

LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2
MODUL IV
PROSEDUR



DISUSUN OLEH

Rizkulloh Alpriansah

2311102142

DOSEN PENGAMPUH

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM PURWOKERTO
2024

I. DASAR TEORI

1. Pengertian Prosedur

Prosedur adalah sekumpulan instruksi program yang dikelompokkan menjadi satu kesatuan. Prosedur bertujuan untuk menyederhanakan kodeprogram yang kompleks, membuatnya lebih terstruktur dan mudah dipahami. Ketika dipanggil, prosedur dapat memengaruhi program secara langsung. Dalam konteks ini, prosedur tidak memiliki nilai kembalian dan tidak menggunakan kata kunci return.

2. Deklarasi Prosedur

Deklarasi prosedur melibatkan penamaan prosedur, mendeklarasikan parameter yang diterima, dan menyusun logika yang akan dijalankan. Berikut adalah contoh deklarasi prosedur dalam pseudocode dan go.

Notasi dalam Go

```
func cetakPesan(flag int, M string) {  
    var jenis string  
    if flag == 0 {  
        jenis = "error"  
    } else if flag == 1 {  
        jenis = "warning"  
    } else {  
        jenis = "informasi"  
    }  
    fmt.Println(M, jenis)  
}
```

3. Pemanggilan Prosedur

Prosedur dipanggil dalam program utama untuk menjalankan logika yang telah didefinisikan. Berikut adalah cara pemanggilan prosedur dalam program

Notasi dalam Go

```
package main  
  
import "fmt"  
  
func main() {  
    var bilangan int  
    var pesan string
```

```
cetakPesan(bilangan, pesan)
}
```

4. Contoh Program

Berikut adalah contoh program lengkap yang menggunakan prosedur untuk menampilkan pesan berdasarkan input dari pengguna.

Berikut contoh pada GoLang:

```
package main

import "fmt"

func main() {
    var bilangan int
    var pesan string

    fmt.Print("Masukkan flag (0-2) dan pesan: ")

    fmt.Scan(&bilangan, &pesan)
    cetakPesan(bilangan, pesan)
}

func cetakPesan(flag int, M string) {
    var jenis string

    if flag == 0 {
        jenis = "error"
    } else if flag == 1 {
        jenis = "warning"
    }
}
```

5. Parameter

Parameter dalam prosedur memungkinkan komunikasi antara prosedur dan bagian lain dari program. Parameter dapat dibagi menjadi dua kategori: parameter formal, yang dideklarasikan dalam prosedur, dan parameter aktual, yang digunakan saat pemanggilan.

II. Guided

Guided 1

```
//Rizkulloh Alpriyansah 2311102142
package main

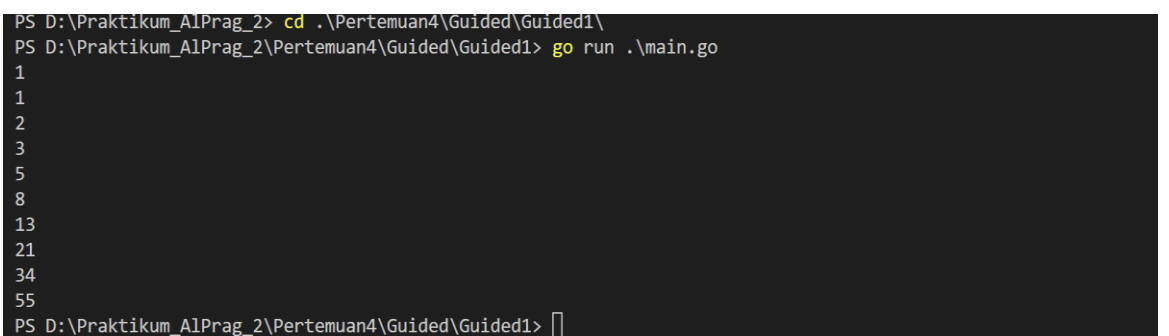
import "fmt"

func cetakFibo(n int) {
    var f1, f2, f3 int
    f2 = 0
    f3 = 1

    for i := 1; i <= n; i++ {
        fmt.Println(f3)
        f1 = f2
        f2 = f3
        f3 = f1 + f2
    }
}

func main() {
    cetakFibo(10)
}
```

Screenshot Output



```
PS D:\Praktikum_AlPrag_2> cd .\Pertemuan4\Guided\Guided1\
PS D:\Praktikum_AlPrag_2\Pertemuan4\Guided\Guided1> go run .\main.go
1
1
2
3
5
8
13
21
34
55
PS D:\Praktikum_AlPrag_2\Pertemuan4\Guided\Guided1> 
```

Deskripsi

Program di atas adalah implementasi fungsi Fibonacci dalam bahasa Go, yang mencetak deret Fibonacci hingga nilai ke-n. Fungsi cetakNFibo menerima satu parameter, n, yang menentukan jumlah angka Fibonacci yang akan ditampilkan. Di dalam fungsi, variabel f1, f2, dan f3 digunakan untuk menyimpan nilai Fibonacci saat ini dan dua nilai sebelumnya. Nilai awal f2 diatur ke 0 dan f3 ke 1. Melalui sebuah loop, program mencetak nilai f3 yang merupakan angka Fibonacci saat ini, lalu memperbarui nilai f1, f2, dan f3 untuk menghitung angka Fibonacci berikutnya. Di dalam fungsi main, fungsi cetakNFibo dipanggil dengan argumen 10, sehingga program mencetak sepuluh angka pertama dari deret Fibonacci.

Guided 2

```
package main

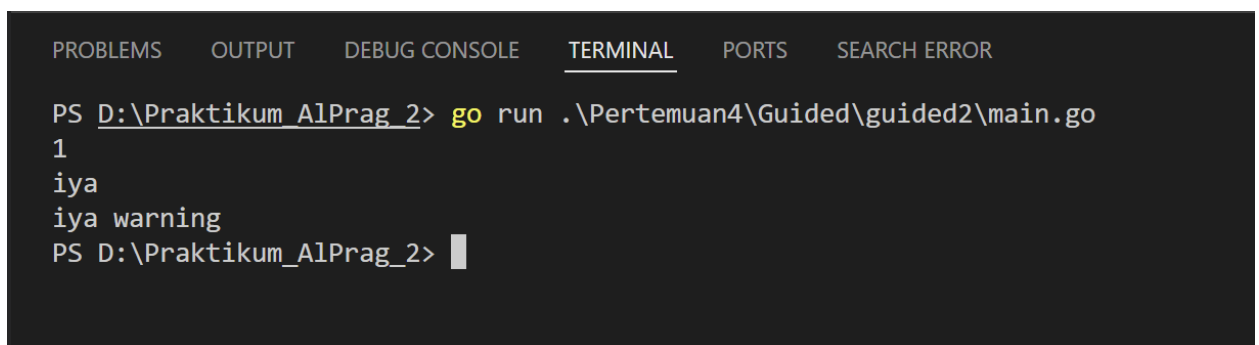
import "fmt"

func main() {
    var bilangan int
    var pesan string
    fmt.Scan(&bilangan)
    fmt.Scan(&pesan)
    cetakPesan(pesan, bilangan) //pemanggilan prosedur langsung
}

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)  
}
```

Screenshot Output



The screenshot shows a Go IDE interface with a terminal window. The terminal displays the command to run a Go program and its output. The command is `go run .\Pertemuan4\Guided\guided2\main.go`. The output consists of three lines: `1`, `iya`, and `iya warning`. The prompt `PS D:\Praktikum_AIPrag_2>` is visible at the end of the output.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  
  
PS D:\Praktikum_AIPrag_2> go run .\Pertemuan4\Guided\guided2\main.go  
1  
iya  
iya warning  
PS D:\Praktikum_AIPrag_2> █
```

Deskripsi

Program ini menerima dua input dari pengguna, yaitu sebuah bilangan (flag) dan sebuah pesan (M). Berdasarkan nilai flag, program akan mencetak pesan yang disertai dengan jenis pesan tertentu: "error" jika flag bernilai 0, "warning" jika 1, dan "informasi" jika 2. Pertama, program meminta pengguna memasukkan nilai untuk bilangan (yang berfungsi sebagai flag) dan pesan. Kemudian, fungsi cetakPesan akan memeriksa nilai flag dan mencetak pesan yang sesuai dengan jenis pesan yang telah ditentukan.

Guided3

Source Code

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Praktikum_AlPrag_2> go run .\Pertemuan4\Guided\Guided3\main.go
volume tabung pertama: 785
volume tabung kedua (gabungan): 7850
volume tabung dengan jari-jari 14 dan tinggi 100: 61544.00000000001
PS D:\Praktikum_AlPrag_2> █
```

Deskripsi

Program ini menghitung volume tabung berdasarkan jari-jari dan tinggi yang diberikan, dengan menggunakan fungsi `volumeTabung`. Fungsi ini menerima dua parameter, yaitu `jari_jari` dan `tinggi`, kemudian menghitung luas alas tabung menggunakan rumus $\pi * r^2$ dan mengalikannya dengan tinggi untuk mendapatkan volume. Program utama (`main`) mendeklarasikan dua variabel `r` dan `t` untuk jari-jari dan tinggi tabung. Volume tabung pertama dihitung dengan nilai jari-jari 5 dan tinggi 10, kemudian volume kedua adalah penjumlahan dari dua tabung (satu dengan jari-jari 5 dan satu lagi dengan jari-jari 15). Terakhir, volume untuk tabung dengan jari-jari 14 dan tinggi 100 juga dihitung. Outputnya menampilkan volume tabung pertama, volume gabungan dari dua tabung, dan volume tabung dengan ukuran yang diberikan.

Guided4

Source Code

```
package main

import "fmt"

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

```

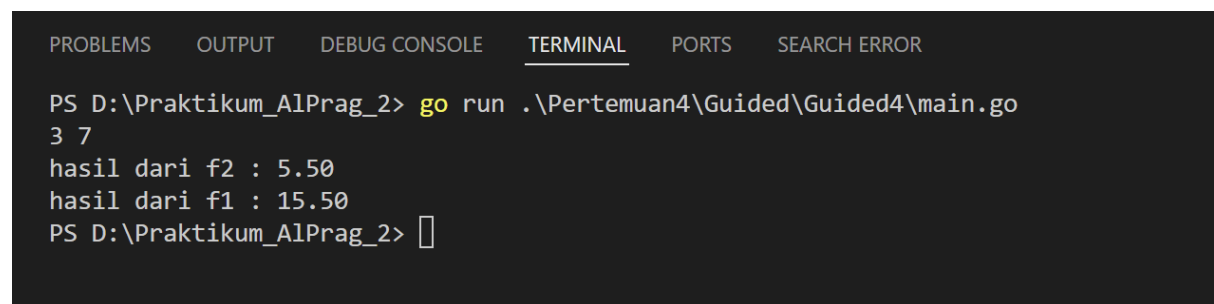
func f2(x, y int, hasil *float64) {
    *hasil = float64(2*x) - 0.5*float64(y) + 3.0
}

func output(c float64, d float64) {
    fmt.Printf("hasil dari f2 : %.2f\n", c)
    fmt.Printf("hasil dari f1 : %.2f\n", d)
}

func main() {
    var a, b int
    var c float64
    fmt.Scan(&a, &b)
    f2(a, b, &c)
    output(c, f1(b, a))
}

```

Screenshot



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Praktikum_AIPrag_2> go run .\Pertemuan4\Guided\Guided4\main.go
3 7
hasil dari f2 : 5.50
hasil dari f1 : 15.50
PS D:\Praktikum_AIPrag_2> 

```

Deskripsi

Program ini mengimplementasikan dua fungsi untuk melakukan perhitungan matematika sederhana, yaitu f1 dan f2, yang keduanya menggunakan operasi aritmatika berbasis dua input (bilangan x dan y). Fungsi f1 mengembalikan nilai hasil perhitungan langsung, sementara f2 menggunakan parameter pointer (*hasil) untuk mengembalikan hasil melalui referensi. Fungsi output bertugas menampilkan hasil dari f2 dan f1 dalam format desimal dua angka di belakang

koma. Pada fungsi main, program membaca dua bilangan integer dari input pengguna (a dan b), lalu memanggil fungsi f2 untuk menghitung hasil dan menyimpannya dalam variabel c menggunakan pointer. Kemudian, hasil dari fungsi f1 dengan argumen yang dibalik ditampilkan melalui output.

UNGUIDED

Unguided1

```
//Rizkulloh Alpriyansah
//2311102142
package main

import (
    "fmt"
)

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

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

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

```

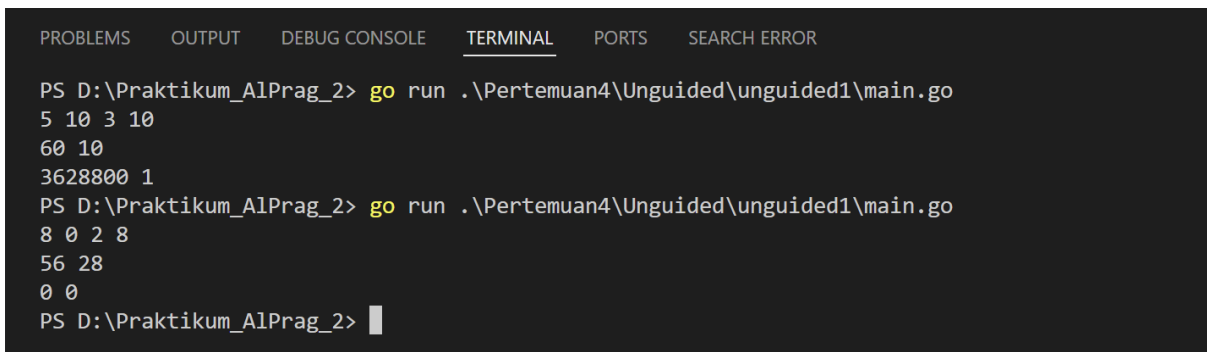
}

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

    fmt.Printf("%d %d\n", permutasi(a, c), kombinasi(a, c))
    fmt.Printf("%d %d\n", permutasi(b, d), kombinasi(b, d))
}

```

Screenshot Output



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Praktikum_AIPrag_2> go run .\Pertemuan4\Unguided\unguided1\main.go
5 10 3 10
60 10
3628800 1
PS D:\Praktikum_AIPrag_2> go run .\Pertemuan4\Unguided\unguided1\main.go
8 0 2 8
56 28
0 0
PS D:\Praktikum_AIPrag_2>

```

Deskripsi

Program ini menentukan pemenang kompetisi soal dengan menghitung berapa soal yang berhasil diselesaikan dan waktu total pengerjaannya. Peserta dianggap menyelesaikan soal kalau waktu pengerjaannya ≤ 300 menit. Program menerima input nama peserta dan waktu untuk 8 soal, terus loop berlanjut sampai nama "Selesai" dimasukkan. Pemenang ditentukan dari jumlah soal terbanyak yang selesai. Kalau ada yang sama, waktu pengerjaan total yang lebih cepat menang. Hasil akhir menampilkan nama pemenang, jumlah soal selesai, dan total waktu pengerjaan.

Unguided2

Source Code

```
package main

import (
    "fmt"
    "strings"
)

type Peserta struct {
    nama      string
    totalSoal int
    totalWaktu int
}

func hitungSkor(soal []int) (int, int) {
    totalSoal := 0
    totalWaktu := 0

    for _, waktu := range soal {
        totalWaktu += waktu
    }

    for _, waktu := range soal {
        if waktu > 0 {
            totalSoal++
        }
    }
}
```

```
    if totalWaktu > 300 {
        return 0, 0
    }

    return totalSoal, totalWaktu
}

func main() {
    var nama string
    var waktuTemp int
    var soal []int

    var pemenang Peserta
    maxSoal := 0

    for {
        fmt.Scan(&nama)

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

        soal = make([]int, 8)

        for i := 0; i < 8; i++ {
            fmt.Scan(&waktuTemp)
            soal[i] = waktuTemp
        }
    }
}
```

```

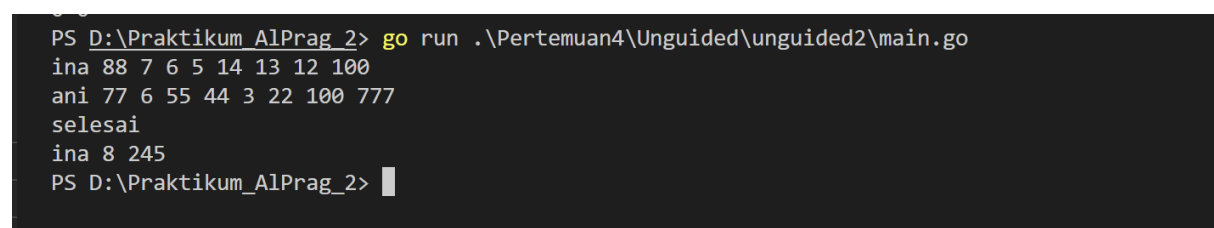
totalSoal, totalWaktu := hitungSkor(soal)

    if totalSoal > maxSoal || (totalSoal == maxSoal && totalWaktu <
pemenang.totalWaktu) {
        pemenang = Peserta{nama, totalSoal, totalWaktu}
        maxSoal = totalSoal
    }
}

if pemenang.totalSoal > 0 {
    fmt.Printf("%s  %d  %d\n", pemenang.nama, pemenang.totalSoal,
pemenang.totalWaktu)
} else {
    fmt.Println("Tidak ada peserta yang valid.")
}
}

```

Screenshot Output



```

PS D:\Praktikum_AlPrag_2> go run .\Pertemuan4\Unguided\unguided2\main.go
ina 88 7 6 5 14 13 12 100
ani 77 6 55 44 3 22 100 777
selesai
ina 8 245
PS D:\Praktikum_AlPrag_2>

```

Deskripsi

Program ini menentukan pemenang kompetisi berdasarkan jumlah soal yang diselesaikan dan total waktu pengerjaan. Setiap peserta memasukkan waktu untuk 8 soal, lalu program menghitung berapa soal yang selesai dan waktu totalnya. Kalau waktu pengerjaan soal lebih dari 300 menit, soal dianggap tidak selesai. Pemenang ditentukan dari jumlah soal terbanyak, dan kalau jumlahnya sama, waktu pengerjaan total yang lebih cepat akan menang. Program terus menerima

data peserta sampai nama "Selesai" dimasukkan, dan di akhir, program menampilkan nama pemenang, jumlah soal selesai, dan total waktu pengerjaan.

Unguided3

Source Code

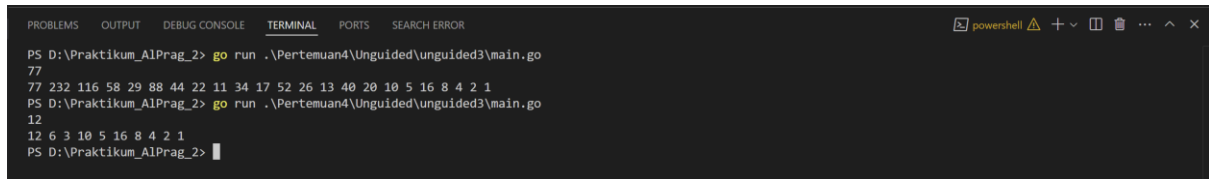
```
package main

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.Print(n)
}

func main() {
    var n int
    fmt.Scan(&n)
    if n > 0 && n < 1000000 {
        cetakDeret(n)
    } else {
        fmt.Println("Masukkan bilangan positif kurang dari 1.000.000")
    }
}
```

Screenshot Output



```
PS D:\Praktikum_AlPrag_2> go run .\Pertemuan4\Unguided\unguided3\main.go
77
77 232 116 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\Praktikum_AlPrag_2> go run .\Pertemuan4\Unguided\unguided3\main.go
12
12 6 3 10 5 16 8 4 2 1
PS D:\Praktikum_AlPrag_2> █
```

Deskripsi

Program ini mencetak deret bilangan sesuai aturan: kalau bilangan genap, dibagi dua; kalau ganjil, dikali tiga terus ditambah satu. Prosesnya terus diulang sampai bilangan jadi 1. Di bagian main, pengguna diminta input bilangan n yang dicek apakah nilainya antara 1 sampai 999.999. Kalau valid, fungsi cetakDeret bakal dijalankan untuk mencetak deret sesuai aturan tadi. Kalau nggak valid, program bakal kasih pesan buat masukin bilangan yang sesuai.

IV. DAFTAR PUSTAKA

1. Asisten praktikum, Akmelia Zahara dan Kyla Azzahra Kinan “Modul IV Prosedur” Learning Management System, 2024