LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 4

PROSEDUR

Oleh:

NAMA: ICHYA ULUMIDDIIN

NIM: 103112400076

KELAS: 12IF-01-A

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. DASAR TEORI

Dalam bahasa pemrograman, prosedur dan fungsi adalah dua konsep yang sering digunakan untuk mengorganisir kode menjadi bagian-bagian yang lebih modular dan terstruktur. Keduanya merupakan subprogram yang dapat dipanggil untuk menjalankan tugas tertentu, namun memiliki perbedaan utama:

- Fungsi: Mengembalikan nilai setelah eksekusi.
- Prosedur: Tidak mengembalikan nilai setelah eksekusi.

Dalam bahasa Go (Golang), tidak ada konsep prosedur yang terpisah seperti dalam beberapa bahasa pemrograman lainnya. Semua subprogram, baik yang mengembalikan nilai maupun tidak, disebut sebagai fungsi. Dengan kata lain, Go tidak membedakan antara prosedur dan fungsi; semuanya diimplementasikan sebagai fungsi.

Keuntungan Menggunakan Prosedur dan Fungsi:

- Modularitas: Memungkinkan pemrogram membagi program menjadi bagian-bagian kecil yang lebih mudah dikelola dan dipahami.
- Reusabilitas: Kode yang ditulis dalam bentuk prosedur atau fungsi dapat digunakan kembali di berbagai bagian program tanpa perlu menulis ulang kode yang sama.
- Pemeliharaan: Dengan memisahkan kode ke dalam prosedur dan fungsi, pemeliharaan dan pembaruan kode menjadi lebih mudah dilakukan tanpa mempengaruhi keseluruhan program.

II. GUIDED

Source Code Guided 1

```
//ICHYA ULUMIDDIIN
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 Rp
50,000)\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.Scanln(&jamLembur)

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

Output

```
Masukkan Nama Karyawan: Udin
Masukkan Gaji Pokok: 1500000
Masukkan Jumlah Jam Lembur: 12

=== SLip Gaji ===
Nama Karyawan:  Udin
Gaji Pokok : Rp1500000.00
Bonus Lembur : Rp600000.00 (12 jam x Rp 50,000)
Total Gaji : Rp2100000.00
```

Penjelasan

Program yang ditulis dalam bahasa Go (Golang) ini bertujuan untuk menghitung total gaji karyawan dengan mempertimbangkan gaji pokok dan bonus lembur. Program ini dirancang dengan pendekatan modular menggunakan sebuah fungsi hitungGaji yang bertanggung jawab dalam melakukan perhitungan serta menampilkan slip gaji karyawan.

Fungsi hitungGaji menerima tiga parameter, yaitu nama karyawan (string), gaji pokok (float64), dan jumlah jam lembur (int). Di dalam fungsi ini, pertama-tama dilakukan perhitungan bonus lembur berdasarkan jumlah jam lembur dikalikan dengan tarif lembur sebesar Rp50.000 per jam. Setelah itu, total gaji dihitung dengan menjumlahkan gaji pokok dan bonus lembur. Fungsi ini kemudian mencetak slip gaji, yang mencakup nama karyawan, gaji pokok, rincian bonus lembur, dan total gaji yang diterima.

Di dalam fungsi main, program meminta inputan dari pengguna, yaitu nama karyawan, gaji pokok, dan jumlah jam lembur. Input ini dikumpulkan menggunakan fmt.Scanln(), yang membaca data dari keyboard. Setelah data diperoleh, program memanggil fungsi hitungGaji untuk melakukan perhitungan dan menampilkan hasilnya.

Source Code Guided 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)
}
```

Output

```
Masukkan Nilai 2 : 89
Masukkan Nilai 3 : 95

=== Hasil Akademik ===
Nama Mahasiswa : Udin
Nilai 1          : 90.00
Nilai 2          : 89.00
Nilai 3          : 95.00
Rata-Rata        : 91.33
Status   :  Lulus
```

Penjelasan

Program ini dibuat menggunakan bahasa Golang dengan tujuan utama untuk menghitung rata-rata nilai akademik mahasiswa berdasarkan tiga nilai yang dimasukkan oleh pengguna, sekaligus menentukan status kelulusan mahasiswa berdasarkan kriteria yang telah ditentukan. Program ini terdiri dari dua bagian utama, yaitu fungsi hitungRataRata yang bertanggung jawab dalam melakukan perhitungan rata-rata nilai serta menentukan status kelulusan, dan fungsi main yang berfungsi untuk menerima input dari pengguna dan memanggil fungsi hitungRataRata untuk memproses data.

Pada fungsi hitungRataRata, program menerima nama mahasiswa dan tiga nilai akademik sebagai parameter. Selanjutnya, program menghitung rata-rata nilai dengan rumus (nilai1 + nilai2 + nilai3) / 3. Setelah itu, program menentukan status kelulusan mahasiswa dengan syarat bahwa jika rata-rata nilai ≥ 60, maka mahasiswa dinyatakan "Lulus", sedangkan jika kurang dari 60, mahasiswa dianggap "Tidak Lulus". Hasil akhir akan ditampilkan dalam format yang rapi, menampilkan nama mahasiswa, nilai individu, rata-rata, dan status kelulusan.

Fungsi main dalam program ini berfungsi untuk meminta input dari pengguna berupa nama mahasiswa dan tiga nilai akademik. Setelah data dimasukkan, fungsi hitungRataRata dipanggil untuk menghitung rata-rata dan menampilkan hasil akhir.

III. UNGUIDED

Source Code Unguided 1

```
//ICHYA ULUMIDDIIN
package main

import (
    "fmt"
)
func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    result := 1
    for i := 2; i <= n; i++ {
        result *= i
    }
    return result
}
func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}
func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-
r))
}
func main() {
    var a, b, c, d int
    fmt.Scan(&a, &b, &c, &d)
    if a >= c && b >= d {
        fmt.Printf("P(%d,%d) = %d, C(%d,%d) = %d\n", a,
c, permutasi(a, c), a, c, kombinasi(a, c))
        fmt.Printf("P(%d,%d) = %d, C(%d,%d) = %d\n", b,
d, permutasi(b, d), b, d, kombinasi(b, d))
    } else {
        fmt.Println("Input tidak valid, pastikan a >= c
dan b >= d")
    }
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Do
ided1-Prosedur\Unguided1-Prosedur.go"
5 10 3 10
P(5,3) = 60, C(5,3) = 10
P(10,10) = 3628800, C(10,10) = 1
```

Penjelasan

Program dalam bahasa Golang ini dirancang untuk menghitung permutasi dan kombinasi dari dua pasang bilangan yang dimasukkan oleh pengguna. Program ini pertama-tama meminta empat bilangan dari pengguna, yaitu a, b, c, dan d, lalu memverifikasi bahwa a ≥ c dan b ≥ d sebelum melakukan perhitungan. Jika syarat ini terpenuhi, program akan menghitung P(a, c) dan C(a, c) untuk pasangan pertama serta P(b, d) dan C(b, d) untuk pasangan kedua. Jika tidak memenuhi syarat, program akan menampilkan pesan "Input tidak valid".

Program ini terdiri dari tiga fungsi utama, yaitu faktorial, permutasi, dan kombinasi. Fungsi faktorial(n) menghitung faktorial dari sebuah bilangan dengan menggunakan perulangan (for loop), sehingga lebih efisien dibandingkan rekursi dalam perhitungan faktorial besar. Fungsi permutasi(n, r) menghitung permutasi dengan rumus P(n, r) = n! / (n-r)!, yang digunakan untuk menghitung kemungkinan susunan elemen yang berbeda dalam suatu himpunan. Sementara itu, fungsi kombinasi(n, r) menghitung kombinasi dengan rumus C(n, r) = n! / (r!(n-r)!), yang digunakan untuk menentukan cara memilih elemen dari suatu himpunan tanpa memperhatikan urutan.

Pada bagian main(), program menerima empat angka yang dimasukkan oleh pengguna melalui fmt.Scan(&a, &b, &c, &d)

Source Code Unguided 2

```
//ICHYA ULUMIDDIIN
package main

import (
	"fmt"
	"strings"
)

func hitungSkor(waktu [8]int) (int, int) {
	jumlahSoal, totalWaktu := 0, 0
	for _, t := range waktu {
		if t <= 300 {
			jumlahSoal++
			totalWaktu += t
		}
	}
	return jumlahSoal, totalWaktu
}

func main() {
	var nama, pemenang string
	var waktu [8]int
	maxSoal, minWaktu := 0, 99999

	fmt.Println("Masukkan peserta dan waktu (akhiri
dengan 'Selesai'): ")

	for {
		fmt.Scan(&nama)
		if strings.ToLower(nama) == "selesai" {
			break
		}
		for i := 0; i < 8; i++ {
			fmt.Scan(&waktu[i])
		}

		jumlahSoal, totalWaktu := hitungSkor(waktu)

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

	fmt.Printf("\nPemenang: %s\nJumlah Soal: %d\nTotal
Waktu: %d menit\n", pemenang, maxSoal, minWaktu)
}
```

Output

```
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai

Pemenang: Bertha
Jumlah Soal: 7
Total Waktu: 294 menit
```

Penjelasan

Program Golang ini bertujuan untuk menentukan pemenang dalam sebuah kompetisi pemrograman berdasarkan jumlah soal yang diselesaikan dan total waktu pengerjaan. Setiap peserta diberikan 8 soal yang harus dikerjakan dalam batas waktu maksimal 300 menit per soal. Peserta yang menyelesaikan soal terbanyak dalam waktu tersingkat akan dinyatakan sebagai pemenang. Jika dua peserta menyelesaikan jumlah soal yang sama, pemenang ditentukan berdasarkan peserta yang memiliki total waktu pengerjaan paling sedikit.

Program ini terdiri dari dua bagian utama, yaitu fungsi hitungSkor dan fungsi utama main. Fungsi hitungSkor bertanggung jawab untuk menghitung jumlah soal yang diselesaikan (soal yang dikerjakan ≤ 300 menit) dan total waktu pengerjaan untuk soal yang berhasil dikerjakan. Pada fungsi utama main, program meminta input dari pengguna dalam bentuk nama peserta dan 8 waktu pengerjaan soal. Program akan terus menerima input hingga pengguna memasukkan kata "Selesai". Setiap kali seorang peserta memasukkan datanya, program akan menghitung jumlah soal yang berhasil diselesaikan serta total waktu yang dibutuhkan, lalu membandingkannya dengan rekor terbaik sebelumnya. Jika peserta tersebut memiliki lebih banyak soal yang diselesaikan atau waktu pengerjaan lebih cepat dalam jumlah soal yang sama, maka data peserta ini disimpan sebagai pemenang sementara.

Source Code Unguided 3

```
//ICHYA ULUMIDDIIN
package main

import (
	"fmt"
)
func cetakDeret(n int) {
	for n != 1 {
		fmt.Printf("%d ", n)
		if n%2 == 0 {
			n /= 2
		} else {
			n = 3*n + 1
		}
	}
	fmt.Println(n)
}
func main() {
	var n int
	fmt.Scan(&n)
	if n < 1 || n >= 1000000 {
		fmt.Println("Bilangan harus di antara 1 dan
999999.")
		return
	}
	cetakDeret(n)
}
```

Output

```
PS C:\Users\ICHYA ULUMIDDIN\OneDrive\Document
ided3-Prosedur\Unguided3-Prosedur.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Penjelasan

Program terdiri dari dua bagian utama, yaitu fungsi cetakDeret(n) dan fungsi utama main. Fungsi cetakDeret(n) bertanggung jawab untuk menjalankan perhitungan berdasarkan aturan yang disebutkan di atas. Setiap bilangan dalam deret akan dicetak ke layar, dipisahkan oleh spasi, hingga mencapai angka 1, yang menjadi titik berhenti perhitungan.

Pada fungsi main, program meminta input bilangan bulat positif dari pengguna, dengan ketentuan bilangan harus berada di antara 1 dan 999999. Jika input tidak memenuhi batasan ini, program akan menampilkan pesan error dan langsung berhenti. Jika input valid, program akan memanggil fungsi cetakDeret(n) untuk mencetak deret angka yang dihasilkan sesuai aturan yang berlaku.

IV. KESIMPULAN

Dalam bahasa Golang, konsep prosedur sebenarnya tidak dipisahkan secara eksplisit seperti dalam beberapa bahasa pemrograman lain. Sebagai gantinya, fungsi digunakan baik untuk operasi yang mengembalikan nilai maupun yang tidak. Prosedur dalam Golang dapat diimplementasikan sebagai fungsi tanpa nilai kembalian (void function)**,** yang berperan dalam menjalankan serangkaian instruksi tanpa memberikan hasil kembali ke pemanggilnya.

Penggunaan prosedur dalam pemrograman bertujuan untuk mempermudah modularisasi kode, mengurangi duplikasi, serta meningkatkan keterbacaan dan efisiensi program. Dengan memisahkan tugas tertentu ke dalam prosedur terpisah, kode menjadi lebih mudah dikelola, diperbarui, dan digunakan kembali**.** Selain itu, pemisahan logika dalam bentuk prosedur juga membantu dalam debugging dan pengujian kode, karena memungkinkan pemrogram untuk mengisolasi bagian-bagian tertentu dari program.

Dalam praktiknya, prosedur digunakan dalam berbagai konteks, seperti mencetak output, memproses data, menangani input pengguna, dan mengontrol alur eksekusi program. Oleh karena itu, pemahaman tentang prosedur dalam Golang sangat penting untuk menulis kode yang lebih terstruktur, efisien, dan mudah dikembangkan dalam skala kecil maupun besar.

REFERENSI

Hadi. (2023, July 31). *Perbedaan Fungsi dan Prosedur*. Retrieved from wargamasyarakat: https://wargamasyarakat.org/perbedaan-fungsi-dan-prosedur/

Kartika. (2024, Desember 7). *Perbedaan Antara Fungsi dan Prosedur*. Retrieved from sridianti: https://www.sridianti.com/2024/perbedaan-antara-fungsi-dan-prosedur/