

**LAPORAN PRAKTIKUM ALOGARITMA**  
**PEMOGRAMAN 2**  
**MODUL IV**  
**PROSEDUR**



**Disusun Oleh :**  
**Erwin Rivaldo Silaban/2311102248**  
**Dosen Pengampu :**  
**Abednego Dwi Septiadi, S.Kom,M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## I. Dasar Teori

Prosedur dalam Go diimplementasikan melalui fungsi (functions), yang merupakan blok kode yang dapat digunakan kembali untuk melakukan tugas tertentu. Go memperlakukan fungsi sebagai first-class citizens, artinya fungsi dapat diperlakukan seperti tipe data lainnya - bisa dijadikan parameter, nilai return, atau disimpan dalam variabel.

Deklarasi fungsi dalam Go menggunakan kata kunci `func`, diikuti nama fungsi, parameter (opsional), dan tipe data return (opsional). Contoh sintaks dasar:

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

Go mendukung beberapa jenis fungsi utama:

1. Fungsi tanpa parameter dan return value
2. Fungsi dengan parameter
3. Fungsi dengan single return value
4. Fungsi dengan multiple return values
5. Method (fungsi yang terikat dengan tipe data tertentu)

Fitur penting dalam prosedur Go meliputi:

- Pass by value sebagai default
- Pointer untuk pass by reference
- Defer untuk menunda eksekusi
- Panic dan recovery untuk penanganan error
- Interface untuk mendefinisikan kontrak method

Best practices dalam penulisan fungsi Go:

- Gunakan camelCase untuk penamaan
- Awali dengan huruf kapital untuk fungsi yang bisa diakses dari package lain
- Selalu tangani error yang dikembalikan
- Dokumentasikan fungsi dengan format godoc

Kombinasi kesederhanaan sintaks dan fitur modern membuat sistem prosedural Go menjadi powerful namun tetap mudah dipahami, menjadikannya pilihan yang baik untuk pengembangan aplikasi berbagai skala.

## II. Guided

### 1. Source code

```
package main

import "fmt"

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.Print(M, jenis)
}

func main() {
    var bilangan int
    var pesan string

    fmt.Scan(&bilangan)
    fmt.Scan(&pesan)

    cetakPesan(pesan, bilangan)
}
```

- Screenshot hasil

```

15 //
16
17 func main() {
18     var bilangan int
19     var pesan string
20 }

```

PROBLEMS 28 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

PS D:\tes\praktek golang> go run "d:\tes\praktek golang\prosedur\materialpro\prog1.go"

1 hello\_word

1 hello\_wordwarning

PS D:\tes\praktek golang>

- Deskripsi Program

Program diatas akan mencetak jenis pesan tertentu berdasarkan nilai yang di berikan. Program ini memiliki dua fungsi utama : fungsi cetak pesan dan fungsi main. Program akan menampilkan output seperti yang ada pada hasil

## 2. Guided 2

- Source code

```

package main

import "fmt"

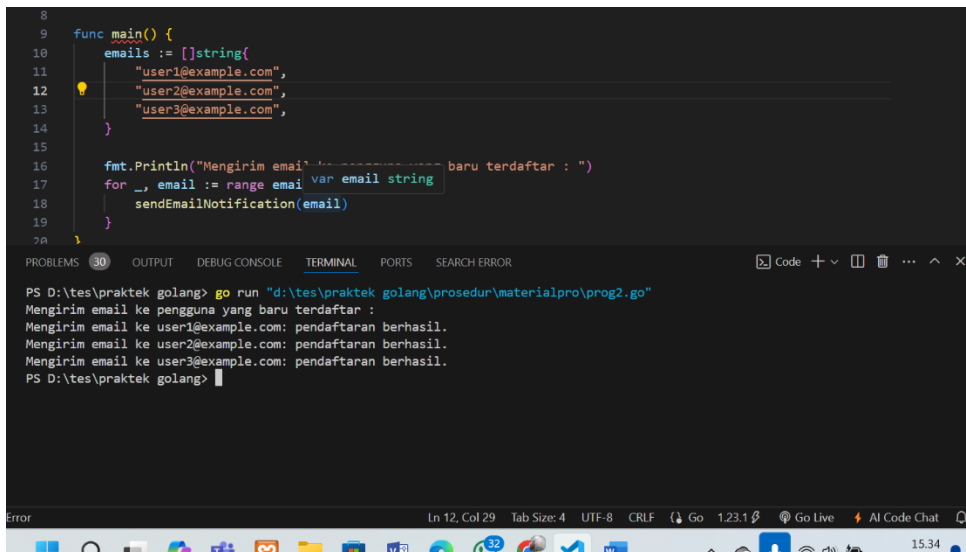
func sendEmailNotification(email string) {
    fmt.Printf("Mengirim email ke %s: pendaftaran berhasil. \n", email)
}

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)
    }
}

```

- Screenshot hasil



```

8
9 func main() {
10     emails := []string{
11         "user1@example.com",
12         "user2@example.com",
13         "user3@example.com",
14     }
15
16     fmt.Println("Mengirim email ke pengguna yang baru terdaftar : ")
17     for _, email := range emails {
18         sendEmailNotification(email)
19     }
20 }

```

PS D:\tes\praktek golang> go run "d:\tes\praktek golang\prosedur\materialpro\prog2.go"

Mengirim email ke pengguna yang baru terdaftar :

Mengirim email ke user1@example.com: pendaftaran berhasil.

Mengirim email ke user2@example.com: pendaftaran berhasil.

Mengirim email ke user3@example.com: pendaftaran berhasil.

PS D:\tes\praktek golang>

- Deskripsi Program

Program diatas akan mengirimkan notifikasi email kepada beberapa pengguna yang baru saja mendaftar

### 3. Guided 3

- Source code

```

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) { //pass by preference
    *hasil = float64(2*x) - 0.5*float64(y) + 3.0
}

func main() {
    var x, y int
    var z float64

    fmt.Print("Enter two intergers : ")
    fmt.Scan(&x, &y)

```

```

    f2(x, y, &z)
    fmt.Println("Result from f2 (stored in z) : ", z)

    resultF1 := f1(y, x)
    fmt.Println("Result from f1 : ", resultF1)
}

```

- Screenshot hasil

```

16  var x, y int
17  var z float64
18
19  fmt.Print("Enter two intergers : ")
20  fmt.Scan(&x, &y)

```

PROBLEMS 38 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

PS D:\tes\praktek golang> go run "d:\tes\praktek golang\prosedur\prog3.go"

Enter two intergers : 2 3

Result from f2 (stored in z) : 5.5

Result from f1 : 8

PS D:\tes\praktek golang>

Error Ln 14, Col 1 Tab Size: 4 UTF-8 CRLF Go 1.23.1 Go Live AI Code Chat

- Deskripsi Program

Program diatas berfungsi untuk menghitung nilai berdasarkan dua input integer yang di masukkan oleh pengguna. Program juga melakukan perhitungan yang sama, tetapi hasilnya disimpan dalam variabel yang di teruskan sebagai pointer.

### III. UNGUIDE

#### 1. K

- Source code

```

package main

import "fmt"

var erwin248, b, c, d int

```

```

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

func permutasi(n, r int) {
    var faktorN, faktorNR int
    faktorial(n, &faktorN)
    faktorial(n-r, &faktorNR)

    hasil := faktorN / faktorNR

    fmt.Print(hasil)
}

func kombinasi(n, r int) {
    var faktorN, faktorR, faktorNR int
    faktorial(n, &faktorN)
    faktorial(r, &faktorR)
    faktorial(n-r, &faktorNR)

    hasil := faktorN / (faktorR * faktorNR)

    fmt.Print(hasil)
}

func main() {

    fmt.Print("Masukkan input = ")
    fmt.Scan(&erwin248, &b, &c, &d)

    if erwin248 >= c && b >= d {
        permutasi(erwin248, c)
        fmt.Print(", ")
        kombinasi(erwin248, c)

        fmt.Print("\n")
        permutasi(b, d)
        fmt.Print(", ")
        kombinasi(b, d)
    } else {
        fmt.Println("Syarat tidak terpenuhi: a harus >= c dan b harus >= d")
    }
}

```

- Screenshot hasil

```

19      hasil := faktorialN / faktorialN
20
PROBLEMS 40 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\tes\praktek golang> go run "d:\tes\praktek golang\prosedur\prouu1.go"
Masukkan input = 5 10 3 10
60, 10
3628800, 1
PS D:\tes\praktek golang>

```

- Deskripsi program

Program ini menghitung permutasi dan kombinasi berdasarkan input pengguna. Terdapat fungsi `faktorial` yang menghitung faktorial, serta dua fungsi utama: `permutasi` untuk menghitung urutan objek dan `kombinasi` untuk memilih objek tanpa memperhatikan urutan. Di dalam fungsi `main`, pengguna diminta memasukkan empat nilai, lalu program memeriksa apakah dua syarat terpenuhi (`erwin248 >= c` dan `b >= d`). Jika ya, program menghitung dan mencetak hasil permutasi serta kombinasi dari pasangan nilai tersebut. Jika tidak, ditampilkan pesan bahwa syarat tidak terpenuhi.

## 2. Unguided 2

- Source code

```

package main

import "fmt"

func hitungSkor(soalSelesai, skor *int) {
    var waktu int
    *soalSelesai = 0
    *skor = 0
    soal := 8

    for i := 1; i <= soal; i++ {
        fmt.Printf("masukkan waktu selesai soal %v : ", i)
        fmt.Scanln(&waktu)

        if waktu < 301 {
            *soalSelesai++
            *skor = *skor + waktu
        }
    }
}

```



```

    }
}

func main() {
    var winn48, erwinn_248 string
    var skor, soalSelesai int

    for {
        var temp, temp1 int

        fmt.Print("\nMasukkan nama peserta : ")
        fmt.Scanln(&winn48)

        if winn48 == "selesai" {
            break
        }

        hitungSkor(&temp, &temp1)

        if temp > soalSelesai || temp == soalSelesai && temp1 < skor {
            erwinn_248 = winn48
            soalSelesai = temp
            skor = temp1
        }
    }

    fmt.Printf("\n%v %v %v", erwinn_248, soalSelesai, skor)
}

```

- Screenshot hasil

```

16      *soalSelesai++
17      *skor = *skor + waktu
18  }
19  }
20  }

PS D:\tes\praktek golang> go run "d:\tes\praktek golang\prosedur\proouu2.go"

Masukkan nama peserta : erwin
masukkan waktu selesai soal 1 : 3
masukkan waktu selesai soal 2 : 45
masukkan waktu selesai soal 3 : 23
masukkan waktu selesai soal 4 : 2
masukkan waktu selesai soal 5 : 1
masukkan waktu selesai soal 6 : 21
masukkan waktu selesai soal 7 : 6
masukkan waktu selesai soal 8 : 76

Masukkan nama peserta : erwin

```

- Deskripsi Program

Program ini bertujuan untuk menentukan peserta dengan jumlah soal selesai terbanyak dan waktu total terkecil. Fungsi `hitungSkor` digunakan untuk menghitung jumlah soal yang diselesaikan oleh seorang peserta dan total waktu yang dihabiskan, di mana hanya soal yang selesai dalam waktu kurang dari 301 detik yang dihitung. Dalam fungsi `main`, program meminta nama peserta secara berulang hingga pengguna memasukkan "selesai". Setiap peserta diminta menginput waktu penyelesaian untuk delapan soal, kemudian program membandingkan hasil peserta tersebut dengan peserta sebelumnya. Jika peserta baru menyelesaikan lebih banyak soal atau menyelesaikan jumlah soal yang sama tetapi dalam waktu yang lebih singkat, data peserta baru akan disimpan sebagai yang terbaik. Pada akhir program, nama peserta terbaik beserta jumlah soal yang selesai dan skor waktu terkecil akan ditampilkan.

### 3. Unguided 3

- Source code

```
package main

import "fmt"

func deretBilangan(n int) {
    win2248 := n

    for win2248 != 1 {
        fmt.Print(win2248, " ")

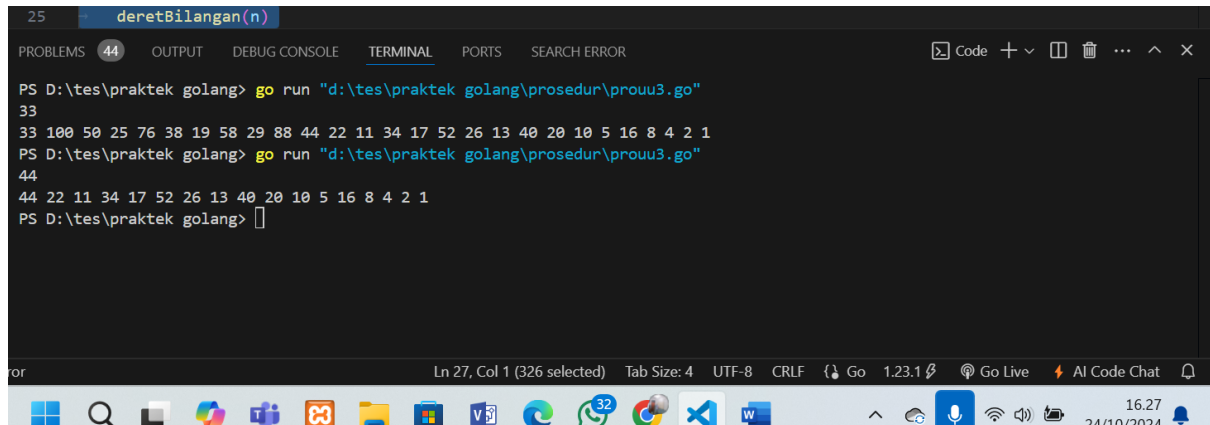
        if win2248%2 == 0 {
            win2248 = win2248 / 2
        } else {
            win2248 = (3 * win2248) + 1
        }
    }
    fmt.Println(win2248)
}

func main() {
    var n int

    fmt.Scan(&n)

    deretBilangan(n)
}
```

- Screenshot hasil



```
25  deretBilangan(n)
PROBLEMS 44 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\tes\praktek golang> go run "d:\tes\praktek golang\prosedur\prouu3.go"
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
PS D:\tes\praktek golang> go run "d:\tes\praktek golang\prosedur\prouu3.go"
44
44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\tes\praktek golang>
```

- Deskripsi Program

Program ini menampilkan deret bilangan berdasarkan algoritma **Collatz**. Pada fungsi `deretBilangan`, bilangan `n` yang diinputkan akan diolah sebagai berikut: jika bilangan tersebut genap, akan dibagi 2; jika ganjil, akan dikalikan 3 lalu ditambah 1. Proses ini berlanjut hingga bilangan mencapai 1. Dalam fungsi `main`, program meminta pengguna memasukkan nilai `n` dan kemudian memanggil fungsi `deretBilangan` untuk menampilkan deret hasil dari proses tersebut.