LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL IV

Prosedur

Disusun Oleh :

Nadhif Atha Zaki / 2311102007

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

Definisi Prosedur

Prosedur adalah potongan instruksi program yang digunakan untuk menyederhanakan dan mengurangi kerumitan dari suatu kode program. Prosedur ini merupakan bagian penting dalam struktur program besar, karena memungkinkan programmer untuk memecah program menjadi bagian-bagian yang lebih kecil dan lebih mudah dipahami. Prosedur tidak mengembalikan nilai apa pun, tetapi memiliki efek langsung pada program utama saat dipanggil.

Suatu subprogram dapat dikategorikan sebagai prosedur apabila memenuhi kriteria berikut:

1. Tidak ada deklarasi tipe nilai yang dikembalikan, artinya prosedur tidak mengembalikan hasil berupa nilai ke pemanggilnya.
2. Tidak terdapat kata kunci return dalam badan subprogram, yang menandakan bahwa prosedur ini tidak memberikan keluaran dalam bentuk return value, tetapi memberikan efek samping terhadap program utama.

Peran Prosedur dalam Program

Kedudukan prosedur dalam program setara dengan instruksi dasar yang sering digunakan sebelumnya, seperti assignment atau operasi input/output dari paket fmt (misalnya, fmt.Scan dan fmt.Println). Prosedur ini bertujuan untuk merepresentasikan suatu proses atau tugas tertentu dalam program, misalnya seperti cetak, hitungRata, cariNilai, dan sebagainya.

Dengan menggunakan prosedur, programmer dapat:

- Memecah kode yang besar menjadi modul-modul kecil yang lebih mudah dipahami dan dikelola.
- Mengurangi duplikasi kode dengan mengelompokkan instruksi yang sering digunakan menjadi satu prosedur yang dapat dipanggil berkali-kali.
- Membuat program lebih terstruktur dan efisien.

Contoh-contoh prosedur mencakup tindakan yang dapat dilakukan secara independen, seperti mencetak data, menghitung nilai rata-rata, atau melakukan tugas-tugas lain yang tidak mengembalikan nilai ke pemanggilnya, namun memberikan efek langsung pada alur program.

II. GUIDED

1.

Sourcecode

```
package main

import "fmt"

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

    if a >= b {
        var result int
        permutasi(a, b, &result)
        fmt.Print(result)
    } else {
        var result int
        permutasi(b, a, &result)
        fmt.Print(result)
    }
}

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

func permutasi(n, r int, result *int) {
    var faktN, faktNR int
    faktorial(n, &faktN)
    faktorial(n-r, &faktNR)
    *result = faktN / faktNR
}
```

Screenshoot Output

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul - IV> go run "d:\Nathh
2 6
30
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul - IV>
```

Deskripsi Program

Program ini berisi prosedur yang digunakan untuk menghitung nilai faktorial dan permutasi, inputan terdiri dari dua buah bilangan positif a dan b dan output berupa sebuah bilangan bulat yang menyatakan nilai a permutasi b apabila a>= b atau b permutasi a untuk kemungkinan yang lain

2.

Sourcecode

```go
package main

import "fmt"

func luas_persegi(s int) {
    luas := s * s
    fmt.Println("Jadi luas persegi adalah", luas)
}

func keliling_persegi(s int) {
    keliling := 4 * s
    fmt.Println("Jadi keliling persegi adalah", keliling)
}

func main() {
    var s int
    fmt.Print("Masukkan sisi persegi:")
    fmt.Scan(&s)
    luas_persegi(s)
    keliling_persegi(s)
}
```

Screenshoot Output

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul - IV> go run "d:\N
Masukkan sisi persegi:13
Jadi luas persegi adalah 169
Jadi keliling persegi adalah 52
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul - IV>
```

Deskripsi Program

Program berisi prosedur untuk menghitung luas persegi dan keliling persegi. Lalu kita memanggil fungsi tersebut untuk mengoutputkan luas dan keliling persegi.

III. UNGUIDED

1.

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? (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!x2!) = 120/12 = 10 P(10,10) = 10!/0! = 3628800/1 = 3628800 C(10,10) = 10!/(10!x0!) = 10!/10! = 1
2	8 0 2 0	56 28 1 1	

Selesaikan program tersebut dengan memanfaatkan prosedure yang diberikan berikut ini!

Sourcecode

```go
package main

import (
    "fmt"
)

// Prosedur untuk menghitung faktorial dan menyimpannya
dalam pointer hasil
func hitungFaktorial(n int, hasil *int) {
    *hasil = 1
    for i := 1; i <= n; i++ {
```

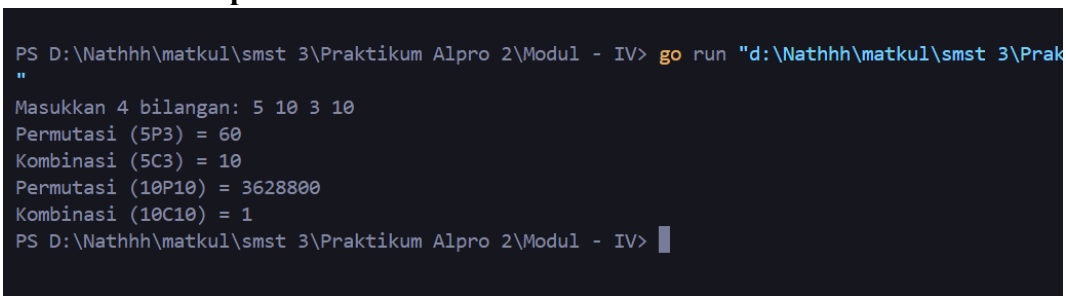
```
        *hasil *= i
    }
}

// Prosedur untuk menghitung dan mencetak permutasi
func cetakPermutasi(n, r int) {
    if n < r {
        fmt.Println("Permutasi tidak dapat dihitung karena n
< r.")
    } else {
        var faktN, faktNR int
        hitungFaktorial(n, &faktN)
        hitungFaktorial(n-r, &faktNR)
        perm := faktN / faktNR
        fmt.Printf("Permutasi (%dP%d) = %d\n", n, r, perm)
    }
}

// Prosedur untuk menghitung dan mencetak kombinasi
func cetakKombinasi(n, r int) {
    if n < r {
        fmt.Println("Kombinasi tidak dapat dihitung karena n
< r.")
    } else {
        var faktN, faktR, faktNR int
        hitungFaktorial(n, &faktN)
        hitungFaktorial(r, &faktR)
        hitungFaktorial(n-r, &faktNR)
        komb := faktN / (faktR * faktNR)
        fmt.Printf("Kombinasi (%dC%d) = %d\n", n, r, komb)
    }
}

func main() {
    // Input empat bilangan: a, b, c, d
    var a, b, c, d int
    fmt.Print("Masukkan 4 bilangan: ")
    fmt.Scan(&a, &b, &c, &d)
    // Baris pertama: permutasi dan kombinasi a terhadap c
    cetakPermutasi(a, c)
    cetakKombinasi(a, c)
    // Baris kedua: permutasi dan kombinasi b terhadap d
    cetakPermutasi(b, d)
    cetakKombinasi(b, d)
}
```

Screenshoot Output



Deskripsi Program

Program meminta memasukakan 4 buath bilangan rill a, b, c , dan d yang dipisahkan oleh spasi dengan syarat a >= c dan b >= d. Output program terdiri dari 2 baris, baris pertama adalah hasil permutasi dan kombinasi a terhadap c dan baris kedua adalah hasil permutasi dan kombinasi b terhadap d. Program ini menggunakan prosedur

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.

Fakultas Informatika
School of Computing
Informatics lab

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

```go
package main

import (
    "fmt"
    "strings"
)

const maxTime = 301 // Waktu maksimal jika soal tidak selesai
// Prosedur untuk menghitung skor (jumlah soal yang diselesaikan dan total waktu)
func hitungSkor(waktu [8]int, soal *int, skor *int) {
    *soal = 0
    *skor = 0
    for _, waktuSoal := range waktu {
```

```
        if waktuSoal < maxTime { // Jika soal diselesaikan
            *soal++
            *skor += waktuSoal
        }
    }
}
func main() {
    var pemenang string
    var soalPemenang, skorPemenang int
    soalPemenang = 0
    skorPemenang = maxTime * 8 // Nilai awal skor pemenang
maksimal
    for {
        var nama string
        var waktu [8]int
        // Membaca input nama peserta
        fmt.Print("Masukkan nama peserta (atau 'Selesai'
untuk mengakhiri): ")
        fmt.Scan(&nama)
        nama = strings.TrimSpace(nama)
        // Jika input adalah 'Selesai', hentikan loop
        if nama == "Selesai" {
            break
        }
        // Membaca waktu penyelesaian 8 soal
        fmt.Print("Masukkan waktu penyelesaian 8 soal (dalam
menit): ")
        for i := 0; i < 8; i++ {
            fmt.Scan(&waktu[i])
        }
        // Hitung soal yang diselesaikan dan total waktu
        var soal, skor int
        hitungSkor(waktu, &soal, &skor)
        // Tentukan pemenang berdasarkan jumlah soal dan
total waktu
        if soal > soalPemenang || (soal == soalPemenang &&
skor <
            skorPemenang) {
            pemenang = nama
            soalPemenang = soal
            skorPemenang = skor
        }
    }
    // Cetak pemenang
```

```
    fmt.Printf("Pemenang: %s, Soal yang diselesaikan: %d,
Total waktu: %d menit\n", pemenang, soalPemenang,
skorPemenang)
}
```

Screenshoot Output

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul - IV> go run "d:\Nathhh\matk
"
Masukkan nama peserta (atau 'Selesai' untuk mengakhiri): Nadhif
Masukkan waktu penyelesaian 8 soal (dalam menit): 20 11 12 42 12 43 54 53
Masukkan nama peserta (atau 'Selesai' untuk mengakhiri): Apalah
Masukkan waktu penyelesaian 8 soal (dalam menit): 20 51 32 42 12 54 56 42
Masukkan nama peserta (atau 'Selesai' untuk mengakhiri): Selesai
Pemenang: Nadhif, Soal yang diselesaikan: 8, Total waktu: 247 menit
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul - IV>
```

Deskripsi Program

Program berfungsi untuk mencari pemenang dari daftar peserta yang diberikan. Program menggunakan proesdur yang mengembaikan total soal dan tital skor yang dikerjakan oleh seorang peserta, melalui parameter formal.

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 ½n, 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 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

```go
package main

import "fmt"

// Prosedur untuk mencetak deret bilangan
func cetakDeret(n int) {
    for n != 1 {
        fmt.Printf("%d ", n) // Cetak nilai n
        if n%2 == 0 {        // Jika n genap
```

```
            n = n / 2
        } else { // Jika n ganjil
            n = 3*n + 1
        }
    }
    fmt.Println(1) // Cetak nilai akhir 1
}
func main() {
    var n int
    fmt.Print("Masukkan nilai awal deret: ")
    fmt.Scan(&n)  // Ambil input dari pengguna
    cetakDeret(n) // Panggil prosedur cetakDeret
}
```

Screenshoot Output

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul - IV> go run "d:\Nathhh\mat
"
Masukkan nilai awal deret: 15
15 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul - IV>
```

Deskripsi Program

Program akan mencetak setiap suku dari deret yang dijelaskan untuk suku nilai awal yang diberikan. Percetakan deret harus dibuat dalam sebuah prosedur. Jika bilangan saat itu genap, maka suku berikutnya adalah ½ n, tetapi jika ganjil maka suku berikutnya bernilai 3n+1 sampai deretterakgur bernilai 1