

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV  
PROSEDUR**



**Disusun Oleh :**

**ALTHAF TEGAR SOFYAN / 2311102217**

**Kelas 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 Definisi Prosedur

Prosedur merupakan beberapa intruksi program menjadi suatu intruksi baru yang dibuat untuk mengurangi kerumitan dari kode program yang kompleks pada suatu program yang besar. Prosedur akan menghasilkan suatu akibat atau efek langsung pada program Ketika dipanggil pada program utama. Suatu subprogram dikatakan prosedur apabila:

- Tidak ada deklarasi tipe nilai yang dikembalikan
- Tidak terdapat kata kunci return dalam badan subprogram

Kedudukan prosedur sama seperti intruksi dasar yang sudah ada sebelumnya dan intruksi yang berasal dari paket (fmt), seperti fmt.Sean dan fmt.Print. Karena itu selalu pilih nama prosedur yang berbentuk kata kerja atau sesuatu yang merepresentasikan proses sebagai nama dari prosedur. Contoh: cetak, hitungRerata, carNilai, belok, mulai.

### 4.2 Deklarasi Prosedur

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

	Notasi Algoritma
1	procedure «nama procedure» (<params>)
2	kamus
3	<i>/deklarasi variabel jika dari procedure</i>
4	
5	algoritma
6	<i>/badan algoritma procedure</i>
7	
8	end procedure
	Notasi dalam bahasa Go
9	func <nama procedure> (<params>) {
10	<i>/* deklarasi variabel jika dari procedure */</i>
11	
12	<i>/* badan algoritma procedure */</i>
13	
14	}

Contoh deklarasi prosedur mencetak n nilai pertama dari fibonanci

	Notas1 A1gor1tma
1	procedure cetakNFibo(in n : integer)
2	kamus
3	f1, f2, f3, 1 : integer
4	algoritma
5	f2 ← 0
6	f3 ← 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
	Notas1 dalam bahasa 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	}

### 4.3 Caze Pemanggil Prosedure

Suatu prosedur hanya akan dieksekusi apabila dipanggil baik secara langsung atau tidak langsung oleh program utama. Pemanggilan suatu procedure cukup mudah, yaitu dengan menuliskan nama beserta parameter atau argument yang diminta dari suatu proseure.

Contoh:

	Notas1 A1gor1tma
7	program contohprosedur
2	kamus
3	x : :integer
4	algoritma
5	x ← 5
6	cetakNFibo(x)                    {ca ra penangg1fan #1}
7	cetakNFibo(100)                {ca ra penangg:11an #2}

8	endp rogram
	Notasi dalam bahasa Go
9	func main() {
10	var x 1nt
11	x 5
12	cetakNFibo(x) {cara pemanggilan #1}
13	cetakNFibo(100) (cara pemanggilan #2}
14	)

Dari contoh di atas terlihat bahwa cara pemanggilan dengan notasi pseudocode dan GoLang adalah sama. Argumen yang digunakan untuk parameter n berupa integer (sesuai deklarasi) yang terdapat pada suatu variabel (cara pemanggilan 11) atau nilainya secara langsung (cara pemanggilan 12).

#### 4.4 Parameter

Suatu subprogram yang dipanggil dapat berkomunikasi dengan pemanggilnya melalui argument yang diberikan melalui parameter yang dideklarasikan pada subprogramnya. Berdasarkan letak penulisannya pada program, maka parameter dikelompokkan menjadi dua, yaitu parameter formal dan parameter actual.

##### 1. Prameter Formal

Parameter formal adalah parameter yang ditulis pada saat deklarasi suatu subprogram, parameter ini berfungsi sebagai petunjuk bahwa argumen apa saja yang diperlukan pada saat pemanggilan subprogram.

##### 2. Parameter Aktual

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.

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

##### 1. Pass by Value

Nilai pada parameter aktual akan disalin ke 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 tidak 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 semua parameter formal pada fungsi adalah pass by value, sedangkan pada prosedur diberi kata kunci in pada saat penulisan parameter formal. Sedangkan pada bahasa pemrograman Go sama seperti fungsi pada pseudocode, tidak terdapat kata kunci khusus untuk parameter formal fungsi dan prosedur

## 2. Pass by Reference

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 parameter formal tersebut akan berdampak pada parameter aktual. Artinya nilai terakhirnya akan dapat diketahui oleh si pemanggil 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 (\*) sebelum tipe data di parameter formal yang menjadi pass by reference.

## II. GUIDED

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

### Source Code

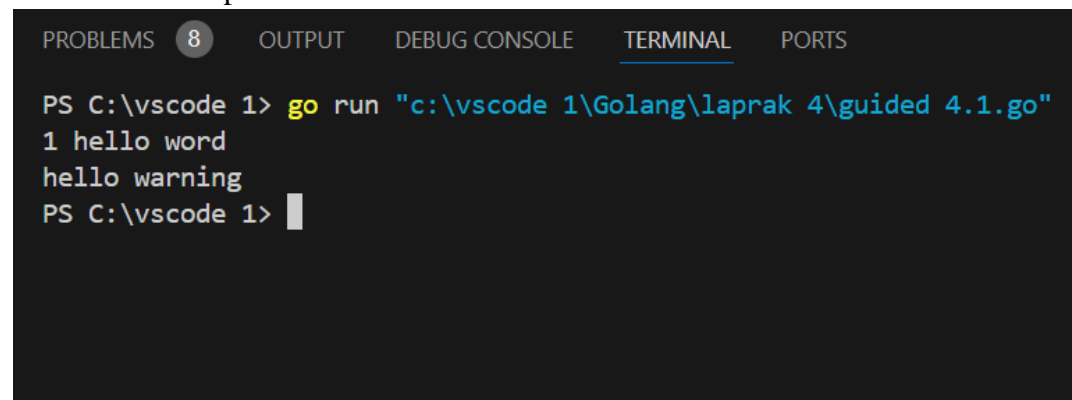
```
package main

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)
}
```

### Screenshoot Output



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\laprak 4\guided 4.1.go"
1 hello word
hello warning
PS C:\vscode 1> 
```

Deskripsi Program: Program diatas bertujuan untuk mencetak pesan dengan jenis berdasarkan nilai integer yang akan dimasukkan oleh pengguna. Program memiliki dua inputan yaitu bilangan dan pesan. Kemudian terdapat fungsi cetakPesan yang dimana menerima dua parameter yaitu pesan(M) dan flag(flag). Flag memiliki implementasi tersendiri, yaitu 0 artinya “error”, 1 artinya “warning”, dan 2 artinya “informasi”. Program akan mencetak pesan yang sesuai dengan pesan yang diberikan.

2. Buatlah sebuah program beserta prosedur yang digunakan untuk mengirimkan pemberitahuan mengenai notifikasi email baru.

#### Source Code

```
package main

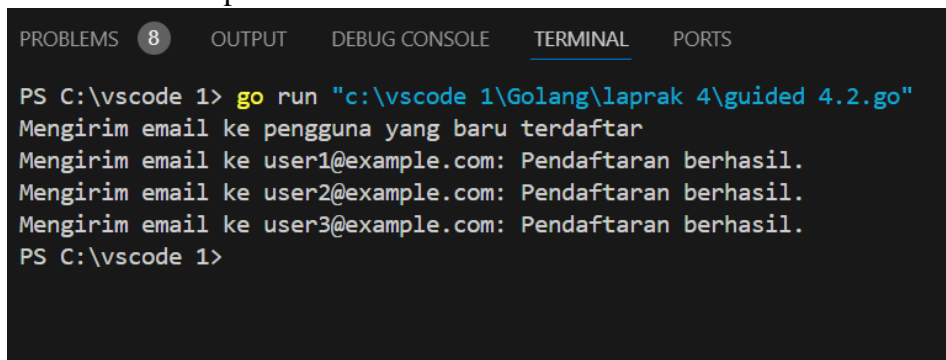
import "fmt"

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

func main() {
    // Daftar email pengguna baru
    emails := []string{"user1@example.com",
        "user2@example.com", "user3@example.com"}

    // Mengirim email pemberitahuan ke setiap
    pengguna
    fmt.Println("Mengirim email ke pengguna yang
    baru terdaftar")
    for _, email := range emails {
        sendEmailNotification(email)
    }
}
```

#### Screenshoot Output



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\vscode 1> go run "c:\vscode 1\Golang\laprak 4\guided 4.2.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:\vscode 1>
```

Deskripsi Program: Program diatas bertujuan untuk mengirim email pemberitahuan kepada pengguna yang baru terdaftar. Terdapat dua fungsi yaitu fungsi sendEmailNotification dan fungsi main. Fungsi sendEmail akan menerima satu parameter email yang dimana merupakan alamat pengguna dan akan mencetak pesan bahwa email pendaftaran sedang dikirim ke alamat email tersebut. Fungsi main akan mendefinisikan alamat email pengguna yang baru dalam bentuk slice dan disini menggunakan loop agar setiap email yang terdaftar dapat memanggil fungsi sendEmail untuk mengirim pemberitahuan. Lalu terakhir program akan mencetak pesan bahwa email sedang dikirim kepada setiap pengguna baru.

3. Buatlah program mengenai pass by value dan pass by reference (pointer)  
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) { //pass by reference
    *hasil = float64(2*x) - 0.5*float64(y) + 3.0
//pass by value
}

func main(){
    var a, b int
    var c float64

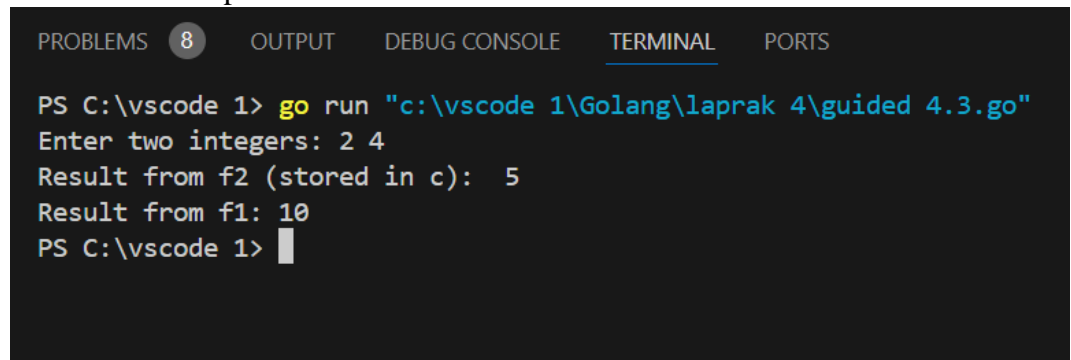
    //take input for a and b
    fmt.Print("Enter two integers: ")
    fmt.Scan(&a, &b)

    //call f2 to calculate and store the result in c
    f2(a, b, &c)

    //Print the result from f2
    fmt.Println("Result from f2 (stored in c): ", c)

    //call f1 and print the result
    resultF1 := f1(b, a) //(6 4)
    fmt.Println("Result from f1:", resultF1)
}
```

#### Screenshoot Output



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\vscode 1> go run "c:\vscode 1\Golang\laprak 4\guided 4.3.go"
Enter two integers: 2 4
Result from f2 (stored in c): 5
Result from f1: 10
PS C:\vscode 1> █
```

Deskripsi Program: Program ini menggunakan tiga fungsi yaitu fungsi f1, fungsi f2, dan fungsi main. Fungsi f1 akan menerima dua integer (x dan y) dan akan mengembalikan hasil perhitungan berupa float64, rumusnya yaitu  $2*x - 0.5*y + 3.0$ . Fungsi f2 juga sama menerima dua parameter integer (x dan y). Fungsi ini akan menghitung nilai yang sama dengan f1 dan menyimpan ke alamat memori. Fungsi main meminta pengguna untuk memasukkan dua integer (a dan b), lalu memanggil fungsi f2 untuk menghitung dan menyimpan hasil perhitungan di variabel c.



### III. UNGUIDED

1. Buatlah sebuah program dengan permasalahan seperti 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 kalian membantu Jonas? 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 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!

#### Source Code

```
package main

import "fmt"

// Menghitung faktorial
func factorial(n_217 int) int {
    if n_217 == 0 {
        return 1
    }
    return n_217 * factorial(n_217-1)
}

// Menghitung permutasi
func permutation(n_217, r_217 int) int {
    return factorial(n_217) / factorial(n_217-r_217)
}

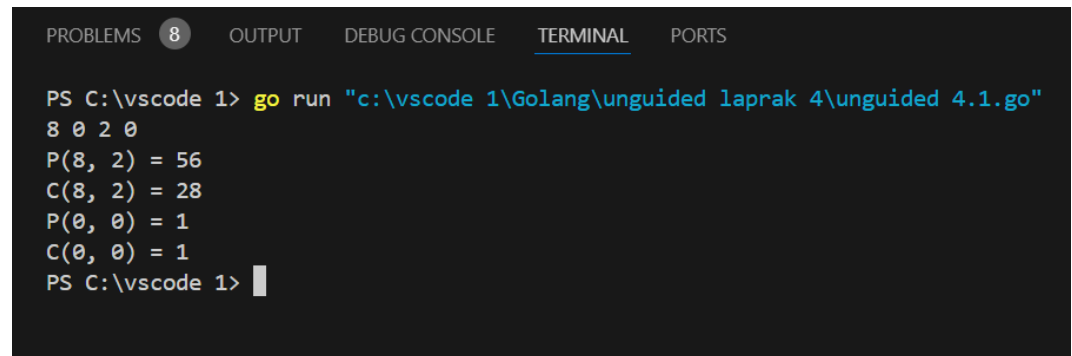
// Menghitung kombinasi
func combination(n_217, r_217 int) int {
    return factorial(n_217) / (factorial(r_217) *
    factorial(n_217-r_217))
}

func main() {
    var a, b, c, d int
    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

    // Permutasi dan kombinasi untuk a terhadap c
    p_a_c := permutation(a, c)
    c_a_c := combination(a, c)
    fmt.Printf("P(%d, %d) = %d\nC(%d, %d) = %d\n", a, c, p_a_c,
    a, c, c_a_c)

    // Permutasi dan kombinasi untuk b terhadap d
    p_b_d := permutation(b, d)
    c_b_d := combination(b, d)
    fmt.Printf("P(%d, %d) = %d\nC(%d, %d) = %d\n", b, d, p_b_d,
    b, d, c_b_d)
}
```

## Screenshoot Output



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 4\unguided 4.1.go"
8 0 2 0
P(8, 2) = 56
C(8, 2) = 28
P(0, 0) = 1
C(0, 0) = 1
PS C:\vscode 1> 
```

**Deskripsi Program:** Program diatas bertujuan untuk menghitung permutasi dan kombinasi berdasarkan input yang dimasukkan oleh pengguna. Di dalam program terdapat 4 fungsi, yaitu fungsi factorial untuk menghitung nilai factorial dari suatu bilangan, fungsi permutasi untuk menghitung nilai permutasi dengan Menyusun banyaknya r benda dari n benda, fungsi kombinasi untuk menghitung nilai kombinasi dengan cara banyaknya cara memilih r benda dari n benda, dan yang terakhir fungsi main. Fungsi main akan menerima input 4 bilangan integer lalu menghitung dan menampilkan nilai permutasi dan kombinasi.

2. Diberikan 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 di dalam prosedur.

procedure hitungscore (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.

### Source Code

```
package main

import "fmt"

func hitungScore(peserta_217 [][]int) (int, int) {
    soal := 0
    skor := 0
    for _, waktu := range peserta_217 {
        if waktu <= 301 {
            soal++
            skor += waktu
        }
    }
    return soal, skor
}

func main() {
    var nama, pemenang string
    var peserta [][]int
    var maxSoal217, minSkor217 int

    for {
        fmt.Print("Masukkan nama peserta ('selesai' untuk berhenti): ")
        fmt.Scan(&nama)
        if nama == "selesai" {
            break
        }

        var waktu [8]int
        fmt.Println("Masukkan waktu pengerjaan soal (dalam menit):")
        for i := 0; i < 8; i++ {
            fmt.Scanf("%d", &waktu[i])
```

```

    }

    peserta = append(peserta, waktu[:])
    soal, skor := hitungScore(waktu[:])
    if soal > maxSoal217 || (soal == maxSoal217 && skor <
minSkor217) {
        maxSoal217 = soal
        minSkor217 = skor
        pemenang = nama
    }
}

fmt.Printf("Pemenang: %s\nJumlah soal: %d\nTotal skor: %d\n",
pemenang, maxSoal217, minSkor217)
}

```

### Screenshoot Output

```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 4\unguided laprak 4.2\unguided 4.2.go"
Masukkan nama peserta ('selesai' untuk berhenti): astuti
Masukkan waktu pengerjaan soal (dalam menit):
20 50 61 71 75 10
Masukkan nama peserta ('selesai' untuk berhenti): bertha
Masukkan waktu pengerjaan soal (dalam menit):
25 47 26 50 60 65
Masukkan nama peserta ('selesai' untuk berhenti): selesai
Pemenang: bertha
Jumlah soal: 8
Total skor: 273
PS C:\vscode 1>

```

**Deskripsi Program:** Program tersebut implementasi sederhana dari sebuah kompetisi pemrograman. Pengguna dapat memasukkan nama peserta dan waktu pengerjaan soal, lalu program akan menentukan pemenang berdasarkan jumlah soal yang berhasil diselesaikan dan total waktu pengerjaan. Fungsi `hitungScore` digunakan untuk menghitung jumlah soal dan skor untuk setiap peserta, sedangkan fungsi `main` bertugas untuk mengumpulkan data, mencari pemenang, dan menampilkan hasil akhir.

3. Skiena dan Ittekkala dalam Programming Challenges mendefinisikan sebuah deret bilangan. Deret dimulai dengan sebuah bilangan bulat  $n$ . Jika bilangan  $n$  saat itu genap, maka suku berikutnya adalah  $1/2 n$ , 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 selalu mencapai suku dengan nilai 1. Buat program skiena yang akan mencetak setiap suku dari deret yang dijelaskan di atas untuk nilai suku awal yang diberikan.

Pencetakan deret harus dibuat dalam prosedur cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal.

procedure 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.

#### Source Code

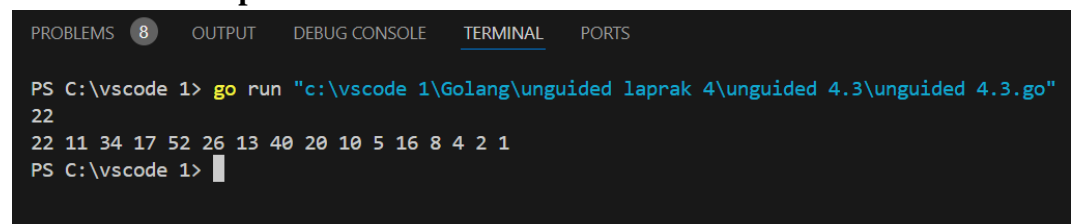
```
package main

import "fmt"

// Prosedur untuk mencetak deret bilangan
func cetakDeret(n int) {
    for n != 1 {
        fmt.Print(n, " ")
        if n%2 == 0 {
            n = n / 2
        } else {
            n = 3*n + 1
        }
    }
    fmt.Println()
}

func main() {
    var n int
    fmt.Scan(&n)
    cetakDeret(n)
}
```

#### Screenshot Output



```
PS C:\vscode 1> go run "c:\vscode 1\Golang\unguided laprak 4\unguided 4.3\unguided 4.3.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\vscode 1>
```

**Deskripsi Program:** Program ini berfungsi untuk menghitung deret bilangan. Pengguna memasukkan sebuah bilangan bulat positif  $n$ , kemudian program akan mencetak deret bilangan yang dimulai dari  $n$ .

Jika bilangan saat ini genap, maka bilangan berikutnya adalah bilangan saat ini dibagi 2. Jika bilangan saat ini ganjil, maka bilangan berikutnya adalah 3 kali bilangan saat ini ditambah 1. Deret akan terus berlanjut sampai mencapai nilai 1, lalu program akan menampilkan keseluruhan deret tersebut.