

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 4
PROSEDUR**



Oleh: Dimas Fanny Hebrasianto Permadi

NAMA: Dimas Ramadhani

NIM: 103112400065

KELAS: IF-12-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

1. Pengertian Prosedur

Prosedur adalah bagian dari program yang berisi kumpulan instruksi untuk menjalankan tugas tertentu. Prosedur membantu menyederhanakan program yang kompleks agar lebih mudah dibaca dan dikelola. Berbeda dengan fungsi, prosedur tidak mengembalikan nilai apa pun.

2. Ciri-Ciri Prosedur

- Tidak menggunakan kata kunci return
- Tidak mengembalikan hasil ke pemanggilan
- Nama prosedur biasanya berupa kata kerja proses

3. Cara Menulis prosedur:

- Notasi Pseudocode:

```
procedure nama_prosedur(parameter)
kamus
    {deklarasi variabel lokal}
algoritma
    {isi atau langkah-langkah prosedur}
endprocedure
```

- Notasi Bahasa Go

```
func namaProsedur(parameter) {
    // deklarasi variabel lokal
    // algoritma atau isi dari prosedur
}
```

4. Cara Memanggil Prosedur

Untuk menjalankan prosedur, cukup tulis nama prosedurnya dan masukkan nilai sesuai parameternya. Contoh pemanggilan prosedur di Go:

```
func main() {
    var x int = 5
    cetakNFibo(x) // Memanggil prosedur
}
```

5. Parameter dalam Prosedur

Parameter adalah data yang dikirim dari program utama ke prosedur.

Ada dua jenis parameter:

- Parameter Formal: Ditulis saat mendeklarasikan prosedur.
- Parameter Aktual: Nilai yang diberikan saat memberikan prosedur.

Berdasarkan cara memori dialokasikan, ada dua tipe parameter:

- Pass by Value: Hanya menyalin nilai, lalu perubahan di dalam prosedur tidak memengaruhi nilai aslinya.
- Pass by Reference: Menggunakan alamat memori, perubahan yang ada didalam prosedur memengaruhi nilai aslinya.

II. GUIDED

1. Nomor 1

- **Source Code:**

```
package main
import "fmt"
func hitungGaji(nama string, gajiPokok float64, jamLembur int) {
    bonusLembur := float64(jamLembur)*50000
    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%.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

    //input dari pengguna
    fmt.Print("Masukkan Nama Karyawan: ")
    fmt.Scanln(&nama)

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

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

    // Memanggil prosedur dengan data dari pengguna
    hitungGaji(nama, gajiPokok, jamLembur)
}
```

- **Screenshoot Hasil Program:**

```
e\Semester 2\Algoritma dan Pemrograman 2\Praktikum
aman 2\Praktikum\Laprak\103112400065_MODUL4\103112
Masukkan Nama Karyawan: Anton
Masukkan Gaji Pokok: 2000000
Masukkan Jumlah Lembur: 3

=== Slip Gaji ===
Nama karyawan : Anton
Gaji Pokok      : Rp2000000.00
Bonus Lembur    : Rp150000.00 (3 jam x Rp50,000)
Total Gaji      : Rp2150000.00
\Semester 2\Algoritma dan Pemrograman 2\Praktikum\
Masukkan Nama Karyawan: Budi
Masukkan Gaji Pokok: 3000000
Masukkan Jumlah Lembur: 0

=== Slip Gaji ===
Nama karyawan : Budi
Gaji Pokok      : Rp3000000.00
Bonus Lembur    : Rp0.00 (0 jam x Rp50,000)
Total Gaji      : Rp3000000.00
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algo
```

- **Penjelasan**

Program ini bertujuan untuk merangkum gaji seseorang dengan memasukkan nama, gaji, dan lama lembur pegawai. Untuk lembur, setiap jam akan dikalikan Rp. 50.000. Untuk output dari program yaitu hasil total gaji pegawai.

Program ini menggunakan prosedur dengan nama HitungGaji, dengan parameter nama tipe string, gajiPokok tipe float64, dan jamLembur tipe integer. Di dalam prosedur akan melakukan deklarasi variabel bonusLembur dengan nilai jamLembur dikalikan 50000. Setelah itu melakukan deklarasi lagi untuk totalGaji dengan gajiPokok+bonusLembur. Lalu akan mencetak hasilnya.

Masuk ke program utama, dengan mendeklarasikan nama tipe string, gajiPokok tipe float64, jamLembur tipe int. Program mencetak kalimat lalu masukkan nama pegawai kita. Program akan mencetak kalimat lalu masukkan gajiPokok. Program akan mencetak kalimat lalu masukkan jamLembur. Lalu program akan memanggil prosedur hitungGaji dengan parameter berurut nama, gajiPokok, jamLembur.

2. Nomor 2

- **Source Code:**

```
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

    // 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)

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

- **Screenshoot Hasil Program:**

```

PS C:\Users\Dimas\OneDrive\Collag
aparak\103112400065_MODUL4> go run
ma dan Pemrograman 2\Praktikum\La
Masukkan Nama Mahasiswa: Permana
Masukkan nilai 1: 87
Masukkan nilai 2: 86
Masukkan nilai 3: 89

=== Hasil Akademik ===
Nama Mahasiswa : Permana
Nilai 1         : 87.00
Nilai 2         : 86.00
Nilai 3         : 89.00
Rata-rata       : 87.33
Status          : Lulus

Masukkan Nama Mahasiswa: Qyink
Masukkan nilai 1: 80
Masukkan nilai 2: 76
Masukkan nilai 3: 81

=== Hasil Akademik ===
Nama Mahasiswa : Qyink
Nilai 1         : 80.00
Nilai 2         : 76.00
Nilai 3         : 81.00
Rata-rata       : 79.00
Status          : Lulus

```

- **Penjelasan:**

Program ini bertujuan untuk merangkum nilai dan lulus atau tidaknya Mahasiswa dengan memasukkan nama, nilai1, nilai2 dan nilai3. Untuk menyatakan lulus, harus menjumlahkan ketiga nilai tersebut lalu dibagi 3. Parameter lulus diberikan jika nilai rata-rata lebih dari sama dengan 60.

Program ini menggunakan prosedur dengan nama `hitungRataRata`, dengan parameter nama tipe string, nilai1, nilai2, nilai3 tipe float64. Di dalam prosedur akan melakukan deklarasi variabel rata-rata dengan menjumlahkan nilai lalu membaginya dengan 3. Setelah itu melakukan deklarasi lagi untuk status dengan kalimat “Tidak Lulus”. Setelah itu ada struktur percabangan dengan kondisi rata-rata di atas sama dengan 60 maka status berubah menjadi “Lulus”. Lalu akan mencetak hasilnya.

Masuk ke program utama, dengan mendeklarasikan nama tipe string, nilai1, nilai2, nilai3 tipe float64. Setiap kali program mencetak kalimat kita dapat menginputkan nama, nilai1, nilai2, dan nilai3 sesuai perintah output. Lalu melakukan pemanggilan prosedur `hitungRataRata` dengan parameter (nama, nilai1, nilai2, nilai3).

III. UNGUIDED

1. Nomor 1

- **Source Code:**

```
// Dimas Ramadhani
// 103112400065
package main
import "fmt"
func main() {
    var a, b, c, d int
    var permut, komb int
    fmt.Scan(&a, &b, &c, &d)
    if a >= c {
        permutasi(a, c, &permut)
        kombinasi(a, c, &komb)
        fmt.Println(permut, komb)
    } else {
        permutasi(c, a, &permut)
        kombinasi(c, a, &komb)
        fmt.Println(permut, komb)
    }
    if b >= d {
        permutasi(b, d, &permut)
        kombinasi(b, d, &komb)
        fmt.Println(permut, komb)
    } else {
        permutasi(d, b, &permut)
        kombinasi(d, b, &komb)
        fmt.Println(permut, komb)
    }
}
func faktorial(n int, hasil *int) {
    *hasil = 1
    for i := 1; i <= n; i++ {
        *hasil *= i
    }
}
```

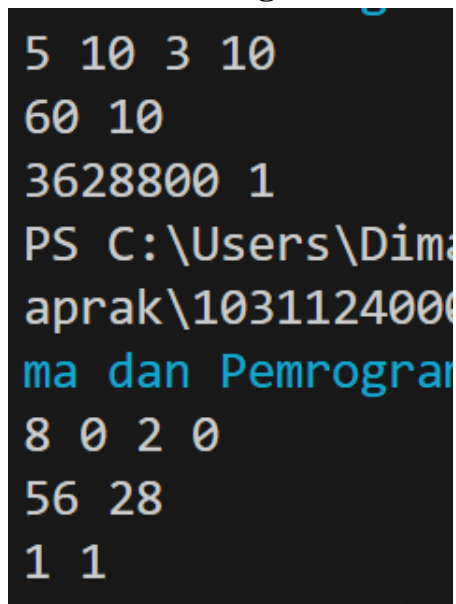


```

}
}
func permutasi(n, r int, hasil *int) {
    var test1, test2 int
    if r > n {
        *hasil = 0
    }
    faktorial(n, &test1)
    faktorial(n-r, &test2)
    *hasil = test1 / test2
}
func kombinasi(n, r int, hasil *int) {
    var test1, test2, test3 int
    if r > n {
        *hasil = 0
    }
    faktorial(n, &test1)
    faktorial(n-r, &test2)
    faktorial(r, &test3)
    *hasil = test1 / (test2 * test3)
}

```

- **Screenshot Hasil Program:**



```

5 10 3 10
60 10
3628800 1
PS C:\Users\Dimas\Documents\1031124000\ma dan Pemrograman
8 0 2 0
56 28
1 1

```

- **Penjelasan:**

Program ini bertujuan untuk menghitung permutasi dan kombinasi dengan memasukkan empat angka dengan variabel a, b, c, dan d tipe integer. Dengan sepasang dua angka a dengan c dan b dengan d.

Program ini menggunakan prosedur dengan nama faktorial, dengan parameter n tipe int dan hasil tipe int. Di dalam prosedur akan melakukan deklarasi variabel hasil dengan 1. Setelah itu melakukan perulangan dengan iterasi 1 dan batasan variabel n, di dalam perulangan akan melakukan perkalian variabel hasil dengan i.

Program ini menggunakan prosedur dengan nama permutasi, dengan parameter n, r tipe int dan hasil tipe int. Di dalam prosedur akan melakukan deklarasi variabel test1, test2 tipe int. Setelah itu melalui struktur percabangan kondisi $r > n$ maka variabel hasil akan bernilai 0. Di luar percabangan akan menjalankan prosedur faktorial(n, &test1), faktorial(n-r, &test2), dan $hasil = \frac{test1}{test2}$

Program ini menggunakan prosedur dengan nama kombinasi, dengan parameter n tipe int dan hasil tipe int. Di dalam prosedur akan melakukan deklarasi variabel test1, test2, test3 tipe int. Setelah itu melalui struktur percabangan kondisi $r > n$ maka variabel hasil = 0. Di luar percabangan akan menjalankan prosedur faktorial(n, &test1), faktorial(n-r, &test2), faktorial(r, &test3), lalu $hasil = \frac{test1}{test2 \times test3}$.

Masuk ke program utama, dengan mendeklarasikan a, b, c, d tipe int lalu permut, komb tipe int. Program membaca masukkan sebagai a, b, c, d. Program akan mencetak kalimat lalu masukkan gajiPokok. Program akan melalui banyaknya struktur percabangan dengan kondisi yang berbeda-beda sesuai pada SOURCE CODE di setiap percabangan akan mencetak hasilnya.

2. Nomor 2

- **Source Code:**

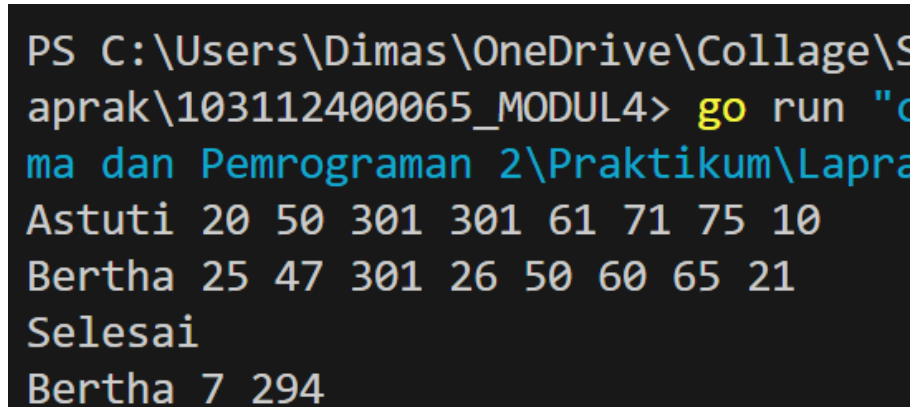
```
// Dimas Ramadhani
// 103112400065
package main
import "fmt"
func hitungSkor(soal int, totalwaktu *int, totalsoal *int) {
    if soal <= 300 {
        *totalsoal++
        *totalwaktu += soal
    }
}
func main() {
    var nama1, nama2 string
    var soal1, soal2 int
    var totalwaktu1, totalwaktu2, totalsoal1, totalsoal2 int
    var totalsoaltemp, totalwaktutemp int
    var namatemp string
    fmt.Scan(&nama1)
    if nama1 != "Selesai" {
        for i := 0; i < 8; i++ {
            fmt.Scan(&soal1)
            hitungSkor(soal1, &totalwaktu1, &totalsoal1)
        }
        for {
            fmt.Scan(&nama2)
            if nama2 == "Selesai" {
                break
            }
            for i := 0; i < 8; i++ {
                fmt.Scan(&soal2)
                hitungSkor(soal2, &totalwaktu2,
&totalsoal2)
            }
            if totalsoal1 < totalsoal2 {
                totalsoaltemp = totalsoal2
                totalwaktutemp = totalwaktu2
                namatemp = nama2
            } else if totalsoal1 > totalsoal2 {
                totalsoaltemp = totalsoal1
            }
        }
    }
}
```

```

        totalwaktutemp = totalwaktu1
        namatemp = nama1
    } else if totalsoal1 == totalsoal2 {
        if totalwaktu1 > totalwaktu2 {
            totalsoaltemp = totalsoal1
            totalwaktutemp = totalwaktu1
            namatemp = nama1
        } else if totalsoal1 < totalsoal2 {
            totalsoaltemp = totalsoal2
            totalwaktutemp = totalwaktu2
            namatemp = nama2
        }
    }
}
fmt.Println(namatemp, totalsoaltemp, totalwaktutemp)
}
}

```

- **Screenshoot Hasil Program:**



```

PS C:\Users\Dimas\OneDrive\Collage\S aprak\103112400065_MODUL4> go run "c:\Users\Dimas\OneDrive\Collage\S aprak\103112400065_MODUL4\main.go"
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294

```

- **Penjelasan:**

Program ini bertujuan untuk mengetahui mahasiswa terbanyak atau tercepat dalam mengerjakan 8 soal dengan variabel inputan kita yaitu nama1, nama2 tipe string, soal1, soal2 tipe int.

Program ini menggunakan prosedur dengan nama hitungSkor, dengan parameter soal tipe int, total waktu tipe int dan totalsoal tipe int. Di dalam prosedur akan melalui percabangan dengan kondisi n kurang dari sama dengan 300, karena pada soal diberitahukan setiap soal hanya dapat dikerjakan selama 300 menit. Di dalam percabangan kan menambahkan nilai untuk variabel totalsoal lalu totalwaktu ditambahkan variabel soal.

Masuk ke program utama, dengan mendeklarasikan nama1, nama2, namatemp tipe string, soal1, soal2, totalwaktu1, totalwaktu2, totalsoal1, totalsoal2, totalsoaltemp, totalwaktutemp tipe int. Program membaca masukkan nama mahasiswa pertama sebagai nama1. Program akan melalaui percabangan dengan kondisi nama1 tidak sama dengan “Selesai”. Di dalamnya akan menjalankan struktur perulangan dengan iterasi 0 dan batas kurang dari 8, di dalamnya program membaca inputan waktu pengerjaan setiap soal lalu menjalankan prosedur hitungSkor(soal1, &totalwaktu1, &totalsoal1). Setelah struktur perulangan pertama, akan masuk lagi ke perulangan kedua, dalam struktur ini program langsung membaca inputan nama selanjutnya dengan variabel nama2. Setelah itu masuk ke percabangan dengan kondisi nama2 sama dengan “Selesai” maka perulangan akan langsung berhenti. Setelah percabangan ada perulangan untuk menghitung waktu pengerjaan soal sama seperti mahasiswa pertama. Seteleha perulangan itu masuk ke percabangan untuk mengetahui mahasiswa mana yang mengerjakan soal terbanyak terlebih dulu, jika jumlah soal sama maka yang tercepat, untuk perhitungan jumlah soal bergantung pada pengerjaan mahasiswa jika lebih dari 300 menit tidak terhitung. Terakhir setelah percabangan akan tercetak nama pemenang beserta total soal dan waktu pengerjaannya.

3. Nomor 3

- **Source Code:**

```
// Dimas Ramadhani
// 103112400065
package main
import "fmt"
func cetakDeret(n int, hasil *float64) {
    var ntemp int
    ntemp = n
    for ntemp != 1 {
        if ntemp%2==0 {
            *hasil = float64(ntemp)/2
            ntemp = int(*hasil)
        } else if ntemp%2!=0 {
            *hasil = 3*float64(ntemp)+1
            ntemp = int(*hasil)
        }
        fmt.Print(*hasil, " ")
    }
}
```

```

}
func main() {
    var n int
    var deret float64
    fmt.Scan(&n)
    if n < 1000000 {
        cetakDeret(n, &deret)
    } else {
        fmt.Print("Error")
    }
}

```

- **Screenshoot Hasil Program:**

```

PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman
Praktikum\Laprak\103112400065_MODUL4> go run "c:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2\Praktikum\Laprak\103112400065_MODUL4\103112400065_Unguided3.go"
22
11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman
Praktikum\Laprak\103112400065_MODUL4> go run "c:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2\Praktikum\Laprak\103112400065_MODUL4\103112400065_Unguided3.go"
100
50 25 76 38 19 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

```

- **Penjelasan:**

Program ini bertujuan untuk membuat deret bilangan, jika bilangan genap akan dikalikan dengan $\frac{1}{2}$, jika ganjil akan dikalikan dengan 3 lalu tambah satu. Program ini hanya menginputkan satu angka saja, untuk membuat deret tersebut akan menghitung bilangan sebelumnya tergantung genap atau ganjilnya.

Program ini menggunakan prosedur dengan nama cetakDeret, dengan parameter n tipe int dan hasil tipe float64. Di dalam prosedur akan melakukan deklarasi variabel ntemp tipe int. Setelah itu melakukan pemberian nilai ntemp = n. Lalu perulangan dengan kondisi ntemp tidak sama dengan 1, di dalam perulangan akan melalui percabangan dengan kondisi pertama ntemp modulus 2 sama dengan 0 maka akan melakukan kalkulasi pada variabel $hasil = \frac{float64(ntemp)}{2}$ lalu pemberian nilai variabel $ntemp = int(hasil)$. Pada kondisi kedua ntemp modulus 2 tidak sama dengan 0 maka akan melakukan kalkulasi $hasil = 3 \times float64(ntemp) + 1$ lalu pemberian nilai $ntemp = int(hasil)$. Setelah percabangan akan mencetak hasil lalu pemberian spasi.

Masuk ke program utama, dengan mendeklarasikan `n` tipe `int` dan deret tipe `float64`. Program membaca masukkan sebagai `n`. Program akan melalui percabangan dengan kondisi `n` kurang dari 1.000.000 maka akan menjalankan prosedur `cetakDeret(n, &deret)`. Selain kondisi itu akan tercetak “Error”.

IV. KESIMPULAN

Prosedur sangat penting dalam pemrograman karena membantu membagi tugas besar menjadi langkah-langkah kecil yang mudah dikelola. Dengan memahami cara kerja prosedur dan jenis parameter yang digunakan, kita dapat menulis program yang lebih efisien, terorganisir, dan mudah dimodifikasi di masa depan. Selain itu, penggunaan prosedur memungkinkan pengembangan program yang lebih modular, sehingga memudahkan kolaborasi dalam tim dan mempercepat proses debugging atau perbaikan kesalahan.

V. REFERENSI

Prayogo, N. A. (2024). *Dasar Pemrograman Golang* (Versi 4.0.20240830). Retrieved from <https://github.com/novalagung/dasarpemrogramangolang>