

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL: 4
MATERI: PROSEDUR



DISUSUN OLEH:

NAMA: JESIKA METANIA RAHMA ARIFIN

NIM: 103112400080

KELAS: 12 IF 01

DOSEN:

Dimas Fanny Hebrisianto Permadi S.ST, M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025/2026

DASAR TEORI

Prosedur

Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu instruksi baru yang dibuat untuk mengurangi kerumitan dari kode program yang kompleks pada suatu program yang besar. Prosedur akan menghasilkan suatu akibat atau efek langsung pada program ketika dipanggil pada program utama. Suatu subprogram dikatakan prosedur apabila:

1. Tidak ada deklarasi tipe nilai yang dikembalikan, dan
2. Tidak terdapat kata kunci return dalam badan subprogram.

Kedudukannya prosedur 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. Contoh: cetak, hitungRerata, cariNilai, belok, mulai,

Deklarasi Program dengan Prosedur:

```
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)
}
```

Penulisan argumen pada parameter cetakPesan(pesan,bilangan) harus sesuai urutan tipe data pada func cetakPesan(M string, flag int), yaitu string kemudian integer.

GUIDED

GUIDED 1

```
package main

import "fmt"

func hitungGaji(nama string, gajiPokok float64, jamLembur int) {
    bonusLembur := float64(jamLembur) * 500
    totalGaji := gajiPokok + bonusLembur

    fmt.Println("\n=== Slip Gaji ===")
    fmt.Println("Nama Karyawan :", nama)
    fmt.Printf("Gaji Pokok :Rp%.2f\n", gajiPokok)
    fmt.Printf("Bonus Lembur :Rp%f (%d jam x Rp50,000)\n", bonusLembur, jamLembur)
    fmt.Printf("Total Gaji :Rp%.2f\n", totalGaji)
}

func main() {
    var nama string
    var gajiPokok float64
    var jamLembur int

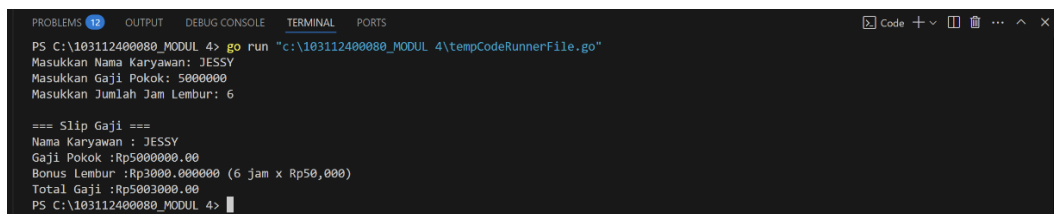
    fmt.Print("Masukkan Nama Karyawan: ")
    fmt.Scanln(&nama)

    fmt.Print("Masukkan Gaji Pokok: ")
    fmt.Scanln(&gajiPokok)

    fmt.Print("Masukkan Jumlah Jam Lembur: ")
    fmt.Scan(&jamLembur)

    hitungGaji(nama, gajiPokok, jamLembur)
}
```

SOURCECODE:



```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 4> go run "c:\103112400080_MODUL 4\tempCodeRunnerFile.go"
Masukkan Nama Karyawan: JESSY
Masukkan Gaji Pokok: 5000000
Masukkan Jumlah Jam Lembur: 6

=== Slip Gaji ===
Nama Karyawan : JESSY
Gaji Pokok :Rp5000000.00
Bonus Lembur :Rp3000.000000 (6 jam x Rp50,000)
Total Gaji :Rp5003000.00
PS C:\103112400080_MODUL 4> █
```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk menghitung jumlah gaji bonus lembur seorang karyawan.

GUIDED 2

```
package main
import "fmt"
//Prosedur untuk menghitung rata-rata dan menentukan kelulusan
func hitungRataRata(nama string, nilai1, nilai2, nilai3 float64) {
    rataRata:=(nilai1+nilai2+nilai3) / 3
    status:= "Tidak Lulus"
    if rataRata >= 60 {
        status="Lulus"
    }

    fmt.Println("\n=== Hasil Akademik ===")
    fmt.Println("Nama Mahasiswa:", nama)
    fmt.Printf("Nilai 1 : %.2f\n", nilai1)
    fmt.Printf("Nilai 2 : %.2f\n", nilai2)
    fmt.Printf("Nilai 3 : %.2f\n", nilai3)
    fmt.Printf("Rata-rata : %.2f\n", rataRata)
    fmt.Println("Status :", status)
}

func main() {
    var nama string
    var nilai1, nilai2, nilai3 float64

    //Input dari pengguna
    fmt.Print("Masukkan Nama Mahasiswa: ")
    fmt.Scanln(&nama)

    fmt.Print("Masukkan Nilai 1: ")
    fmt.Scanln(&nilai1)

    fmt.Print("Masukkan Nilai 2: ")
    fmt.Scanln(&nilai2)

    fmt.Print("Masukkan Nilai 3: ")
    fmt.Scanln(&nilai3)

    // Memanggil prosedur dengan data dari pengguna
    hitungRataRata(nama, nilai1, nilai2, nilai3)
}
```

SOURCECODE:

```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 4> go run "c:\103112400080_MODUL 4\GuidedProcedur2.go"
Masukkan Nama Mahasiswa: JESSY
Masukkan Nilai 1: 100
Masukkan Nilai 2: 90
Masukkan Nilai 3: 89

=== Hasil Akademik ===
Nama Mahasiswa: JESSY
Nilai 1 : 100.00
Nilai 2 : 90.00
Nilai 3 : 89.00
Rata-rata : 93.00
Status : Lulus
PS C:\103112400080_MODUL 4>
```

DESKRIPSI PROGRAM:

Program ini adalah program yang dibuat untuk menghitung nilai mahasiswa untuk melihat apakah dia lulus atau tidak.

UNGUIDED

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 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 $a \geq b$ dan $c \geq d$. Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi a terhadap b , sedangkan baris kedua adalah hasil permutasi dan kombinasi c terhadap d .

```
// JESIKA METANIA RAHMA ARIFIN
// 103112400080
package main

import (
    "fmt"
)

// Faktorial menghitung n!
func Faktorial(n int) int {
    if n == 0 {
        return 1
    }
    result := 1
    for i := 2; i <= n; i++ {
        result *= i
    }
    return result
}

// Permutasi  $P(n, r) = n! / (n-r)!$ 
```

```

func Permutasi(n, r int) int {
    if r > n {
        return 0
    }
    return Faktorial(n) / Faktorial(n-r)
}

// Kombinasi  $C(n, r) = n! / (r! * (n-r)!)$ 
func Kombinasi(n, r int) int {
    if r > n {
        return 0
    }
    return Faktorial(n) / (Faktorial(r) * Faktorial(n-r))
}

// Prosedur untuk mencetak hasil permutasi dan kombinasi
func CetakHasil(n, r int) {
    fmt.Println(Permutasi(n, r), Kombinasi(n, r))
}

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

    if a >= c && b >= d {
        CetakHasil(a, c)
        CetakHasil(b, d)
    } else {
        fmt.Println("Input tidak valid: pastikan a >= c dan b >= d.")
    }
}

```

SOURCECODE:

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400000_MODULE 4> go run "c:\103112400000_MODULE 4\Unguided1Prosedur.go"
5 10 3 10
60 10
3628800 1
PS C:\103112400000_MODULE 4> go run "c:\103112400000_MODULE 4\Unguided1Prosedur.go"
8 0 2 0
56 28
1 1
PS C:\103112400000_MODULE 4>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk menghitung nilai permutasi dan kombinasi pada suatu bilangan n.

UNGUIDED 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. prosedur 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.

```
//JESIKA METANIA RAHMA ARIFIN
// 103112400080
package main

import (
    "fmt"
)

// Fungsi untuk menghitung jumlah soal yang diselesaikan dan total waktu
func hitungSkor(waktu []int) (int, int) {
    batasWaktu := 301
    jumlahSoal, totalWaktu := 0, 0
    for _, t := range waktu {
        if t < batasWaktu {
            jumlahSoal++
            totalWaktu += t
        }
    }
    return jumlahSoal, totalWaktu
}

func main() {
    var pemenang string
    var maxSoal, minWaktu int = 0, 1000000

    for {
        var nama string
        var waktu [8]int
        _, err := fmt.Scan(&nama)
        if err != nil || nama == "Selesai" {
            break
        }

        for i := 0; i < 8; i++ {
            _, err = fmt.Scan(&waktu[i])
            if err != nil {
                break
            }
        }
    }
}
```

```

    }
}

jumlahSoal, totalWaktu := hitungSkor(waktu[:])

if jumlahSoal > maxSoal || (jumlahSoal == maxSoal && totalWaktu < minWaktu) {
    maxSoal = jumlahSoal
    minWaktu = totalWaktu
    pemenang = nama
}
}

fmt.Println(pemenang, maxSoal, minWaktu)
}

```

SOURCECODE:

```

PS C:\103112400080_MODUL 4> go run "c:\103112400080_MODUL 4\Unguided2Prosedur.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:\103112400080_MODUL 4>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk menentukan pemenang dalam mengerjakan soal kompetisi. Dan Bertha menang karena mengerjakan soal lebih banyak dan waktu yang lebih cepat daripada Astuti.

UNGUIDED 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 $1/2n$, 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. 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.

```

// JESIKA METANIA RAHMA ARIFIN
// 103112400080
package main

import (

```



```

    "fmt"
)

func cetakDeret(n int) {
    for n != 1 {
        fmt.Print(n, " ")
        if n%2 == 0 {
            n /= 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Println(1)
}

func main() {
    var n int
    fmt.Scan(&n)
    cetakDeret(n)
}

```

SOURCECODE:

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODULE 4> go run "c:\103112400080_MODULE 4\Unguided3Prosedur.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\103112400080_MODULE 4> go run "c:\103112400080_MODULE 4\Unguided3Prosedur.go"
45
45 136 68 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\103112400080_MODULE 4> go run "c:\103112400080_MODULE 4\Unguided3Prosedur.go"
87
87 262 131 394 197 592 296 148 74 37 112 56 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\103112400080_MODULE 4>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang dibuat untuk menghitung deret suatu bilangan.

KESIMPULAN

Prosedur di Golang dikenal sebagai fungsi (function), yang berfungsi untuk mengorganisir kode sehingga lebih terstruktur, modular, dan dapat dimanfaatkan kembali. Fungsi di Golang bisa dideklarasikan dengan memakai kata kunci func, diikuti nama fungsi, parameter (apabila ada), tipe kembalian (jika ada), serta isi fungsi tersebut dalam kurung kurawal {}. Golang menyediakan berbagai fitur dalam prosedur/fungsi, seperti:

REFERENSI

Dari modul yang diberikan