

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROSEDUR**



Disusun Oleh :

Rafi Bintang Maulana / 2311102327

Kelas IF-11-06

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Prosedur dalam pemrograman adalah blok kode yang dikelompokkan untuk menjalankan tugas tertentu dan dapat dipanggil berkali-kali di berbagai tempat dalam program. Prosedur biasanya digunakan untuk menghindari pengulangan kode yang sama dan meningkatkan modularitas serta keterbacaan program. Di beberapa bahasa pemrograman, prosedur disebut juga dengan istilah function atau subroutine. Prosedur dapat menerima parameter sebagai input, menjalankan instruksi tertentu, dan memberikan hasil atau melakukan aksi tanpa mengembalikan nilai.

Keuntungan utama dari penggunaan prosedur adalah pemeliharaan dan pengorganisasian kode yang lebih baik. Dengan membagi program menjadi prosedur-prosedur yang lebih kecil, programmer dapat memusatkan perhatian pada satu tugas tertentu dalam setiap prosedur tanpa harus memperhatikan keseluruhan program. Selain itu, prosedur memudahkan pengujian dan debugging, karena kesalahan bisa dilokalisasi dalam blok yang lebih kecil. Prosedur juga mendukung reusability (penggunaan ulang kode) sehingga dapat mengurangi pengulangan instruksi dalam kode.

II. GUIDED

1. 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

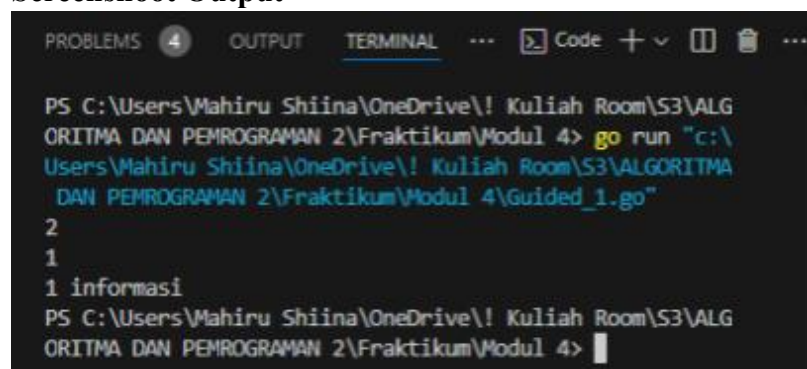
```
package main

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

Screenshot Output



```
PROBLEMS 4 OUTPUT TERMINAL ... Code + - [ ] [ ] ...

PS C:\Users\Mahiru Shiina\OneDrive\! Kuliah Room\S3\ALG
ORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 4> go run "c:\
Users\Mahiru Shiina\OneDrive\! Kuliah Room\S3\ALGORITMA
DAN PEMROGRAMAN 2\Fraktikum\Modul 4\Guided_1.go"
2
1
1 informasi
PS C:\Users\Mahiru Shiina\OneDrive\! Kuliah Room\S3\ALG
ORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 4> |
```

Deskripsi Program

Program ini menerima dua input dari pengguna: sebuah bilangan integer dan sebuah pesan string. Program kemudian memanggil fungsi “**cetakpesan**” dengan kedua nilai tersebut. Fungsi “**cetakpesan**” menentukan jenis pesan berdasarkan nilai bilangan. Setelah itu, program mencetak pesan yang diberikan beserta jenisnya.

2. Pengirim Notifikasi Email

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)
    }
}
```

Screenshot Output



```
PS C:\Users\Mahiru Shiina\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 4>
ORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 4\Guided_2.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 C:\Users\Mahiru Shiina\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 4>
```

Deskripsi Program

Program mendefinisikan sebuah fungsi “**sendEmailNotification**” yang menerima parameter berupa alamat email dan menampilkan pesan konfirmasi bahwa email telah dikirim. Di dalam fungsi “**main**”, terdapat sebuah slice “**emails**” yang berisi daftar alamat email pengguna. Menggunakan perulangan “**for**” untuk iterasi melalui setiap alamat email dalam slice tersebut dan memanggil fungsi “**sendEmailNotification**” untuk mengirim notifikasi ke masing-masing alamat.

3. Perhitungan dan Penggunaan Fungsi Return dan Pointer

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)
}
```

Screenshot Output

Deskripsi Program

Program ini untuk mengilustrasikan penggunaan fungsi dengan pengembalian nilai dan penggunaan pointer untuk mengubah nilai variabel dalam fungsi. Di dalam fungsi “**main**”, program meminta pengguna memasukkan dua bilangan integer, lalu memanggil “**f2**” untuk menyimpan hasil perhitungan ke variabel “**c**” menggunakan pointer. Setelah itu, program juga memanggil “**f1**” untuk menghitung nilai yang sama namun dengan urutan parameter yang dibalik dan mencetak hasil dari kedua fungsi.

III. 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 kalian membantu Jonas?

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 .

Sourcecode

```
package main

import (
    "fmt"
)

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

func RumusPermutasi(n, r int) int {
    return RumusFaktorial(n) / RumusFaktorial(n-r)
}

func RumusKombinasi(n, r int) int {
    return RumusFaktorial(n) / (RumusFaktorial(r) * RumusFaktorial(n-r))
}

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

    fmt.Print("Masukkan empat bilangan (a b c d): ")
    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

    if a >= c && b >= d {

        fmt.Printf("=====
=====\\n")

        pa := RumusPermutasi(a, c)
        ca := RumusPermutasi(a, c)
        fmt.Printf("Permutasi(a dan c): %d, Kombinasi(a dan c): %d\\n", pa, ca)
```

```

        fmt.Printf("-----\n")
        pb := RumusPermutasi(b, d)
        cb := RumusKombinasi(b, d)
        fmt.Printf("Permutasi(b, d): %d, Kombinasi(b, d): %d\n", pb, cb)

        fmt.Printf("=====\n")
    } else {
        fmt.Println("Input Salah. Pastikan a >= c dan b >= d.")
    }
}

```

Screenshoot Output

```

PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum
\Modul 4> go run "c:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRA
MAN 2\Fraktikum\Modul 4\Unguided_1.go"
Masukkan empat bilangan (a b c d): 5 10 3 10
=====
Permutasi(a dan c): 60, Kombinasi(a dan c): 60
-----
Permutasi(b, d): 3628800, Kombinasi(b, d): 1
=====
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum
\Modul 4>

```

Deskripsi Program

Program ini menghitung permutasi dan kombinasi dari dua pasang bilangan dengan menggunakan fungsi rekursif untuk menghitung faktorial. Program memiliki tiga fungsi. Sesuai namanya, Fungsi “**RumusFaktorial**” untuk menghitung faktorial suatu bilangan secara rekursif, “**RumusPermutasi**” untuk menghitung permutasi menggunakan faktorial, dan “**RumusKombinasi**” untuk menghitung kombinasi.

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 di dalam prosedur.

```

procedure 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"
    "strings"
)

func HitungSkor(waktu []int) (int, int) {
    totalPenyelesaian := 0
    totalWaktu := 0
    for _, w := range waktu {
        if w <= 300 {
            totalPenyelesaian++
            totalWaktu += w
        }
    }
    return totalPenyelesaian, totalWaktu
}

func main() {
    pemindai := bufio.NewScanner(os.Stdin)
    var namaPemenang string
    maksPenyelesaian := 0
    minWaktu := 999999
    adaPesertaMenyelesaikan := false

    for {
        pemindai.Scan()
        masukan := pemindai.Text()

        if masukan == "Selesai" {
            break
        }

        bagian := strings.Fields(masukan)
        nama := bagian[0]
        waktu := make([]int, len(bagian)-1)
```



```

        for i := 1; i < len(bagian); i++ {
            fmt.Sscanf(bagian[i], "%d", &waktu[i-1])
        }

        totalPenyelesaian, totalWaktu := HitungSkor(waktu)

        if totalPenyelesaian > 0 {
            adaPesertaMenyelesaikan = true
        }

        if totalPenyelesaian > maksPenyelesaian || (totalPenyelesaian ==
maksPenyelesaian && totalWaktu < minWaktu) {
            namaPemenang = nama
            maksPenyelesaian = totalPenyelesaian
            minWaktu = totalWaktu
        }
    }

    if adaPesertaMenyelesaikan {
        fmt.Printf("%s %d %d\n", namaPemenang, maksPenyelesaian,
minWaktu)
    } else {
        fmt.Println("Tidak ada peserta yang menyelesaikan masalah dalam
batas waktu.")
    }
}

```

Screenshoot Output

```

Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294
PS C:\Users\Rinn chocomint\OneDrive\! Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 4>

```

Deskripsi Program

Program ini berguna untuk menghitung pemenang dari sebuah kompetisi berdasarkan jumlah soal yang diselesaikan dan waktu yang dibutuhkan. dimulai dengan membaca input dari pengguna menggunakan “**bufio.Scanner.**” Input berisi nama peserta diikuti dengan daftar waktu penyelesaian soal. program mengolah waktu penyelesaian soal peserta dengan memanggil fungsi `HitungSkor()`, yang menghitung berapa banyak soal yang diselesaikan dalam waktu kurang dari atau sama dengan 300 detik serta menjumlahkan total waktu yang dibutuhkan untuk menyelesaikan soal tersebut. Proses ini berulang sampai pengguna memasukkan "Selesai," yang menandakan akhir input.

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 $1/2n$, 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. 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 selalu mencapai suku dengan nilai 1.

Buat program **sklena** yang akan mencetak setiap suku dari deret yang dijelaskan di atas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetak Deret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal.

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.Printf("%d ", n)
        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Println(1)
}

func main() {
    var input int
    fmt.Scan(&input)

    if input > 0 && input < 1000000 {
        CetakDeret(input)
    } else {
        fmt.Println("Masukan harus bilangan positif kurang dari 1000000")
    }
}
```

Screenshoot Output



```
PS C:\Users\Rinn chocomint\OneDrive\Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 4> go run "c:\Users\Rinn chocomint\OneDrive\Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 4\Unguided_3.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\Rinn chocomint\OneDrive\Kuliah Room\S3\ALGORITMA DAN PEMROGRAMAN 2\Fraktikum\Modul 4>
```

Deskripsi Program

Program ini mencetak deret bilangan, Deret dimulai dari bilangan bulat positif “ n ”, di mana jika “ n ” genap, suku berikutnya adalah “ $n/2$ ”, dan jika “ n ” ganjil, suku berikutnya adalah “ $3n + 1$ ”. Proses ini diulangi hingga suku mencapai nilai 1. Program menerima input bilangan dari pengguna, memastikan bahwa input tersebut adalah bilangan positif kurang dari 1.000.000, dan kemudian mencetak deret tersebut dalam satu baris dengan setiap suku dipisahkan oleh spasi.