

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROSEDUR**



Disusun Oleh :

RAKHA YUDHISTIRA / 2311102010

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

1. Definisi Prosedur

Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu Instruksi baru yang dibuat untuk mengurangi kerumitan dari kode program yang kompleks pada suatu program yang besar. Prosedur akan menghasilkan suatu akibat atau efek langsung pada program ketika dipanggil pada program utama. Suatu subprogram dikatakan prosedur apabila:

1. Tidak ada deklarasi tipe nilai yang dikembalikan, dan
2. Tidak terdapat kata kunci return dalam badan subprogram.

Kedudukannya prosedur sama seperti instruksi dasar yang sudah ada sebelumnya (**assignment**) dan/atau instruksi yang berasal dari paket (**fmt**), seperti **fmt.Scan** dan **fmt.Prntn**. Karena itu selalu pilih nama prosedur yang berbentuk kata kerja atau sesuatu yang merepresentasikan proses sebagai nama cetak, hitungRerata, cariNilai, belok, mulai, dari prosedur.

2. Karakteristik Prosedur

- **Tidak Mengembalikan Nilai (Void):** Seperti yang disebutkan sebelumnya, prosedur tidak memiliki nilai kembalian, yang membedakannya dari fungsi. Jika program memerlukan hasil dari sebuah operasi, lebih tepat menggunakan fungsi daripada prosedur.
- **Mempermudah Debugging:** Dengan membagi program besar menjadi prosedur-prosedur kecil, proses debugging menjadi lebih mudah. Ketika terjadi kesalahan, pengembang hanya perlu memeriksa prosedur yang terkait, bukan seluruh program.
- **Penggunaan Parameter:** Prosedur biasanya memerlukan parameter untuk melakukan tugasnya. Parameter dapat berupa nilai input yang akan digunakan di dalam prosedur. Ada dua tipe utama parameter:
 - 1) **Parameter Masukan (Input Parameter):** Digunakan untuk menerima nilai yang diberikan ke prosedur dari pemanggil.
 - 2) **Parameter Keluaran (Output Parameter):** Digunakan untuk mengembalikan hasil operasi yang dilakukan di dalam prosedur.

II. GUIDED

1. 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 M

Keluaran berupa String pesan M beserta jenis pesannya, yaitu error, warning atau informasi berdasarkan nilai flag 0, 1 dan 2 secara berturut-turut.

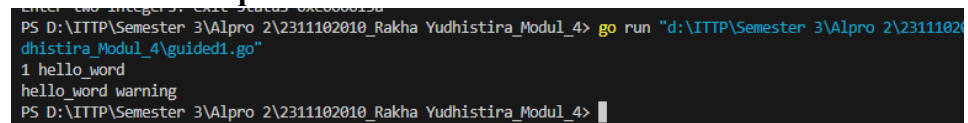
Sourcecode

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

Screenshoot Output



```
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4> go run "d:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4\guided1.go"
1 hello word
hello word warning
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4>
```

Deskripsi Program

Kode di atas dimulai dengan mendeklarasikan dua variabel, yaitu bilangan (tipe int) dan pesan (tipe string), yang nilainya akan diinput oleh pengguna melalui `fmt.Scan`. Setelah menerima input, fungsi `cetakPesan` dipanggil dengan parameter pesan dan bilangan. Fungsi ini mengevaluasi nilai flag (yang merupakan nilai bilangan yang diterima) untuk menentukan jenis pesan, yaitu "error" (jika flag 0), "warning" (jika flag 1), atau "informasi"

(jika flag 2), kemudian mencetak pesan dan jenisnya menggunakan `fmt.Println`.

2. Soal Studi Case

Sourcecode

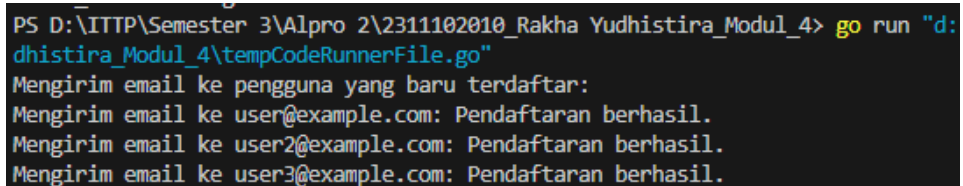
```
package main
import "fmt"

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

func main() {
    emails := []string{"user@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



```
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4> go run "d:
dhistira_Modul_4\tempCodeRunnerFile.go"
Mengirim email ke pengguna yang baru terdaftar:
Mengirim email ke user@example.com: Pendaftaran berhasil.
Mengirim email ke user2@example.com: Pendaftaran berhasil.
Mengirim email ke user3@example.com: Pendaftaran berhasil.
```

Deskripsi Program

Kode di atas mendefinisikan fungsi `sendEmailNotification` yang mencetak pesan "Mengirim email" ke alamat email tertentu yang diterima sebagai parameter. Dalam fungsi `main`, sebuah slice `emails` berisi daftar alamat email pengguna baru. Kemudian, melalui loop `for`, setiap email dalam slice tersebut dikirim ke fungsi `sendEmailNotification`, yang mencetak pesan notifikasi bahwa email pendaftaran berhasil dikirim ke masing-masing alamat.

3. Soal Studi Case

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
}

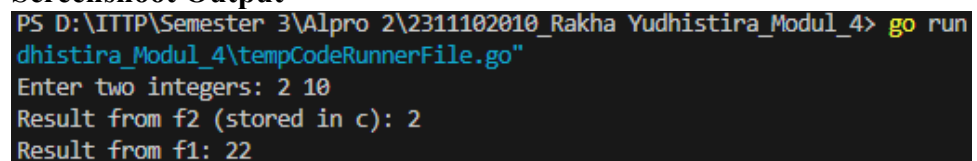
func main() {
    var a, b int
    var c float64

    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



```
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4> go run
dhistira_Modul_4\tempCodeRunnerFile.go
Enter two integers: 2 10
Result from f2 (stored in c): 2
Result from f1: 22
```

Deskripsi Program

Kode di atas mendefinisikan dua fungsi: f1 dan f2. Fungsi f1 mengambil dua parameter integer (x dan y), menghitung hasil dengan formula $(2*x - 0.5*y + 3.0)$, dan mengembalikan hasil sebagai float64. Fungsi f2 melakukan perhitungan yang sama tetapi hasilnya disimpan di alamat memori variabel hasil menggunakan pointer. Dalam fungsi main, pengguna memasukkan dua bilangan bulat, lalu fungsi f2 dipanggil untuk menyimpan hasil ke variabel c. Setelah itu, hasil dari f1 dicetak terpisah, menggunakan nilai b dan a sebagai argumen.

III. UNGUIDED

1. 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)

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!

$$P(n, r) = \frac{n!}{(n-r)!}, \text{ Sedangkan } C(n, r) = \frac{n!}{r!(n-r)!}$$

Sourcecode

```
package main

import (
    "fmt"
)

func factorial(n int) int {
    if n == 0 {
        return 1
    }
    return n * factorial(n-1)
}

func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

func combination(n, r int) int {
    return permutation(n, r) / factorial(r)
}

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

    fmt.Print("Masukkan nilai a, b, c, d: ")
    fmt.Scan(&a, &b, &c, &d)

    if a >= c && b >= d {
        p1 := permutation(a, c)
```

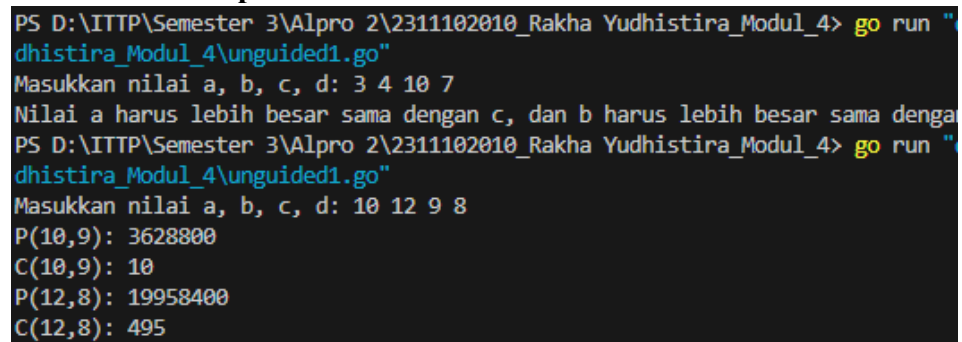
```

        c1 := combination(a, c)
        p2 := permutation(b, d)
        c2 := combination(b, d)

        fmt.Printf("P(%d,%d): %d\n", a, c, p1)
        fmt.Printf("C(%d,%d): %d\n", a, c, c1)
        fmt.Printf("P(%d,%d): %d\n", b, d, p2)
        fmt.Printf("C(%d,%d): %d\n", b, d, c2)
    } else {
        fmt.Println("Nilai a harus lebih besar
sama dengan c, dan b harus lebih besar sama dengan d.")
    }
}

```

Screenshoot Output



```

PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4> go run "
dhistira_Modul_4\unguided1.go"
Masukkan nilai a, b, c, d: 3 4 10 7
Nilai a harus lebih besar sama dengan c, dan b harus lebih besar sama dengan d
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4> go run "
dhistira_Modul_4\unguided1.go"
Masukkan nilai a, b, c, d: 10 12 9 8
P(10,9): 3628800
C(10,9): 10
P(12,8): 19958400
C(12,8): 495

```

Deskripsi Program

Kode di atas mendefinisikan fungsi untuk menghitung faktorial, permutasi, dan kombinasi. Fungsi factorial menggunakan rekursi untuk menghitung faktorial dari bilangan n. Fungsi permutation menghitung permutasi dengan rumus $n! / (n-r)!$, sedangkan combination menghitung kombinasi dengan rumus permutasi dibagi $r!$. Dalam fungsi main, pengguna diminta memasukkan empat nilai (a, b, c, dan d). Jika $a \geq c$ dan $b \geq d$, maka program menghitung dan menampilkan permutasi serta kombinasi dari nilai-nilai tersebut untuk pasangan (a, c) dan (b, d). Jika kondisi tidak terpenuhi, pesan kesalahan akan ditampilkan.

2. 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.

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"
)

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

```

PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4> go run "d:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4\unguided2.go"
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294

```

Deskripsi Program

Kode di atas membaca input nama peserta dan waktu pengerjaan soal menggunakan bufio.Reader. Setiap peserta memiliki beberapa waktu pengerjaan soal, dan program menghitung jumlah soal yang diselesaikan dalam waktu maksimal 300 detik menggunakan fungsi hitungSkor. Program membandingkan jumlah soal yang diselesaikan dan total waktu untuk menentukan pemenang, dengan prioritas jumlah soal terbanyak dan waktu terkecil jika jumlah soal sama. Jika ditemukan pemenang, program mencetak nama pemenang, jumlah soal yang diselesaikan, dan total waktu; jika tidak ada peserta, program menampilkan pesan "Tidak ada peserta."

3. 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 $\frac{1}{2}n$, 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"
)

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

```
Input tidak valid. Harap masukkan bilangan bulat positif yang lebih kecil
PS D:\ITTP\Semester 3\Alpro 2\2311102010_Rakha Yudhistira_Modul_4> go run
dhistira_Modul_4\unguided3.go
Masukkan nilai suku awal (n): 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Deskripsi Program

Kode di atas mendefinisikan fungsi cetakDeret yang mencetak deret Collatz, di mana setiap bilangan n akan dibagi dua jika genap, atau diubah menjadi $3*n + 1$ jika ganjil, hingga mencapai angka 1. Dalam fungsi main, pengguna diminta memasukkan nilai awal n , dan jika n adalah bilangan positif yang kurang dari 1 juta, program akan memanggil fungsi cetakDeret untuk mencetak deret tersebut. Jika input tidak valid, program akan menampilkan pesan kesalahan.