

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROCEDURE**



Disusun Oleh :

Arjun Ahmad Santoso / 2311102211

S1IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

4.1 Definisi Prosedur

Prosedur adalah sebuah blok program yang dapat digunakan untuk menjalankan suatu langkah-langkah tertentu. Prosedur dapat mengambil parameter seperti fungsi pada umumnya. Namun prosedur tidak memiliki nilai return seperti fungsi.

Suatu sub-program dikatakan prosedur jika tidak ada deklarasi tipe nilai yang dikembalikan dan tidak ada kata kunci return di dalam badan sub program tersebut.

4.2 Pendeklarasian Prosedur

Pendeklarasian Prosedur hampir sama dengan pendeklarasian fungsi pada umumnya yang membedakan hanyalah tidak terdapat tipe nilai yang dikembalikan karena prosedur tidak mereturn nilai.

4.3 Pemanggilan Prosedur

Sama seperti fungsi pada umumnya, prosedur dapat dipanggil dengan menuliskan namanya pada baris program dan memasukkan argumen(parameter aktual) jika diperlukan. Parameter (parameter formal) adalah kata kunci yang dituliskan dalam pendeklarasian fungsi atau prosedur, sedangkan argumen(parameter aktual) adalah nilai yang dimasukkan untuk parameter tersebut saat pemanggilan fungsi atau prosedur.

II. GUIDED

1.

Soal Guided

Soal Guided No. 1

Sourcecode

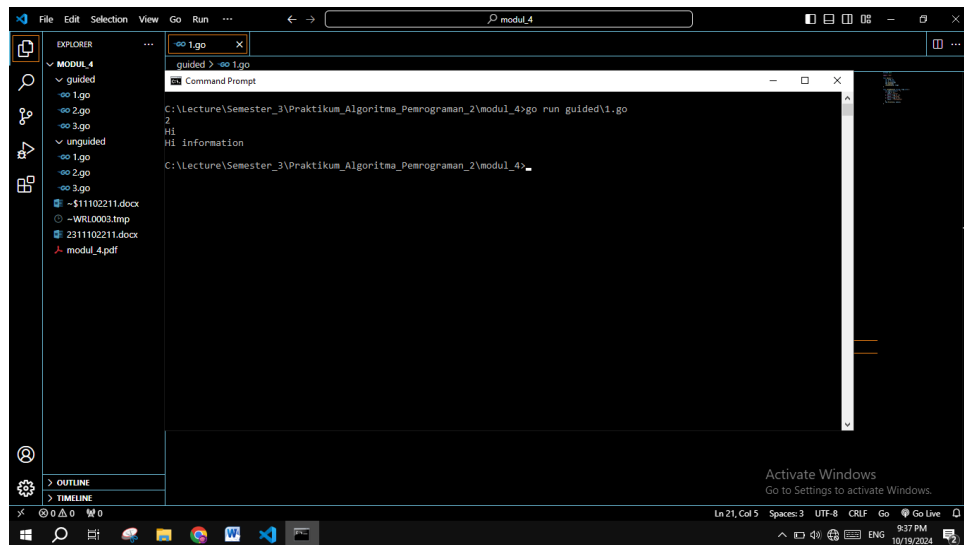
```
package main

import "fmt"

func main() {
    var flag int
    var m string
    fmt.Scan(&flag)
    fmt.Scan(&m)
    cetakPesan(m, flag)
}

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 = "information"
    }
    fmt.Println(m, jenis);
}
```

Screenshoot Output



Deskripsi Program

Program di atas memuat sebuah fungsi cetakPesan yang dapat digunakan untuk mencetak pesan bersama dengan tipe pesannya.

2.

Soal Guided

Soal Guided No. 2

Sourcecode

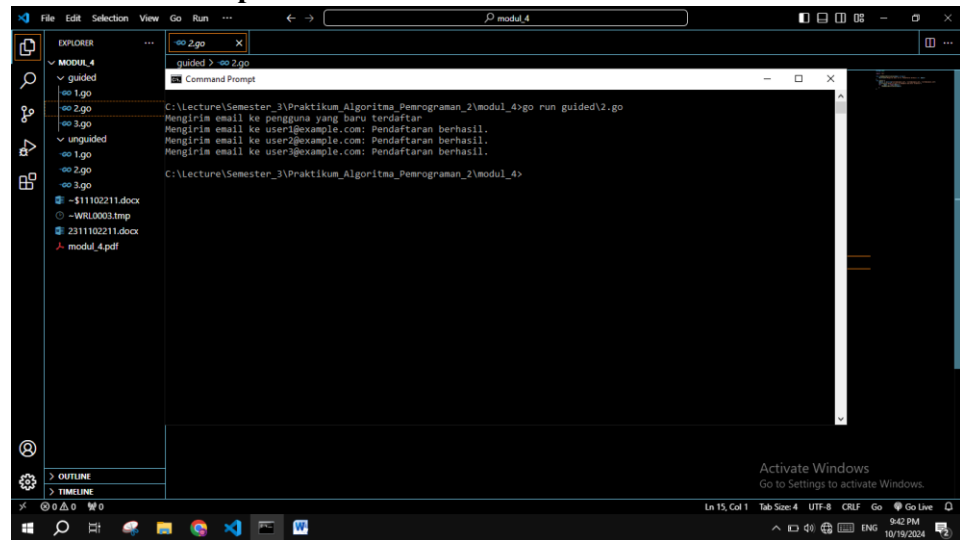
```
package main

import "fmt"

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

func main() {
    emails := []string{"user1@example.com",
"user2@example.com", "user3@example.com"}
    fmt.Println("Mengirim email ke pengguna yang baru terdaftar")
    for _, email := range emails {
        sendEmailNotification(email)
    }
}
```

Screenshoot Output



Deskripsi Program

Program di atas memuat sebuah fungsi `sendEmailNotification` yang berfungsi untuk mencetak pesan pendaftaran berhasil.

3.

Soal Guided

Soal Guided No. 2

Sourcecode

```
package main

import "fmt"

func f1(a, b int) float64 {
    result := 2 * float64(a) - 0.5 * float64(b) + 0.3
    return result
}

func f2(a, b int, c *float64) {
    *c = 2 * float64(a) - 0.5 * float64(b) + 0.3
}

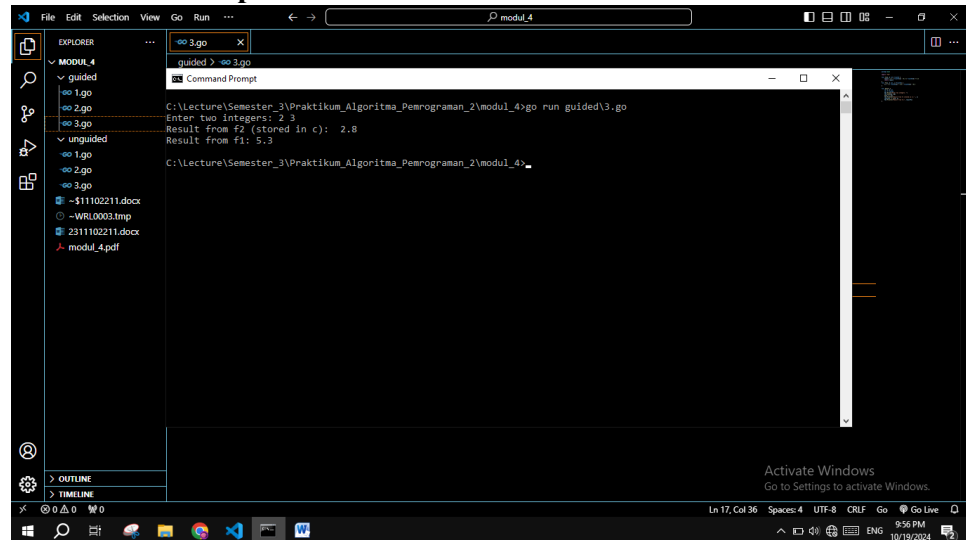
func main() {
    var a, b int
    var c float64
    fmt.Print("Enter two integers: ")
    fmt.Scan(&a, &b)
```

```

    f2(a, b, &c)
    fmt.Println("Result from f2 (stored in c): ", c)
    resultF1 := f1(b, a)
    fmt.Println("Result from f1:", resultF1)
}

```

Screenshoot Output



Deskripsi Program

Program di atas memuat dua fungsi, *f1* dan *f2*. Semua parameter pada *f1* bertipe *value*, sedangkan *f2* memiliki satu parameter bertipe *pointer*. Program diatas menunjukkan perbedaan *pass by value* dan *pass by pointer/pass by reference* secara umum.

III. UNGUIDED

1.

Soal Latihan Modul 4

Soal Latihan Modul 4 No. 1

Sourcecode

```
package main

import "fmt"

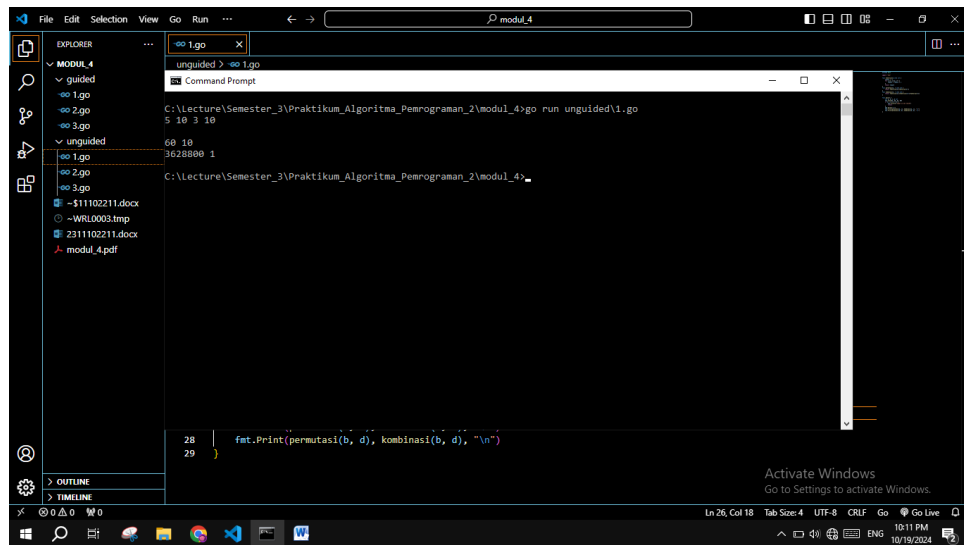
func faktorial(a int) int {
    result := 1
    for i:=1; i<=a; i++ {
        result = result * i
    }
    return result
}

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

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

func main() {
    var a, b, c, d int
    fmt.Scan(&a, &b, &c, &d)
    if a < c || b < d {
        fmt.Println("Input is not valid")
        return
    }
    fmt.Print("\n")
    fmt.Print(permutasi(a, c), kombinasi(a, c), "\n")
    fmt.Print(permutasi(b, d), kombinasi(b, d), "\n")
}
```

Screenshoot Output



Deskripsi Program

Program di atas memuat sebuah fungsi faktorial, permutasi, dan kombinasi yang dapat digunakan.

2.

Soal Latihan Modul 4

Soal Latihan Modul 4 No.2

Sourcecode

```
package main

import "fmt"

func hitungSkor(jumlah_soal_selesai *int, skor *int) bool {
    const jumlah_soal int = 8
    var jumlah_soal_selesai_lokal int = jumlah_soal
    var t = [jumlah_soal] int {}
    skor_lokal := 0
    for i := 0; i < jumlah_soal; i++ {
        fmt.Scan(&t[i])
        if t[i] != 301 {
            skor_lokal += t[i]
        } else {
            jumlah_soal_selesai_lokal --
        }
    }
    if jumlah_soal_selesai_lokal > *jumlah_soal_selesai {
        *skor = skor_lokal
    }
}
```



```

        *jumlah_soal_selesai = jumlah_soal_selesai_lokal
        return true
    } else if jumlah_soal_selesai_lokal ==
*jumlah_soal_selesai {
        if skor_lokal > *skor {
            *skor = skor_lokal
            *jumlah_soal_selesai = jumlah_soal_selesai_lokal
            return true
        }
    }
    return false
}

func main() {
    var jumlah_soal_selesai, skor int // variabel untuk
pemenang
    var nama_pemenang, nama string
    for {
        fmt.Scan(&nama)
        if nama == "Selesai" { break }
        if hitungSkor(&jumlah_soal_selesai, &skor) {
            nama_pemenang = nama
        }
    }
    fmt.Print("\n")
    fmt.Println(nama_pemenang, jumlah_soal_selesai, skor)
}

```

Screenshoot Output

```

File Edit Selection View Go Run ...
modul_4
EXPLORER
  modul_4
    guided
    1.go
    2.go
    3.go
    unguided
    1.go
    2.go
    3.go
    4.go
    11100211.docx
    WA10003.tmp
    231100211.docx
    modul_4.pdf
  OUTLINE
  TIMELINE
Command Prompt
C:\Lecture\Semester_3\Praktikum_Algoritma_Pemrograman_2\modul_4>go run unguided\2.go
Astuti 20 50 301 201 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294
C:\Lecture\Semester_3\Praktikum_Algoritma_Pemrograman_2\modul_4>
44 }

```

Activate Windows
Go to Settings to activate Windows.

Ln 31, Col 1 Tab Size 4 UTF-8 CRLF Go Go Live
10/19/2024

Deskripsi Program

Program di atas berfungsi untuk menentukan pemenang suatu kompetisi pemrograman berdasarkan jumlah soal yang diselesaikan setiap peserta dan waktu yang diperlukan untuk menyelesaikan soal-soal tersebut.

3.

Soal Latihan Modul 4

Soal Latihan Modul 4 No.3

Sourcecode

```
package main

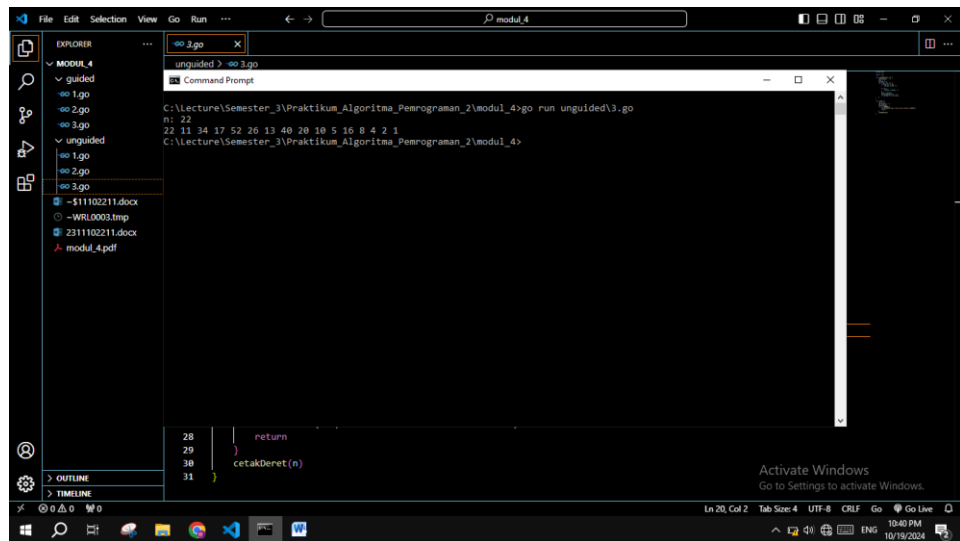
import "fmt"

func cetakDeret(n int) {
    fmt.Print(n, " ")
    prev := n
    current := n
    for i:=0; i<n; i++ {
        if prev % 2 == 0 {
            current = prev / 2

        } else {
            current = 3*prev + 1
        }
        prev = current
        fmt.Print(current, " ")
        if current == 1 { break }
    }
}

func main() {
    var n int
    fmt.Print("n: ")
    fmt.Scan(&n)
    if n > 1000000 {
        fmt.Println("Input tidak boleh melebihi 1000000")
        return
    }
    cetakDeret(n)
}
```

Screenshoot Output



The screenshot shows a Visual Studio Code editor with a Go file named `3.go` open. The file contains a recursive function `cetakDeret(n)` that prints a sequence of numbers. A Command Prompt window is open, showing the execution of the program. The output is a sequence of numbers: 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. The editor's Explorer pane on the left shows a project structure with files like `1.go`, `2.go`, `3.go`, `guided`, `unguided`, `1.go`, `2.go`, `3.go`, `511102211.docx`, `WRL0003.tmp`, `2311102211.docx`, and `modul_4.pdf`. The bottom status bar indicates the file is in UTF-8 encoding and the cursor is at line 31, column 2.

```
28 |     return
29 | }
30 | cetakDeret(n)
31 | }
```

```
C:\Lecture\Semester_3\Praktikum_Algoritma_Pemrograman_2\modul_4> run unguided\3.go
n1: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
C:\Lecture\Semester_3\Praktikum_Algoritma_Pemrograman_2\modul_4>
```

Deskripsi Program

Program di atas memuat sebuah fungsi yang dapat mencetak suatu deret bilangan berdasarkan nilai suku pertama yang dimasukkan. Setiap nilai pada deret ini ditentukan berdasarkan nilai pada suku sebelumnya. Jika nilai suku sebelumnya genap, maka nilai pada suku saat ini adalah nilai pada suku sebelumnya dibagi dua. Namun jika nilai suku sebelumnya ganjil, maka nilai pada suku saat ini adalah tiga kali nilai suku sebelumnya ditambah 1. Proses penentuan suku-suku berikutnya berdasarkan nilai suku saat ini akan terus berulang hingga suku berikutnya bernilai 1. Untuk nilai suku pertama lebih besar dari 1 dan lebih kecil dari 1000000, diketahui suku pada deret akan selalu mencapai nilai 1.