

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROSEDUR**



Disusun Oleh :

Avrizal Setyo Aji Nugroho

2311102145

IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Prosedur dalam bahasa pemrograman adalah subprogram atau blok kode yang memiliki tugas tertentu dan dapat dipanggil dari berbagai bagian program. Di beberapa bahasa lain, prosedur disebut juga sebagai fungsi tanpa nilai balik. Pada dasarnya, prosedur digunakan untuk mengurangi duplikasi kode, mempermudah pemeliharaan, dan meningkatkan keterbacaan program.

Dalam Go, prosedur secara teknis tidak ada sebagai konsep yang terpisah dari fungsi, tetapi kita dapat membentuk prosedur dengan menggunakan fungsi tanpa nilai balik (tanpa return) atau dengan memodifikasi nilai melalui pointer.

II. GUIDED

1. Berikut ini adalah contoh penulisan fungsi pada suatu program lengkap.
Buatlah sebuah program beserta fungsi yang digunakan untuk menghitung nilai faktorial dan permutasi. Masukan terdiri dari dua buah bilangan positif a dan b. Keluaran berupa sebuah bilangan bulat yang menyatakan nilai a permutasi b apabila $a \geq b$ atau b permutasi a untuk kemungkinan yang lain.

Source code

```
package main

import "fmt"

// Prosedur untuk menghitung faktorial
func faktorial(n int, hasil *int) {
    *hasil = 1
    for i := 1; i <= n; i++ {
        *hasil *= i
    }
}

// Prosedur untuk menghitung permutasi
func permutasi(n, r int, hasil *int) {
    var faktN, faktNR int
    faktorial(n, &faktN)
    faktorial(n-r, &faktNR)
    *hasil = faktN / faktNR
}

func main() {
    var a, b int
    fmt.Scan(&a, &b)
```

```

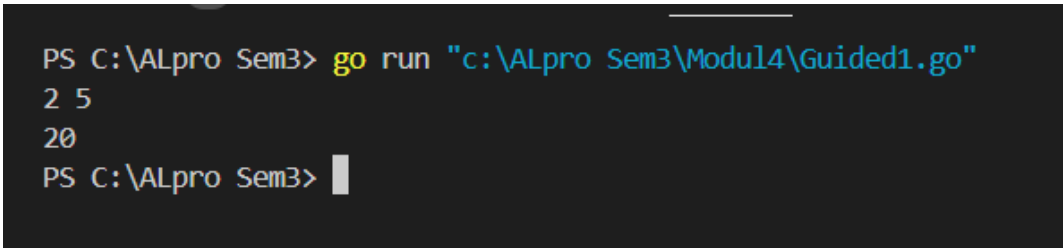
var hasilPermutasi int

if a >= b {
    permutasi(a, b, &hasilPermutasi)
} else {
    permutasi(b, a, &hasilPermutasi)
}

fmt.Println(hasilPermutasi)
}

```

Screenshoot Output



```

PS C:\ALpro Sem3> go run "c:\ALpro Sem3\Modul4\Guided1.go"
2 5
20
PS C:\ALpro Sem3>

```

Deskripsi Program

Program diatas adalah program prosedur yang menerima dua bilangan bulat **a** dan **b** sebagai input, kemudian menghitung dan mencetak hasil permutasi dari dua angka tersebut. Permutasi dihitung dengan rumus ($P(n, r) = \frac{n!}{(n-r)!}$), di mana **n** adalah angka yang lebih besar atau sama dengan **r**. Jika input **a** lebih besar atau sama dengan **b**, maka program menghitung permutasi **P(a, b)**, jika tidak, maka program menghitung **P(b, a)**. Fungsi **faktorial** digunakan untuk menghitung faktorial dari angka, yang kemudian digunakan dalam fungsi **permutasi** untuk menyelesaikan perhitungan permutasi.

2. Membuat program mencari Luas dan Keliling Persegi

Sourcecode

```
package main

import (
    "fmt"
)

// Prosedur untuk menghitung luas
func Luas(s int, L *int) {
    *L = s * s
}

// Prosedur untuk menghitung keliling
func Keliling(s int, K *int) {
    *K = 4 * s
}

func main() {
    var s int
    var L, K int

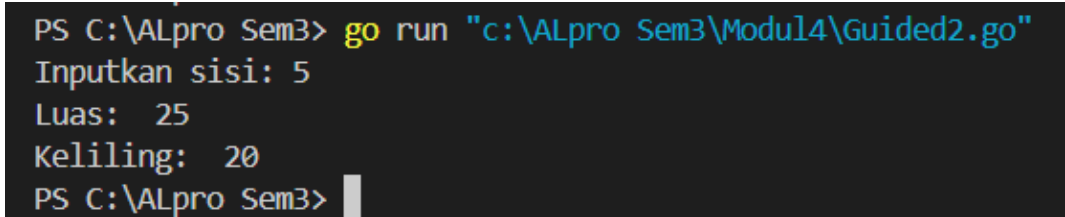
    fmt.Print("Inputkan sisi: ")
    fmt.Scan(&s)

    // Memanggil prosedur untuk menghitung luas
    dan keliling
    Luas(s, &L)
    Keliling(s, &K)

    // Menampilkan hasil luas dan keliling
    fmt.Println("Luas: ", L)
```

```
        fmt.Println("Keliling: ", K)
    }
```

Screenshot Output



```
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\Modul4\Guided2.go"
Inputkan sisi: 5
Luas: 25
Keliling: 20
PS C:\ALpro Sem3>
```

Deskripsi Program

Program di atas menghitung luas dan keliling sebuah persegi berdasarkan input panjang sisi yang diberikan oleh pengguna. Fungsi **Luas(s int)** digunakan untuk menghitung luas persegi dengan rumus ($L = s * s$), sedangkan fungsi **Keliling(s int)** menghitung keliling persegi dengan rumus ($K = 4 * s$). Setelah pengguna memasukkan panjang sisi melalui input, program akan memanggil kedua fungsi tersebut dan mencetak hasil luas dan keliling persegi ke layar.

III. UNGUIDED

1. Soal Studi Case

diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersedia kah 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!

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

Contoh

No	Masukan	Keluaran	Penjelasan
1	5 10 3 10	60 10 3628800 1	$P(5, 3) = 5!/2! = 120/2 = 60$ $C(5, 3) = 5!/(3! \times 2!) = 120/12 = 10$ $P(10, 10) = 10!/0! = 3628800/1 = 3628800$ $C(10, 10) = 10!/(10! \times 0!) = 10!/10! = 1$
2	8 0 2 0	56 28 1 1	

Selesaikan program tersebut dengan memanfaatkan prosedur yang diberikan berikut ini!

```
procedure factorial(in n: integer, in/out hasil:integer)
{I.S. terdefinisi bilangan bulat positif n
 F.S. hasil berisi nilai faktorial dari n}

procedure permutation(in n,r : integer, in/out hasil:integer)
{I.S. terdefinisi bilangan bulat positif n dan r, dan n >= r
 F.S. hasil berisi nilai dari n permutasi r}

procedure combination(in n,r : integer, in/out hasil:integer)
{I.S. terdefinisi bilangan bulat positif n dan r, dan n >= r
 F.S. hasil berisi nilai dari n kombinasi r}
```

Sourcecode

```
package main

import "fmt"

// Prosedur untuk menghitung faktorial
func faktorial(n_145 int, result *int) {
    *result = 1
    for i := 1; i <= n_145; i++ {
        *result *= i
    }
}

// Prosedur untuk menghitung permutasi
func permutasi(n_145, r int, result *int) {
    if n_145 < r {
        *result = 0
    }

    var fact_n_145, fact_n_r int
    faktorial(n_145, &fact_n_145)
    faktorial(n_145-r, &fact_n_r)
    *result = fact_n_145 / fact_n_r
}

// Prosedur untuk menghitung kombinasi
func kombinasi(n_145, r int, result *int) {
    if n_145 < r {
        *result = 0
    }

    var fact_n_145, fact_r, fact_n_r int
    faktorial(n_145, &fact_n_145)
    faktorial(r, &fact_r)
    faktorial(n_145-r, &fact_n_r)
    *result = fact_n_145 / (fact_r * fact_n_r)
}
```



```

func main() {
    var a, b, c, d int
    fmt.Println("Masukkan empat bilangan asli a, b, c,
dan d, dengan syarat a >= c dan b >= d:")
    fmt.Scanln(&a, &b, &c, &d)

    var p1, c1, p2, c2 int

    // Menghitung permutasi dan kombinasi
    permutasi(a, c, &p1)
    kombinasi(a, c, &c1)
    permutasi(b, d, &p2)
    kombinasi(b, d, &c2)

    // Menampilkan hasil
    fmt.Println(p1, c1)
    fmt.Println(p2, c2)
}

```

Screenshoot Output

```

PS C:\ALpro Sem3> go run "c:\ALpro Sem3\Modul4\Unguided1.go"
Masukkan empat bilangan asli a, b, c, dan d, dengan syarat a >= c dan b >= d:
5 10 3 10
60 10
3628800 1
PS C:\ALpro Sem3> 

```

Deskripsi Program

Program di atas adalah program prosedur yang menerima empat bilangan bulat positif **a**, **b**, **c**, dan **d** dengan syarat bahwa **a** >= **c** dan **b** >= **d**, kemudian menghitung serta mencetak hasil permutasi dan kombinasi dari pasangan bilangan tersebut. Permutasi dihitung dengan rumus ($P(n, r) = \frac{n!}{(n-r)!}$), sedangkan kombinasi dihitung dengan rumus ($C(n, r) = \frac{n!}{r!(n-r)!}$). Program ini menggunakan fungsi rekursif `faktorial` untuk menghitung faktorial, serta fungsi *permutation* dan *combination* untuk

menghitung permutasi dan kombinasi. Setelah itu, hasil dari perhitungan permutasi dan kombinasi untuk kedua pasangan angka tersebut dicetak sebagai output.

2. Soal Studi Case

- 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.

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.

No	Masukan	Keluaran
1	Astuti 20 50 301 301 61 71 75 10 Bertha 25 47 301 26 50 60 65 21 Selesai	Bertha 7 294

Keterangan:

Astuti menyelesaikan 6 soal dalam waktu 287 menit, sedangkan Bertha 7 soal dalam waktu 294 menit. Karena Bertha menyelesaikan lebih banyak, maka Bertha menang. Jika keduanya menyelesaikan sama banyak, maka pemenang adalah yang menyelesaikan dengan total waktu paling kecil.

Sourcecode

```
package main

import (
```

```

    "bufio"
    "fmt"
    "os"
    "strings"
)

// Procedure untuk menghitung skor
func hitungSkor_145(soal_145 *int, skor_145 *int,
waktu_145 int) {
    if waktu_145 <= 300 {
        *skor_145 += 1
    }
    *soal_145 += 1
}

func main() {
    var pemenang_145 string
    var soalMax_145, skorMax_145, waktuTotalMax_145 int

    reader := bufio.NewReader(os.Stdin) // Menggunakan
reader untuk input nama

    for {
        fmt.Print("\nMasukkan nama peserta (atau
'Selesai' untuk mengakhiri): ")
        nama_145, _ := reader.ReadString('\n') //
Membaca hingga newline
        nama_145 = strings.TrimSpace(nama_145) // Trim
spasi dan newline

        // Stop the input if the user types "Selesai"
        if nama_145 == "Selesai" {
            break
        }

        var totalSoal_145, skor_145, waktuTotal_145 int

```

```

        for j := 0; j < 8; j++ { // Asumsi ada 8 soal
            var waktu_145 int
            fmt.Printf("Masukkan waktu pengerjaan soal
ke-%d (dalam menit): ", j+1)
            fmt.Scan(&waktu_145)
            fmt.Scanln() // Tambahkan ini untuk flush
input buffer setelah setiap Scan

            // Hitung skor berdasarkan waktu pengerjaan
hitungSkor_145(&totalSoal_145, &skor_145,
waktu_145)

            // Hitung total waktu
waktuTotal_145 += waktu_145
        }

        // Tentukan pemenang berdasarkan skor tertinggi
dan total waktu terendah
        if skor_145 > skorMax_145 || (skor_145 ==
skorMax_145 && waktuTotal_145 < waktuTotalMax_145) {
            pemenang_145 = nama_145
            skorMax_145 = skor_145
            soalMax_145 = totalSoal_145
            waktuTotalMax_145 = waktuTotal_145
        }
    }

    // Output hasil
    fmt.Printf("\n%s %d %d\n", pemenang_145,
soalMax_145, waktuTotalMax_145)
}

```

Screenshoot Output

```

PS C:\ALpro Sem3> go run "c:\ALpro Sem3\Modul4\Unguided2.go"

Masukkan nama peserta (atau 'Selesai' untuk mengakhiri): Astuti
Masukkan waktu pengerjaan soal ke-1 (dalam menit): 20
Masukkan waktu pengerjaan soal ke-2 (dalam menit): 50
Masukkan waktu pengerjaan soal ke-3 (dalam menit): 301
Masukkan waktu pengerjaan soal ke-4 (dalam menit): 301
Masukkan waktu pengerjaan soal ke-5 (dalam menit): 61
Masukkan waktu pengerjaan soal ke-6 (dalam menit): 71
Masukkan waktu pengerjaan soal ke-7 (dalam menit): 75
Masukkan waktu pengerjaan soal ke-8 (dalam menit): 10

Masukkan nama peserta (atau 'Selesai' untuk mengakhiri): Bertha
Masukkan waktu pengerjaan soal ke-1 (dalam menit): 25
Masukkan waktu pengerjaan soal ke-2 (dalam menit): 47
Masukkan waktu pengerjaan soal ke-3 (dalam menit): 301
Masukkan waktu pengerjaan soal ke-4 (dalam menit): 26
Masukkan waktu pengerjaan soal ke-5 (dalam menit): 50
Masukkan waktu pengerjaan soal ke-6 (dalam menit): 60
Masukkan waktu pengerjaan soal ke-7 (dalam menit): 65
Masukkan waktu pengerjaan soal ke-8 (dalam menit): 21

Masukkan nama peserta (atau 'Selesai' untuk mengakhiri): Selesai

Bertha 8 595
PS C:\ALpro Sem3>

```

Deskripsi Program

Program di atas adalah sebuah sistem sederhana untuk menentukan pemenang dalam kompetisi berbasis pengerjaan soal. Setiap peserta diberikan 8 soal, dan waktu yang mereka butuhkan untuk menyelesaikan tiap soal dicatat. Jika peserta menyelesaikan sebuah soal dalam waktu 300 menit atau kurang, mereka mendapatkan skor 1. Program menghitung total soal yang diselesaikan dan skor masing-masing peserta, lalu menentukan pemenang berdasarkan skor tertinggi. Jika terdapat peserta dengan skor yang sama, pemenang dipilih berdasarkan waktu total yang lebih sedikit. Input nama peserta dilakukan hingga pengguna memasukkan "Selesai". Hasil akhir berupa nama pemenang, jumlah soal yang diselesaikan, dan waktu total yang dihabiskan.

3. Soal Studi Case

- 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 selalu mencapai suku dengan nilai 1.

Buat program **skiena** yang akan mencetak setiap suku dari deret yang dijelaskan di atas 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.

No	Masukan	Keluaran
1	22	22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Sourcecode

```
package main

import (
    "fmt"
)

func cetakDeret_145(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 n int

        for {
            fmt.Print("Masukkan satu bilangan integer
positif yang lebih kecil dari 1000000: ")
            fmt.Scan(&n)

            if n > 0 && n < 1000000 {
                break
            } else {
                fmt.Println("Bilangan harus positif dan
lebih kecil dari 1000000")
            }
        }

        cetakDeret_145(n)
    }

```

Screenshoot Output

```

PS C:\ALpro Sem3> go run "c:\ALpro Sem3\Modul4\Unguided3.go"
Masukkan satu bilangan integer positif yang lebih kecil dari 1000000: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\ALpro Sem3> █

```

Deskripsi Program

Program di atas merupakan program Sklena. Program meminta input berupa sebuah bilangan bulat positif yang lebih kecil dari 1.000.000. Setelah menerima input yang valid, program akan mencetak deret bilangan yang dihasilkan dari aturan: jika bilangan genap, bagi dua; jika bilangan ganjil, kalikan tiga lalu tambahkan satu. Proses ini terus diulang hingga mencapai angka 1. Program menggunakan perulangan **for** untuk terus memproses angka hingga mencapai nilai 1, dan setiap bilangan yang dihasilkan selama proses dicetak di layar.