

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4
PROSEDUR**



Disusun Oleh :

HANIF REYHAN ZHAFRAN ADRYTONA / 2311102266

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 komputer, prosedur adalah serangkaian instruksi yang dikelompokkan untuk mencapai tujuan tertentu, sementara fungsi adalah bentuk khusus dari prosedur yang mengembalikan nilai setelah dieksekusi.

Prosedur dalam bahasa pemrograman Go (Golang) direpresentasikan sebagai fungsi yang tidak mengembalikan nilai, atau bisa juga digunakan tanpa memedulikan hasil kembalian. Fungsi dalam Go dideklarasikan menggunakan kata kunci `func` diikuti dengan nama fungsi, parameter (jika ada), dan tipe nilai yang dikembalikan (jika ada). Fungsi di Go dapat dipanggil berkali-kali untuk mengeksekusi blok kode yang telah didefinisikan, membuatnya fleksibel dan efisien untuk digunakan dalam berbagai konteks. Selain itu, Go mendukung fitur seperti fungsi anonim (anonymous functions), closure, dan kemampuan untuk meneruskan fungsi sebagai argumen, yang memperkaya penerapan konsep prosedural dalam pengembangan program.`

II. GUIDED

Guided 1.0

Sourcecode

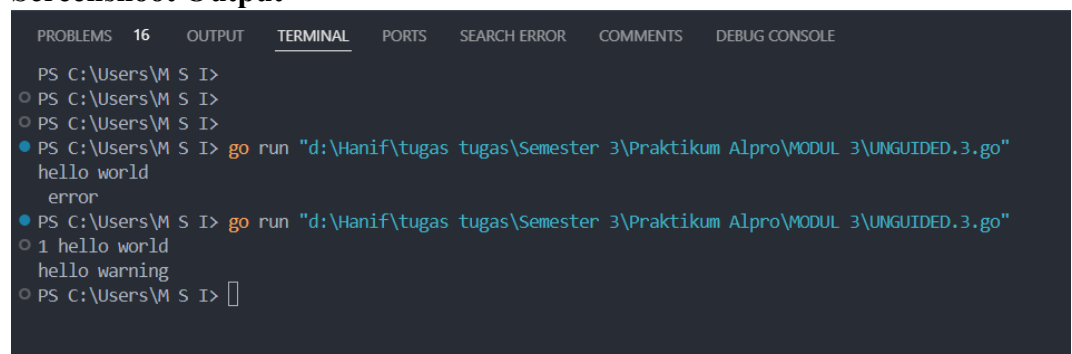
```
package main
import "fmt"

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

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)
}

// 2311102266 - Hanip
```

Screenshoot Output



```
PROBLEMS 16 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE
PS C:\Users\M S I>
o PS C:\Users\M S I>
o PS C:\Users\M S I>
o PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 3\UNGUIDED.3.go"
hello world
error
o PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 3\UNGUIDED.3.go"
o 1 hello world
hello warning
o PS C:\Users\M S I> []
```

Deskripsi Program

Program di atas adalah sebuah program bahasa Go dan berfungsi untuk menerima dua input dari pengguna, yaitu sebuah bilangan bertipe integer dan sebuah pesan bertipe string. Input tersebut dibaca menggunakan fungsi `fmt.Scan`. Bilangan yang dimasukkan

berfungsi sebagai indikator untuk menentukan jenis pesan yang akan ditampilkan, sedangkan pesan string digunakan sebagai teks utama. Setelah input diterima, program memanggil fungsi `cetakPesan` yang bertanggung jawab untuk menampilkan pesan pengguna bersama jenisnya. Jika bilangan yang dimasukkan adalah `0`, maka jenis pesan yang ditampilkan adalah "error", jika `1` maka jenisnya "warning", dan jika `2` maka jenisnya "informasi".

Namun, ada beberapa hal yang perlu diperhatikan dalam program ini. Fungsi `fmt.Scan` digunakan untuk menerima input, tetapi tidak cocok untuk membaca pesan dengan spasi karena hanya mengambil string hingga spasi pertama. Ini akan menjadi masalah jika pesan yang dimasukkan terdiri dari beberapa kata. Selain itu, program tidak menangani kasus ketika nilai bilangan yang dimasukkan bukan `0`, `1`, atau `2`, yang membuatnya tidak fleksibel untuk menangani input di luar angka tersebut.

Guided 1.2

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)
    }
}

// 2311102266 - Hanip
```

Screenshot Output

```
PROBLEMS 17 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE

PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 4\GUIDED.2.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:\Users\M S I> █
```

Deskripsi Program

Program di atas adalah program Go sederhana yang mensimulasikan pengiriman notifikasi email kepada beberapa pengguna. Program dimulai dengan mendeklarasikan fungsi `sendEmailNotification` yang menerima parameter berupa alamat email dan menampilkan pesan bahwa pendaftaran berhasil dengan menggunakan format string. Di dalam fungsi `main`, terdapat slice `emails` yang berisi tiga alamat email. Program kemudian mencetak pesan awal untuk menunjukkan bahwa proses pengiriman email dimulai, dan menggunakan perulangan `for` untuk mengiterasi setiap alamat email dalam slice. Pada setiap iterasi, fungsi `sendEmailNotification` dipanggil untuk setiap alamat email, dan pesan yang menandakan notifikasi pendaftaran berhasil dikirimkan dicetak ke layar.

Guided 1.3

Sourcecode

```
package main

import "fmt"

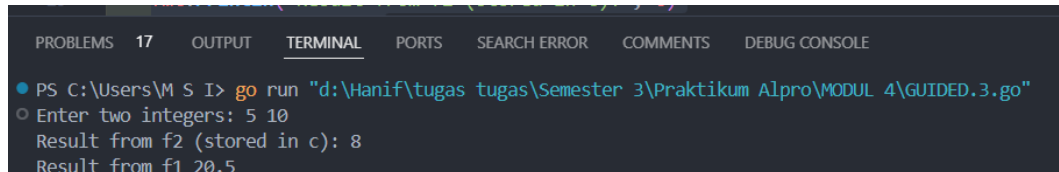
func f1(x, y int) float64 {
    var hasil float64
    hasil = float64(2*x) - 0.5*float64(y) + 3.0
    return hasil
}

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

func 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)
```

```
}  
// 2311102266 - Hanip
```

Screenshoot Output



```
PROBLEMS 17 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE  
● PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 4\GUIDED.3.go"  
○ Enter two integers: 5 10  
Result from f2 (stored in c): 8  
Result from f1 20.5
```

Deskripsi Program

Program di atas adalah sebuah program Go yang mendemonstrasikan dua cara perhitungan sebuah persamaan matematika sederhana melalui fungsi dengan dan tanpa penggunaan pointer. Fungsi f1 dan f2 memiliki perhitungan yang sama, yaitu $2*x - 0.5*y + 3.0$, namun cara pengembalian hasilnya berbeda. Fungsi f1 menerima dua parameter integer x dan y, kemudian mengonversi hasil perhitungan ke tipe float64 dan mengembalikannya sebagai nilai return. Fungsi ini melakukan perhitungan secara langsung di dalamnya dan mengembalikan nilai ke pemanggil. Di sisi lain, fungsi f2 menggunakan pendekatan yang melibatkan pointer, di mana ia menerima parameter ketiga berupa pointer hasil (pointer ke float64). Hasil perhitungan tidak dikembalikan secara eksplisit, melainkan disimpan pada alamat memori yang ditunjuk oleh pointer tersebut, yang kemudian memodifikasi nilai variabel di luar fungsi.

Di dalam fungsi main, program meminta pengguna untuk memasukkan dua bilangan integer yang disimpan dalam variabel a dan b. Setelah menerima input, program pertama-tama memanggil fungsi f2, yang menyimpan hasil perhitungan pada variabel c melalui pointer. Hasil perhitungan dari f2 kemudian ditampilkan. Setelah itu, fungsi f1 dipanggil dengan urutan parameter yang dibalik, dan hasilnya disimpan di variabel resultF1, yang kemudian dicetak. Dengan demikian, program ini memperlihatkan dua pendekatan berbeda dalam menghitung hasil dan cara memanipulasi variabel di luar cakupan fungsi melalui pointer.

III. UNGUIDED

Soal Studi Case

Minggu ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakah kalkan membantu Jonas? (tidak tentunya ya :p) Masukan terdiri dari empat buah bilangan asli a , b , c , dan d yang dipisahkan oleh spasi, dengan syarat $a \geq c$ dan $b \geq d$ Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi a terhadap c , sedangkan baris kedua adalah hasil permutasi dan kombinasi b terhadap d . Catatan : permutasi (P) dan kombinasi (C) dari n terhadap r ($n \geq r$) dapat dihitung dengan menggunakan persamaan berikut! $P(n, r) = \frac{n!}{(n-r)!}$, sedangkan $C(n, r) = \frac{n!}{r!(n-r)!}$

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var x, y, z, w int
    fmt.Print("Masukkan x, y, z, w: ")
    fmt.Scanf("%d %d %d %d", &x, &y, &z, &w)

    if x >= z && y >= w {

        permutasi1_2311102266 := permutasi(x, z)
        kombinasi1_2311102266 := kombinasi(x, z)
        permutasi2_2311102266 := permutasi(y, w)
        kombinasi2_2311102266 := kombinasi(y, w)

        fmt.Printf("Permutasi P(x, z) = %d, Kombinasi C(x, z) = %d\n", permutasi1_2311102266, kombinasi1_2311102266)
        fmt.Printf("Permutasi P(y, w) = %d, Kombinasi C(y, w) = %d\n", permutasi2_2311102266, kombinasi2_2311102266)
    } else {
        fmt.Println("Syarat x >= z dan y >= w harus dipenuhi.")
    }
}

func faktorial(n int) int {
    if n == 0 {
```

```

        return 1
    }
    return n * faktorial(n-1)
}
func permutasi(n, r int) int {
    if n < r {
        return 0
    }
    return faktorial(n) / faktorial(n-r)
}
func kombinasi(n, r int) int {
    if n < r {
        return 0
    }
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}

// 2311102266 - Hanip 11-if-06

```

Screenshoot Output

```

PROBLEMS 6 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE

PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 4\UNGUIDED.1.go"
Masukkan x, y, z, w: 3 8 0 2
Permutasi P(x, z) = 1, Kombinasi C(x, z) = 1
Permutasi P(y, w) = 56, Kombinasi C(y, w) = 28
PS C:\Users\M S I>
fwd-i-search: _

```

Deskripsi Program

Program di atas adalah program Go yang menghitung permutasi dan kombinasi dari dua pasang bilangan yang dimasukkan oleh pengguna. Program meminta input empat bilangan x, y, z, dan w, kemudian memeriksa apakah nilai $x \geq z$ dan $y \geq w$. Jika kondisi ini terpenuhi, program menghitung permutasi dan kombinasi dari pasangan x, z dan y, w menggunakan fungsi permutasi dan kombinasi. Kedua fungsi ini menghitung permutasi dan kombinasi berdasarkan rumus faktorial, di mana faktorial dihitung menggunakan fungsi rekursif faktorial. Jika syarat input terpenuhi, hasil permutasi dan kombinasi ditampilkan dengan format yang telah ditentukan. Jika syarat tidak terpenuhi, program mencetak pesan kesalahan. Nama variabel untuk hasil perhitungan diberi label khusus (misalnya, permutasi1_2311102266), tetapi titik dalam nama

variabel telah diganti dengan underscore agar sesuai dengan aturan penamaan dalam Go.

Unguided 2

SOAL :

Kompetisi pemrograman tingkat nasional berlangsung ketat. Setiap peserta diberikan 8 soal yang harus dapat diselesaikan dalam waktu 5 jam saja. Peserta yang berhasil menyelesaikan soal paling banyak dalam waktu paling singkat adalah pemenangnya. Buat program gema yang mencari pemenang dari daftar peserta yang diberikan. Program harus dibuat modular, yaitu dengan membuat prosedur `hitungSkor` yang mengembalikan total soal dan total skor yang dikerjakan oleh seorang peserta, melalui parameter formal. Pembacaan nama peserta dilakukan di program utama, sedangkan waktu pengerjaan dibaca didalam prosedur. prosedur `hitungSkor`(in/out soal, skor : integer) Setiap baris masukan dimulai dengan satu string nama peserta tersebut diikuti dengan adalah 8 integer yang menyatakan berapa lama (dalam menit) peserta tersebut menyelesaikan soal. Jika tidak berhasil atau tidak mengirimkan jawaban maka otomatis dianggap menyelesaikan dalam waktu 5 jam 1 menit (301 menit). Satu baris keluaran berisi nama pemenang, jumlah soal yang diselesaikan, dan nilai yang diperoleh. Nilai adalah total waktu yang dibutuhkan untuk menyelesaikan soal yang berhasil diselesaikan.

Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "math"
    "os"
    "strings"
)

func hitungSkor(waktu []int, soal *int, skor *int) {
    *soal = 0
    *skor = 0
    for _, w := range waktu {
        if w <= 300 {
            *soal++
            *skor += w
        }
    }
}
```

```

}

func main() {
    var pemenang string
    var maxSoal, minSkor int
    maxSoal = -1
    minSkor = math.MaxInt32
    reader := bufio.NewReader(os.Stdin)
    for {
        fmt.Println("Masukkan nama dan waktu :")
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

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

        if input == "" {
            continue
        }

        parts := strings.Fields(input)
        if len(parts) != 9 {
            fmt.Println("Input tidak valid.")
            continue
        }

        nama := parts[0]
        var waktu [8]int

        for i := 0; i < 8; i++ {
            fmt.Sscanf(parts[i+1], "%d", &waktu[i])
        }

        var soal, skor int
        hitungSkor(waktu[:], &soal, &skor)

        if soal > maxSoal || (soal == maxSoal && skor < minSkor) {
            pemenang = nama
            maxSoal = soal
            minSkor = skor
        }
    }

    if pemenang != "" {

```

```

        fmt.Printf("%s %d %d\n", pemenang, maxSoal, minSkor)
    } else {
        fmt.Println("Tidak ada peserta.")
    }
}

// 2311102266 - hanip

```

Screenshoot Output

```

PROBLEMS 8 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE

PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 4\UNGUIDED.2.go"
Masukkan nama dan waktu :
hanif 20 20 29
Input tidak valid.
Masukkan nama dan waktu :
Astuti 20 50 301 301 61 71 75 10
Masukkan nama dan waktu :
Bertha 25 47 301 26 50 60 65 21
Masukkan nama dan waktu :
selesai
● Bertha 7 294
○ PS C:\Users\M S I>

```

Deskripsi Program

Program di atas adalah program Go yang menghitung skor peserta berdasarkan waktu pengerjaan soal dan menentukan pemenang berdasarkan kriteria tertentu. Program ini menggunakan input dari pengguna yang terdiri dari nama peserta diikuti oleh delapan waktu pengerjaan soal dalam satu baris. Pertama, program menginisialisasi variabel untuk menyimpan nama pemenang, jumlah soal maksimum yang diselesaikan, dan skor minimum yang diperoleh. Program kemudian membaca input secara berulang menggunakan `bufio.NewReader`, memeriksa apakah pengguna mengetik "Selesai" untuk menghentikan input. Jika input tidak kosong dan valid (memiliki tepat sembilan elemen), program memisahkan nama peserta dan waktu pengerjaan, lalu menyimpan waktu dalam array. Fungsi `hitungSkor` digunakan untuk menghitung jumlah soal yang berhasil diselesaikan (soal dengan waktu pengerjaan ≤ 300 menit) dan total waktu yang digunakan untuk soal-soal tersebut. Jika jumlah soal yang diselesaikan peserta melebihi jumlah maksimum sebelumnya atau sama tetapi dengan skor yang lebih rendah, nama peserta tersebut akan menjadi pemenang baru. Setelah semua input selesai, program menampilkan nama pemenang beserta jumlah soal yang diselesaikan dan total waktu yang digunakan; jika tidak ada peserta yang valid, program mencetak pesan bahwa tidak ada peserta.

UNGUIDED 3

Skiena dan Revilla dalam Programming Challenges mendefinisikan sebuah deret bilangan. Deret dimulai dengan sebuah bilangan bulat n . Jika bilangan n saat itu genap, maka suku berikutnya adalah $\frac{1}{2}n$, tetapi jika ganjil maka suku berikutnya bernilai $3n+1$. Rumus yang sama digunakan terus menerus untuk mencari suku berikutnya. Deret berakhir ketika suku terakhir bernilai 1. Sebagai contoh jika dimulai dengan $n=22$, maka deret bilangan yang diperoleh adalah: 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1 Untuk suku awal sampai dengan 1000000, diketahui deret ini selalu mencapai suku dengan nilai 1. Buat program skiena yang akan mencetak setiap suku dari deret yang dijelaskan diatas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal. prosedur cetakDeret(in n : integer) Masukan berupa satu bilangan integer positif yang lebih kecil dari 1000000. Keluaran terdiri dari satu baris saja. Setiap suku dari deret tersebut dicetak dalam baris yang dan dipisahkan oleh sebuah spasi.

Sourcecode

```
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.Print("Masukkan nilai suku awal (n): ")
    fmt.Scan(&n)
    if n > 0 && n < 1000000 {
```

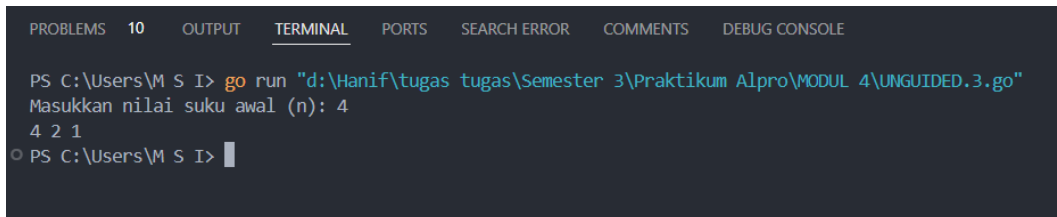
```

        cetakDeret(n)
    } else {
        fmt.Println("Input tidak valid. Harap masukkan bilangan
        bulat positif yang lebih kecil dari 1000000.")
    }
}

// 2311102266 - hanip

```

Screenshoot Output



```

PROBLEMS 10 OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CONSOLE

PS C:\Users\M S I> go run "d:\Hanif\tugas tugas\Semester 3\Praktikum Alpro\MODUL 4\UNGUIDED.3.go"
Masukkan nilai suku awal (n): 4
4 2 1
PS C:\Users\M S I>

```

Deskripsi Program

Program di atas adalah implementasi dari algoritma Collatz dalam bahasa pemrograman Go, yang menghasilkan deret angka berdasarkan nilai awal yang dimasukkan oleh pengguna. Pengguna diminta untuk memasukkan bilangan bulat positif `n`, dan program akan memvalidasi input untuk memastikan bahwa `n` berada dalam rentang yang benar, yaitu lebih besar dari 0 dan kurang dari 1.000.000. Jika input valid, fungsi `cetakDeret` akan dipanggil, yang mencetak nilai `n` dan melakukan perhitungan berdasarkan aturan berikut: jika `n` genap, maka dibagi dua, dan jika `n` ganjil, maka dihitung dengan rumus $3*n + 1$. Proses ini berlanjut hingga `n` menjadi 1, yang juga dicetak sebagai suku terakhir dalam deret. Jika pengguna memasukkan nilai yang tidak valid, program akan memberikan pesan kesalahan yang menjelaskan bahwa input harus berupa bilangan bulat positif yang sesuai dengan ketentuan. Dengan cara ini, program tidak hanya menghitung deret Collatz tetapi juga memberikan validasi yang tepat untuk input pengguna.