

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 2**  
**MODUL IV**  
**PROSEDUR**



**Disusun Oleh :**

**Rasyid Nafsyarie / 2311102011**

**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

Prosedur atau fungsi adalah bagian dari program yang mengelompokkan sekumpulan instruksi untuk menjalankan tugas tertentu. Dalam pemrograman, prosedur memungkinkan kita untuk menghindari pengulangan kode, membuat kode lebih terstruktur, modular, dan mudah dikelola. Prosedur juga dapat menerima parameter dan mengembalikan nilai, yang membantu meningkatkan fleksibilitas kode.

Di Golang, prosedur dikenal sebagai **fungsi** (function). Fungsi dalam Go didefinisikan dengan kata kunci `func`, diikuti oleh nama fungsi, parameter (opsional), tipe data yang dikembalikan (jika ada), dan badan fungsi.

```
func namaFungsi(parameter1 tipe, parameter2 tipe) (returnType) {
```

```
    // Logika fungsi
```

```
    return nilai
```

```
}
```

```
package main
```

```
import "fmt"
```

```
// Definisi fungsi
```

```
func tambah(a int, b int) int {
```

```
    return a + b
```

```
}
```

```
func main() {  
  
    // Memanggil fungsi  
  
    hasil := tambah(3, 4)  
  
    fmt.Println("Hasil penjumlahan:", hasil)  
  
}
```

Golang memiliki fitur menarik, yaitu fungsi dapat mengembalikan lebih dari satu nilai. Fitur ini sangat berguna saat kita ingin mengembalikan hasil sekaligus pesan error atau status lainnya.

## II. GUIDED

### 1. Guided 1

#### Soal Studi Case

XXXXXXXXXXXXXXXXXXXX

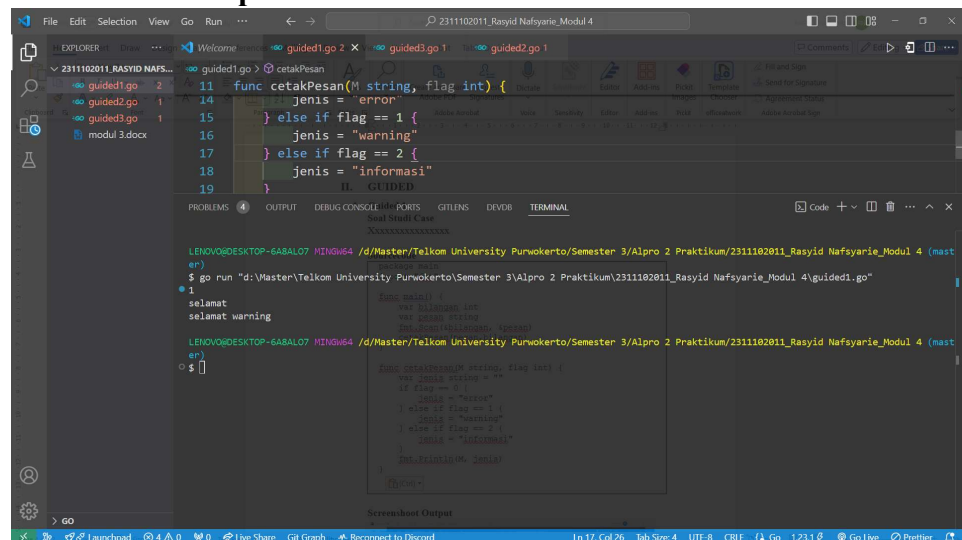
#### Sourcecode

```
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)
}
```

#### Screenshot Output



The screenshot shows a Go IDE with the source code and terminal output. The source code is the same as provided in the previous block. The terminal output shows the execution of the program with the following commands and results:

```
LENOVODESKTOP-GABAL07 MINGW64 /d:/Master/Telkom University Purwokerto/Semester 3/Alpro 2 Praktikum/2311102011_Rasyid Nafsyarie_Modul 4 (mast
er)
$ go run "d:/Master/Telkom University Purwokerto/Semester 3/Alpro 2 Praktikum/2311102011_Rasyid Nafsyarie_Modul 4/guided1.go"
1
selamat
selamat warning
LENOVODESKTOP-GABAL07 MINGW64 /d:/Master/Telkom University Purwokerto/Semester 3/Alpro 2 Praktikum/2311102011_Rasyid Nafsyarie_Modul 4 (mast
er)
$ []
```

#### Deskripsi Program

Di sini ada dua variabel yang dideklarasikan pada fungsi main yaitu bilangan: variabel dengan tipe int yang akan menyimpan input angka dari pengguna, pesan: variabel dengan tipe string yang akan menyimpan input teks (pesan) dari pengguna. Setelah mendapatkan input dari pengguna, program memanggil fungsi cetakPesan dengan memberikan dua argumen pesan: teks yang dimasukkan pengguna. bilangan: angka yang dimasukkan pengguna. Selanjutnya program akan memproses input dan memanggil fungsi cetakPesan.

## 2. Guided 2

### Soal Studi Case

XXXXXXXXXXXXXXXXXX

### Sourcecode

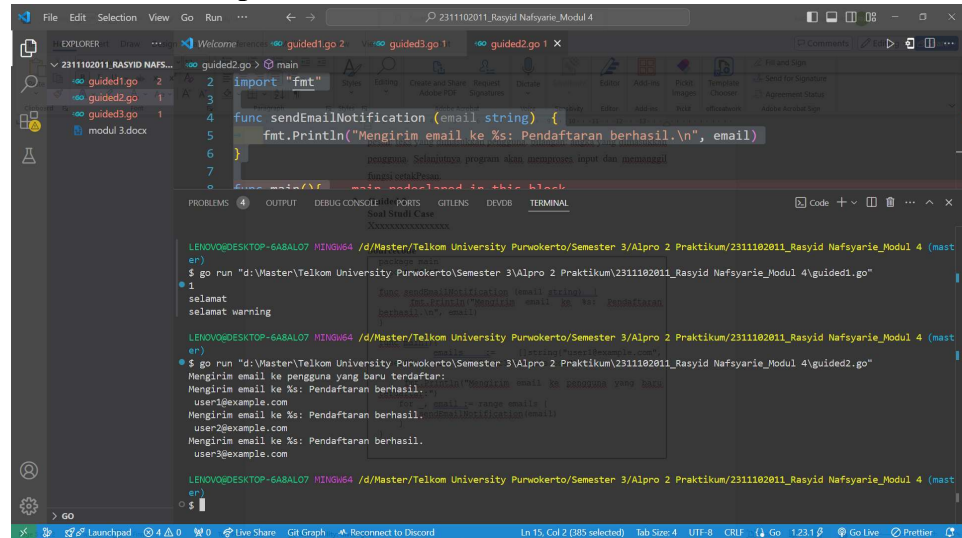
```
package main
import "fmt"

func sendEmailNotification (email string) {
    fmt.Println("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)
    }
}
```

## Screenshoot Output



## Deskripsi Program

Fungsi sendEmailNotification jika fungsi ini dipanggil dengan email = "user1@example.com", pesan yang akan dicetak seharusnya adalah: *Mengirim email ke user1@example.com: Pendaftaran berhasil.*

### 3. Guided 3

#### Soal Studi Case

XXXXXXXXXXXXXXXXXXXX

#### 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)

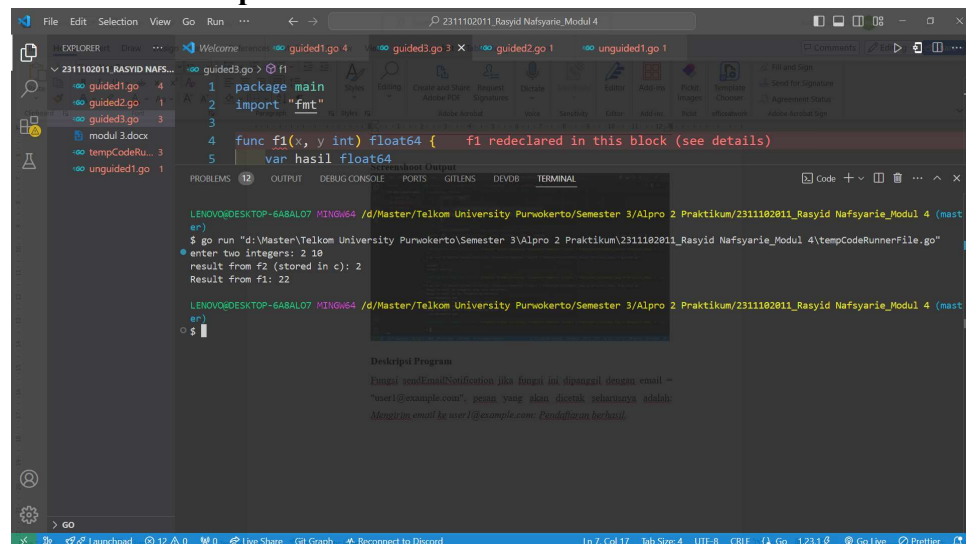
    f2(a, b, &c)

    fmt.Println("result from f2 (stored in c):", c)

    resultF1 := f1(b, a)
    fmt.Println("Result from f1:", resultF1)
}

```

## Screenshoot Output



## Deskripsi Program

Fungsi `f1` menerima dua parameter (`x` dan `y`), yang merupakan bilangan bulat (`int`), dan mengembalikan hasil perhitungan sebagai nilai dengan tipe `float64` (bilangan pecahan). Fungsi `f2` mirip dengan `f1`, tetapi hasilnya dikembalikan melalui pointer. Fungsi ini tidak secara langsung mengembalikan hasil, melainkan memperbarui nilai variabel yang dikirimkan melalui alamatnya (pointer).

### III. UNGUIDED

#### Unguided 1

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 kalla 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"

//Rasyid Nafsyarie 2311102011
var a, b, c, d int

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

func hitungPermutasi(n, r int) {
    hasil := faktorial(n) / faktorial(n-r)
    fmt.Printf("Permutasi(%d, %d) = %d\n", n, r, hasil)
}

func hitungKombinasi(n, r int) {
    hasil := faktorial(n) / (faktorial(r) * faktorial(n-r))
    fmt.Printf("Kombinasi(%d, %d) = %d\n", n, r, hasil)
}

func main() {
    fmt.Print("Masukkan input (a, b, c, d) = ")
}
```



```

    fmt.Scan(&a, &b, &c, &d)
    if a >= c && b >= d {
        hitungPermutasi(a, c)
        hitungKombinasi(a, c)
        hitungPermutasi(b, d)
        hitungKombinasi(b, d)
    } else {
        fmt.Println("Syarat tidak terpenuhi: a harus >= c
dan b harus >= d")
    }
}

```

## Screenshoot Output

```

package main

import "fmt"

func main() {
    var a, b, c, d int
    fmt.Print("Masukkan nilai: ")
    fmt.Scan(&a, &b, &c, &d)

    if a >= c && b >= d {
        hitungPermutasi(a, c)
        hitungKombinasi(a, c)
        hitungPermutasi(b, d)
        hitungKombinasi(b, d)
    } else {
        fmt.Println("Syarat tidak terpenuhi: a harus >= c
dan b harus >= d")
    }
}

```

Terminal Output:

```

$ go run "d:\Master\Telkom University Purwokerto\Semester 3\Alpro 2 Praktikum\2311102011_Rasyid Nafsyarie_Modul 3\unguided1.go"
Masukkan nilai: 10 20 21 22
Tidak memenuhi kondisi

$ go run "d:\Master\Telkom University Purwokerto\Semester 3\Alpro 2 Praktikum\2311102011_Rasyid Nafsyarie_Modul 3\unguided1.go"
Masukkan nilai: 6 12 4 8
Permutasi: 360
Kombinasi: 15
Permutasi: 19958400
Kombinasi: 495

```

## Deskripsi Program

Program diatas menghitung permutasi dan kombinasi berdasarkan input yang diberikan oleh pengguna. Program ini juga memeriksa apakah syarat input terpenuhi sebelum melakukan perhitungan. Fungsi ini menghitung faktorial dari suatu bilangan n. Faktorial adalah hasil perkalian semua bilangan bulat positif kurang dari atau sama dengan n. Fungsi ini digunakan untuk menghitung permutasi, yaitu banyaknya cara mengatur r elemen dari n elemen yang berbeda. Program ini meminta empat angka dari pengguna dan menghitung permutasi serta kombinasi untuk dua pasang angka (a, c dan b, d), tetapi hanya jika syarat  $a \geq c$  dan  $b \geq d$  terpenuhi.

## Unguided 2

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 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

//Rasyid Nafsyarie 2311102011

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

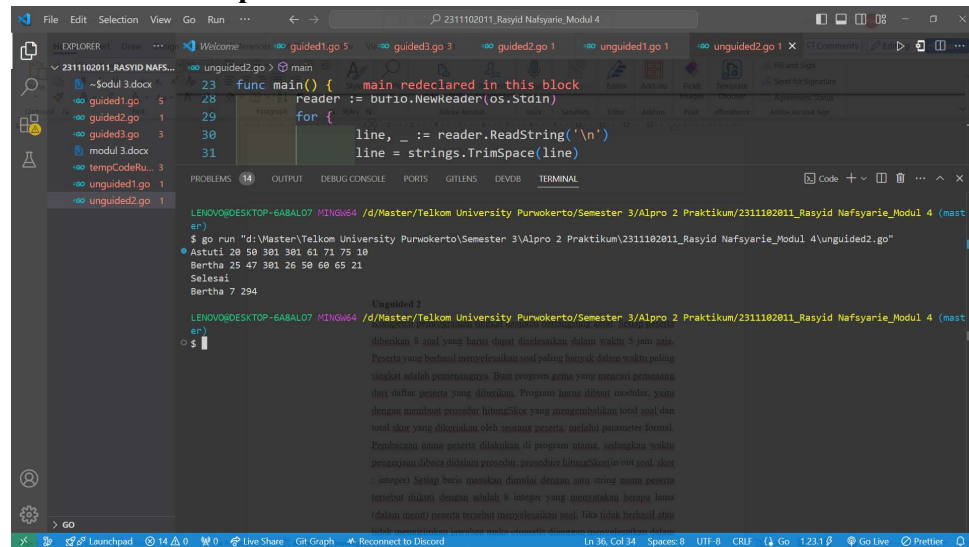
func hitungSkor(jumlahSoal *int, totalSkor *int, waktuPeserta []int) {
    *jumlahSoal = 0
    *totalSkor = 0
    for _, waktu := range waktuPeserta {
        if waktu <= 300 {
            *totalSkor += waktu
        }
    }
}
```

```

                                *jumlahSoal += 1
                                }
                                }
                                }
func main() {
    var namaPeserta string
    var waktuPeserta []int
    var pemenang string
    var maxSoal, minWaktu int
    reader := bufio.NewReader(os.Stdin)
    for {
        line, _ := reader.ReadString('\n')
        line = strings.TrimSpace(line)
        if line == "Selesai" {
            break
        }
        input := strings.Fields(line)
        namaPeserta = input[0]
        waktuPeserta = make([]int, 0)
        for _, w := range input[1:] {
            waktu, err := strconv.Atoi(w)
            if err == nil {
                waktuPeserta =
                append(waktuPeserta, waktu)
            }
        }
        var jumlahSoal, totalSkor int
        hitungSkor(&jumlahSoal, &totalSkor,
waktuPeserta)
        if jumlahSoal > maxSoal || (jumlahSoal ==
maxSoal &&
                                totalSkor < minWaktu) {
                                pemenang = namaPeserta
                                maxSoal = jumlahSoal
                                minWaktu = totalSkor
                                }
        }
        if pemenang != "" {
            fmt.Printf("%s %d %d\n", pemenang, maxSoal,
minWaktu)
        } else {
            fmt.Println("Tidak ada peserta.")
        }
    }
}

```

## Screenshoot Output



```
23 func main() {
28     reader := bufio.NewReader(os.Stdin)
29     for {
30         line, _ := reader.ReadString('\n')
31         line = strings.TrimSpace(line)
    }
}

$ go run "d:\Master\Telkom University Purwokerto\Semester 3\Alpro 2 Praktikum\2311182011_Rasyid Nafsyarie_Modul 4\unguided2.go"
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294

Diperoleh 2
Diberikan 8 soal yang harus dapat diselesaikan dalam waktu 5 jam saja.
Peserta yang berhasil menyelesaikan soal paling banyak dalam waktu paling
sangat adalah pemenangnya. Buat program game yang mencari pemenang
dari daftar peserta yang diberikan. Program harus dibuat modular, yang
dengan membuat prosedur hitungSkor yang mengembalikan total soal dan
total skor yang dikerjakan oleh seorang peserta, melalui parameter formal.
Prosedurnya harus peserta dilakukan di program utama, sedangkan waktu
penyelesaian dibaca melalui prosedur prosedur hitungSkor() yang soal, skor
... (text continues) ...
Selesai
Bertha 7 294
```

## Deskripsi Program

Program ini adalah program sederhana yang digunakan untuk menentukan pemenang dari sebuah kompetisi berdasarkan seberapa banyak soal yang dijawab peserta dan waktu yang mereka habiskan untuk menjawabnya. Program membaca input baris demi baris hingga pengguna mengetik "Selesai". Fungsi hitungSkor digunakan untuk menghitung jumlah soal yang dijawab peserta dalam waktu yang lebih kecil atau sama dengan 300 detik. Hanya soal yang selesai dalam waktu  $\leq 300$  detik yang dihitung. Nama peserta dengan skor terbanyak dan waktu total paling sedikit disimpan dalam variabel pemenang. Ketika input "Selesai" diberikan, program akan menampilkan nama pemenang, jumlah soal yang dijawab, dan total waktu yang dihabiskan untuk soal-soal tersebut.

### 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

//Rasyid Nafsyarie 2311102011
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.")
    }
}

```

## Screenshoot Output

The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Go program with a function `cetakDeret` that prints a sequence of numbers based on the Collatz conjecture. The terminal shows the output of running the program with different inputs. For input 20, it prints the sequence: 20 10 5 16 8 4 2 1. For input 25, it prints: 25 12 6 3 10 5 16 8 4 2 1. For input 7, it prints: 7 10 5 16 8 4 2 1. The terminal also shows the error message for an invalid input: "Input tidak valid. Harap masukkan bilangan bulat positif yang lebih kecil dari 1000000."

## Deskripsi Program

Program ini adalah program yang menghasilkan deret angka berdasarkan aturan dari Collatz conjecture, yaitu aturan matematika yang mengatakan bahwa jika kita mengambil bilangan positif apa pun, mengikuti langkah-langkah tertentu, akhirnya kita akan selalu mencapai angka 1. Jika  $n$  adalah bilangan genap, maka kita bagi  $n$  dengan 2 ( $n = n / 2$ ). Jika  $n$  adalah bilangan ganjil, kita kalikan  $n$  dengan 3 lalu ditambah 1 ( $n = 3 * n + 1$ ). Program membatasi input  $n$  pada rentang 1 hingga 999.999. Hal ini dilakukan untuk memastikan bahwa input valid dan menghindari masalah performa jika pengguna memasukkan bilangan yang sangat besar.