

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4
PROSEDUR**



Oleh:

MUHAMMAD FAUZAN

103112400064

12 IF 01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

4.1 Definisi Procedure

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.
2. Tidak terdapat kata kunci `return` dalam badan subprogram.

Kedudukan prosedur sama seperti instruksi dasar yang sudah ada sebelumnya (assignment) dan/atau instruksi yang berasal dari paket (fmt), seperti `fmt.Scan` dan `fmt.Print`. Oleh karena itu, selalu pilih nama prosedur yang berbentuk kata kerja atau sesuatu yang merepresentasikan proses sebagai nama dari prosedur, misalnya: `cetak`, `hitungRerata`, `cariNilai`, `belok`, `mulai`, `dsb`.

4.2 Deklarasi Procedure

Berikut ini adalah cara penulisan deklarasi prosedur dalam notasi Pseudocode dan GoLang.

Notasi Algoritma

```
procedure <nama procedure> (<params>)  
kamus  
{deklarasi variabel lokal dari procedure}  
...  
algoritma  
{badan algoritma procedure}  
...  
Endprocedure
```

Notasi dalam bahasa Go

```
func <nama procedure> (<params>) {  
    /* deklarasi variabel lokal dari procedure */  
    ...  
    /* badan algoritma procedure */  
    ...  
}
```

Penulisan deklarasi ini berada di luar blok program utama (`func main()`) pada suatu program Go dan bisa ditulis sebelum atau setelah dari blok program utama tersebut.

4.3 Cara Pemanggilan Procedure

Suatu prosedur hanya akan dieksekusi apabila dipanggil, baik secara langsung atau tidak langsung oleh program utama. Pemanggilan prosedur cukup mudah, yaitu dengan menuliskan nama beserta parameter atau argumen yang diminta oleh prosedur tersebut.

Sebagai contoh, prosedur `cetakNFibo` dipanggil dengan menuliskan namanya dan memberikan nilai sebagai argumen untuk parameter `n`:

Notasi Algoritma

```
cetakNFibo(10)
```

Notasi dalam bahasa Go

```
func main() {  
    var x int  
    x = 5  
    cetakNFibo(x) // cara pemanggilan #1  
    cetakNFibo(100) // cara pemanggilan #2  
}
```

Argumen yang digunakan dapat berupa variabel atau langsung menggunakan nilai konstanta.

4.4 Contoh Program dengan Procedure

Berikut ini adalah contoh program menggunakan prosedur untuk menampilkan pesan berdasarkan input pengguna.

Notasi dalam bahasa Go

```
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)
}
```

4.5 Parameter dalam Procedure

Suatu subprogram yang dipanggil dapat berkomunikasi dengan pemanggilnya melalui argumen yang diberikan melalui parameter yang dideklarasikan pada subprogram tersebut. Parameter dalam prosedur dapat dikategorikan berdasarkan letaknya dalam program, yaitu:

1. **Parameter formal** – Dideklarasikan dalam definisi prosedur.
2. **Parameter aktual** – Argumen yang diberikan saat pemanggilan prosedur.

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

Contoh 1

```
package main

import "fmt"

//Prosedur untuk menghitung total gaji karyawan
func hitungGaji(nama string, gajiPokok float64, jamLembur int) {
    bonusLembur := float64(jamLembur) * 50000
    totalGaji := gajiPokok + bonusLembur

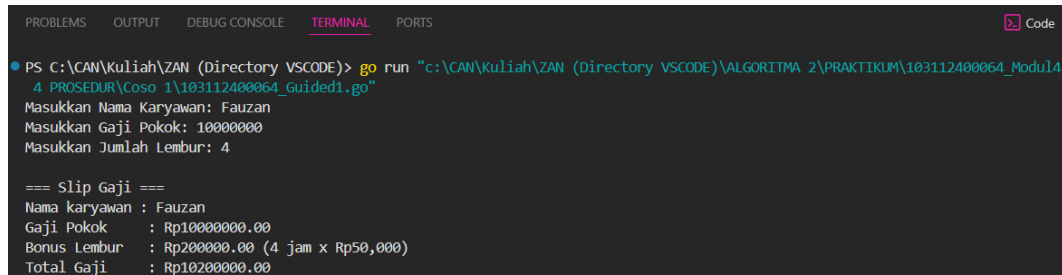
    fmt.Println("\n=== Slip Gaji ===")
    fmt.Printf("Nama Karyawan : %s\n", nama)
    fmt.Printf("Gaji Pokok    : Rp%.2f\n", gajiPokok)
    fmt.Printf("Bonus Lembur   : Rp%.2f (%d jam x Rp50,000)\n", bonusLembur,
jamLembur)
    fmt.Printf("Total Gaji     : Rp%.2f\n", totalGaji)
}

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

    //Menerima input dari user
    fmt.Print("Masukkan Nama Karyawan : ")
    fmt.Scanln(&nama)
    fmt.Println("Masukkan Gaji Pokok : ")
    fmt.Scanln(&gajiPokok)
    fmt.Println("Masukkan Jam Lembur : ")
    fmt.Scanln(&jamLembur)

    //Menghitung total gaji
    hitungGaji(nama, gajiPokok, jamLembur)
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code
PS C:\CAN\Kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\Kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\PRAKTIKUM\103112400064_Modul4
4_PROSEDUR\Coso 1\103112400064_Guided1.go"
Masukkan Nama Karyawan: Fauzan
Masukkan Gaji Pokok: 10000000
Masukkan Jumlah Lembur: 4

=== Slip Gaji ===
Nama karyawan : Fauzan
Gaji Pokok    : Rp10000000.00
Bonus Lembur  : Rp2000000.00 (4 jam x Rp50,000)
Total Gaji    : Rp12000000.00
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini digunakan untuk menghitung total gaji seorang karyawan berdasarkan gaji pokok dan jumlah jam lembur.

Contoh 2

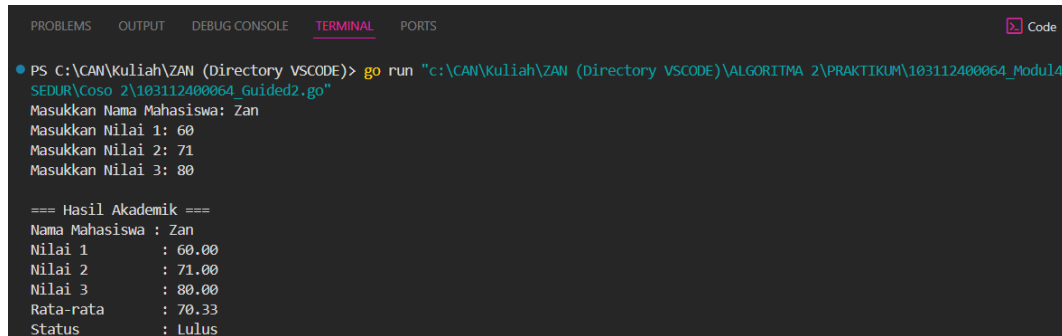
```
package main

import "fmt"

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
    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)
    hitungRataRata(nama, nilai1, nilai2, nilai3)
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code
● PS c:\CAN\kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\PRAKTIKUM\103112400064_Modul4
SEDUR\Coso 2\103112400064_Guided2.go"
Masukkan Nama Mahasiswa: Zan
Masukkan Nilai 1: 60
Masukkan Nilai 2: 71
Masukkan Nilai 3: 80

=== Hasil Akademik ===
Nama Mahasiswa : Zan
Nilai 1       : 60.00
Nilai 2       : 71.00
Nilai 3       : 80.00
Rata-rata     : 70.33
Status       : Lulus
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini digunakan untuk menghitung **rata-rata nilai mahasiswa** dan menentukan apakah mahasiswa tersebut **lulus atau tidak** berdasarkan ambang batas 60. Nilai dihitung dengan menjumlahkan ketiga angka yang dimasukkan, kemudian membaginya dengan tiga. Jika hasilnya mencapai atau melebihi 60, mahasiswa dinyatakan lulus, jika tidak, maka tidak lulus.

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

Soal 1

```
// MUHAMMAD FAUZAN
//103112400064

package main

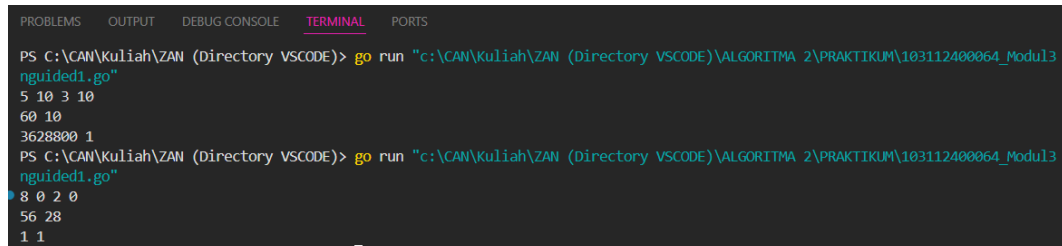
import "fmt"
func main() {
    var a, b, c, d int
    fmt.Scan(&a, &b, &c, &d)
    if a >= c && b >= d {
        fmt.Println(permutasi(a, c), kombinasi(a, c))
        fmt.Println(permutasi(b, d), kombinasi(b, d))
    } else {
        fmt.Println("tidak sesuai")
    }
}

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

func permutasi(n, r int) int {
    if r > n {
        return 0
    }
    return faktorial(n) / faktorial(n-r)
}

func kombinasi(n, r int) int {
    if r > n {
        return 0
    }
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\CAN\kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\PRAKTIKUM\103112400064_Modul3\nguided1.go"
5 10 3 10
60 10
3628800 1
PS C:\CAN\kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\PRAKTIKUM\103112400064_Modul3\nguided1.go"
8 0 2 0
56 28
1 1
```

// Foto hasil dari menjalankan code

Deskripsi: Program diatas adalah Program untuk menghitung hasil permutasi dan kombinasi dari dua pasang bilangan yang diberikan sebagai input 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*.

Soal 2

```
//MUHAMMAD FAUZAN
//103112400064

package main

import "fmt"

func hitungSkor(waktu int, totalWaktu, totalSoal *int) {
    if waktu <= 300 {
        *totalSoal++
        *totalWaktu += waktu
    }
}

func main() {
    var namaPemenang, peserta1, peserta2 string
    var waktu1, waktu2, totalWaktu1, totalWaktu2, totalSoal1, totalSoal2,
    maxSoal, maxWaktu int
    fmt.Scan(&peserta1)
    if peserta1 == "Selesai" {
        return
    }
    for i := 0; i < 8; i++ {
        fmt.Scan(&waktu1)
        hitungSkor(waktu1, &totalWaktu1, &totalSoal1)
    }
    for {
        fmt.Scan(&peserta2)
        if peserta2 == "Selesai" {
            break
        }
        for i := 0; i < 8; i++ {
            fmt.Scan(&waktu2)
            hitungSkor(waktu2, &totalWaktu2, &totalSoal2)
        }
        switch {
        case totalSoal1 < totalSoal2:
            maxSoal, maxWaktu, namaPemenang = totalSoal2,
totalWaktu2, peserta2
        case totalSoal1 > totalSoal2:
            maxSoal, maxWaktu, namaPemenang = totalSoal1,
totalWaktu1, peserta1
        }
    }
}
```

```

        case totalWaktu1 > totalWaktu2:
            maxSoal, maxWaktu, namaPemenang = totalSoal1,
totalWaktu1, peserta1
        default:
            maxSoal, maxWaktu, namaPemenang = totalSoal2,
totalWaktu2, peserta2
    }
}
fmt.Println(namaPemenang, maxSoal, maxWaktu)
}

```

Screenshots Output

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS c:\CAN\kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\PRAKTIKUM\103112400064_Modul4
2\103112400064_Unguided2.go"
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294

```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menilai peserta berdasarkan jumlah soal yang dikerjakan dalam waktu ≤ 300 detik. Pemenang ditentukan berdasarkan jumlah soal terbanyak. Jika jumlahnya sama, peserta dengan waktu lebih cepat menang. Output: Nama pemenang, jumlah soal, total waktu.

Soal 3

```
//MUHAMMAD FAUZAN
//103112400064

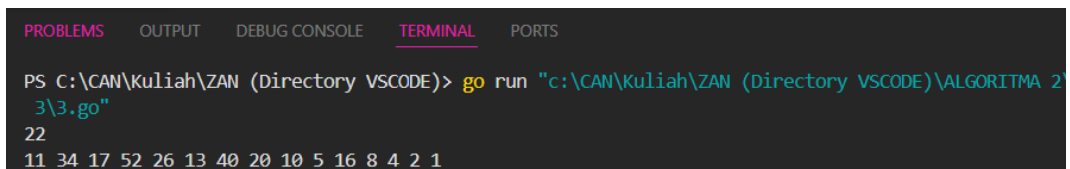
package main

import "fmt"

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

func main() {
    var n int
    fmt.Scan(&n)
    if n < 1000000 {
        cetakDeret(n)
    } else {
        fmt.Print("Error")
    }
}
```

Screenshots Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\CAN\Kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\Kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\3\3.go"
22
11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini digunakan untuk mencetak deret bilangan berdasarkan aturan, jika bilangan genap dibagi 2, jika bilangan ganjil dikali 3 lalu ditambah 1, proses berlanjut hingga bilangan mencapai 1; jika input lebih dari 999999 program akan menampilkan Error.

IV. KESIMPULAN

V. REFERENSI

Modul Praktikum Algoritma dan Pemrograman 2. (2025). *Modul 4: Prosedur*.
Fakultas Informatika, Telkom University.