

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROSEDUR**



Disusun Oleh :

Fahrial Aufa Ramadhan / 2311102241

IF-11-6

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Prosedur dan function adalah program kecil yang berada dalam program utama yang digunakan untuk menyelesaikan masalah khusus dengan parameter yang diberikan. Perbedaan prosedur dan function ada pada pengembalian nilai. Pada prosedur, nilai yang diproses tidak dapat dikembalikan atau diubah, nilainya hanya diproses di dalam prosedur dan kembali kepada keadaan semula ketika keluar dari prosedur. Sedangkan pada function, nilai yang diproses akan berubah dan memiliki nilai keluaran. Bisa disimpulkan bahwa prosedur itu hanya berisi metode, langkah-langkah atau tahap-tahap penyelesaian masalah sedangkan function adalah berisi metode yang memproses nilai dan mengubah nilai itu sendiri.

PEMANGGILAN PROSEDUR

Prosedur bukan program yang berdiri sendiri, jadi ia tidak dapat dieksekusi secara langsung. (ini berarti, instruksi-instruksi di dalam prosedur baru dapat dilaksanakan hanya bila prosedur tersebut diakses. Untuk memanggil sebuah fungsi tanpa nilai balik, yang perlu dilakukan hanyalah menuliskan nama fungsi diikuti dengan daftar parameternya sama persis dengan pemanggilan fungsi dengan nilai balik.

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

Ketika nama prosedur dipanggil, kendali program berpindah secara otomatis pada prosedur tersebut. Seluruh instruksi di dalam badan prosedur dilaksanakan. Setelah semua instruksi selesai dilaksanakan, kendali program berpindah secara otomatis kembali ke instruksi sesudah pemanggil prosedur.

I. GUIDED

Soal Studi Case

Buatlah sebuah program beserta prosedur yang digunakan untuk menampilkan suatu pesan error, warning atau Informasi berdasarkan masukan dari user. Masukan terdiri dari sebuah bilangan bulat flag (0 s.d. 2) dan sebuah string pesan M. Keluaran berupa string pesan M beserta jenis pesannya, yaitu error, warning atau informasi berdasarkan nilai flag 0, 1 dan 2 secara berturut-turut.

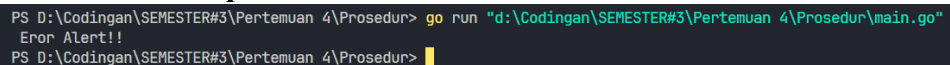
Sourcecode

```
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 = "Alert error"
    }else if flag == 1{
        jenis = "warning"
    }else if flag ==2 {
        jenis = "informasi"
    }
    fmt.Println(M, jenis)
}
```

Screenshoot Output



```
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur> go run "d:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur\main.go"
Error Alert!!
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur> 
```

Deskripsi Program

fungsi utama main, yang minta pengguna untuk memasukkan sebuah bilangan bulat dan sebuah string. Kode ini juga mendefinisikan fungsi cetakPesan, yang menerima dua parameter: sebuah string dan sebuah integer (flag).

II. GUIDED

Soal Studi Case

Sourcecode

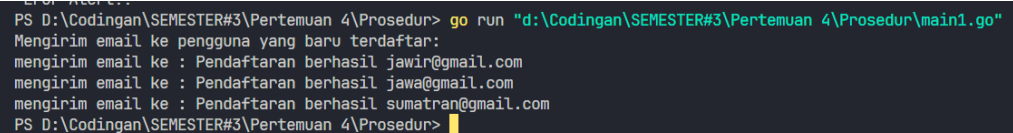
```
package main

import "fmt"

func sendEmailNotification(email string) {
    fmt.Println("mengirim email ke : Pendaftaran
    berhasil", email)
}

func main() {
    emails := []string{"jawir@gmail.com",
    "jawa@gmail.com", "sumatran@gmail.com"}
    fmt.Println("Mengirim email ke pengguna yang
    baru terdaftar: ")
    for _, email := range emails {
        sendEmailNotification(email)
    }
}
```

Screenshot Output



```
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur> go run "d:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur\main1.go"
Mengirim email ke pengguna yang baru terdaftar:
mengirim email ke : Pendaftaran berhasil jawir@gmail.com
mengirim email ke : Pendaftaran berhasil jawa@gmail.com
mengirim email ke : Pendaftaran berhasil sumatran@gmail.com
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur>
```

Deskripsi Program

Program menghasilkan pesan pengantar sebelum menggunakan loop untuk iterasi melalui setiap alamat email dalam slice, memanggil fungsi sendEmailNotification untuk masing-masing email.

III. GUIDED

Soal Studi Case

Suatu subprogram yang dipanggil dapat berkomunikasi dengan pemanggilnya melalui argumen yang diberikan melalui parameter yang dideklarasikan pada subprogramnya. Berikut ini jenis atau pembagian dari parameter. Berdasarkan letak penulisannya pada program, maka parameter dapat dikelompokkan menjadi dua, yaitu parameter formal dan parameter aktual.

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

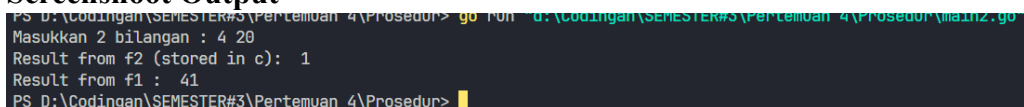
    fmt.Print("Masukkan 2 bilangan : ")
    fmt.Scan(&a, &b)

    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



```
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur> go run d:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur\main2.go
Masukkan 2 bilangan : 4 20
Result from f2 (stored in c): 1
Result from f1 : 41
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur>
```

Deskripsi Program

Program ini meminta input dua bilangan bulat dari pengguna, lalu menggunakan kedua fungsi f1 f2 tersebut untuk menghitung dan mencetak hasil berdasarkan input yang diberikan.

IV. UNGUIDED

Soal Studi Case

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 kalian membantu Jonas? (tidak tentunya ya :p)

Sourcecode

```
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:!!!")
    }
}
```

Screenshoot Output

```
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur> go run "d:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur\unguided1.go"
Masukkan input (a, b, c, d) = 5 10 3 10
Permutasi(5, 3) = 60
Kombinasi(5, 3) = 10
Permutasi(10, 10) = 3628800
Kombinasi(10, 10) = 1
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur> █
```

Deskripsi Program

Program diatas menghitung kombinasi dan permutasi dari dua pasang angka,Pengguna melakukan inputan 4 inputan program memiliki funtion 3 untuk factorial,satu untuk permutasi,dan satu lagi untuk kombinasi

V. UNGUIDED

Soal Studi Case

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 di dalam prosedur.

Sourcecode

```
package main

import (
    "fmt"
    "strings"
)

func hitungSkor(waktu []int, soal *int, skor *int) {
    *soal = 0
    *skor = 0
    for _, t := range waktu {
        if t < 301 {
            *soal++
            *skor += t
        }
    }
}

func main() {
    var pemenang string
    var soalMax, skorMin int
    soalMax = 0
    skorMin = 999999

    for {
        var nama string
        var waktu [8]int

        fmt.Print("Masukkan nama peserta dan waktu penyelesaian soal (ketik 'Selesai' untuk berhenti): ")
        fmt.Scan(&nama)

        if strings.ToLower(nama) == "selesai" {
            break
        }

        for i := 0; i < 8; i++ {
            fmt.Scan(&waktu[i])
        }
    }
}
```



```

        var soal, skor int
        hitungSkor(waktu[:], &soal, &skor)

        if soal > soalMax || (soal == soalMax && skor
< skorMin) {
            pemenang = nama
            soalMax = soal
            skorMin = skor
        }

        if pemenang != "" {
            fmt.Printf("Pemenang: %s dengan %d soal
diselesaikan dan total waktu %d menit\n", pemenang,
soalMax, skorMin)
        } else {
            fmt.Println("Tidak ada peserta.")
        }
    }
}

```

Screenshoot Output

```

PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur> go run "d:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur\tempCodeRunnerFile.go"
Masukkan nama peserta dan waktu penyelesaian soal (ketik 'Selesai' untuk berhenti): Astuti 20 50 301 301 61 71 75 10
Masukkan nama peserta dan waktu penyelesaian soal (ketik 'Selesai' untuk berhenti): Bertha 25 47 301 26 50 60 65 21
Masukkan nama peserta dan waktu penyelesaian soal (ketik 'Selesai' untuk berhenti): selesai
Pemenang: Bertha dengan 7 soal diselesaikan dan total waktu 294 menit
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur>

```

Deskripsi Program

Program ini digunakan untuk pemenang dari kompetisi pemrograman berdasarkan waktu penyelesaian soal. Peserta memasukkan nama mereka diikuti oleh waktu yang diperlukan untuk menyelesaikan 8 soal

VI. UNGUIDED

Soal Studi Case

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 $1/2n$, 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.

Sourcecode

```
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

```
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur> go run "d:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur\unguided3.go"
Masukkan nilai suku awal (n): 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\Codingan\SEMESTER#3\Pertemuan 4\Prosedur> |
```

Deskripsi Program

Program ini dirancang untuk mencetak deret bilangan berdasarkan aturan tertentu. Deret mulai dengan sebuah bilangan positif n yang lebih kecil dari 1.000.000.