LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 4

PROSEDUR

Disusun Oleh : Brian Farrel Evandhika

2311102037

IF 11 06

Dosen Pengampu :

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

4.1 Definlsl Procedure

Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu Instruksl baru

yang dibuat untuk mengurangi kerumitan dari kode program yang kompleks pada suatu program yang

besar. Prosedur akan menghasllkan suatu akibat atau efek langsung pada program ketika dipanggil

pada program Utama. Suatu subprogram dikatakan prosedur apabila:

1. Tldakada deklarasi tipe nilai yang dikembalikan, danTldak terdapat kata kunci return dalam badan subprogram.

2.Kedudukannya prosedur sama seperti instruksi dasar yang sudah ada sebelumnya (assignment

dan/atau instruksi yang berasal dari paket (fmt), seperti fmtScan dan fmt.Prtnt. Karena itu selalu pilih

nama prosedur yang berbentuk atau sesuatu yang merepresentasikan proses sebagai nama

dari prosedur. Contoh: cetak, hitung rata rata, cari nilaiI, belok, mulai,

4.2 Deklarasl Procedure

Berikut ini adalah cara penulisan deklarasi prosedur pada notasi Pseudocode dan GoLang.

Notasi Algoritma

```
1 procedure <nama procedure> (<params>)
2 kamus
3  {deklarasi variabel lokal dari procedure}
4 algoritma
5  {badan algoritma procedure}
6  ...
7 endprocedure
```

Notasi dalam bahasa Go

```
9 func <nama procedure> (<params>) {
10  /* deklarasi variabel lokal dari procedure */
```

```
11  ...
12  /* badan algoritma procedure */
13  ...
14 }
```

Penulisan deklarasi ini berada di luar blok yang dari program utama atau func malno pada suatu

program Go, dan bisa ditulis sebelum atau setelah dari blok program utama tersebut.

Contoh deklarasi prosedur mencetak n nilai pertama dari deret Fibonacci.

Notasi Algoritma

```
1 procedure cetakNFibo(n : integer)
2   kamus
3     f1, f2, f3, i : integer
4   algoritma
5     f1 <- 0
6     f2 <- 1
7     for i <- 1 to n do
8       output(f3)
9       f1 <- f2
10      f2 <- f3
11      f3 <- f1 + f2
12     endfor
13 endprocedure
```

Notasi dalam bahasa Go

Go

```
14 func cetakNFibo(n int) {
15   var f1, f2, f3 int
16   f2 = 0
17   f3 = 1
18   for i := 1; i <= n; i++ {
19     fmt.Println(f3)
20     f1 = f2
21     f2 = f3
22     f3 = f1 + f2
23   }
24 }
```

Catatan: Kata kunci In pada contoh di atas akan dijelaskan pada materi parameter di modul 5 ini.

4.3 Cara Pemangglan Procedure

prosedur hanya akan dleksekusl apablla dlpanggtl

Seperti yang sudah dijelaskan sebelumnya, suatu

baik secara langsung atau tidak langsung Oleh program utama. Tidak langsung di sini maksudnya adalah

prosedur dipanggil Oleh program utama melalu perantara subprogram yang lain.

Pernanggilan suatu prosedure cukup mudah, yaitu dengan hanya menullskan nama beserta parameter

atau argumen yang dlrntnta darl suatu prosedur. Sebagai contoh prosedur cetakNFibo di atas dipa nggil

dengan menuliskan namanya, kemudian sebuah variabel atau nilai integer tertentu sebagai argumen

untuk paramter n, Contoh:

Notasi Algoritma

```
1 program contohprosedur
2   kamus
3     x : integer
4   algoritma
5     x <- 5
6     cetakNFibo(x)  {cara pemanggilan #1}
7     cetakNFibo(100) {cara pemanggilan #2}
8 endprogram
```

Notasi dalam bahasa Go

Go

```
9 func main() {
10   var x int
11   x = 5
12   cetakNFibo(x)  {cara pemanggilan #1}
13   cetakNFibo(100) {cara pemanggilan #2}
14 }
```

Dari contoh di atas terlihat bahwa pernanggilan dengan notasi pseudocode dan Gol-ang adalah

sama. Argumen yang digunakan untuk parameter n berupa integer (sesuai deklarasi) yang terdapat

pada suatu variabel (cara pemanggilan #1) atau nilainya secara langsung (cara pemanggilan #2).

4.4 Contoh Program dengan Procedure

Berikut ini adalah contoh penulisan prosedur pada suatu program lengkap.

Buatlah sebuah program beserta prosedur yang digunakan untuk menampilkan suatu pesan error,

warning atau informasi berdasarkan masukan dari user.

Masukan terdiri dari sebuah bilangan bulatfiag (O s.d. 2) dan sebuah string pesan

M.

Keluaran berupa string pesan M besertajenis pesannya, yaitu error, warning atau informasi berdasarkan

nilai flag O, I dan 2 secara berturut•turut.

Penulisan argumen pada parameter cetakPesan(pesanbilangan) harus sesual urutan upe dau pada

cetakPesan(M flag yaitu string kemudian integer.

4.5 Parameter

Suatu subprogram yang dipanggil dapat berkomunikasi dengan pemanggilnya melalui argumen yang

diberikan melalui parameter yang dideklarasihan pada subprogramnya. Berihut ini jenis atau pembagian

dari parameter.

Berdasarkan letak penulisannya pada program, maka parameter dapat dikelompokkan menjadi dua,

yaitu parameter formal dan parameter aktual.

1. Parameter Formal

Parameter formal adalah parameter yang ditulis pada saat deklarasi suatu subprogram,

parameter ini berfungsi sebagai petunjuk bahwa argumen apa sala yang diperlukan pada saat

pemanggilan subprogram.

Sebagai contoh parameter Jarl_Jarl, tinggl pada deklarasi fungsl volumeTabung adalah

parameter formal (tehs berwarna merah). Artinya ketika memanggil volumeTabung maka kita

harus mempersiapkan dua integer (berapapun nilainya) sebagaiJari_jari dan tinggi.

2. Parameter Aktual

Sedangkan parameter aktual adalah argumen yang digunakan pada bagian parameter Saat

pemanggilan suatu subprogram. Banyaknya argumen dan tipe data yang terdapat pada

paramter aktual harus mengikuti parameter formal.

Sebagai contoh argumen r, t, 15, 14 dan 100 pada contoh kode di atas (teks berwarna biru)

adalah parameter aktual, yang menyatakan nilal yang kita berikan sebagai Jari-Jari dan tinggi.

Selain itu parameter juga dikelompokkan berdasarkan alokasi memorinya, yaitu pass by value dan pass

by reference.

1. Pass by Value

Nilai pada parameter aktual akan disalin he variabel lokal (parameter formal) pada subprogram.

Artinya parameter aktual dan formal dialokasikan di dalam memori komputer dengan alamat

memori yang berbeda, Subprogram dapat menggunakan nilai pada parameter formal tersebut

untuk proses apapun, tetapi tidal? dapat mengembalikan informasinya ke pemanggil melalui

parameter aktual karena pemanggil tidak dapat mengakses memori yang digunakan Oleh

subprogram. pass by value bisa digunakan baik Oleh fungsi ataupun prosedur.

Pada notasi pseudocode, secara sernua parameter formal pada fungsi adalah pass by value,

sedangkan pada prosedur diberi kata kunci in pada saat penulisan parameter formal. Sedangkan

pada bahasa pemrograman Co sama seperti fungsi pada pseudocode, tidak terdapat kata kunci

khusus untuk parameter formal fungsi dan prosedur.

2. Pass by Reference (Pointer)

c

Ketika parameter didefinisikan sebagai pass by reference, maka pada Saat pemanggilan

parameter formal akan berperan sebagai pointer yang menyimpan alamat memori dari

parameter aktual. Sehingga perubahan nilai yang terjadi pada paramter formal tersebut akan

berdampak pada parameter aktual. Artinya nilai terakhirnya akan dapat diketahui Oleh si

pernanggil setelah subprogram tersebut selesai dieksekusi. pass by reference sebaiknya

digunakan hanya untuk prosedur.

Penulisan parameter pass by reference pada prosedur baik pseudocode dan GO menggunakan

kata kunci atau identifier khusus. Pada pseudocode menggunakan kata kunci in/out, sedangkan

pada bahasa Go diberi identifier asterik (k) sebelum tipe data di parameter formal yang menJadi

pass by reference.

Catatan

Parameter pada fungsi sebaiknya adalah pass by value, hal ini dikarenakan fungsi bisa

mengembalikan (return) nilai he pemanggil dan tidak memberikan efek langsung pada

program, walaupun tidak menutup kemungkinan menggunakan pass by reference.

Penggunaan pass by reference sebaiknya pada prosedur karena prosedur tidak bisa

mengembalikan nilai ke pernanggil. Dengan memanfaatkan pass by reference maka

prosedur seolah-olah bisa mengirimkan nilai kepada si pemanggil.

untuk lebih jelas perhatikan contoh sebuah subprogram yang digunakan untuk menghitung

persamaan berikut ini:

$$f(x,y) = 2x - \frac{y}{2} + 3$$

Notasi Algoritma

```
function f1(x, y: integer) → real
    kamus
        hasil : real
    algoritma
        hasil ← 2*x - 0.5*y + 3
        return hasil
    endfunction

procedure f2(in x, y : integer, in/out hasil: real)
    algoritma
        hasil ← 2*x - 0.5*y + 3
    endprocedure

program Contoh
    kamus
        a, b : integer
        c : real
    algoritma
        input(a, b)
        f2(a, b, c)
        output(c, f1(b, a))
    endprogram
```

Notasi dalam bahasa Go

Go

```
package main

import "fmt"

func f1(x, y int) 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.Scan(&a, &b)
    f2(a, b, &c)
    fmt.Println(c, f1(b, a))
}
```

GUIDED

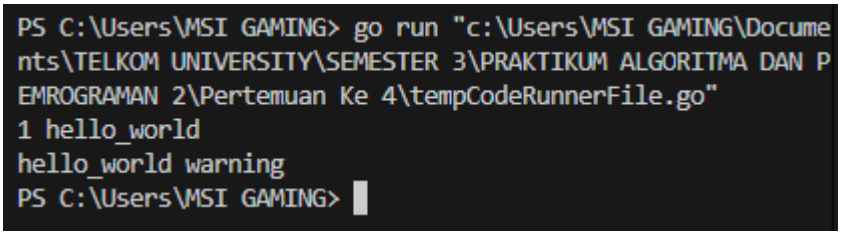
Soal Studi Case

Buatlah sebuah program beserta prosedur yang digunakan untuk menampilkan suatu pesan error,
warning atau informasi berdasarkan masukan dari user.

Sourcecode

```go
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
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)
}
```

Screenshot Program



Deskripsi Program

Kodingan Go tersebut adalah sebuah program sederhana yang menerima dua input dari pengguna, yaitu sebuah angka (bilangan) dan sebuah string (pesan). Program ini kemudian memanggil fungsi cetakPesan yang akan mencetak pesan tersebut diikuti dengan jenis pesan berdasarkan nilai dari bilangan. Jika bilangan adalah 0, jenis pesannya "error"; jika 1, jenisnya "warning"; dan jika 2, jenisnya "informasi". Program menggunakan fmt.Scan untuk membaca input dari pengguna, dan fmt.Println untuk mencetak hasilnya. Singkat, sederhana, dan tepat guna.

Soal Studi Case
Buatlah program untuk mengirim notifikasi ke email

Sourcecode

```go
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
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)
    }
}
```

Screenshot Program

```
PS C:\Users\MSI GAMING> go run "c:\Users\MSI GAMING\Docume
nts\TELKOM UNIVERSITY\SEMESTER 3\PRAKTIKUM ALGORITMA DAN P
EMROGRAMAN 2\Pertemuan Ke 4\tempCodeRunnerFile.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\MSI GAMING>
```

Deskripsi Program

Kodingan Go tersebut adalah sebuah program yang mengirimkan notifikasi email kepada pengguna baru. Fungsi sendEmailNotification mengambil satu parameter email dan mencetak pesan konfirmasi bahwa email tersebut telah berhasil didaftarkan. Dalam fungsi main, sebuah slice emails berisi daftar alamat email pengguna baru, lalu program mencetak pesan awal dan menggunakan loop for untuk mengiterasi setiap email dalam slice. Pada setiap iterasi, program memanggil fungsi sendEmailNotification untuk setiap alamat email, mengirimkan notifikasi pendaftaran berhasil. Bersih, jelas, dan langsung ke tujuan.
Soal Studi Case

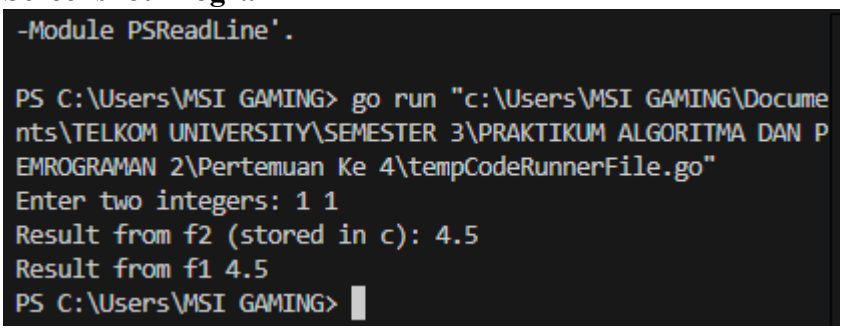
Soal Studi Case
Buatlah program untuk menghitung parameter volume tabung

Sourcecode

```go
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
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
}
```

```go
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)
}
```

Screenshot Program



Deskripsi Program

Kodingan Go tersebut adalah sebuah program yang menghitung nilai dengan dua cara berbeda. Fungsi f1 menerima dua bilangan bulat (x dan y) dan mengembalikan hasil float64 dari operasi matematis tertentu. Fungsi f2 menerima dua bilangan bulat yang sama serta sebuah pointer ke variabel float64, lalu menyimpan hasil operasi matematis yang sama ke dalam variabel tersebut. Dalam fungsi main, dua bilangan bulat dimasukkan oleh pengguna, hasil dari fungsi f2 disimpan dalam variabel c, dan hasil dari fungsi f1 langsung dicetak. Program ini mendemonstrasikan penggunaan pointer dan perhitungan matematis dalam bahasa Go. Ringkas dan efisien!

II. UNGUIDED

Soal Studi Case 1

Minggu ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah maternatika diskrit untuk mempetajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng
untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakah kalian
membantu Jonas? (tidak tentunya ya
Masukan terdiri dari empat buah bilangan asli a, b, c, dan d yang dipisahkan Oleh spasi, dengan
Syarat a > c dan b > 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.
permutasi (P) dan kombinasi (C) dari n terhadap r (n > r) dapat dihitung dengan
menggunakan persamaan berikut!

$$P(n,r) = \frac{n!}{(n-r)!}, \text{ sedangkan } C(n,r) = \frac{n!}{r!(n-r)!}$$

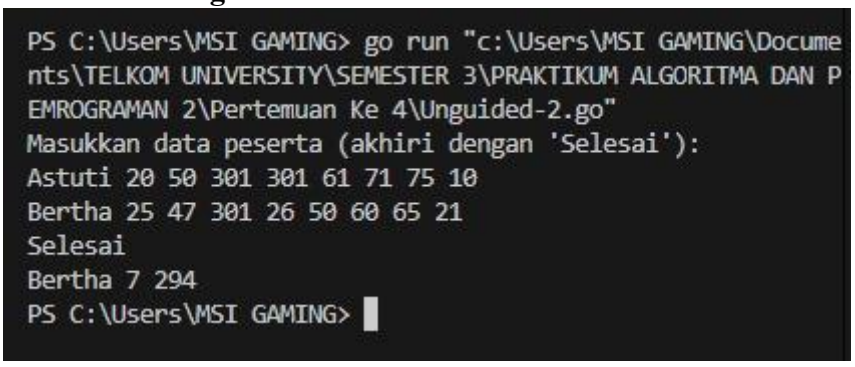
Source Code

```
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
package main

import (
```

```
    "fmt"
)


func factorial(n int, hasil *int) {
    *hasil = 1
    for i := 2; i <= n; i++ {
        *hasil *= i
    }
}


func permutation(n, r int, hasil *int) {
    var nFact, nMinRFact int
    factorial(n, &nFact)
    factorial(n-r, &nMinRFact)
    *hasil = nFact / nMinRFact
}


func combination(n, r int, hasil *int) {
    var nFact, rFact, nMinRFact int
    factorial(n, &nFact)
    factorial(r, &rFact)
    factorial(n-r, &nMinRFact)
    *hasil = nFact / (rFact * nMinRFact)
}

func main() {

    var a, b, c, d int
    fmt.Print("Enter values for a, b, c, d: ")
    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)


    var P_ac, C_ac, P_bd, C_bd int


    permutation(a, c, &P_ac)
    combination(a, c, &C_ac)
    fmt.Printf("%d %d\n", P_ac, C_ac)


    permutation(b, d, &P_bd)
    combination(b, d, &C_bd)
    fmt.Printf("%d %d\n", P_bd, C_bd)
```

```
}
```

Screenshot Program



Deskripsi Program

Kodingan Go ini menghitung faktorial, permutasi, dan kombinasi dari angka-angka yang diberikan pengguna. Fungsi factorial menghitung faktorial suatu bilangan, sementara fungsi permutation dan combination menggunakan faktorial untuk menghitung permutasi dan kombinasi dari dua bilangan. Dalam fungsi main, program meminta pengguna memasukkan empat nilai, kemudian menghitung dan mencetak hasil permutasi dan kombinasi dari dua pasangan angka tersebut. Program ini dengan jelas memperlihatkan penggunaan faktorial dalam perhitungan permutasi dan kombinasi di bahasa Go. Komprehensif dan elegan!

Soal Studi Case 2

Kompetisi pemrograman tingkat nasional bedangsung ketat. Setiap peserta diberihan 8 soal

yang harus dapat diselesaikan dalam waktu 5 jam saja. Peserta yang berhasil menyelesaikan

soal paling banyak dalam Waktu paling Singkat adalah pemenangnya.

yang rnencari pemenang dari daftar peserta yang diberikan. Program harus

Buat program

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 di dalam prosedur.prosedure soal, skor : Integer)Setiap baris masukan dimulai dengan satu string nama peserta tersebut diikutl dengan adalah8 integer yang menyatakan berapa lama (dalam menit) peserta tersebut menyelesaikan soal.Jika tidak berhasil atau tidak mengirimkan jawaban maka otomatis dianggap menyelesaikandalam waktu 5 jam I menit (301 menit).Satu baris keluaranberisi nama pemenang, jumlah soal yang diselesaikan, dan nilai yangdiperoleh. Nilai adalah total wahtu yang dibutuhkan untuk menyelesaikan soal yang berhasildiselesaikan.

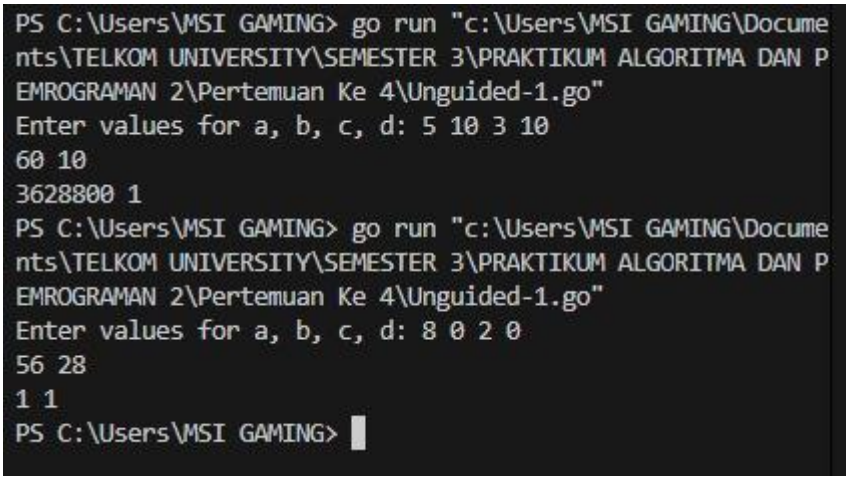
Source Code

```go
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
package main

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

// hitungSkor menghitung jumlah soal yang diselesaikan dan total waktu yang dibutuhkan.
func hitungSkor(waktu []int) (int, int) {
    const waktuMaks = 301 // Waktu maksimum yang diperbolehkan dalam menit
    soalTerselesaikan := 0
    totalWaktu := 0
```

```
    for _, w := range waktu {
        if w < waktuMaks { // hanya soal yang diselesaikan dalam
waktu kurang dari 301 menit yang dihitung
            soalTerselesaikan++
            totalWaktu += w
        }
    }
    return soalTerselesaikan, totalWaktu
}

func main() {
    scanner := bufio.NewScanner(os.Stdin)
    var namaPemenang string
    var maxSoal, minTotalWaktu int

    // Inisialisasi nilai pemenang
    maxSoal = 0
    minTotalWaktu = 1000000 // Angka yang sangat besar untuk
perbandingan

    fmt.Println("Masukkan data peserta (akhiri dengan 'Selesai'): ")
    for scanner.Scan() {
        input := scanner.Text()

        // Memeriksa akhir dari input
        if strings.ToLower(input) == "selesai" {
            break
        }

        // Memisahkan input menjadi bagian-bagian
        parts := strings.Fields(input)

        // Mendapatkan nama peserta
        nama := parts[0]

        // Mengonversi string waktu ke dalam bentuk integer
        var waktu []int
        for i := 1; i < len(parts); i++ {
            w, _ := strconv.Atoi(parts[i])
            waktu = append(waktu, w)
        }

        // Menghitung jumlah soal yang diselesaikan dan total waktu
        soalTerselesaikan, totalWaktu := hitungSkor(waktu)
```

```
        // Menentukan apakah peserta ini adalah pemenang baru
        if soalTerselesaikan > maxSoal || (soalTerselesaikan ==
maxSoal && totalWaktu < minTotalWaktu) {
            namaPemenang = nama
            maxSoal = soalTerselesaikan
            minTotalWaktu = totalWaktu
        }
    }

    // Menampilkan hasil
    fmt.Printf("%s %d %d\n", namaPemenang, maxSoal, minTotalWaktu)
}
```

Screenshot Program



Deskripsi Program

Kodingan Go ini adalah program untuk menentukan pemenang berdasarkan jumlah soal yang diselesaikan dan total waktu yang dibutuhkan. Fungsi hitungSkor menghitung berapa banyak soal yang diselesaikan dalam waktu kurang dari 301 menit dan mengakumulasikan total waktu. Dalam fungsi main, pengguna memasukkan data peserta yang berisi nama dan waktu penyelesaian soal, berakhir dengan kata "Selesai". Program memproses input tersebut, menghitung skor tiap peserta, dan menentukan pemenang berdasarkan kriteria jumlah soal tertinggi dan waktu total terendah. Program ini menampilkan hasil akhir dengan nama pemenang, jumlah soal yang diselesaikan, dan total waktu yang dibutuhkan. Sistem yang rapi dan informatif untuk lomba pemrograman.

Soal Studi Case 3

Skiena dan Revilla dalam Programming Challenges mendefinisikan sebuah deret bilangan. Deretdimulai dengan sebuah bilangan bulat n. Jika bilangan n Saat itu genap, maka suku berikutnyaadalah 1/2n, tetapijika ganjil maka suku berikutnya bernilai 3n+l. Rumus yang sama digunakanterus menerus untuk mencari suku berikutnya. Deret berakhir ketika suku terakhir bernilai 1.Sebagai contohJika 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

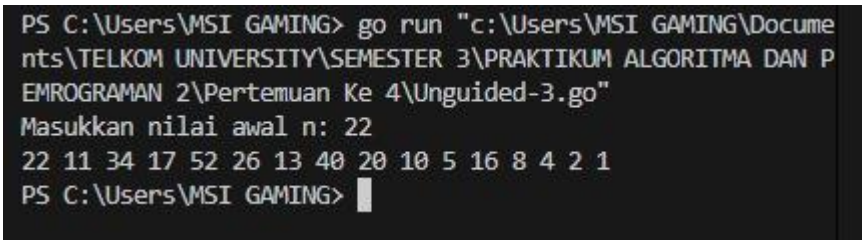
Untuk suku awat sampai dengan 1000000, diketahui deret selalu mencapai suku dengan nilaiBuat program sklena yang akan mencetak setiap suku dari deret yang dijelaskan di atas untuknilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetakDeret yangmempunyai I parameter formal, yaitu nilai dari suku awal.prosedure n : inr.eg.e.r)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 yangdan dipisahkan Oleh sebuah spasi.

Source Code

```go
//2311102037 BRIAN FARREL EVANDHIKA IF 11 06
package main

import (
    "fmt"
)

// cetakDeret mencetak setiap suku dari deret berdasarkan aturan yang diberikan.
func cetakDeret(n int) {
    for n != 1 {
        // Mencetak suku saat ini
        fmt.Printf("%d ", n)

        // Menghitung suku berikutnya
        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
```

```
    // Mencetak angka 1 sebagai suku terakhir
    fmt.Println(1)
}

func main() {
    var n int

    // Meminta input dari pengguna
    fmt.Print("Masukkan nilai awal n: ")
    fmt.Scan(&n)

    // Memastikan n valid (kurang dari 1.000.000)
    if n < 1 || n >= 1000000 {
        fmt.Println("Nilai n harus lebih dari 0 dan kurang dari
1.000.000")
        return
    }

    // Memanggil prosedur cetakDeret untuk mencetak deret
    cetakDeret(n)
}
```

Screenshot Program



Deskripsi Program

Kodingan Go ini mencetak deret angka berdasarkan nilai awal yang diberikan oleh pengguna. Fungsi cetakDeret mencetak setiap suku dari deret berdasarkan aturan tertentu: jika suku saat ini genap, maka dibagi dua; jika ganjil, maka dikalikan tiga dan ditambah satu. Deret ini akan terus berlanjut sampai mencapai angka 1. Fungsi main meminta input nilai awal dari pengguna dan memastikan nilainya valid, yaitu lebih dari 0 dan kurang dari 1.000.000. Setelah validasi, fungsi cetakDeret dipanggil untuk mencetak deret tersebut. Program ini menunjukkan penerapan aturan matematis sederhana dalam mencetak deret angka.