

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROSEDUR**



Disusun Oleh :

Marsep Trianto Pakondo / 2311102251

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

Pengertian Prosedur dalam Golang

Prosedur dalam Golang merupakan sebuah unit atau blok program yang berisi serangkaian perintah atau instruksi yang dapat dipanggil untuk menjalankan tugas tertentu. Berbeda dengan fungsi yang mengembalikan nilai, prosedur dalam Golang dirancang untuk melakukan serangkaian aksi atau tugas tanpa mengembalikan nilai. Prosedur membantu dalam mengorganisasi program menjadi bagian-bagian yang lebih kecil dan dapat dikelola dengan lebih baik.

Karakteristik Prosedur dalam Golang

Prosedur dalam Golang memiliki sifat modular, yang berarti:

- Dapat berdiri sendiri sebagai unit program yang independen
- Dapat dipanggil dari bagian program manapun
- Memiliki batasan yang jelas dengan bagian program lainnya
- Memudahkan dalam pemeliharaan dan pengujian kode

Prosedur memiliki ruang lingkup yang menentukan aksesibilitas variabel:

- Variabel yang dideklarasikan di dalam prosedur bersifat lokal
- Hanya dapat diakses di dalam prosedur tersebut
- Memiliki lifetime selama prosedur tersebut dieksekusi
- Tidak dapat diakses dari luar prosedur

Prosedur dapat menerima parameter yang memungkinkan:

- Pertukaran data antara pemanggil dan prosedur
- Penggunaan prosedur yang lebih fleksibel
- Pemrosesan data yang berbeda-beda
- Interaksi dengan bagian program lainnya

II. GUIDED

Soal Studi Case

XXXXXXXXXXXXXXXXXXXX

Sourcecode

```
package main

import "fmt"

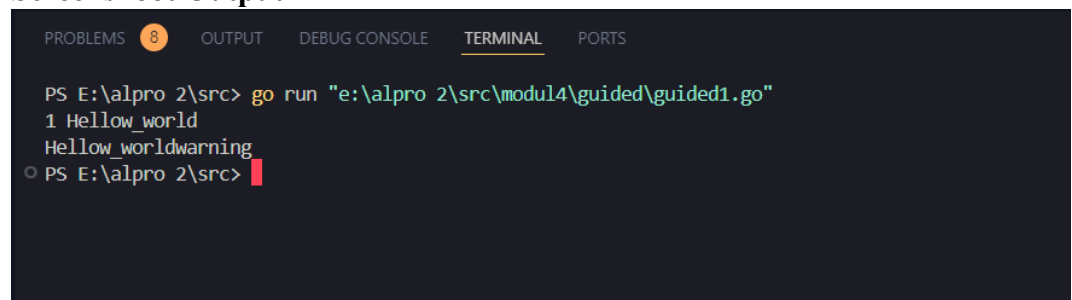
func cetakPesan(M string, flag int) {
    var jenis string = ""
    if flag == 0 {
        jenis = "error"
    } else if flag == 1 {
        jenis = "warning"
    } else if flag == 2 {
        jenis = "informasi"
    }
    fmt.Print(M, jenis)
}

func main() {
    var bilangan int
    var pesan string

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

    cetakPesan(pesan, bilangan)
}
```

Screenshoot Output



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS E:\alpro 2\src> go run "e:\alpro 2\src\modul4\guided\guided1.go"
1 Hellow_world
Hellow_worldwarning
PS E:\alpro 2\src>
```

Deskripsi Program

Program diatas digunakan untuk mencetak pesan dengan jenis tertentu berdasarkan nilai yang diberikan. Program ini memiliki dua fungsi utama: cetakPesan dan main. Fungsi cetakPesan menerima dua argumen, yaitu sebuah string (pesan) dan sebuah bilangan bulat (flag). Nilai flag menentukan jenis pesan yang akan dicetak, yaitu "error" jika flag adalah 0, "warning" jika flag adalah 1, dan "informasi" jika flag adalah 2. Fungsi main meminta pengguna untuk memasukkan sebuah bilangan dan sebuah pesan, kemudian memanggil fungsi cetakPesan dengan nilai-nilai yang telah dimasukkan.

Sourcecode

```
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)
    }
}
```

Screenshoot Output

```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
• PS E:\alpro 2\src> go run "e:\alpro 2\src\modul4\guided\tempCodeRunnerFile.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 E:\alpro 2\src>
```

Deskripsi Program

Program diatas digunakan untuk mengirimkan notifikasi email kepada beberapa pengguna yang baru saja mendaftar. Program ini memiliki dua fungsi utama: `sendEmailNotification` dan `main`. Fungsi `sendEmailNotification` menerima alamat email sebagai input dan mencetak pesan ke konsol yang mensimulasikan pengiriman email. Fungsi `main` berisi sebuah slice (array dinamis) yang berisi alamat email beberapa pengguna. Dalam loop `for`, setiap alamat email dalam slice tersebut akan dipanggil satu per satu sebagai argumen untuk fungsi `sendEmailNotification`, sehingga program akan mencetak pesan notifikasi untuk setiap pengguna yang ada dalam slice. Jadi singkatnya, kode ini itu mendemonstrasikan cara mengirim notifikasi email secara massal ke beberapa pengguna.

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) { //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)
    }

```

Screenshoot Output

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS E:\alpro 2\src> go run "e:\alpro 2\src\modul4\guided\tempCodeRunnerFile.go"
Enter two intergers : 5 10
Result from f2 (stored in z) : 8
Result from f1 : 20.5
● PS E:\alpro 2\src> go run "e:\alpro 2\src\modul4\guided\tempCodeRunnerFile.go"
Enter two intergers : 10 5
● Result from f2 (stored in z) : 20.5
● Result from f1 : 8
○ PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk mengimplementasikan dua fungsi matematika yang menerima dua bilangan bulat sebagai input dan mengembalikan hasil perhitungan sebagai bilangan desimal. Fungsi f1 menggunakan mekanisme "return by value", di mana hasil perhitungan dikembalikan sebagai nilai dari fungsi. Fungsi f2 menggunakan mekanisme "pass by reference", di mana hasil perhitungan disimpan dalam alamat memori yang ditunjuk oleh pointer yang di-pass sebagai argumen. Dalam fungsi main, pengguna diminta untuk memasukkan dua bilangan bulat, kemudian kedua fungsi tersebut dipanggil dengan bilangan-bilangan tersebut sebagai input. Hasil perhitungan dari kedua fungsi kemudian dicetak ke layar. Perbedaan utama antara kedua fungsi terletak pada cara mereka mengembalikan hasil perhitungan. Fungsi f1 mengembalikan hasil secara langsung, sedangkan fungsi f2 mengubah nilai yang ditunjuk oleh pointer yang di-pass sebagai argumen.

III. UNGUIDED

Soal Studi Case

XXXXXXXXXXXXXXXXXXXX

Sourcecode

```
package main

import "fmt"

var a,b,c,d int

// Fungsi untuk menghitung faktorial dari bilangan n
func faktorial(n int, hasil *int) {
    *hasil = 1
    // Loop untuk menghitung faktorial, dimulai dari 1
    hingga n
    for i := 1; i <= n; i++ {
        *hasil = *hasil * i
    }
}

// Fungsi untuk menghitung permutasi  $P(n, r) = n! / (n-r)!$ 
func permutasi(n, r int) {
    var faktorN, faktorNR int
    faktorial(n, &faktorN)
    faktorial(n-r, &faktorNR)

    hasil := faktorN / faktorNR

    fmt.Print(hasil)
}

// Fungsi untuk menghitung kombinasi  $C(n, r) = n! / (r! * (n-r)!)$ 
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(&a,&b,&c,&d)

        if a >= c && b >= d {
            // Baris pertama: Permutasi dan Kombinasi a
            terhadap c

            permutasi(a, c)
            fmt.Print(", ")
            kombinasi(a, c)

            fmt.Print("\n")
            // Baris kedua: Permutasi dan Kombinasi b
            terhadap d

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

```

Screenshoot Output

```

PS E:\alpro 2\src> go run "e:\alpro 2\src\modul4\unguided\unguided1.go"
Masukkan input = 5 10 3 10
60, 10
3628800, 1
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul4\unguided\unguided1.go"
Masukkan input = 8 0 2 0
56, 28
1, 1
PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk menghitung faktorial, permutasi, dan kombinasi dari dua inputan bilangan bulat. Fungsi faktorial menghitung faktorial dari sebuah bilangan dengan menggunakan pointer untuk memodifikasi nilai variabel di luar fungsi. Fungsi permutasi dan kombinasi menggunakan fungsi faktorial untuk menghitung nilai permutasi dan kombinasi, kemudian mencetak hasilnya langsung ke layar.

Fungsi main meminta pengguna untuk memasukkan empat bilangan bulat, kemudian memanggil fungsi permutasi dan kombinasi untuk menghitung nilai permutasi dan kombinasi dari dua pasang bilangan tersebut kalau syarat $a \geq c$ dan $b \geq d$ terpenuhi.

Sourcecode

```
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 namaPeserta, namaPemenang string
    var skor, soalSelesai int

    for {
        var temp, temp1 int

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

        if namaPeserta == "selesai" {
            break
        }
    }
}
```

```

        hitungSkor(&temp, &temp1)

        if temp > soalSelesai || temp == soalSelesai &&
temp1 < skor {
            namaPemenang = namaPeserta
            soalSelesai = temp
            skor = temp1
        }
    }

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

```

Screenshot Output

```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul4\unguided\unguided2.go"

Masukkan nama peserta : Astuti
masukkan waktu selesai soal 1 : 22
masukkan waktu selesai soal 2 : 30
masukkan waktu selesai soal 3 : 301
masukkan waktu selesai soal 4 : 20
masukkan waktu selesai soal 5 : 10
masukkan waktu selesai soal 6 : 33
masukkan waktu selesai soal 7 : 50
masukkan waktu selesai soal 8 : 21

Masukkan nama peserta : Bertha
masukkan waktu selesai soal 1 : 23
masukkan waktu selesai soal 2 : 301
masukkan waktu selesai soal 3 : 40
masukkan waktu selesai soal 4 : 21
masukkan waktu selesai soal 5 : 55
masukkan waktu selesai soal 6 : 61
masukkan waktu selesai soal 7 : 50
masukkan waktu selesai soal 8 : 15

Masukkan nama peserta : selesai

Astuti 7 186
PS E:\alpro 2\src>

```

Deskripsi Program

Program diatas digunakan untuk menghitung skor dan jumlah soal yang selesai dikerjakan oleh beberapa peserta ujian. Fungsi `hitungSkor` menerima dua pointer sebagai argumen untuk menyimpan jumlah soal yang selesai dan total skor. kemudian, program akan meminta pengguna untuk memasukkan waktu yang dibutuhkan untuk menyelesaikan setiap soal. Jika waktu yang dibutuhkan kurang dari 301 detik, maka soal dianggap selesai dan skor akan ditambahkan dengan waktu yang digunakan. Fungsi `main` akan meminta pengguna untuk memasukkan nama peserta secara berulang hingga pengguna memasukkan "selesai". Untuk setiap peserta, fungsi `hitungSkor` akan dipanggil untuk menghitung

skor dan jumlah soal yang selesai. Program akan terus melacak peserta dengan jumlah soal selesai terbanyak, dan apabila kedua peserta menyelesaikan jumlah soal sama, maka yang menang yaitu mengerjakan dengan waktu yang lebih cepat. kemudian mencetak nama pemenang, jumlah soal yang selesai, dan total skornya(setiap waktu yang ditambahkan).

Sourcecode

```
package main

import "fmt"

func deretBilangan(n int) {
    hasil := n

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

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

func main() {
    var n int

    fmt.Scan(&n)

    deretBilangan(n)
}
```

Screenshoot Output

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS E:\alpro 2\src> go run "e:\alpro 2\src\modul4\unguided\unguided3.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS E:\alpro 2\src> go run "e:\alpro 2\src\modul4\unguided\unguided3.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 E:\alpro 2\src>
```

Deskripsi Program

Program diatas digunakan untuk menghasilkan deret bilangan dengan aturan apabila bilangan yang dimasukkan pengguna berupa bilangan genap maka menggunakan rumus $\frac{1}{2}n$, dan apabila ganjil menggunakan rumus $3n+1$. Fungsi deretBilangan menerima sebuah bilangan bulat n sebagai input dan kemudian melakukan iterasi terus-menerus sampai nilai hasil mencapai 1. Pada setiap iterasi, nilai hasil akan dicetak ke layar. Jika hasil adalah bilangan genap, maka nilai hasil akan dibagi 2. Sebaliknya, jika hasil adalah bilangan ganjil, maka nilai hasil akan dikalikan 3 dan ditambah 1. Proses ini akan terus diulang sampai hasil bernilai 1. Dalam fungsi main, pengguna diminta untuk memasukkan sebuah bilangan bulat dan memanggil fungsi deretBilangan untuk menghasilkan deret bilangan sesuai dengan aturannya.