

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4
PROCEDUR**



Oleh:

Raja Muhammad Lufhti

103112400027

12 IF 01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. DASAR TEORI

Penjelasan tentang Prosedur dalam Pemrograman

Apa Itu Prosedur?

Prosedur adalah bagian dari program yang dirancang untuk menjalankan tugas tertentu secara terpisah dari program utama. Prosedur memungkinkan program dibagi menjadi bagian-bagian kecil yang lebih modular, sehingga kode lebih terstruktur, mudah dibaca, dan dapat digunakan kembali.

Fungsi dan Manfaat Prosedur

1. **Memecah masalah kompleks** – Dengan membagi program menjadi prosedur-prosedur kecil, masalah dapat dipecah menjadi bagian yang lebih sederhana.
2. **Mempermudah pemeliharaan kode** – Jika ada kesalahan atau perubahan, cukup memperbaiki satu prosedur tanpa harus mengubah seluruh program.
3. **Menghindari duplikasi kode** – Jika suatu tugas dilakukan berulang kali, cukup buat satu prosedur dan panggil prosedur tersebut kapan pun diperlukan.
4. **Meningkatkan efisiensi** – Program menjadi lebih modular dan terorganisir, sehingga lebih mudah dipahami dan dikelola.

Cara Kerja Prosedur

- **Deklarasi** – Prosedur harus dideklarasikan terlebih dahulu sebelum digunakan dalam program.
- **Pemanggilan** – Prosedur tidak akan berjalan sendiri, melainkan harus dipanggil oleh program utama atau prosedur lain.
- **Eksekusi** – Saat dipanggil, prosedur menjalankan perintah di dalamnya sesuai dengan parameter yang diberikan.

Parameter dalam Prosedur

Prosedur dapat menerima **parameter** untuk menerima input dari pemanggil. Parameter ini memungkinkan prosedur bekerja secara fleksibel dengan berbagai nilai masukan.

Parameter dibagi menjadi dua jenis:

1. **Parameter Formal** – Dideklarasikan dalam definisi prosedur sebagai placeholder untuk nilai yang akan diberikan.
2. **Parameter Aktual** – Nilai yang diberikan ke prosedur saat pemanggilan.

Kesimpulan

Prosedur adalah bagian penting dalam pemrograman yang membantu mengorganisir kode dengan lebih baik. Dengan menggunakan prosedur, program menjadi lebih modular, mudah dikelola, dan efisien. Selain itu, prosedur juga memungkinkan penggunaan kembali kode tanpa perlu menulis ulang perintah yang sama berulang kali.

II. GUIDED

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

```
ive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4\GUIDED1_103112400072\G1.go"
Masukkan Nama Karyawan : udin
Masukkan Gaji Pokok :
50000
Masukkan Jam Lembur :
3

=== Slip Gaji ===
Nama Karyawan : udin
Gaji Pokok : Rp50000.00
Bonus Lembur : Rp15000.00 (3 jam x Rp50,000)
Total Gaji : Rp20000.00
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menghitung total gaji karyawan dengan menambahkan bonus sebesar Rp50.000 per jam lembur ke gaji pokok, kemudian menampilkan slip gaji yang berisi detail perhitungan.

Contoh 2

```
//RAJA MUHAMMAD LUFHTI 103112400027
```

```
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("Nama mahasiswa: ")
    fmt.Scanln(&nama)

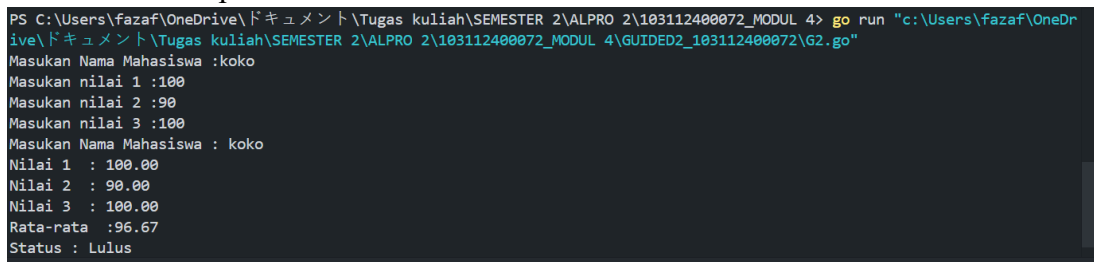
    fmt.Print("Nilai 1: ")
    fmt.Scanln(&nilai1)

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

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

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

Screenshots Output



```
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4> go run "c:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4\GUIDED2_103112400072\G2.go"
Masukan Nama Mahasiswa :koko
Masukan nilai 1 :100
Masukan nilai 2 :90
Masukan nilai 3 :100
Masukan Nama Mahasiswa : koko
Nilai 1 : 100.00
Nilai 2 : 90.00
Nilai 3 : 100.00
Rata-rata :96.67
Status : Lulus
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menghitung rata-rata dari tiga nilai mahasiswa dan menentukan status kelulusannya. Pengguna memasukkan nama mahasiswa beserta tiga nilai, kemudian program menghitung rata-rata dan menampilkan hasilnya. Jika rata-rata nilai mencapai 60 atau lebih, mahasiswa dinyatakan "Lulus", sedangkan jika kurang dari 60, statusnya "Tidak Lulus". Hasil akhir mencakup nama mahasiswa, nilai yang dimasukkan, rata-rata, serta status kelulusan.

III. UNGUIDED

Soal 1

```
// Muahamad faza fahri aziz || 103112400072
package main

import "fmt"

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

    fmt.Print(permutasi(a, c), " ", combination(a, c), "\n")
    fmt.Print(permutasi(b, d), " ", combination(b, d), "\n")
}

func factorial(n int) int {
    var hasil 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 factorial(n) / factorial(n-r)
}

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

Screenshots Output

```
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072
ive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4\UNGUIDED1_MODU
5 10 3 10
60 10
3628800 1
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072
```

// Foto hasil dari menjalankan code

Deskripsi: Program ini menghitung nilai permutasi dan kombinasi dari dua pasangan bilangan yang dimasukkan oleh pengguna. Setelah menerima empat angka sebagai input, program menentukan nilai permutasi $P(n,r)$ dan kombinasi $C(n,r)$ dengan menghitung faktorial sesuai rumus matematika yang berlaku. Hasil perhitungan ditampilkan dalam format yang jelas untuk setiap pasangan angka yang diberikan.

Soal 2

```
//RAJA MUHAMMAD LUFHTI
103112400027

package main

import "fmt"

func hitungNilai(waktu [8]int, soal *int,
nilai *int) {
    *soal = 0
    *nilai = 0
    for _, waktuSoal := range waktu {
        if waktuSoal < waktuMaksimal {
            *soal++
            *nilai += waktuSoal
        }
    }
}

const waktuMaksimal = 301

func main() { var
    Nama string
    var nilaiPemenang, soalPemenang int
    soalPemenang = 0
    nilaiPemenang = waktuMaksimal * 8
```

```
package main

import (
    "fmt"
    "strings"
)

func hitungSkor() (int, int) {
    var jumlahSoal, totalWaktu, waktu int
    jumlahSoal, totalWaktu = 0, 0

    for i := 0; i < 8; i++ {
        fmt.Scan(&waktu)

        if waktu < 301 {
            jumlahSoal++
            totalWaktu += waktu
        }
    }

    return jumlahSoal, totalWaktu
}

func main() {
    var pemenang string
    var soalTerbanyak, skorTerendah int
    soalTerbanyak, skorTerendah = -1, 99999999

    fmt.Println("Masukkan data peserta (ketik 'Selesai' untuk berhenti):")

    for {
        var nama string
        fmt.Scan(&nama)

        if strings.ToLower(nama) == "selesai" {
            break
        }

        jumlahSoal, totalWaktu := hitungSkor()

        if jumlahSoal > soalTerbanyak || (jumlahSoal == soalTerbanyak && totalWaktu <
skorTerendah) {
            pemenang = nama
        }
    }
}
```

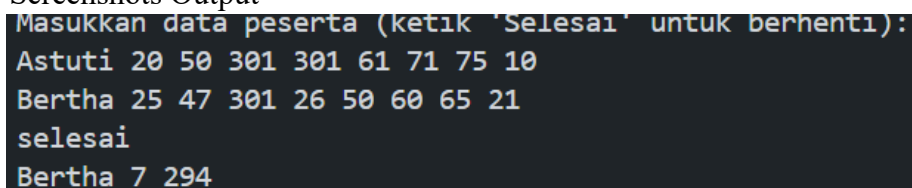
```

        soalTerbanyak = jumlahSoal
        skorTerendah = totalWaktu
    }
}

if pemenang != "" {
    fmt.Printf("%s %d %d\n", pemenang, soalTerbanyak, skorTerendah)
} else {
    fmt.Println("Tidak ada peserta.")
}
}

```

Screenshots Output



```

Masukkan data peserta (ketik 'Selesai' untuk berhenti):
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 menentukan pemenang dalam kompetisi pemrograman berdasarkan jumlah soal yang berhasil diselesaikan dan total waktu yang digunakan. Setiap peserta memasukkan namanya, diikuti oleh waktu pengerjaan untuk 8 soal. Jika...

Soal 3

```
package main

import (
    "fmt"
)

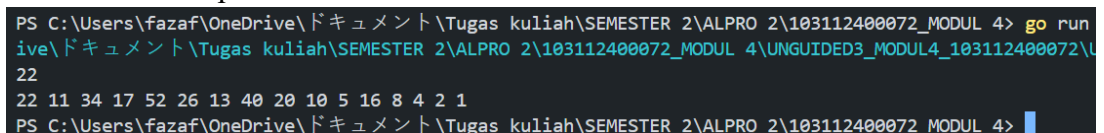
func cetakDeret(n int) {
    for n != 1 {
        fmt.Print(n, " ")

        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Println(n)
}

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

    if n > 0 && n < 1000000 {
        cetakDeret(n)
    } else {
        fmt.Println("Input harus bilangan positif lebih kecil dari 1000000.")
    }
}
```

Screenshots Output



```
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4> go run
ive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4\UNGUIDED3_MODUL4_103112400072\U
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_MODUL 4>
```

// Foto hasil dari menjalankan code

Deskripsi Program ini mencetak deret $(3n + 1)$ berdasarkan angka yang dimasukkan oleh pengguna. Jika angka tersebut genap, maka akan dibagi dua, sedangkan jika...

ganjil, akan dikalikan tiga dan ditambah satu. Proses ini berulang hingga angka mencapai 1, lalu hasilnya ditampilkan dalam satu baris.

IV. KESIMPULAN

Penjelasan tentang Prosedur dalam Pemrograman

- **Prosedur** adalah blok kode yang berisi serangkaian instruksi untuk menjalankan tugas tertentu. Prosedur membantu menyusun program agar lebih modular dan mudah dikelola.
- **Deklarasi prosedur** dilakukan dengan memberikan nama, parameter (jika diperlukan), dan instruksi dalam blok kode.
- **Pemanggilan prosedur** dilakukan dengan menyebutkan nama prosedur serta memberikan argumen yang sesuai dengan parameter yang dideklarasikan.
- **Parameter** dalam prosedur memungkinkan komunikasi antara pemanggil dan prosedur itu sendiri. Parameter dibagi menjadi:
 - **Parameter formal** – Dideklarasikan saat prosedur dibuat.
 - **Parameter aktual** – Nilai yang diberikan saat prosedur dipanggil.
- Berdasarkan cara alokasi memorinya, parameter dapat berupa:
 - **Pass by value** – Nilai dikirim dalam bentuk salinan, sehingga perubahan tidak memengaruhi data asli.
 - **Pass by reference/pointer** – Menggunakan alamat memori, sehingga perubahan dalam prosedur akan berdampak langsung pada data asli.
- Materi ini juga mencakup berbagai contoh implementasi prosedur dalam pemrograman, seperti:
 - Perhitungan deret Fibonacci
 - Penanganan pesan
 - Perhitungan permutasi dan kombinasi
 - Penentuan pemenang dalam kompetisi pemrograman
 - Deret bilangan Skiena

V. REFERENSI

MODUL 4 PROCEDUR ALGORITMA PEMROGRAMAN