

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



**Disusun Oleh :**

**Andika Indra Prastawa / 2311102033**

**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

Golang, atau Go Language, adalah bahasa pemrograman yang dikembangkan oleh Google pada tahun 2009. Bahasa ini dirancang untuk meningkatkan efisiensi dalam pengembangan perangkat lunak, terutama untuk aplikasi berskala besar. Berikut adalah beberapa konsep dasar dan prosedur yang penting dalam pemrograman menggunakan Golang. Golang adalah bahasa pemrograman open-source yang menawarkan kesederhanaan dan kecepatan. Tujuan utama dari pengembangan Golang adalah untuk mengatasi kompleksitas dari bahasa pemrograman lain seperti C++ dan memudahkan proses pengembangan software

```
func namaFungsi(parameter tipeData) tipeReturn { // kode fungsi return nilai }
```

Karakteristik Fungsi di Go:

- Dapat mengembalikan beberapa nilai
- Mendukung fungsi anonim (closure)
- Bisa dijadikan sebagai tipe data
- Mendukung metode (functions bound to types)
- Dapat menerima variadic parameters

Praktikum ini telah mendemonstrasikan berbagai jenis fungsi dalam Go dan penggunaannya. Beberapa poin penting:

- Go menyediakan sintaks yang bersih dan mudah dibaca untuk penulisan fungsi
- Multiple return values sangat berguna untuk error handling
- Variadic functions memberikan fleksibilitas dalam jumlah parameter
- Named return values meningkatkan readability kode

## II. GUIDED

### 1. Guided 1

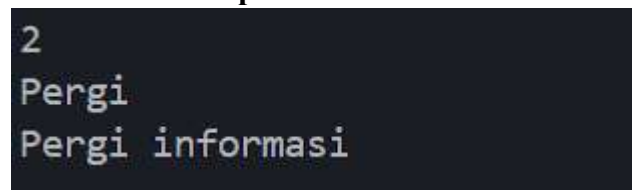
#### Sourcecode

```
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
Pergi
Pergi informasi
```

#### Deskripsi Program

Fungsi Scan() digunakan untuk membaca input dari keyboard. Parameter yang diberikan adalah alamat memory (&) dari kedua variables yang telah dideklarasikan sebelumnya (bilangan, dan pesan). Artinya, saat user diminta untuk memasukkan nilai, nilainya akan disimpan langsung ke dalam variabel yang telah didefinisikan.

## 2. Guided 2

### Sourcecode

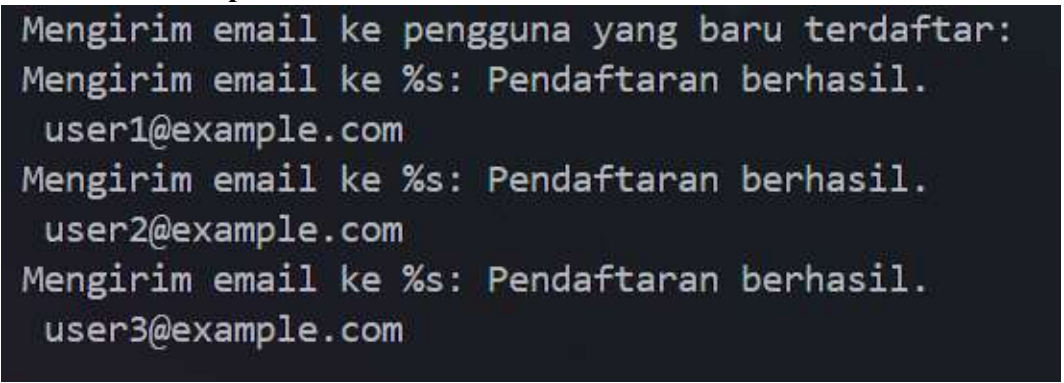
```
package main
import "fmt"

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

Program menggunakan loop for range untuk iterasi atas elemen-elemen dalam array slice `emails`. Loop ini akan melewati setiap item dalam array dan melakukan operasi tertentu pada setiap item tersebut.

### 3. Guided 3

#### 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

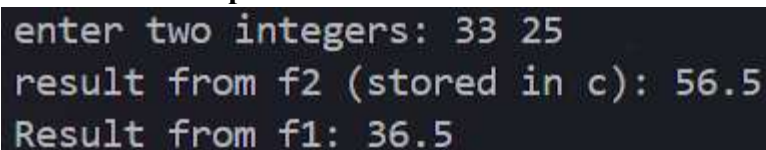
    fmt.Print("enter two integers: ")
    fmt.Scan(&a, &b)

    f2(a, b, &c)

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

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

#### Screenshoot Output



```
enter two integers: 33 25
result from f2 (stored in c): 56.5
Result from f1: 36.5
```

### **Deskripsi Program**

Fungsi f1 menerima dua parameter bertipe integer (x dan y). Proses:  
Menghitung hasil dengan rumus:  $hasil = 2 \cdot x - 0.5 \cdot y + 3.0$   
Hasil perhitungan tersebut dikonversi menjadi tipe data float64. Return:  
Mengembalikan nilai hasil perhitungan sebagai tipe float64.

.

### III. UNGUIDED

#### Unguided 1

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

```
package main
import "fmt"
var a, b, c, d int
//Andika Indra Prastawa
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) = 10 20 8 18
Permutasi(10, 8) = 1814400
Kombinasi(10, 8) = 45
Permutasi(20, 18) = 1216451004088320000
Kombinasi(20, 18) = 190
```

### Deskripsi Program

Faktorial adalah hasil perkalian semua bilangan bulat positif kurang dari atau sama dengan n. Fungsi ini digunakan untuk menghitung permutasi, yaitu banyaknya cara mengatur r elemen dari n elemen yang berbeda.

### 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

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

//Andika Indra Prastawa

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

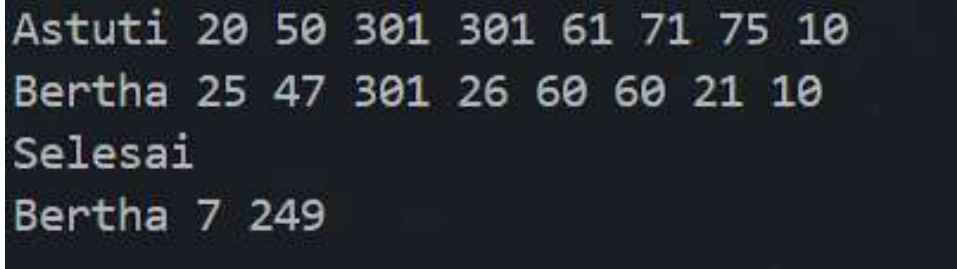
func main() {
    var namaPeserta string
    var waktuPeserta []int
    var pemenang string
    var maxSoal, minWaktu int
    reader := bufio.NewReader(os.Stdin)
    for {
        line, _ := reader.ReadString('\n')
        line = strings.TrimSpace(line)
        if line == "Selesai" {
            break
        }
        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 47 301 26 60 60 21 10
Selesai
Bertha 7 249

```

### Deskripsi Program

Pertama-tama, nama partisipannya akan disimpan dalam variabel namaPartisipan. Kemudian, sisanya akan dibagi-bagikan lagi menjadi list angka yang merepresentasikan waktu tempuh masing-masing partisipan. Lalu, fungsi hitungSoker dipanggil untuk menghitung banyaknya soal yang sudah diselesaikan dan total waktu yang digunakan.

### Unguided 3

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

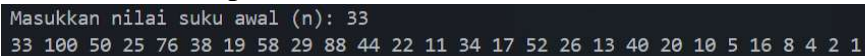
```
package main
import (
    "fmt"
)

//Andika Indra Prastawa
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): 33  
33 100 50 25 76 38 19 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

### Deskripsi Program

Program mendeklarasikan variabel `n` untuk menyimpan input dari pengguna. Menggunakan `fmt.Scan()` untuk membaca input dari pengguna dan menyimpannya dalam variabel `n`. Program memeriksa apakah nilai `n` berada dalam rentang yang valid (lebih besar dari 0 dan kurang dari 1.000.000): Jika valid, fungsi `cetakDeret(n)` dipanggil untuk mencetak deret. Jika tidak valid, program menampilkan pesan kesalahan yang meminta pengguna untuk memasukkan bilangan bulat positif yang sesuai.