

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL IV

Peosedur



Disusun Oleh :

Alfin Adriansyah/ 2311102264

S1IF_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 serangkaian instruksi yang dikelompokkan bersama untuk mencapai tujuan tertentu. Dengan menggunakan prosedur, pengembang dapat memecah kode menjadi bagian-bagian yang lebih kecil dan lebih terkelola. Setiap prosedur dapat memiliki input dan output yang memungkinkan interaksi dengan bagian-bagian lain dari program. Dengan menggunakan prosedur, pengembang dapat meningkatkan efisiensi dan keterbacaan kode, serta memfasilitasi pemeliharaan dan penggunaan kembali kode yang sama di berbagai bagian programan.

Perbedaan fungsi dan prosedur :

Prosedur (dalam pengertian umum) biasanya digunakan hanya untuk menjalankan sebuah aksi, seperti mencetak sesuatu di layar, mengubah nilai variabel, atau memanggil fungsi lain.

Fungsi (dalam pengertian umum) digunakan ketika Anda memerlukan perhitungan atau proses yang menghasilkan nilai yang akan digunakan lebih lanjut dalam program.

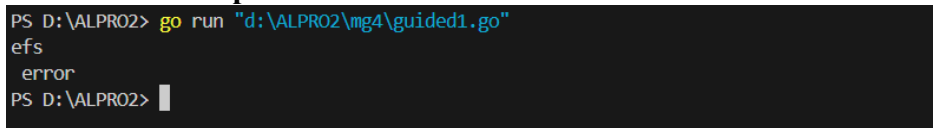
II. GUIDED

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)
}
```

Screenshoot Output



```
PS D:\ALPRO2> go run "d:\ALPRO2\mg4\guided1.go"
efs
error
PS D:\ALPRO2>
```

Deskripsi Program

Program di atas menerima input berupa bilangan dan pesan, lalu mencetak pesan tersebut disertai jenis status (error, warning, atau informasi) berdasarkan nilai bilangan yang diinput.

2. Soal studi kasus
Mencoba mengirim pesan atau notification

```

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)

    }

}

```

Screenshot Output

```

PS D:\ALPRO2> go run "d:\ALPRO2\mg4\guided2.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.
PS D:\ALPRO2>

```

Deskripsi Program

Program tersebut merupakan aplikasi sederhana yang mengirimkan notifikasi email kepada pengguna yang baru terdaftar. Program ini menyimpan daftar email dalam array `emails` dan kemudian menggunakan loop untuk memanggil fungsi `sendEmailNotification` untuk setiap email dalam daftar. Fungsi tersebut mencetak pesan ke konsol yang

mensimulasikan pengiriman notifikasi pendaftaran berhasil ke setiap alamat email yang ada di daftar.

3. 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)

```
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.Printf("Enter two integers: ")

    fmt.Scan(&a, &b)

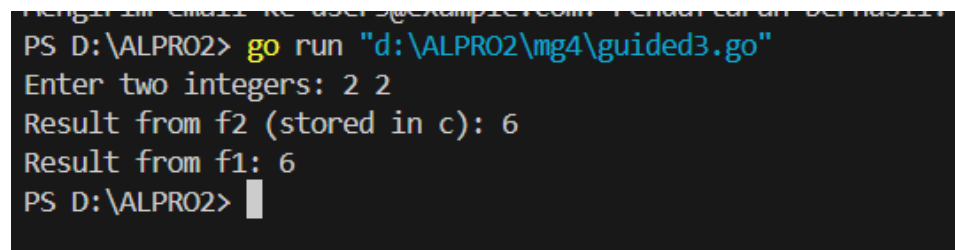
    f2(a, b, &c)

    fmt.Println("Result from f2 (stored in
c):", c)
```

```
    resultF1 := f1(b, a)

    fmt.Println("Result from f1:", resultF1)
}
```

Screenshot Output



```
PS D:\ALPRO2> go run "d:\ALPRO2\mg4\guided3.go"
Enter two integers: 2 2
Result from f2 (stored in c): 6
Result from f1: 6
PS D:\ALPRO2>
```

Deskripsi Program

Program ini adalah aplikasi sederhana dalam Go yang meminta dua bilangan bulat dari pengguna dan menghitung hasilnya menggunakan dua fungsi, `f1` dan `f2`, dengan rumus yang sama: $(2x - 0.5y + 3.0)$. Fungsi `f1` mengembalikan hasil langsung, sedangkan `f2` menyimpan hasilnya dalam variabel yang dipassing melalui pointer. Program ini kemudian mencetak hasil dari kedua fungsi, dengan hasil dari `f2` disimpan dalam variabel `c`, dan hasil dari `f1` ditampilkan setelah pemanggilan.

III. UNGUIDED

1. Minggu ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika

an 44 | Modul Praktikum Algoritma dan Pemrograman 2

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"
)

func factorial(n int) int {
    f := 1
    for i := 2; i <= n; i++ {
        f *= i
    }
    return f
}

func permComb(n, r int) (int, int) {
    if n < r {
        return 0, 0
    }
    perm := factorial(n) / factorial(n-r)
    comb := perm / factorial(r)
    return perm, comb
}

func main() {
    var a, b, c, d int
    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

    Pac, Cac := permComb(a, c)
    Pbd, Cbd := permComb(b, d)

    fmt.Println(Pac, Cac)
```

```
        fmt.Println(Pbd, Cbd)
    }
```

Screenshot Output

```
PS D:\ALPRO2> go run "d:\ALPRO2\mg4\unguided\unguided1.go"
5 10 3 10
60 10
3628800 1
PS D:\ALPRO2> |
```

Deskripsi Program

Program ini menghitung permutasi dan penggabungan dua bilangan genap yang dimasukkan oleh pengguna. Pengguna memasukkan empat angka: "a", "b", "c" dan "d". Program menghitung permutasi dan kombinasi "a" untuk "c" dan "b" untuk "d" menggunakan fungsi penugasan sederhana. Hasil persilangan dan kombinasi setiap pasangan angka dicetak pada dua baris terpisah, baris pertama menunjukkan hasil "a" dan "c", baris kedua menunjukkan hasil "b" dan "d".

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.

Sourcecode

```
package main

import (
    "fmt"
)

func hitungSkor(waktu []int, jumlahSoal *int, totalSkor *int) {
    *jumlahSoal = 0
    *totalSkor = 0
    for _, t := range waktu {
        if t < 301 {
```



```

        *jumlahSoal++
        *totalSkor += t
    }
}

func main() {
    var pemenang string
    var maxSoal, minSkor int
    maxSoal = 0
    minSkor = 999999

    for {
        var namaPeserta string
        var waktuPenyelesaian [8]int

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

        if namaPeserta == "Selesai" || namaPeserta == "selesai" {
            break
        }

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

        var jumlahSoal, totalWaktu int
        hitungSkor(waktuPenyelesaian[:], &jumlahSoal, &totalWaktu)

        if jumlahSoal > maxSoal || (jumlahSoal == maxSoal && totalWaktu < minSkor) {
            pemenang = namaPeserta
            maxSoal = jumlahSoal
            minSkor = totalWaktu
        }

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

```

Screenshoot Output

```
Pemenang: Baresti dengan 7 soal diselesaikan dan total waktu 294 menit
PS D:\ALPRO2> go run "d:\ALPRO2\mg4\unguided\unguided2.go"
Masukkan nama dan waktu penyelesaian ('Selesai' untuk berhenti): Astuti
20 50 301 301 61 71 75 10
Masukkan nama dan waktu penyelesaian ('Selesai' untuk berhenti): Bertha
25 47 301 26 50 60 65 21
Masukkan nama dan waktu penyelesaian ('Selesai' untuk berhenti): Selesai
Pemenang: Bertha dengan 7 soal diselesaikan dan total waktu 294 menit
PS D:\ALPRO2> █
```

Deskripsi Program

Program ini menghitung skor dan jumlah soal yang diselesaikan oleh peserta dalam sebuah kompetisi. Pengguna diminta untuk memasukkan nama peserta dan waktu penyelesaian untuk masing-masing dari delapan soal. Program akan terus menerima input hingga peserta mengetik "Selesai". Setiap waktu yang kurang dari 301 detik dihitung sebagai soal yang berhasil diselesaikan, dan total waktu dihitung untuk menentukan pemenang. Pemenang adalah peserta dengan jumlah soal terbanyak yang diselesaikan, dan jika ada yang sama, pemenang ditentukan berdasarkan total waktu penyelesaian terendah. Jika tidak ada peserta yang mendaftar, program akan menampilkan pesan bahwa tidak ada peserta.

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 $\frac{1}{2}n$, 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

Sourcecode

```
package main

import "fmt"

func cetakDeret(n int) {
    for n != 1 {
        fmt.Printf("%d ", n)
        if n%2 == 0 {
            n = n / 2
        } else {

```

```

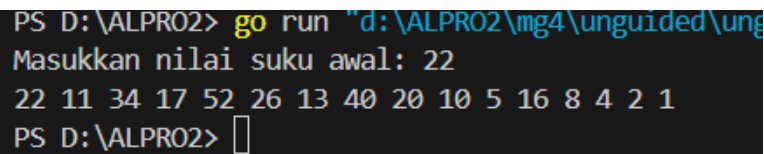
        n = 3*n + 1
    }
    }
    fmt.Printf("1\n")
}

func main() {
    var n int
    fmt.Print("Masukkan nilai suku awal: ")
    fmt.Scan(&n)

    if n > 0 && n < 1000000 {
        cetakDeret(n)
    } else {
        fmt.Println("Nilai harus antara 1 dan
1000000")
    }
}

```

Screenshoot Output



```

PS D:\ALPRO2> go run "d:\ALPRO2\mg4\unguided\ung
Masukkan nilai suku awal: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\ALPRO2>

```

Deskripsi Program

Program ini menghitung dan mencetak deret berdasarkan **Collatz Conjecture** dimulai dari bilangan bulat positif `n`. Jika `n` genap, nilainya dibagi dua; jika ganjil, nilainya dikalikan tiga dan ditambah satu, berulang hingga deret mencapai angka 1. Program meminta input bilangan positif yang lebih kecil dari 1.000.000, lalu mencetak deret tersebut dalam satu baris dengan elemen dipisahkan spasi. Jika input tidak valid, program akan menampilkan pesan kesalahan.