

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROSEDUR**



Disusun Oleh :

Muhammad Ihab Aufa Rafi / 2311102226

S1IF-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

1.1 Definisi Procedure

Prosedur dapat dianggap sebagai potongan beberapa Instruksi program menjadi suatu Instruksi 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:

1. Tidak ada deklarasi tipe nilai yang dikembalikan, dan
2. Tidak terdapat kata kunci **return** dalam badan subprogram.

Kedudukannya prosedur sama seperti Instruksi dasar yang sudah ada sebelumnya (assignment) dan/atau instruksi yang berasal dari paket (**fmt**), seperti **fmt.Scan** 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**, **cariNilai**, **belok**, **mulai**,...

1.2 Deklarasi Procedure

Deklarasi Cara deklarasi prosedur di pemrograman melibatkan beberapa langkah dasar, tergantung pada bahasa yang digunakan. Secara umum, berikut adalah tahapan deklarasi procedure:

1. Nama Procedure: Diawali dengan nama yang menggambarkan tujuan dari procedure tersebut.
2. Parameter (jika ada): Menyertakan parameter yang menjadi masukan untuk procedure tersebut.
3. Blok Kode: Di dalamnya berisi instruksi atau langkah-langkah yang akan dijalankan oleh procedure tersebut.

Ini adalah contoh deklarasi prosedur dalam bahasa golang.

```
package main

import "fmt"
```

```
// Deklarasi Procedure di Golang
func CetakPesan(nama string) {
    fmt.Println("Halo, " + nama + "!")
}

func main() {
    // Memanggil procedure
    CetakPesan("Andi")
}
```

1.3 Cara Pemanggilan Procedure

Seperti yang sudah dijelaskan sebelumnya, suatu prosedur hanya akan dieksekusi apabila dipanggil baik secara langsung atau tidak langsung oleh program utama. Tidak langsung di sini maksudnya adalah prosedur dipanggil oleh program utama melalui perantara subprogram yang lain.

Pemanggilan suatu prosedur cukup mudah, yaitu dengan hanya menuliskan nama beserta parameter atau argumen yang diminta dari suatu prosedur. Sebagai contoh prosedur cetakNFibo di atas dipanggil dengan menuliskan namanya, kemudian sebuah variabel atau nilai integer tertentu sebagai argumen untuk parameter n.

1.4 Contoh Program dengan Fungsi

Contoh program dengan prosedur ada pada bagian Guided nomor 1.

II. GUIDED

1. Soal Studi Case

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

Masukan terdiri dari sebuah bilangan bulat flag (0 s.d. 2) dan sebuah string pesan **M**.

Keluaran berupa string pesan M beserta jenis pesannya, yaitu error, warning, atau informasi berdasarkan nilai flag 0, 1, dan 2 secara berturut-turut.

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)
}
```

Screenshoot Output

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Guided4-1\Guided4-1.go"
2 20
20 informasi
PS D:\Alpro2Golang> |
```

Deskripsi Program

Program ini merupakan sebuah program dalam bahasa Go yang bertujuan untuk mencetak sebuah pesan dengan kategori tertentu berdasarkan input yang diberikan oleh pengguna.

Program memiliki dua fungsi:

1. Fungsi main yang berfungsi untuk menerima input dari pengguna berupa sebuah bilangan dan sebuah pesan. Input ini kemudian diteruskan ke fungsi cetakPesan.
2. Fungsi cetakPesan yang bertugas untuk menentukan jenis pesan berdasarkan nilai flag (bilangan) yang diterima. Fungsi ini menggunakan percabangan if-else untuk menentukan jenis pesan:
 - Jika flag bernilai 0, jenis pesan adalah "error"
 - Jika flag bernilai 1, jenis pesan adalah "warning"
 - Jika flag bernilai 2, jenis pesan adalah "informasi"

2. Soal Studi Case

Sebuah perusahaan memiliki sistem pendaftaran pengguna baru, di mana setiap kali pengguna baru terdaftar, sistem harus mengirimkan email konfirmasi secara otomatis. Perusahaan ingin memastikan bahwa setiap pengguna yang baru terdaftar menerima pesan konfirmasi melalui email. Untuk memenuhi kebutuhan ini, seorang developer diminta membuat program sederhana yang secara otomatis mengirimkan notifikasi email ke daftar pengguna baru setelah pendaftaran berhasil.

Sourcecode

```
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  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Guided4-2\Guided4-2.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 D:\Alpro2Golang>

```

Deskripsi Program

Program ini merupakan sebuah program dalam bahasa Go yang mensimulasikan sistem pengiriman email pemberitahuan kepada pengguna yang baru terdaftar.

Program memiliki dua fungsi:

1. Prosedur `sendEmailNotification` yang bertugas untuk mensimulasikan pengiriman email pemberitahuan. Prosedur ini menerima parameter berupa string email dan akan mencetak pesan konfirmasi pengiriman email ke alamat tersebut.
2. Fungsi `main` yang berisi:
 - Deklarasi slice `emails` yang menyimpan daftar alamat email pengguna baru.
 - Perulangan menggunakan `for range` untuk mengirim email ke setiap alamat dalam slice `emails`.

Program akan mengoutputkan pesan konfirmasi pengiriman email untuk setiap alamat email yang ada dalam daftar. Program ini bisa digunakan sebagai dasar untuk sistem notifikasi email yang sebenarnya, meskipun dalam contoh ini hanya mensimulasikan pengiriman dengan mencetak pesan ke layar.

Struktur program ini mendemonstrasikan penggunaan slice dan perulangan dalam Go, serta menunjukkan bagaimana prosedur dapat digunakan untuk memisahkan logika pengiriman email dari fungsi utama program. layar.

3. Soal Studi Case

Seorang mahasiswa diminta untuk membuat program yang mengimplementasikan dua fungsi matematika sederhana yang menerima dua bilangan bulat sebagai input dan mengembalikan hasil perhitungan. Fungsi pertama (f1) mengembalikan hasil secara langsung, sedangkan fungsi kedua (f2) memodifikasi hasil menggunakan pointer.

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) { //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
```

```

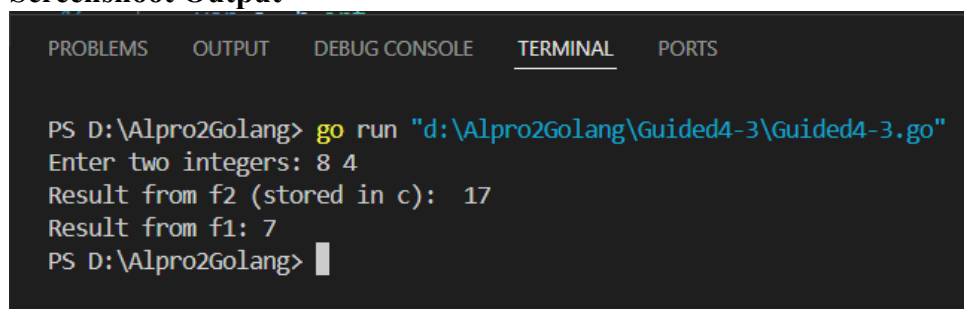
    f2(a, b, &c)

    //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  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Guided4-3\Guided4-3.go"
Enter two integers: 8 4
Result from f2 (stored in c): 17
Result from f1: 7
PS D:\Alpro2Golang>

```

Deskripsi Program

Program ini merupakan sebuah program dalam bahasa Go yang bertujuan untuk melakukan perhitungan matematika sederhana menggunakan dua metode berbeda yaitu fungsi dengan nilai kembalian (return) dan prosedur menggunakan pointer. Program tersebut memiliki fungsi **f1** untuk melakukan perhitungan dengan rumus $2x - 0.5y + 3.0$ dan mengembalikan hasil dalam bentuk **float64**. Prosedur **f2** memiliki fungsi yang sama namun menggunakan pointer untuk menyimpan hasil perhitungannya. Pada fungsi **main**, program akan meminta input dua bilangan integer dari pengguna, kemudian menggunakan kedua fungsi tersebut untuk melakukan perhitungan. Program akan mengoutputkan hasil perhitungan dari kedua fungsi tersebut, dimana hasil f2 disimpan terlebih dahulu dalam variabel c sebelum dicetak, sedangkan hasil f1 langsung dicetak setelah fungsi dipanggil.

III. UNGUIDED

1. 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 kalian 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!

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

Sourcecode

```
package main

import "fmt"

var a, b, c, d int

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

func permutasi(n, r int, hasil *int) {
    var faktN, faktNR int
    faktorial(n, &faktN)
    faktorial(n-r, &faktNR)
    *hasil = faktN / faktNR
}

func kombinasi(n, r int, hasil *int) {
    var faktN, faktR, faktNR int
```

```

    faktorial(n, &faktN)
    faktorial(r, &faktR)
    faktorial(n-r, &faktNR)
    *hasil = faktN / (faktR * faktNR)
}

func main() {
    fmt.Printf("Masukkan Input (dipisahkan oleh spasi) = ")
    fmt.Scan(&a, &b, &c, &d)

    if a >= c && b >= d {
        var perm1_226, komb1_226, perm2_226, komb2_226 int

        permutasi(a, c, &perm1_226)
        kombinasi(a, c, &komb1_226)
        fmt.Printf("%d, %d\n", perm1_226, komb1_226)

        permutasi(b, d, &perm2_226)
        kombinasi(b, d, &komb2_226)
        fmt.Printf("%d, %d\n", perm2_226, komb2_226)
    } else {
        fmt.Println("Syarat tidak terpenuhi: a harus >= c
dan b harus >= d")
    }
}

```

Screenshoot Output

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided3-1\Unguided3-1.go"
Masukkan Input (dipisahkan oleh spasi) = 6 5 4 3
360, 15
60, 10
PS D:\Alpro2Golang>

```

Deskripsi Program

Program ini merupakan sebuah program dalam bahasa Go yang bertujuan untuk menghitung nilai permutasi dan kombinasi dari dua pasang angka yang diinputkan. Program ini menggunakan prosedur-prosedur untuk melakukan perhitungan matematika yang diperlukan.

Program memiliki tiga prosedur utama:

1. Prosedur **faktorial** yang berfungsi untuk menghitung nilai faktorial dari sebuah bilangan dengan menggunakan perulangan for.

2. Prosedur **permutasi** yang menggunakan prosedur faktorial untuk menghitung nilai permutasi dengan rumus $n!/(n-r)!$
3. Prosedur **kombinasi** yang juga menggunakan prosedur faktorial untuk menghitung nilai kombinasi dengan rumus $n!/(r!(n-r)!)$

Pada fungsi **main**, program akan meminta input empat bilangan (a, b, c, d) yang dipisahkan oleh spasi. Program kemudian melakukan pengecekan syarat dimana nilai a harus lebih besar atau sama dengan c, dan nilai b harus lebih besar atau sama dengan d. Jika syarat terpenuhi, program akan menghitung dan menampilkan nilai permutasi dan kombinasi untuk pasangan (a,c) dan (b,d). Jika syarat tidak terpenuhi, program akan menampilkan pesan error.

Program menggunakan pointer untuk menyimpan hasil perhitungan dari setiap prosedur, sehingga tidak menggunakan nilai kembalian (return). Metode ini memungkinkan prosedur untuk mengubah nilai variabel yang berada di luar scope prosedur tersebut.

2. Soal Studi Case

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 hitung Skor (in/out soal, skor: integer)

Setiap baris **masukan** dimulai dengan satu string nama peserta tersebut diikuti dengan 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 (
    "fmt"
)

func hitungSkor(soal *int, skor *int, waktu []int) {
    *soal = 0
    *skor = 0

    // Menghitung jumlah soal yang diselesaikan dan total waktu
    for _, t := range waktu {
        if t != 301 { // Jika waktu bukan 301 atau berhasil diselesaikan
            *soal++
            *skor += t
        }
    }
}

func main() {
    var peserta []string
    var waktu [][]int

    // Akan membaca input sampai nama "Selesai"
    for {
        var nama string
        fmt.Scan(&nama)
        if nama == "Selesai" {
            break
        }

        // Membaca 8 waktu pengerjaan
        var times []int
        for i := 0; i < 8; i++ {
            var t int
            fmt.Scan(&t)
            times = append(times, t)
        }

        peserta = append(peserta, nama)
        waktu = append(waktu, times)
    }
}

```

```

// Mencari pemenang
var maxSoal_226, minWaktu_226 int = -1, 10000
var pemenang_226 string

for i := 0; i < len(peserta); i++ {
    var jmlSoal, totalWaktu int
    hitungSkor(&jmlSoal, &totalWaktu, waktu[i])

    if jmlSoal > maxSoal_226 || (jmlSoal == maxSoal_226
&& totalWaktu < minWaktu_226) {
        maxSoal_226 = jmlSoal
        minWaktu_226 = totalWaktu
        pemenang_226 = peserta[i]
    }
}

// Cetak hasil
fmt.Printf("%s %d %d\n", pemenang_226, maxSoal_226,
minWaktu_226)
}

```

Screenshoot Output

```

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided4-2\Unguided4-2.go"
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Ihab 10 30 301 100 9 35 20 40
Selesai
Ihab 7 244
PS D:\Alpro2Golang>

```

Deskripsi Program

Program ini merupakan sebuah program dalam bahasa Go yang bertujuan untuk menentukan pemenang dari sebuah kompetisi pemrograman tingkat nasional dimana setiap peserta diberikan 8 soal yang harus diselesaikan dalam waktu maksimal 5 jam. Program ini dibuat secara modular dengan menggunakan prosedur **hitungSkor** untuk menghitung total soal yang berhasil diselesaikan dan total waktu pengerjaan dari setiap peserta.

Prosedur **hitungSkor** menerima parameter berupa pointer untuk menyimpan jumlah soal dan total waktu, serta array yang berisi waktu

pengerjaan 8 soal. Prosedur ini menggunakan perulangan untuk mengiterasi array waktu dan menghitung jumlah soal yang berhasil diselesaikan (waktu tidak sama dengan 301) serta mengakumulasi total waktu pengerjaannya.

Pada program utama, data peserta dan waktu pengerjaan disimpan dalam slice yang akan dibaca secara berulang sampai input "Selesai" diterima. Setiap data peserta terdiri dari nama dan 8 nilai waktu pengerjaan untuk masing-masing soal. Program menggunakan perulangan untuk mencari pemenang berdasarkan dua kriteria: jumlah soal terbanyak, atau jika jumlah soal sama maka waktu pengerjaan tersingkat yang akan menang.

Program akan mengoutputkan nama pemenang beserta jumlah soal yang berhasil diselesaikan dan total waktu pengerjaannya. Jika peserta tidak berhasil menyelesaikan suatu soal atau tidak mengirimkan jawaban, maka secara otomatis waktu pengerjaan untuk soal tersebut dianggap 301 menit (5 jam 1 menit) dan tidak dihitung sebagai soal yang berhasil diselesaikan.

3. Soal Studi Case

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 $1/2 n$, 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.

Sourcecode

```
package main

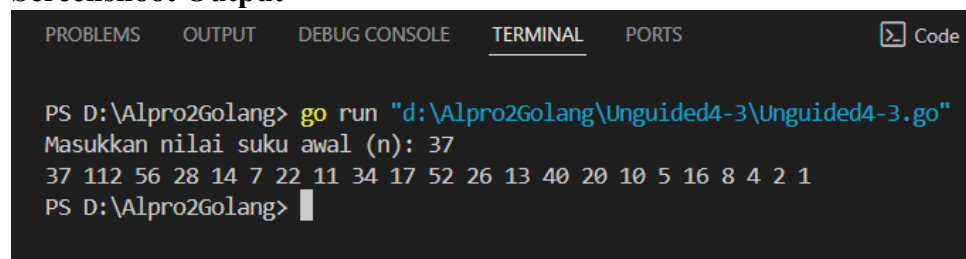
import "fmt"

func cetakDeret(n_226 int) {
    fmt.Printf("%d", n_226)

    for n_226 != 1 {
        if n_226 % 2 == 0 {
            n_226 = n_226 / 2
        } else {
            n_226 = 3 * n_226 + 1
        }
        fmt.Printf(" %d", n_226)
    }
    fmt.Println()
}

func main() {
    var n int
    fmt.Printf("Masukkan nilai suku awal (n): ")
    fmt.Scan(&n)
    if n > 0 && n < 1000000{
        cetakDeret(n)
    }else {
        fmt.Println("Input tidak valid. Silakan masukkan
bilangan bulat positif lebih kecil dari 1000000.")
    }
}
```

Screenshoot Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Code

PS D:\Alpro2Golang> go run "d:\Alpro2Golang\Unguided4-3\Unguided4-3.go"
Masukkan nilai suku awal (n): 37
37 112 56 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\Alpro2Golang> 
```

Deskripsi Program

Program ini merupakan sebuah program dalam bahasa Go yang bertujuan untuk menghasilkan dan mencetak deret bilangan berdasarkan definisi Skiena dan Revilla dalam Programming Challenges. Program ini dibuat secara modular dengan menggunakan prosedur **cetakDeret** yang menerima satu parameter berupa bilangan bulat positif sebagai nilai suku awal.

Prosedur **cetakDeret** menggunakan perulangan for dan percabangan if-else untuk menghasilkan suku-suku deret. Perulangan akan terus berlanjut selama nilai suku belum mencapai 1. Di dalam perulangan, program menggunakan percabangan if-else untuk menentukan rumus yang akan digunakan: jika suku saat itu genap maka suku berikutnya adalah setengah dari nilai saat ini, sedangkan jika ganjil maka suku berikutnya adalah tiga kali nilai saat ini ditambah satu.

Pada program utama, nilai suku awal dibaca dari input pengguna. Program kemudian memanggil prosedur **cetakDeret** dengan parameter nilai suku awal tersebut. Prosedur akan mencetak setiap suku dari deret dalam satu baris, dengan setiap suku dipisahkan oleh sebuah spasi. Program akan berhenti ketika deret mencapai nilai 1, yang menurut spesifikasi soal pasti akan tercapai untuk setiap nilai awal yang lebih kecil dari 1.000.000.