

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4
PROSEDUR**



Disusun Oleh :

Aby Hakim Al Yasiry Faozi/2311102208

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Prosedur adalah potongan kode yang dapat digunakan kembali untuk melakukan tugas tertentu. Tidak seperti kode linear tradisional, prosedur membantu memecah masalah kompleks menjadi bagian-bagian yang lebih mudah dikelola. Prosedur dapat dipanggil berkali-kali dalam suatu program, sehingga kode menjadi lebih modular dan mudah dibaca serta dipelihara.

Komponen:

- Nama: Setiap prosedur memiliki nama yang mengidentifikasinya.
- Parameter: Prosedur mungkin menerima input (parameter) yang mempengaruhi operasinya.
- Badan Prosedur: Ini adalah blok kode yang mendefinisikan fungsi prosedur.
- Tipe Kembali: Beberapa prosedur mengembalikan nilai setelah selesai, sementara yang lain tidak.

Jenis:

- Prosedur: Biasanya melakukan tugas tertentu tetapi tidak mengembalikan nilai.
- Fungsi: Melakukan tugas tertentu dan mengembalikan nilai.

Keuntungan:

- Penggunaan Ulang: Setelah didefinisikan, prosedur dapat digunakan berulang kali dalam program.
- Pemeliharaan: Perubahan yang dilakukan pada prosedur akan tercermin di mana pun prosedur itu dipanggil, mengurangi redundansi dan memudahkan pemeliharaan.
- Modularitas: Prosedur membantu dalam menyusun kode menjadi unit logis, meningkatkan keterbacaan dan organisasi.
- Abstraksi: Prosedur memungkinkan programmer untuk mengabstraksi operasi kompleks di balik panggilan prosedur yang sederhana.

Keamanan dan Validasi:

- Prosedur dapat dan seharusnya melakukan validasi pada input yang diterima untuk memastikan data yang diproses adalah valid.
- Memastikan keamanan data selama pemrosesan adalah aspek penting dalam merancang prosedur, terutama dalam aplikasi yang memerlukan autentikasi dan otorisasi.

II. GUIDED

1. Soal Studi Case

Sourcecode

```
package main

import "fmt"

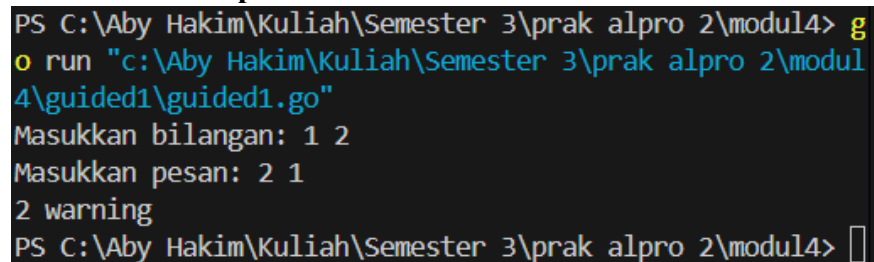
func main() {
    var bilangan int
    var pesan string

    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&bilangan)
    fmt.Print("Masukkan pesan: ")
    fmt.Scanln(&pesan)
    fmt.Scanln(&pesan)

    cetakPesan(pesan, bilangan)
}

func cetakPesan(M string, flag int) {
    var jenis string = ""
    switch flag {
    case 0:
        jenis = "eror"
    case 1:
        jenis = "warning"
    case 2:
        jenis = "informasi"
    default:
        jenis = "tidak dikenal"
    }
    fmt.Println(M, jenis)
}
```

Screenshoot Output



```
PS C:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4\guided1\guided1.go"
Masukkan bilangan: 1 2
Masukkan pesan: 2 1
2 warning
PS C:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4>
```

Deskripsi Program

Program ini merupakan program sederhana yang meminta input berupa bilangan dan pesan, lalu mencetak pesan tersebut dengan tambahan label berdasarkan bilangan yang dimasukkan. Berikut penjelasannya.

1. Deklarasi Variabel

- `bilangan`: Menampung input bilangan bulat.
- `pesan`: Menampung input pesan string.
- `jenis`: Menampung jenis label yang akan ditampilkan.

2. Meminta Input

- `fmt.Print("Masukkan bilangan: ")`: Menampilkan teks "Masukkan bilangan: " ke layar.
- `fmt.Scan(&bilangan)`: Membaca input bilangan dari pengguna dan menyimpannya ke variabel `bilangan`.
- `fmt.Print("Masukkan pesan: ")`: Menampilkan teks "Masukkan pesan: " ke layar.
- `fmt.Scanln(&pesan)`: Membaca input pesan dari pengguna dan menyimpannya ke variabel `pesan`.

3. Fungsi cetakPesan()

- Fungsi `cetakPesan(M string, flag int)` menerima dua argumen:
 - `M`: Pesan yang akan dicetak.
 - `flag`: Bilangan yang menentukan jenis label.
- Fungsi ini menggunakan switch-case untuk menentukan jenis label berdasarkan nilai `flag`.
 - case 0: Label "error".
 - case 1: Label "warning".
 - case 2: Label "informasi".
 - default: Label "tidak dikenal" jika `flag` tidak sesuai dengan case di atas.
- `fmt.Println(M, jenis)`: Mencetak pesan `M` dan label `jenis` ke layar.

4. Memanggil Fungsi cetakPesan()

- `cetakPesan(pesan, bilangan)`: Memanggil fungsi `cetakPesan()` dengan argumen `pesan` dan `bilangan` yang telah diinputkan.

2. Soal Studi Case

Sourcecode

```
package main

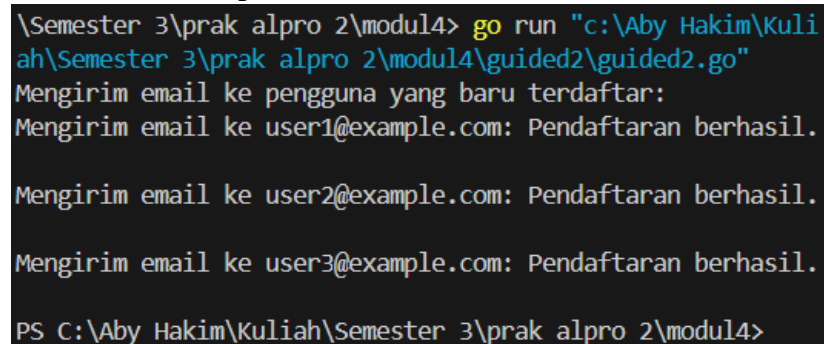
import "fmt"

// procedure untuk mengirim email
func sendEmailNotification(email string) {
    fmt.Printf("Mengirim email ke %s: Pendaftaran
berhasil.\n", email)
}

func main() {
    // Daftar email pengguna baru
    emails := []string{"user1@example.com",
"user2@example.com", "user3@example.com"}

    // Mengirim email pemberitahuan ke setiap pengguna
    fmt.Println("Mengirim email ke pengguna yang baru
terdaftar:")
    for _, email := range emails {
        sendEmailNotification(email)
    }
}
```

Screenshoot Output



```
\Semester 3\prak alpro 2\modul4> go run "c:\Aby Hakim\Kuli
ah\Semester 3\prak alpro 2\modul4\guided2\guided2.go"
Mengirim email ke pengguna yang baru terdaftar:
Mengirim email ke user1@example.com: Pendaftaran berhasil.
Mengirim email ke user2@example.com: Pendaftaran berhasil.
Mengirim email ke user3@example.com: Pendaftaran berhasil.
PS C:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4>
```

Deskripsi Program

1. Package dan Import:
 - Program ini dimulai dengan mendeklarasikan package utama main, dan mengimpor paket fmt yang digunakan untuk format I/O (Input/Output).
2. Prosedur sendEmailNotification
 - Prosedur ini menerima sebuah string email sebagai parameter.

- Di dalam prosedur, ada perintah `fmt.Printf` yang mencetak pesan ke layar, menyatakan bahwa email telah dikirim ke alamat email yang diterima sebagai parameter. Pesannya adalah "Mengirim email ke {email}: Pendaftaran berhasil."

3. Fungsi main:

- Fungsi utama dari program di mana eksekusi dimulai.
- Mendeklarasikan slice emails yang berisi daftar email pengguna baru. Dalam hal ini, slice tersebut berisi tiga alamat email: `user1@example.com`, `user2@example.com`, dan `user3@example.com`.
- Menampilkan pesan "Mengirim email ke pengguna yang baru terdaftar:" ke layar dengan menggunakan `fmt.Println`.
- Menggunakan loop `for` untuk iterasi melalui setiap email dalam slice emails. Untuk setiap email, prosedur `sendEmailNotification` dipanggil, yang mengirim (menampilkan) pesan email notifikasi ke pengguna terkait.

3. Soal Studi Case

Sourcecode

```
package main

import "fmt"

func f1(x, y int) float64 { // pass by value
    var hasil float64
    hasil = float64(2*x) - 0.5*float64(y) + 3.0
    return hasil
}

func f2(x, y int, hasil *float64) { //pas by reference
    *hasil = float64(2*x) - 0.5*float64(y) + 3.0
}

func main() {
    var a, b int
    var c float64

    // Take input for a and b
    fmt.Print("Enter two integers: ")
    fmt.Scan(&a, &b)

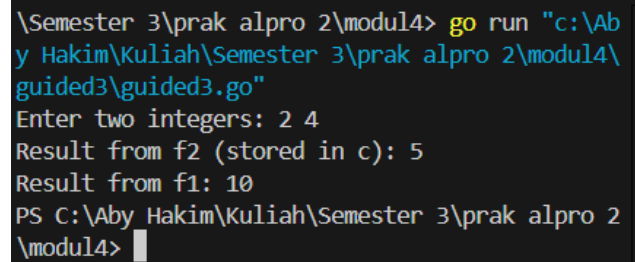
    // Call f2 to calculate and store result in
    f2(a, b, &c)

    // Print the result from f2
```

```
fmt.Println("Result from f2 (stored in c):", c)

// Call f1 and print the result
resultF1 := f1(b, a)
fmt.Println("Result from f1:", resultF1)
}
```

Screenshot Output



```
\Semester 3\prak alpro 2\modul4> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4\guided3\guided3.go"
Enter two integers: 2 4
Result from f2 (stored in c): 5
Result from f1: 10
PS C:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4>
```

Deskripsi Program

Algoritma

1. Input Pengguna: Pengguna diminta untuk memasukkan dua bilangan bulat a dan b.
2. Perhitungan dengan Fungsi f2:
 - f2 menerima nilai a dan b, serta pointer ke variabel float64 hasil.
 - Fungsi ini menghitung hasil menggunakan rumus:
$$\text{hasil} = 2 \times a - 0.5 \times b + 3.0$$
 - Hasil perhitungan disimpan dalam variabel c melalui pointer hasil.
3. Menampilkan Hasil dari f2: Nilai yang disimpan dalam c (hasil dari f2) ditampilkan.
4. Perhitungan dengan Fungsi f1:
 - f1 menerima nilai b dan a (parameter ditukar).
 - Fungsi ini juga menghitung hasil menggunakan rumus yang sama:
$$\text{hasil} = 2 \times b - 0.5 \times a + 3.0$$
 - Hasil perhitungan dikembalikan sebagai nilai float64.
5. Menampilkan Hasil dari f1

III. UNGUIDED

1. Soal Studi Case

Sourcecode

```
package main

import (
    "fmt"
)

func faktorial(n int) int {
    if n == 0 {
        return 1
    }
    return n * faktorial(n-1)
}

func Permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

func combination(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}

func main() {
    var a, b, c, d int

    fmt.Print("Masukkan nilai a (bilangan asli): ")
    fmt.Scan(&a)

    fmt.Print("Masukkan nilai b (bilangan asli): ")
    fmt.Scan(&b)

    fmt.Print("Masukkan nilai c (dengan syarat a >= c): ")
    fmt.Scan(&c)

    fmt.Print("Masukkan nilai d (dengan syarat b >= d): ")
    fmt.Scan(&d)

    fmt.Println("Keluaran:")
    fmt.Println("Permutasi(a, c):", Permutasi(a, c),
    "Kombinasi(a, c):", combination(a, c))
    fmt.Println("Permutasi(b, d):", Permutasi(b, d),
    "Kombinasi(b, d):", combination(b, d))
}
```

Screenshoot Output

```
\Semester 3\prak alpro 2\modul4> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4\unguided1\unguided1.go"
Masukkan nilai a (bilangan asli): 5 10 3 10
Masukkan nilai b (bilangan asli): Masukkan nilai c (dengan syarat a >= c):
Masukkan nilai d (dengan syarat b >= d): Keluaran:
Permutasi(a, c): 60 Kombinasi(a, c): 10
Permutasi(b, d): 3628800 Kombinasi(b, d): 1
PS C:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4>
```

Deskripsi Program

Program ini bertujuan untuk menghitung permutasi dan kombinasi dari bilangan yang dimasukkan oleh pengguna. Berikut adalah langkah-langkah program ini:

1. Fungsi Faktorial:
 - faktorial(n int) int: Menghitung faktorial dari bilangan n secara rekursif. Jika n adalah 0, maka mengembalikan 1. Jika tidak, mengalikan n dengan faktorial dari n-1.
2. Fungsi Permutasi:
 - Permutasi(n, r int) int: Menghitung permutasi dari n objek yang diambil r objek pada satu waktu. Ini dilakukan dengan membagi faktorial n dengan faktorial (n-r).
3. Fungsi Kombinasi:
 - combination(n, r int) int: Menghitung kombinasi dari n objek yang diambil r objek pada satu waktu. Ini dilakukan dengan membagi faktorial n dengan hasil kali dari faktorial r dan faktorial (n-r).
4. Fungsi Utama:
 - Mendeklarasikan variabel a, b, c, dan d sebagai bilangan bulat.
 - Meminta pengguna untuk memasukkan nilai a, b, c, dan d.
 - Menampilkan hasil permutasi dan kombinasi menggunakan fungsi Permutasi dan combination untuk nilai a dan c, serta nilai b dan d.

2. Soal Studi Case

Sourcecode

```
package main

import (
    "fmt"
)

func hitungSkor(soal []int, skor *int) int {
    totalSkor := 0
    totalSoal := 0
    for _, waktu := range soal {
        if waktu < 301 {
            totalSkor += waktu
            totalSoal++
        }
    }
    *skor = totalSkor
    return totalSoal
}

func main() {
    var peserta1, peserta2 [8]int
    var skor1, skor2 int

    fmt.Println("Masukkan waktu pengerjaan soal Astuti:")
    fmt.Scanln(&peserta1[0], &peserta1[1], &peserta1[2],
        &peserta1[3], &peserta1[4], &peserta1[5], &peserta1[6],
        &peserta1[7])

    fmt.Println("Masukkan waktu pengerjaan soal Bertha:")
    fmt.Scanln(&peserta2[0], &peserta2[1], &peserta2[2],
        &peserta2[3], &peserta2[4], &peserta2[5], &peserta2[6],
        &peserta2[7])

    soalTerselesaikan1 := hitungSkor(peserta1[:],
        &skor1)
    soalTerselesaikan2 := hitungSkor(peserta2[:],
        &skor2)

    fmt.Println("Keluaran:")

    if soalTerselesaikan1 > soalTerselesaikan2 {
        fmt.Printf("Pemenang: Astuti\nJumlah soal diselesaikan: %d\nTotal waktu: %d\n",
            soalTerselesaikan1, skor1)
    } else if soalTerselesaikan2 > soalTerselesaikan1 {
        fmt.Printf("Pemenang: Bertha\nJumlah soal diselesaikan: %d\nTotal waktu: %d\n",
            soalTerselesaikan2, skor2)
    }
```

```

    } else {
        if skor1 < skor2 {
            fmt.Printf("Pemenang: Astuti\nJumlah soal
diselesaikan: %d\nTotal waktu: %d\n",
soalTerselesaikan1, skor1)
        } else if skor2 < skor1 {
            fmt.Printf("Pemenang: Bertha\nJumlah soal
diselesaikan: %d\nTotal waktu: %d\n",
soalTerselesaikan2, skor2)
        } else {
            fmt.Println("Hasil seri!")
        }
    }
}
}

```

Screenshoot Output

```

\Semester 3\prak alpro 2\modul4> go run "
c:\Aby Hakim\Kuliah\Semester 3\prak alpro
2\modul4\unguided2\unguided2.go"
Masukkan waktu pengerjaan soal Astuti:
20 50 301 301 61 71 75 10
Masukkan waktu pengerjaan soal Bertha:
25 47 301 26 50 60 65 21
Keluaran:
Pemenang: Bertha
Jumlah soal diselesaikan: 7
Total waktu: 294
PS C:\Aby Hakim\Kuliah\Semester 3\prak al
pro 2\modul4>

```

Deskripsi Program

Program ini bertujuan untuk menentukan pemenang dari dua peserta berdasarkan waktu pengerjaan soal-soal mereka. Berikut adalah penjelasan langkah-langkahnya:

1. Fungsi hitungSkor:

- Menerima array soal yang berisi waktu pengerjaan soal dari peserta, dan pointer skor untuk menyimpan total skor.
- Menginisialisasi totalSkor dan totalSoal dengan nilai 0.
- Menggunakan loop untuk memeriksa setiap waktu dalam array soal. Jika waktu kurang dari 301, tambahkan ke totalSkor dan tingkatan totalSoal.
- Menyimpan hasil perhitungan di variabel yang ditunjuk oleh skor dan mengembalikan jumlah soal yang terselesaikan (totalSoal).

2. Fungsi Utama (main):

- Mendeklarasikan variabel untuk menyimpan waktu pengerjaan soal dari dua peserta (peserta1 dan peserta2) serta skor masing-masing (skor1 dan skor2).
- Meminta pengguna untuk memasukkan waktu pengerjaan soal untuk Astuti dan Bertha.
- Menghitung jumlah soal yang terselesaikan dan skor total untuk masing-masing peserta menggunakan fungsi hitungSkor.
- Menampilkan hasil perhitungan dan menentukan pemenang berdasarkan jumlah soal yang terselesaikan. Jika jumlah soal yang terselesaikan sama, pemenang ditentukan berdasarkan skor total yang lebih rendah (waktu pengerjaan lebih cepat). Jika hasilnya sama, maka dinyatakan seri.

3. Soal Studi Case

Sourcecode

```
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.Print("Masukan: ")
    fmt.Scanln(&n)
    cetakDeret(n)
}
```

Screenshoot Output

```
\Semester 3\prak alpro 2\modul4> go run "c:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4\unguided3\unguided3.go"
Masukan: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\Aby Hakim\Kuliah\Semester 3\prak alpro 2\modul4> █
```

Deskripsi Program

Program ini bertujuan untuk mencetak deret angka berdasarkan aturan tertentu, yang dikenal sebagai Collatz Conjecture atau $3n + 1$ problem. Berikut adalah penjelasan langkah-langkahnya:

1. Fungsi cetakDeret:

- Menerima input n berupa bilangan bulat.
- Menggunakan loop for untuk mencetak nilai n selama n tidak sama dengan 1.
- Jika n adalah bilangan genap, maka n dibagi 2.
- Jika n adalah bilangan ganjil, maka n dihitung dengan rumus $3*n + 1$.
- Setelah keluar dari loop, mencetak angka 1 sebagai akhir dari deret.

2. Fungsi Utama (main):

- Mendeklarasikan variabel n untuk menyimpan input dari pengguna.
- Meminta pengguna untuk memasukkan nilai n .
- Memanggil fungsi cetakDeret dengan nilai n yang dimasukkan oleh pengguna.