

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROSEDUR**



Disusun Oleh :

Egi Umar Ferdhika / 2311102277

11-IF-06

Dosen Pengampu :

Abednego Dwi Septiadi, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dalam pemrograman Go, tidak ada konsep prosedur yang terpisah dari fungsi seperti dalam beberapa bahasa pemrograman lainnya. Pada dasarnya, prosedur dalam Go adalah fungsi yang tidak mengembalikan nilai. Deklarasi prosedur menggunakan kata kunci `func`, sama seperti fungsi pada umumnya, tetapi tanpa tipe kembalian dan tanpa pernyataan `return`. Prosedur digunakan untuk menjalankan serangkaian perintah atau operasi, seperti mencetak data atau melakukan manipulasi, tanpa memerlukan hasil balik. Misalnya, prosedur bisa digunakan untuk menampilkan pesan atau mengubah variabel, di mana hasilnya tidak perlu dikembalikan ke pemanggil.

Meskipun dalam Go prosedur tidak disebut secara khusus, kita bisa menganggap fungsi yang tidak memiliki nilai kembalian sebagai prosedur. Prosedur sangat berguna untuk tugas-tugas seperti logging, menampilkan data ke layar, atau melakukan operasi yang tidak memerlukan pengolahan lebih lanjut dari hasilnya.

II. GUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan.

Source Code diberi penjelasan maka akan menjadi nilai ++

GUIDED 1

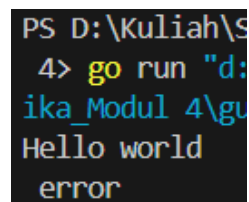
Sourcecode

```
package main

import "fmt"
// Nama : Egi Umar Ferdhika
// NIM : 2311102277
func main() {
    var bilangan int
    var pesan string
    fmt.Scan(&bilangan, &pesan)
    cetak_Pesan(pesan, bilangan)
}

func cetak_Pesan(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)
}
```

Screenshoot Output



```
PS D:\Kuliah\S
4> go run "d:\
ika_Modul 4\gu
Hello world
error
```

Deskripsi Program

Program diatas adalah program yang meminta pengguna untuk memasukkan sebuah bilangan dan pesan. Bilangan tersebut berfungsi sebagai "flag" yang menentukan jenis pesan: jika bilangan adalah 0, jenisnya adalah "error"; jika 1, jenisnya "warning"; dan jika 2, jenisnya "informasi". Setelah input diterima, program akan mencetak pesan yang dimasukkan oleh pengguna beserta jenisnya

sesuai dengan nilai bilangan. Misalnya, jika pengguna memasukkan 1 untuk bilangan dan Ini adalah pesan untuk pesan, output yang dihasilkan akan menampilkan "Ini adalah pesan warning".

GUIDED 2

Sourcecode

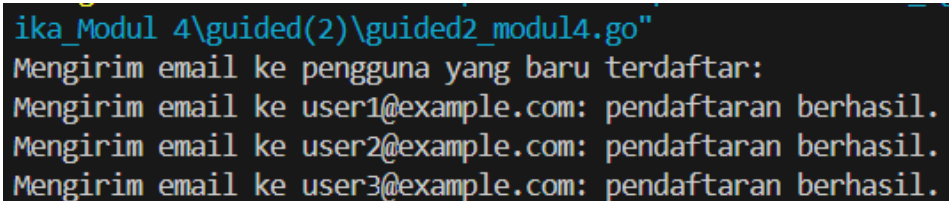
```
package main

import "fmt"
// Nama : Egi Umar Ferdhika
// NIM : 2311102277
func SendEmailNotif(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 {
        SendEmailNotif(email)
    }
}
```

Screenshoot Output



```
ika Modul 4\guided(2)\guided2_modul4.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.
```

Deskripsi Program

Program diatas adalah program dalam bahasa Go yang berfungsi untuk mengirim notifikasi email kepada pengguna yang baru terdaftar. Dalam program ini, fungsi SendEmailNotif didefinisikan untuk mencetak pesan yang menunjukkan bahwa email telah berhasil dikirim ke alamat email yang diberikan. Fungsi main kemudian membuat daftar alamat email yang akan menerima notifikasi, yaitu user1@example.com, user2@example.com, dan user3@example.com. Program ini mencetak pesan bahwa email sedang dikirim ke pengguna yang baru terdaftar,

dan menggunakan loop untuk mengiterasi setiap alamat email dalam daftar, memanggil fungsi SendEmailNotif untuk mengirimkan notifikasi ke masing-masing pengguna. Output dari program ini akan menunjukkan bahwa email dikirim ke setiap pengguna dalam daftar, menampilkan pesan seperti "Mengirim email ke user1@example.com: pendaftaran berhasil."

GUIDED 3

Sourcecode

```
package main

import "fmt"
// Nama : Egi Umar Ferdhika
// NIM : 2311102277
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 main() {
    var a, b int
    var c float64

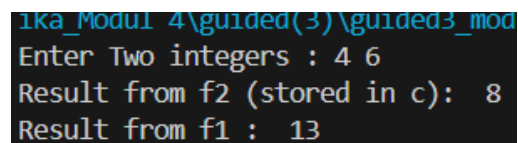
    fmt.Print("Enter Two integers : ")
    fmt.Scan(&a, &b)

    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



```
ika_Modul 4\guided(3)\guided3_mod
Enter Two integers : 4 6
Result from f2 (stored in c): 8
Result from f1 : 13
```

Deskripsi Program

Kode ini adalah program dalam bahasa Go yang mendemonstrasikan penggunaan fungsi untuk melakukan perhitungan matematis dengan dua bilangan bulat sebagai input. Terdapat dua fungsi: f1 dan f2. Fungsi f1 menerima dua parameter bilangan bulat (x dan y), menghitung hasil berdasarkan rumus $2 \cdot x - 0.5 \cdot y + 3$, dan mengembalikan hasil sebagai tipe data float64. Fungsi f2 juga melakukan perhitungan yang sama tetapi menggunakan pointer untuk menyimpan hasilnya dalam variabel yang dioper dari fungsi main. Di dalam fungsi main, pengguna diminta untuk memasukkan dua bilangan bulat, yang disimpan dalam variabel a dan b. Fungsi f2 kemudian dipanggil dengan a, b, dan alamat dari variabel c, sehingga hasil perhitungan disimpan di dalam c. Program mencetak hasil yang disimpan di c setelah pemanggilan f2. Selanjutnya, fungsi f1 dipanggil dengan parameter b dan a, dan hasilnya dicetak.

III. UNGUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan.

Source Code diberi penjelasan maka akan menjadi nilai ++

UNGUIDED 1

Sourcecode

```
package main

import (
    "fmt"
    // Nama : Egi Umar Ferdhika
    // NIM : 2311102277
)

func factorial(n int, result *int) {
    *result = 1
    if n == 0 {
        return
    }
    for i := 2; i <= n; i++ {
        *result *= i
    }
}

func permutasi(n, r int, result *int) {
    var faktorialN, faktorialNR int
    factorial(n, &faktorialN)
    factorial(n-r, &faktorialNR)
    *result = faktorialN / faktorialNR
}
```

```

func kombinasi(n, r int, result *int) {
    var faktorialN, faktorialR, faktorialNR int
    factorial(n, &faktorialN)
    factorial(r, &faktorialR)
    factorial(n-r, &faktorialNR)
    *result = faktorialN / (faktorialR * faktorialNR)
}

func main() {
    var a, b, c, d int
    fmt.Print("Masukkan 4 input(dibedakan dengan spasi)
: ")
    fmt.Scan(&a, &b, &c, &d)

    if a >= c && b >= d {
        var permutasiA, kombinasiA, permutasiB,
kombinasiB int

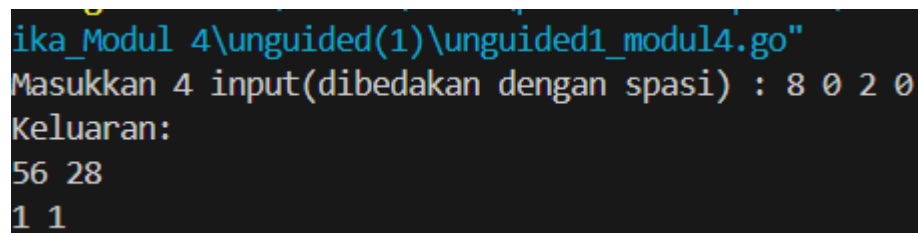
        permutasi(a, c, &permutasiA)
        kombinasi(a, c, &kombinasiA)

        permutasi(b, d, &permutasiB)
        kombinasi(b, d, &kombinasiB)

        fmt.Println("Keluaran:")
        fmt.Println(permutasiA, kombinasiA)
        fmt.Println(permutasiB, kombinasiB)
    } else {
        fmt.Println("Syarat tidak terpenuhi")
    }
}

```

Screenshoot Output



```

ika_Modul 4\unguided(1)\unguided1_modul4.go
Masukkan 4 input(dibedakan dengan spasi) : 8 0 2 0
Keluaran:
56 28
1 1

```

Deskripsi Program

Kode ini adalah program dalam bahasa Go yang menghitung nilai permutasi dan kombinasi berdasarkan input pengguna. Terdapat beberapa Prosedur yaitu: factorial untuk menghitung faktorial dari bilangan bulat, permutasi untuk menghitung permutasi menggunakan nilai faktorial, dan kombinasi untuk menghitung kombinasi juga berdasarkan faktorial. Fungsi utama main meminta

pengguna memasukkan empat bilangan bulat dan memeriksa syarat apakah dua bilangan pertama lebih besar atau sama dengan dua bilangan kedua. Jika syarat terpenuhi, program akan menghitung permutasi dan kombinasi untuk pasangan tersebut dan mencetak hasilnya. Jika tidak, program akan menampilkan pesan bahwa syarat tidak terpenuhi. Dengan demikian, program ini menunjukkan cara menggunakan fungsi dalam Go untuk melakukan perhitungan matematis dan validasi input.

UNGUIDED 2

Sourcecode

```
package main

import (
    "fmt"
    "strings"
    // Nama : Egi Umar Ferdhika
    // NIM : 2311102277
)

func hitungSkor(namaPemain *string, jumlahSoal *int,
totalSkor *int) {
    var waktuUjian [8]int
    *jumlahSoal = 0
    *totalSkor = 0

    fmt.Print("Masukkan nama peserta: ")
    fmt.Scan(namaPemain)
    if strings.ToLower(*namaPemain) == "selesai" {
        return
    }

    for i := 0; i < 8; i++ {
        fmt.Printf("Masukkan waktu ujian ke-%d: ", i+1)
        fmt.Scan(&waktuUjian[i])
        if waktuUjian[i] != 301 {
            (*jumlahSoal)++
            *totalSkor += waktuUjian[i]
        }
    }
}

func main() {
    var pemenang string
    var soalTertinggi, waktuTerendah int

    for {
        var namaPeserta string
        var jumlahSoal, skorTotal int
```



```

        hitungSkor(&namaPeserta, &jumlahSoal,
&skorTotal)

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

        fmt.Printf("Peserta: %s, Jumlah Soal: %d, Skor:
%d\n", namaPeserta, jumlahSoal, skorTotal)

        if jumlahSoal > soalTertinggi || (jumlahSoal ==
soalTertinggi && skorTotal < waktuTerendah) {
            pemenang = namaPeserta
            soalTertinggi = jumlahSoal
            waktuTerendah = skorTotal
        }
    }

    fmt.Printf("Pemenang: %s, Jumlah Soal: %d, Skor:
%d\n", pemenang, soalTertinggi, waktuTerendah)
}

```

Screenshoot Output

```

Masukkan nama peserta: Wagiman
Masukkan waktu ujian ke-1: 335
Masukkan waktu ujian ke-2: 65
Masukkan waktu ujian ke-3: 12
Masukkan waktu ujian ke-4: 36
Masukkan waktu ujian ke-5: 775
Masukkan waktu ujian ke-6: 3
Masukkan waktu ujian ke-7: 45
Masukkan waktu ujian ke-8: 10
Peserta: Wagiman, Jumlah Soal: 8, Skor: 1281
Masukkan nama peserta: Tardi
Masukkan waktu ujian ke-1: 65
Masukkan waktu ujian ke-2: 87
Masukkan waktu ujian ke-3: 235
Masukkan waktu ujian ke-4: 66
Masukkan waktu ujian ke-5: 115
Masukkan waktu ujian ke-6: 345
Masukkan waktu ujian ke-7: 22
Masukkan waktu ujian ke-8: 98
Peserta: Tardi, Jumlah Soal: 8, Skor: 1033
Masukkan nama peserta: selesai
Pemenang: Tardi, Jumlah Soal: 8, Skor: 1033

```

Deskripsi Program

Program diatas adalah program dalam bahasa Go yang digunakan untuk menghitung skor peserta dalam ujian berdasarkan waktu yang dihabiskan untuk

menyelesaikan delapan soal. Program ini terdiri dari prosedur hitungSkor, yang menerima pointer untuk nama pemain, jumlah soal, dan total skor. Fungsi ini meminta nama peserta dan waktu yang dihabiskan untuk setiap soal. Jika waktu yang dimasukkan sama dengan 301, soal tersebut tidak dihitung sebagai diselesaikan. Setelah menerima input, Prosedur ini mengupdate jumlah soal yang diselesaikan dan total skor. Fungsi utama main menjalankan loop yang terus meminta input peserta hingga nama yang dimasukkan adalah "selesai". Setelah setiap input, program mencetak jumlah soal yang diselesaikan dan skor untuk peserta tersebut. Program juga membandingkan hasil untuk menentukan pemenang berdasarkan jumlah soal terbanyak yang diselesaikan dan, jika sama, berdasarkan skor terendah. Setelah semua peserta selesai, program mencetak nama pemenang beserta jumlah soal yang diselesaikan dan skor mereka.

UNGUIDED 3

Sourcecode

```
package main

import (
    "fmt"
    // Nama : Egi Umar Ferdhika
    // NIM : 2311102277
)

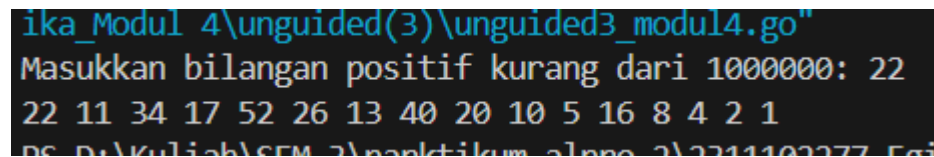
// cetakDeret mencetak setiap suku dari deret
// berdasarkan aturan yang diberikan
func cetakDeret(n int) {
    for n != 1 {
        fmt.Printf("%d ", n)
        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Println(1) // Tambahkan 1 di akhir karena deret
    // selalu berakhir dengan 1
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan positif kurang dari
    1000000: ")
    fmt.Scan(&n)

    if n > 0 && n < 1000000 {
        cetakDeret(n)
    }
}
```

```
    } else {  
        fmt.Println("Input harus berupa bilangan positif  
kurang dari 1000000.")  
    }  
}
```

Screenshoot Output



```
ika_Modul 4\unguided(3)\unguided3_modul4.go"  
Masukkan bilangan positif kurang dari 1000000: 22  
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

Deskripsi Program

Program diatas adalah program dalam bahasa Go yang mencetak deret angka berdasarkan aturan yang dikenal sebagai Konjectur Collatz. Program ini terdiri dari prosedur cetakDeret yang menerima bilangan bulat positif n dan mencetak setiap suku dari deret hingga mencapai 1. Aturan yang digunakan adalah: jika n adalah genap, maka n dibagi dua; jika n adalah ganjil, maka n diperoleh dengan rumus $3n+1$. Proses ini berulang hingga n menjadi 1, yang selalu menjadi akhir dari deret. Di dalam fungsi main, program meminta pengguna untuk memasukkan bilangan positif kurang dari 1.000.000. Jika input valid, fungsi cetakDeret dipanggil untuk mencetak deret berdasarkan nilai tersebut. Jika input tidak valid, program akan mencetak pesan kesalahan yang menjelaskan bahwa input harus berupa bilangan positif dan kurang dari 1.000.000.