

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROSEDUR**



Disusun Oleh :

Rendi Widya Anggita/2311102278

S1IF-11-06

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Prosedur adalah sekumpulan instruksi atau perintah yang dirancang untuk melakukan tugas tertentu yang dikelompokkan menjadi satu kesatuan. Dalam bahasa Go, prosedur diwakili oleh fungsi, yang memungkinkan programmer untuk menulis kode yang lebih terstruktur dan efisien. Fungsi dapat menerima input dalam bentuk parameter dan menghasilkan output, sehingga memudahkan pemrograman. Dengan menggunakan prosedur, kita dapat menghindari pengulangan kode, dan mempermudah dalam maintenance program.

Deklarasi Prosedur :

Cara mendeklarasikan sebuah prosedur hampir sama seperti fungsi, yaitu menggunakan kata kunci *func*, diikuti dengan nama prosedurnya, kemudian tambahkan parameter di dalam tanda kurung(), parameter bisa lebih dari satu dan harus ada tipe data.

	Notasi Algoritma
1	procedure <nama procedure> (<params>)
2	kamus
3	{deklarasi variabel lokal dari procedure}
4	...
5	algoritma
6	{badan algoritma procedure}
7	...
8	endprocedure
	Notasi dalam bahasa Go
9	func <nama procedure> (<params>) {
10	/* deklarasi variabel lokal dari procedure */
11	...
12	/* badan algoritma procedure */
13	...
14	}

	Notasi Algoritma
1	procedure cetakNFibo(in n : integer)
2	kamus
3	f1, f2, f3, i : integer
4	algoritma
5	f2 ← 0
6	f3 ← 1
7	for i ← 1 to n do
8	output(f3)
9	f1 ← f2
10	f2 ← f3
11	f3 ← f1 + f2
12	endfor
13	endprocedure
	Notasi dalam bahasa Go
14	func cetakNFibo(n int) {
15	var f1, f2, f3 int
16	f2 = 0
17	f3 = 1
18	for i := 1; i <= n; i++ {
19	fmt.Println(f3)
20	f1 = f2
21	f2 = f3
22	f3 = f1 + f2
23	}
24	}

Pemanggilan prosedur :

Cara memanggil prosedur juga sama seperti fungsi, yaitu dengan memanggil nama prosedur beserta parameter atau argumen yang diminta.

	Notasi Algoritma
1	program contohprosedur
2	kamus
3	x : integer
4	algoritma
5	x ← 5
6	cetakNFibo(x) {cara pemanggilan #1}
7	cetakNFibo(100) {cara pemanggilan #2}

8	endprogram
	Notasi dalam bahasa Go
9	func main() {
10	var x int
11	x = 5
12	cetakNFibo(x) {cara pemanggilan #1}
13	cetakNFibo(100) {cara pemanggilan #2}
14	}

Perbedaannya adalah jika fungsi selalu mengembalikan nilai, baik itu tipe data tunggal ataupun beberapa tipe data, sedangkan prosedur tidak mengembalikan nilai apapun.

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.

Masukkan 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 informassi berdasarkan nilai flag 0, 2 dan 2 secara berturut - turut.

Sourcecode

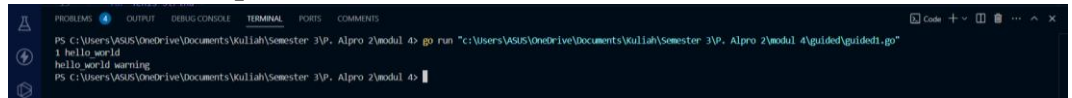
```
package main

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 C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 4> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 4\guided\guided.go"
1 hello world
hello world warning
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 4>
```

Deskripsi Program

Program diatas adalah program sederhana untuk menampilkan sebuah pesan berdasarkan input dari pengguna. Dimana pengguna harus menginputkan satu bilangan bulat(integer) dan satu teks(string). Input pertama akan digunakan sebagai **flag** untuk menentukan jenis pesan(error, warning, atau informasi), input kedua adalah teks yang akan ditampilkan. Jika flag bernilai 0, maka pesan dianggap sebagai **error**, Jika flag bernilai 1, maka pesan dianggap sebagai **warning**, Jika flag bernilai 2, maka pesan dianggap sebagai **informasi**.

2. Soal Studi Case

Sourcecode

```
package main

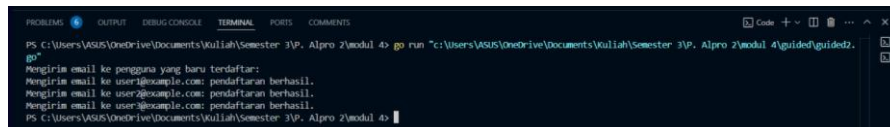
import "fmt"

func sendEmailNotification(email string){
    fmt.Print("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)
    }
}
```

Screenshoot Output



Deskripsi Program

Program tersebut adalah program untuk mengirimkan notifikasi email untuk pengguna yang baru terdaftar, dimana nanti Setiap pengguna yang ada dalam daftar email mendapatkan pesan konfirmasi bahwa proses pendaftaran mereka berhasil.

Program ini memiliki satu fungsi yaitu fungsi sendEmailNotification yang memiliki parameter email yang bertipe data string, yang berisi alamat email. Alamat email tersebut nantinya akan disimpan kedalam array emails. Jika program dijalankan, Setiap pengguna yang ada dalam daftar email mendapatkan pesan konfirmasi bahwa proses pendaftaran mereka berhasil.

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.Println("Masukkan 2 bilangan : ")
    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 C:\Users\VASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 4> go run "c:\Users\VASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 4\guided\guided3.go"
Masukkan 2 bilangan : 2 9
Result from f2 (stored in c): 2.5
Result from f1 : 20
PS C:\Users\VASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 4>
```

Deskripsi Program

Program tersebut adalah program untuk melakukan operasi perhitungan dengan angka yang diinputkan oleh pengguna, program ini memiliki dua fungsi yaitu f1 dan d2 yang masing-masing memiliki output yang berbeda.

Fungsi f1 mengembalikan hasil secara langsung, sedangkan fungsi f2 mengubah dahulu nilai variable diluar fungsi menggunakan pointer. Pada fungsi f1 dan f2 sama-sama memiliki 2 parameter, dan melakukan operasi matematika yang sama, tetapi pada f1 hasil perhitungan dikonversi ke float64 dan kemudian mengembalikan nilai hasilnya dalam tipe float64, sedangkan pada f2 hasilnya akan disimpan dalam variable yang ditunjuk oleh pointer hasil.

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.

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"
)

// Fungsi untuk menghitung faktorial
func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    return n * faktorial(n-1)
}

// Fungsi untuk menghitung permutasi
func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

// Fungsi untuk menghitung kombinasi
func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}

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

    fmt.Println("Masukkan 4 bilangan : ")
    fmt.Scan(&a, &b, &c, &d)

    // Menghitung permutasi dan kombinasi
    pC := permutasi(a, c)
    pD := permutasi(b, d)
    kC := kombinasi(a, c)
    kD := kombinasi(b, d)

    //output
    fmt.Printf("Permutasi P(%d, %d): %d\n", a, c, pC)
    fmt.Printf("Kombinasi C(%d, %d): %d\n", a, c, kC)
    fmt.Printf("Permutasi P(%d, %d): %d\n", b, d, pD)
    fmt.Printf("Kombinasi C(%d, %d): %d\n", b, d, kD)
}
```

Screenshoot Output



```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 4> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 4\unguided\unguided.go"

Masukkan 4 bilangan :
5 10 3 10
Permutasi P(5, 3): 60
Kombinasi C(5, 3): 10
Permutasi P(10, 10): 3628800
Kombinasi C(10, 10): 1
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 4>
```

Deskripsi Program

Program diatas adalah program untuk menghitung permutasi dan kombinasi dari 4 bilangan yang diinputkan oleh pengguna, dengan syarat bilangan pertama harus lebih besar dari atau sama dengan bilangan ke tiga, dan bilangan kedua harus lebih besar dari atau sama dengan bilangan keempat.

Program ini menggunakan tiga fungsi yaitu fungsi faktorial, permutasi dan kombinasi. Kemudian ketika fungsi ini dipanggil pada main program beserta parameternya.

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.

```
prosedure hitungSkor(in/out soal, skor : integer)
```

Setiap baris **masukkan** 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

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 (
    "fmt"
    "strings"
)

// Fungsi hitungSkor untuk menghitung jumlah soal yang
// diselesaikan dan total waktu
func hitungSkor(soal []int) (jumlahSoal int, totalWaktu int) {
    {
        for _, waktu := range soal {
            if waktu <= 300 { // Soal dianggap selesai jika waktu
<= 300 menit
                jumlahSoal++
                totalWaktu += waktu
            }
        }
        return
    }
}

func main() {
    var peserta string
    var jumlahSoal int
    var namaPemenang string
    var maxSoal, minWaktu int

    for {
        // Input nama peserta
        fmt.Print("Masukkan nama peserta: ")
        fmt.Scan(&peserta)
        if strings.ToLower(peserta) == "selesai" {
            break
        }

        // Input jumlah soal yang diselesaikan oleh peserta
        fmt.Printf("Masukkan jumlah soal yang diselesaikan
oleh %s: ", peserta)
        fmt.Scan(&jumlahSoal)

        // Input waktu untuk setiap soal
        soal := make([]int, jumlahSoal) // membuat array
dinamis untuk waktu soal
        fmt.Printf("Masukkan waktu penyelesaian %d soal
(dalam menit) untuk %s (dipisahkan dengan spasi):\n",
jumlahSoal, peserta)
        for i := 0; i < jumlahSoal; i++ {
```

```

        fmt.Scan(&soal[i])
    }

    // Hitung jumlah soal yang diselesaikan dan total waktu
    solvedSoal, totalWaktu := hitungSkor(soal)

    // Tentukan pemenang
    if solvedSoal > maxSoal || (solvedSoal == maxSoal && totalWaktu < minWaktu) {
        namaPemenang = peserta
        maxSoal = solvedSoal
        minWaktu = totalWaktu
    }
}

// Output hasil pemenang
fmt.Printf("Pemenangnya adalah: %s dengan %d soal selesai dan total waktu %d menit\n", namaPemenang, maxSoal, minWaktu)
}

```

Screenshoot Output

```

PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Alpro 2\modul 4> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Alpro 2\modul 4\unguided\unguided2.go"
Masukkan nama peserta: astuti
Masukkan jumlah soal yang diselesaikan oleh astuti: 8
Masukkan waktu penyelesaian 8 soal (dalam menit) untuk astuti (dipisahkan dengan spasi):
20 50 301 301 61 71 75 10
Masukkan nama peserta: berthia
Masukkan jumlah soal yang diselesaikan oleh berthia: 8
Masukkan waktu penyelesaian 8 soal (dalam menit) untuk berthia (dipisahkan dengan spasi):
25 47 301 26 50 60 65 21
Masukkan nama peserta: selesai
Pemenangnya adalah: berthia dengan 7 soal selesai dan total waktu 294 menit
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Alpro 2\modul 4>

```

Deskripsi Program

Program ini adalah simulasi dari kompetisi pemrograman yang bertujuan untuk menentukan pemenang berdasarkan jumlah soal yang diselesaikan dan total waktu yang dihabiskan oleh peserta dalam menyelesaikan soal-soal tersebut. Program akan meminta input dari pengguna berupa nama peserta, jumlah soal yang diselesaikan, dan waktu yang dibutuhkan untuk menyelesaikan setiap soal. Setelah itu, program akan menentukan pemenangnya berdasarkan kriteria berikut:

1. Peserta yang menyelesaikan soal terbanyak.
2. Jika ada peserta yang menyelesaikan jumlah soal yang sama, pemenangnya adalah peserta yang menyelesaikan soal dengan waktu total yang lebih sedikit.

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 $1/4n$, 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.

```
procedure 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(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
    fmt.Print("Masukkan bilangan positif kurang dari
1.000.000: ")
    fmt.Scan(&n)

    if n > 0 && n < 1000000 {
        cetakDeret(n)
    } else {
        fmt.Println("bilangan harus berupa bilangan positif
kurang dari 1.000.000.")
    }
}
```

Screenshoot Output

Deskripsi Program

Program diatas adalah program untuk membuat sebuah deret bilangan yang dimulai dari bilangan bulat n positif kurang dari 1.000.000, kemudian bilangan berikutnya tergantung pada nilai n , apabila nilai n adalah genap maka suku berikutnya adalah $\frac{1}{2}n$, tetapi jika ganjil maka suku berikutnya adalah $3n+1$.

Program ini memiliki satu fungsi yaitu fungsi cetakDeret yang memiliki parameter n dengan tipe data integer, pada fungsi ini merupakan perulangan untuk mencetak deret angka dari n hingga 1, dengan syarat yaitu n bilangan ganjil ataupun n bilangan genap.