

LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2
MODUL IV
PROSEDUR



Oleh :

Geranada Saputra Priambudi

2311102008

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi, S.Kom., M.Kom.

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY

I. DASAR TEORI

Definisi Prosedure

Prosedur dapat dianggap sebagai pembentukan suatu instruksi baru yang dibuat untuk mempermudah pemahaman algoritma program yang lebih besar. Kedudukannya sama seperti instruksi dasar yang sudah ada sebelumnya (assignment) dan/atau instruksi yang berasal dari paket (fmt), seperti fmt.Scan dan fmt.Print. Karena itu selalu pilih nama prosedur yang berbentuk kata kerja atau sesuatu yang merepresentasikan proses sebagai nama dari prosedur. Go hanya mempunyai kata kunci func untuk deklarasi fungsi dan prosedur. Perbedaan keduanya ada pada:

- Ada/tidak ada deklarasi tipe nilai yang dikembalikan, dan
- Digunakan / tidak digunakan kata kunci return dalam badan subprogram tersebut.

Parameter

Berdasarkan letak penulisannya pada program, maka parameter dapat dikelompokkan menjadi dua yaitu parameter formal dan parameter aktual. Perbedaan parameter aktual dengan parameter formal:

- Parameter aktual
Parameter aktual adalah parameter (tidak selalu berupa variabel) yang dipakai dalam pemanggilan fungsi. Parameter aktual ini akan menggantikan parameter formal dan digunakan dalam eksekusi fungsi.
- Parameter formal
Parameter formal adalah variabel yang ada pada daftar parameter dalam definisi fungsi. Parameter ini hanya merupakan "placeholder" yang akan diisi dengan nilai dari parameter aktual ketika fungsi tersebut dipanggil. Parameter formal memiliki tipe data yang spesifik, yang berarti harus digunakan sesuai dengan tipe data yang telah dideklarasikan. Pada contoh program di atas misalnya, maka dalam fungsi jumlah() variabel x dan y dinamakan sebagai parameter formal.

Selain itu parameter juga dikelompokkan berdasarkan alokasi memorinya yaitu *pass by value* dan *pass by reference*.

- *Pass by Value*
Pass by Value adalah metode di mana nilai dari parameter aktual disalin ke dalam variabel lokal (parameter formal) pada subprogram. Ini berarti parameter aktual dan formal berada di lokasi memori yang berbeda. Akibatnya, perubahan yang terjadi pada parameter formal tidak mempengaruhi nilai dari parameter aktual di luar fungsi atau prosedur. Dalam Go, semua parameter fungsi secara *default* adalah *pass by value*. Ketika sebuah nilai dikirimkan ke fungsi, fungsi tersebut hanya bekerja dengan salinan dari nilai tersebut. Ini berguna karena tidak ada risiko fungsi tersebut mengubah nilai asli dari parameter.
- *Pass by Reference (Pointer)*
Pass by Reference adalah metode di mana alamat memori dari parameter aktual diberikan ke parameter formal. Dengan demikian, parameter formal berperan sebagai *pointer* yang menunjuk ke lokasi memori yang sama dengan parameter

aktual. Oleh karena itu, setiap perubahan yang dilakukan pada parameter formal akan mempengaruhi langsung nilai dari parameter aktual. Dalam Go, *pass by reference* dicapai dengan menggunakan *pointer*. Parameter formal yang menjadi *pointer* akan ditandai dengan simbol * (*asterisk*), dan untuk melewati alamat memori digunakan simbol & (*ampersand*).

II. GUIDED

1. Source Code:

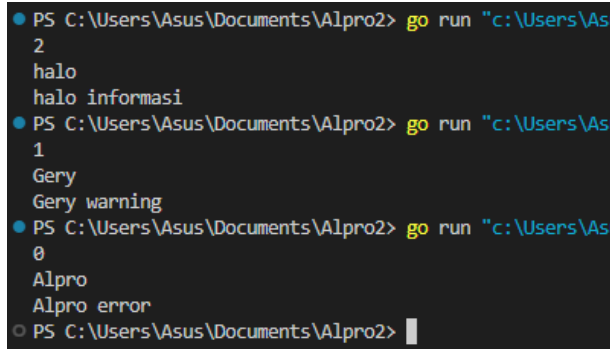
```
package main

import "fmt"

func main() {
    var bilangan int
    var pesan string
    fmt.Scan(&bilangan)
    fmt.Scan(&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)
}
```

Output:



```
PS C:\Users\Asus\Documents\Alpro2> go run "c:\Users\Asus\Documents\Alpro2\main.go"
2
halo
halo informasi
PS C:\Users\Asus\Documents\Alpro2> go run "c:\Users\Asus\Documents\Alpro2\main.go"
1
Gery
Gery warning
PS C:\Users\Asus\Documents\Alpro2> go run "c:\Users\Asus\Documents\Alpro2\main.go"
0
Alpro
Alpro error
PS C:\Users\Asus\Documents\Alpro2>
```

Penjelasan:

Program diatas berfungsi untuk menampilkan pesan dengan jenis yang ditentukan berdasarkan nilai input bilangan yang diberikan. Pertama, program meminta pengguna memasukkan dua input: bilangan dan pesan. Berdasarkan nilai bilangan tersebut, program akan mengklasifikasikan pesan menjadi salah satu dari tiga jenis: "error" jika nilainya 0, "warning" jika nilainya 1, dan "informasi" jika nilainya 2. Fungsi cetakPesan bertugas untuk menggabungkan pesan yang dimasukkan dengan jenis pesan yang sesuai dan menampilkannya. Algoritma ini bekerja dengan menggunakan kondisi if untuk menentukan jenis pesan yang akan ditampilkan, dan akhirnya mengeluarkan output pesan yang disesuaikan sesuai dengan input

bilangan. Output yang dihasilkan terdiri dari pesan pengguna yang diikuti dengan jenis pesannya.

2. Source Code:

```
package main

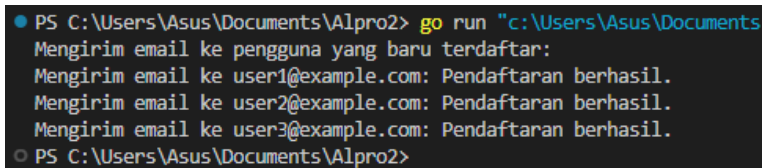
import "fmt"

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)
    }
}
```

Output:



```
PS C:\Users\Asus\Documents\Alpro2> go run "c:\Users\Asus\Documents\
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\Asus\Documents\Alpro2>
```

Penjelasan:

Program diatas berfungsi untuk mengirim notifikasi pendaftaran melalui email kepada beberapa pengguna yang baru terdaftar. Program dimulai dengan mendeklarasikan sebuah slice yang berisi daftar alamat email dari tiga pengguna. Fungsi sendEmailNotification digunakan untuk mencetak pesan konfirmasi bahwa email berhasil dikirim ke setiap alamat. Dalam fungsi main, program menampilkan pesan awal, kemudian menggunakan perulangan for untuk mengiterasi setiap email dalam slice. Setiap kali perulangan terjadi, fungsi sendEmailNotification dipanggil dengan argumen email saat itu untuk mencetak pesan bahwa email pendaftaran berhasil dikirim. Output yang dihasilkan adalah pesan konfirmasi yang menyebutkan bahwa email pendaftaran berhasil dikirim ke masing-masing alamat email yang terdaftar.

3. Source Code:

```
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)

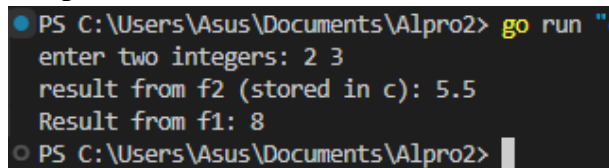
    f2(a, b, &c)

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

    resultF1 := f1(b, a)
    fmt.Println("Result from f1:", resultF1)
}

```

Output:



```

PS C:\Users\Asus\Documents\Alpro2> go run "C:\Users\Asus\Documents\Alpro2\main.go"
enter two integers: 2 3
result from f2 (stored in c): 5.5
Result from f1: 8
PS C:\Users\Asus\Documents\Alpro2>

```

Penjelasan:

Program diatas menghitung hasil dari dua fungsi matematika berbeda yang memanipulasi dua bilangan bulat yang dimasukkan pengguna. Fungsi f1 mengembalikan hasil perhitungan $(2 * x) - 0.5 * y + 3.0$, sedangkan fungsi f2 menggunakan pointer untuk menyimpan hasil perhitungan yang sama ke variabel yang ditunjuk. Di dalam fungsi main, pengguna diminta untuk memasukkan dua bilangan bulat yang kemudian diproses oleh fungsi f2, yang hasilnya disimpan dalam variabel c dan dicetak. Setelah itu, fungsi f1 dihitung dengan urutan parameter yang berbeda, dan hasilnya juga dicetak. Output program menunjukkan hasil perhitungan 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 kalla 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!

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

Source Code:

```
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: a harus >= c dan b harus
>= d")
    }
}

```

Output:

```

PS C:\Users\Asus\Documents\Alpro2> go run "c:\Us
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 C:\Users\Asus\Documents\Alpro2> go run "c:\Us
go"
Masukkan input (a, b, c, d) = 8 0 2 0
Permutasi(8, 2) = 56
Kombinasi(8, 2) = 28
Permutasi(0, 0) = 1
Kombinasi(0, 0) = 1

```

Penjelasan:

Program diatas untuk menghitung permutasi dan kombinasi dari dua pasang bilangan berdasarkan input yang diberikan oleh pengguna. Pertama, program meminta empat bilangan asli, yaitu a, b, c, dan d, dengan syarat bahwa a harus lebih besar atau sama dengan c, dan b harus lebih besar atau sama dengan d. Dalam program ini, prosedur faktorial digunakan untuk menghitung nilai faktorial dari suatu bilangan, yang merupakan langkah penting dalam menghitung permutasi dan kombinasi. Prosedur hitungPermutasi digunakan untuk menghitung dan mencetak hasil permutasi dari a terhadap c, menggunakan rumus $P(n, r) = n! / (n-r)!$, sedangkan prosedur hitungKombinasi digunakan untuk menghitung dan mencetak hasil kombinasi dari a terhadap c serta b terhadap d, sesuai dengan rumus $C(n, r) = n! / (r!(n-r)!)$. Setelah menerima input, program memeriksa apakah syarat yang ditentukan terpenuhi. Jika terpenuhi, program akan mengeksekusi kedua prosedur tersebut untuk menghitung dan menampilkan hasilnya; jika tidak, program akan menampilkan pesan kesalahan.

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 didalam prosedur.

prosedure 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.

Source Code:

```
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)

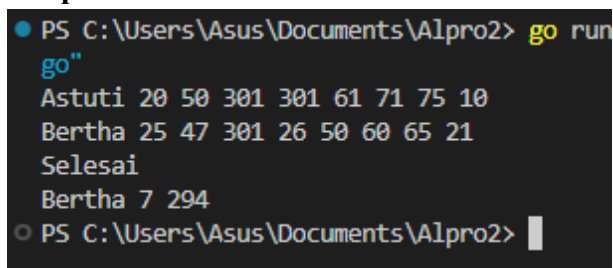
    for _, w := range input[1:] {
        waktu, err := strconv.Atoi(w)
        if err == nil {
            waktuPeserta = append(waktuPeserta, waktu)
        }
    }

    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.")
}
}

```

Output:



```

PS C:\Users\Asus\Documents\Alpro2> go run go
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\Asus\Documents\Alpro2>

```

Penjelasan:

Program diatas untuk menentukan pemenang dari kompetisi pemrograman tingkat nasional dengan mengevaluasi performa peserta dalam menyelesaikan soal. Peserta diberikan delapan soal yang harus diselesaikan dalam waktu lima jam (300 menit). Program menggunakan pendekatan modular dengan prosedur hitungSkor yang bertugas menghitung jumlah soal yang berhasil diselesaikan dan total waktu yang

dihabiskan oleh peserta. Dalam prosedur ini, waktu pengerjaan untuk setiap soal dibaca dan jika seorang peserta tidak menyelesaikan soal dalam waktu yang ditentukan, waktu tersebut dianggap sebagai 301 menit. Dalam fungsi utama, program menerima input nama peserta diikuti dengan waktu yang dihabiskan untuk menyelesaikan setiap soal, hingga menerima input "Selesai" untuk menandai akhir data peserta. Program kemudian membandingkan jumlah soal yang diselesaikan dan total waktu di antara peserta untuk menentukan pemenang. Peserta dengan jumlah soal terbanyak dan waktu terpendek dinyatakan sebagai pemenang, dan hasilnya dicetak ke layar. Jika tidak ada peserta yang terdaftar, program akan memberikan pesan bahwa tidak ada peserta.

Pada contoh di atas: Peserta Astuti menyelesaikan soal dalam waktu 20, 50, dan waktu tidak sah (301) untuk dua soal. Peserta Bertha menyelesaikan soal dalam waktu 25, 47, 26, 50, 60, 65, dan waktu tidak sah (301). Bertha adalah pemenang yang berhasil menyelesaikan 7 soal dalam total waktu 294 menit. Output mencerminkan peserta yang menyelesaikan soal terbanyak dalam waktu terpendek, sesuai dengan ketentuan kompetisi.

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 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.

prosedure 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.

Source Code:

```
package main

import (
    "fmt"
)

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

```

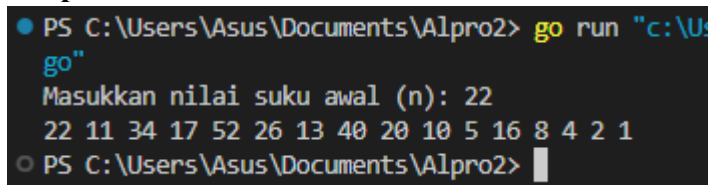
        n = 3*n + 1
    }
}
fmt.Print(n)
}

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.")
    }
}

```

Output:



```

PS C:\Users\Asus\Documents\Alpro2> go run "c:\Users\Asus\Documents\Alpro2\main.go"
Masukkan nilai suku awal (n): 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\Asus\Documents\Alpro2>

```

Penjelasan:

Program diatas adalah program dalam bahasa Go yang mencetak deret bilangan berdasarkan aturan yang ditetapkan oleh Skiena dan Revilla. Deret ini dimulai dari sebuah bilangan bulat positif n , di mana jika n genap, suku berikutnya adalah setengah dari n ($\frac{1}{2}n$), dan jika n ganjil, suku berikutnya adalah $3n + 1$. Proses ini berlanjut hingga suku terakhir bernilai 1. Program ini menggunakan prosedur `cetakDeret` yang menerima parameter n dan mencetak setiap suku dari deret tersebut dalam satu baris, dipisahkan oleh spasi. Cara kerja program dimulai dengan meminta pengguna memasukkan nilai awal n , kemudian program memeriksa validitas input. Jika valid, program akan memanggil prosedur `cetakDeret`, yang akan menghitung dan mencetak deret sesuai aturan hingga mencapai nilai 1. Sebagai contoh, jika pengguna memasukkan nilai 22, program akan menghasilkan output: "22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1", yang mencerminkan setiap suku dalam deret berdasarkan aturan yang telah ditentukan.