menggunakan persamaan berikut!  $P(n, r) = \frac{n!}{(n-r)!}$ , sedangkan  $C(n, r) = \frac{n!}{r!(n-r)!}$

#### Sourcecode

```
//Reza Alvonzo//
package main
import "fmt"
var a, b, c, d int

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

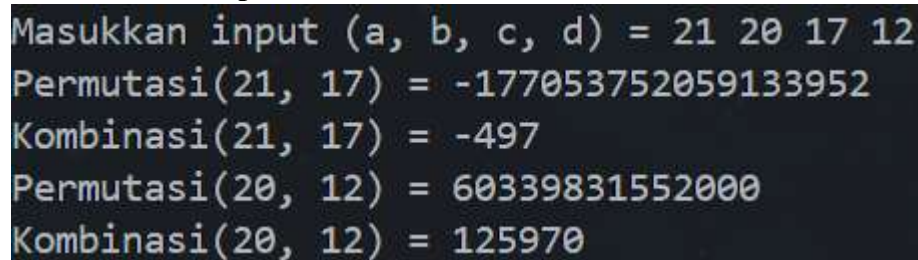
func hitungPermutasi(n, r int) {
    hasil := faktorial(n) / faktorial(n-r)
    fmt.Printf("Permutasi(%d, %d) = %d\n", n, r, hasil)
}

func hitungKombinasi(n, r int) {
    hasil := faktorial(n) / (faktorial(r) * faktorial(n-r))
    fmt.Printf("Kombinasi(%d, %d) = %d\n", n, r, hasil)
}

func main() {
    fmt.Print("Masukkan input (a, b, c, d) = ")
    fmt.Scan(&a, &b, &c, &d)
    if a >= c && b >= d {
        hitungPermutasi(a, c)
        hitungKombinasi(a, c)
    }
}
```

```
        hitungPermutasi(b, d)
        hitungKombinasi(b, d)
    } else {
        fmt.Println("Syarat tidak terpenuhi: a harus >= c
dan b harus >= d")
    }
}
```

### Screenshoot Output



```
Masukkan input (a, b, c, d) = 21 20 17 12
Permutasi(21, 17) = -177053752059133952
Kombinasi(21, 17) = -497
Permutasi(20, 12) = 60339831552000
Kombinasi(20, 12) = 125970
```

### Deskripsi Program

Variabel global a, b, c, dan d digunakan untuk menyimpan input pengguna. Penggunaan variabel global umumnya tidak disarankan karena dapat menyebabkan masalah dalam program yang lebih besar, namun dalam kasus sederhana ini, penggunaannya masih dapat diterima. Fungsi hitungKombinasi ini menghitung kombinasi dari n elemen diambil r elemen. Rumus kombinasi adalah  $n! / (r! * (n-r)!)$ .

### UNGUIDED 2

Kompetisi pemrograman tingkat nasional berlangsung ketat. Setiap peserta diberikan 8 soal yang harus dapat diselesaikan dalam waktu 5 jam saja. Peserta yang berhasil menyelesaikan soal paling banyak dalam waktu paling singkat adalah pemenangnya. 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 didalam prosedur. prosedur hitungSkor(in/out soal, skor : integer) Setiap baris masukan dimulai dengan satu string nama peserta tersebut diikuti dengan adalah 8 integer yang menyatakan berapa lama

(dalam menit) peserta tersebut menyelesaikan soal. Jika tidak berhasil atau tidak mengirimkan jawaban maka otomatis dianggap menyelesaikan dalam waktu 5 jam 1 menit (301 menit). Satu baris keluaran berisi nama pemenang, jumlah soal yang diselesaikan, dan nilai yang diperoleh. Nilai adalah total waktu yang dibutuhkan untuk menyelesaikan soal yang berhasil diselesaikan.

### Sourcecode

```
//Reza Alvonzo//
package main
import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

func hitungSkor(jumlahSoal *int, totalSkor *int, waktuPeserta []int) {
    *jumlahSoal = 0
    *totalSkor = 0
    for _, waktu := range waktuPeserta {
        if waktu <= 300 {
            *totalSkor += waktu
            *jumlahSoal += 1
        }
    }
}

func main() {
    var namaPeserta string
    var waktuPeserta []int
    var pemenang string
    var maxSoal, minWaktu int
    reader := bufio.NewReader(os.Stdin)
    for {
        line, _ := reader.ReadString('\n')
        line = strings.TrimSpace(line)
        if line == "Selesai" {
            break
        }
        input := strings.Fields(line)
        namaPeserta = input[0]
        waktuPeserta = make([]int, 0)
        for _, w := range input[1:] {
            waktu, err := strconv.Atoi(w)
            if err == nil {
                waktuPeserta = append(waktuPeserta, waktu)
            }
        }
    }
}
```



```

    }
}
var jumlahSoal, totalSkor int
hitungSkor(&jumlahSoal, &totalSkor, waktuPeserta)
if jumlahSoal > maxSoal || (jumlahSoal == maxSoal
&&
    totalSkor < minWaktu) {
    pemenang = namaPeserta
    maxSoal = jumlahSoal
    minWaktu = totalSkor
}
}
if pemenang != "" {
    fmt.Printf("%s %d %d\n", pemenang, maxSoal,
minWaktu)
} else {
    fmt.Println("Tidak ada peserta.")
}
}

```

### Screenshoot Output

```

Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 391 26 50 60 65 21
Selesai
Bertha 7 294

```

### Deskripsi Program

Fungsi `hitungSkor` menghitung jumlah soal yang dijawab benar dan total waktu yang digunakan oleh seorang peserta. Menggunakan `bufio.NewReader` untuk membaca input dari pengguna secara baris per baris. Setiap baris berisi nama peserta diikuti oleh waktu yang digunakan untuk setiap soal.

### UNGUIDED III

Skiena dan Revilla dalam Programming Challenges mendefinisikan sebuah deret bilangan. Deret dimulai dengan sebuah bilangan bulat  $n$ . Jika bilangan  $n$  saat itu genap, maka suku berikutnya adalah  $\frac{1}{2}n$ , 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. Sebagai contoh jika dimulai dengan  $n=22$ , maka deret bilangan yang diperoleh adalah: 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 Untuk suku awal sampai dengan 1000000, diketahui deret ini selalu mencapai suku dengan nilai 1. Buat program skiena yang akan mencetak setiap suku dari deret yang dijelaskan diatas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal. prosedur cetakDeret(in  $n$  : integer ) Masukan berupa satu bilangan integer positif yang lebih kecil dari 1000000. Keluaran terdiri dari satu baris saja. Setiap suku dari deret tersebut dicetak dalam baris yang dan dipisahkan oleh sebuah spasi.

#### Sourcecode

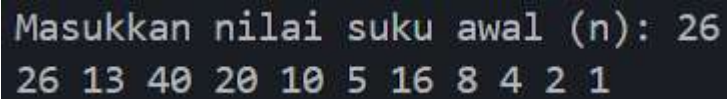
```
//Reza Alvonzo//
package main
import (
    "fmt"
)

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

func main() {
    var n int
    fmt.Print("Masukkan nilai suku awal (n): ")
    fmt.Scan(&n)
    if n > 0 && n < 1000000 {
        cetakDeret(n)
    }
}
```

```
    } else {  
        fmt.Println("Input tidak valid. Harap masukkan  
        bilangan bulat positif yang lebih kecil dari 1000000.")  
    }  
}
```

### Screenshoot Output



```
Masukkan nilai suku awal (n): 26  
26 13 40 20 10 5 16 8 4 2 1
```

### Deskripsi Program

Fungsi cetakderet mencetak deret Collatz dimulai dari bilangan n. Meminta input dari pengguna, memanggil fungsi cetakDeret, dan memberikan pesan kesalahan jika input tidak valid.