

**LAPORAN PRAKTIKUM
ALGORITME DAN PEMROGRAMAN 2**

**MODUL 4
PROSEDUR**



Oleh:

MUHAMMAD AMIR SALEH

2311102233

IF - 11 - 06

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. Dasar Teori

Di Golang, tidak ada perbedaan khusus antara prosedur dan fungsi, karena keduanya didefinisikan menggunakan kata kunci `func`. Namun, prosedur biasanya merujuk pada fungsi yang tidak mengembalikan nilai. Prosedur di Go dideklarasikan sama seperti fungsi lainnya, hanya saja tanpa tipe pengembalian. Secara umum, baik prosedur maupun fungsi di Go memiliki sintaks yang sama, dan perbedaannya hanya terletak pada apakah mereka mengembalikan nilai atau tidak.

II. Guided Guided 1

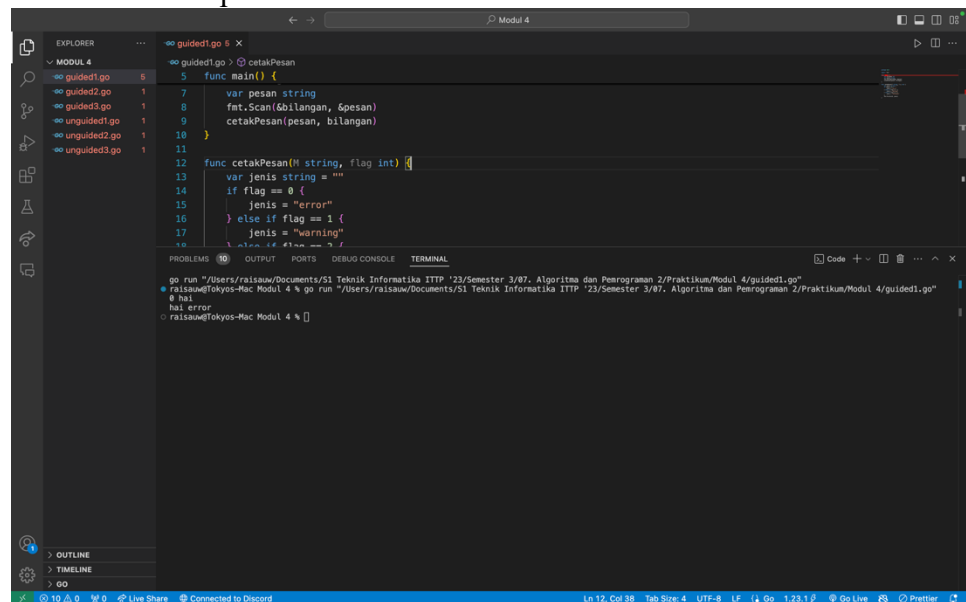
```
package main

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)
}
```

Screenshots Output



Deskripsi:

Program ini adalah program untuk mencetak pesan dengan jenis berdasarkan nilai flag tertentu. Fungsi cetakPesan menerima dua parameter: M sebagai pesan teks dan flag sebagai penanda jenis pesan. Jika flag bernilai 0, jenis pesan adalah “error”; jika bernilai 1, jenisnya “warning”; dan jika bernilai 2, jenisnya “informasi”. Di dalam fungsi main(), program meminta input dari pengguna berupa sebuah bilangan (bilangan) dan pesan teks (pesan). Kemudian, fungsi cetakPesan dipanggil dengan parameter tersebut untuk mencetak pesan dan jenis yang sesuai berdasarkan nilai flag.

Guided 2

```
package main

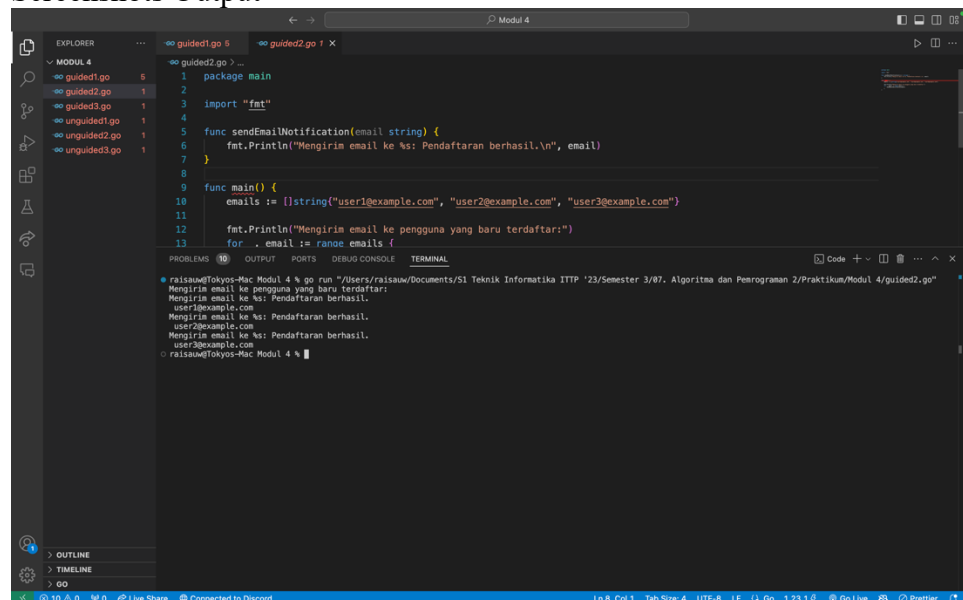
import "fmt"

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)
    }
}
```

Screenshots Output

The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORER' sidebar shows a project named 'MODUL 4' with files 'guided1.go', 'guided2.go', 'guided3.go', 'unguided1.go', 'unguided2.go', and 'unguided3.go'. The main editor area displays the Go code from the 'Guided 2' section. Below the code, the 'TERMINAL' tab is active, showing the output of running the program. The output consists of three lines: 'Mengirim email ke pengguna yang baru terdaftar:', 'Mengirim email ke %s: Pendaftaran berhasil. user1@example.com', 'Mengirim email ke %s: Pendaftaran berhasil. user2@example.com', and 'Mengirim email ke %s: Pendaftaran berhasil. user3@example.com'. The status bar at the bottom indicates the file is at 'Ln 8, Col 1' and the editor is using 'UTF-8' encoding.

Deskripsi:

Program ini adalah program untuk mencetak pesan dengan jenis berdasarkan nilai flag tertentu. Fungsi cetakPesan menerima dua parameter: M sebagai pesan teks dan flag sebagai penanda jenis pesan. Jika flag bernilai 0, jenis pesan adalah “error”; jika bernilai 1, jenisnya “warning”; dan jika

bernilai 2, jenisnya “informasi”. Di dalam fungsi `main()`, program meminta input dari pengguna berupa sebuah bilangan (bilangan) dan pesan teks (pesan). Kemudian, fungsi `cetakPesan` dipanggil dengan parameter tersebut untuk mencetak pesan dan jenis yang sesuai berdasarkan nilai flag.

Guided 3

```
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

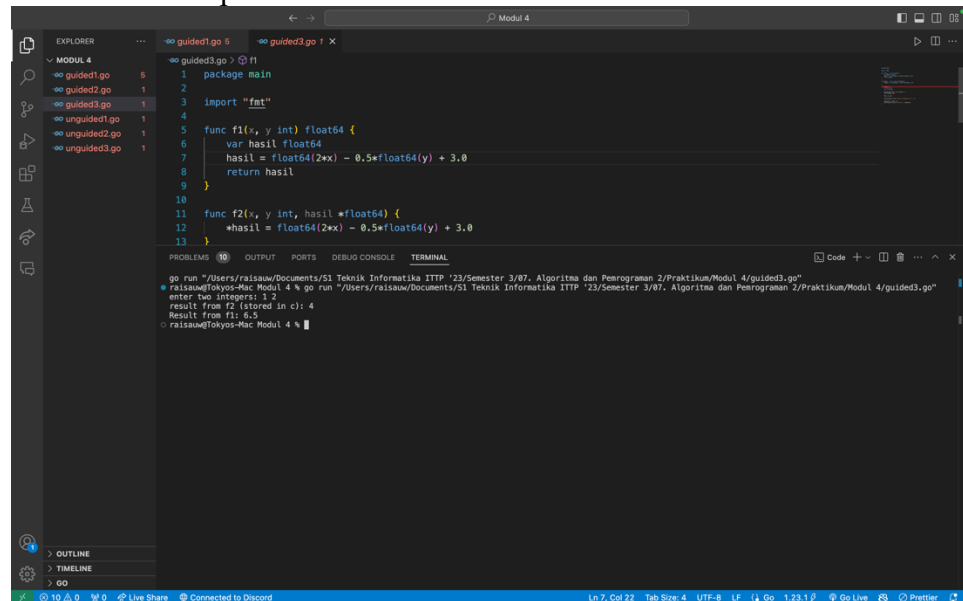
    fmt.Print("enter two integers: ")
    fmt.Scan(&a, &b)

    f2(a, b, &c)

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

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

Screenshots Output



```
1 package main
2
3 import "fmt"
4
5 func f1(x, y int) float64 {
6     var hasil float64
7     hasil = float64(2*x) - 0.5*float64(y) + 3.0
8     return hasil
9 }
10
11 func f2(x, y int, hasil *float64) {
12     *hasil = float64(2*x) - 0.5*float64(y) + 3.0
13 }
14
15 func main() {
16     var a, b int
17     fmt.Println("Enter two integers: ")
18     fmt.Scanln(&a, &b)
19     f1(a, b)
20     f2(a, b, &hasil)
21     fmt.Println("Result from f1: ", hasil)
22 }
```

go run "/Users/raisaaw/Documents/S1 Teknik Informatika ITPP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 4/guided3.go"

enter two integers: 1 2

result from f1 (stored in c): 4

Result from f1: 8.5

raisaaw@Tokyo-Mac Modul 4 %

Deskripsi:

Program ini adalah program untuk menghitung hasil dari suatu persamaan menggunakan dua fungsi berbeda, f1 dan f2. Fungsi f1 menerima dua bilangan bulat (x dan y) dan mengembalikan hasil persamaan $2*x - 0.5*y + 3.0$ dalam bentuk float64. Fungsi f2 juga menerima dua bilangan bulat yang sama, tetapi menyimpan hasil perhitungannya langsung di alamat variabel hasil menggunakan pointer. Di dalam main, program meminta input dari pengguna berupa dua bilangan bulat (a dan b). Fungsi f2 dipanggil untuk menghitung dan menyimpan hasilnya di variabel c. Hasil dari f1 kemudian dihitung dengan memasukkan b dan a sebagai argumen, dan kedua hasil dari f1 dan f2 dicetak ke layar.

III. Unguided Unguided 1

```
package main

import (
    "fmt"
)

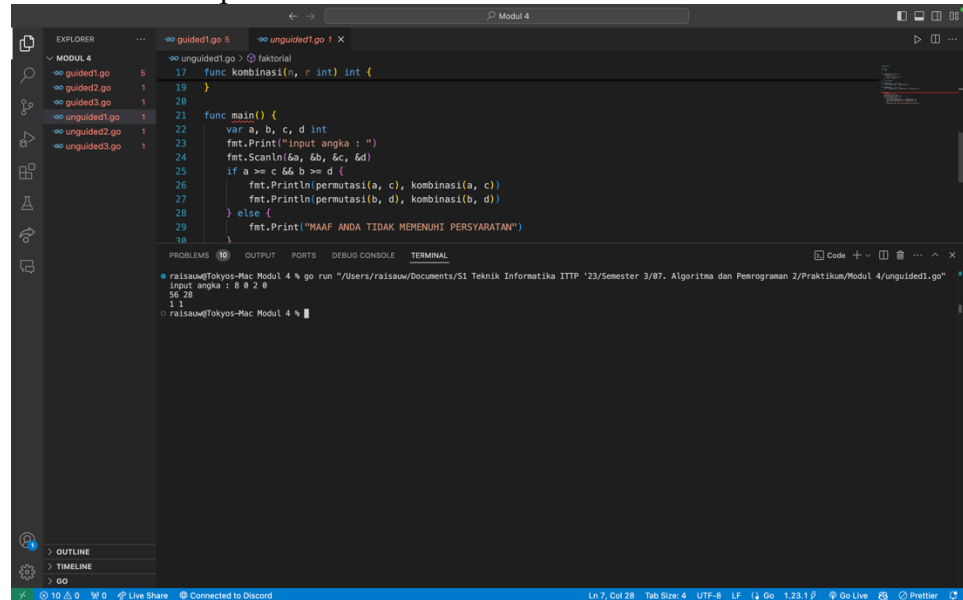
func faktorial(n int) int {
    var hasil int = 1
    for i := 1; i <= n; i++ {
        hasil = hasil * i
    }
    return hasil
}

func permutasi(n, r int) int {
    return faktorial(n) / faktorial(n-r)
}

func kombinasi(n, r int) int {
    return faktorial(n) / (faktorial(r) * faktorial(n-r))
}

func main() {
    var a, b, c, d int
    fmt.Print("input angka : ")
    fmt.Scanln(&a, &b, &c, &d)
    if a >= c && b >= d {
        fmt.Println(permutasi(a, c), kombinasi(a, c))
        fmt.Println(permutasi(b, d), kombinasi(b, d))
    } else {
        fmt.Print("MAAF ANDA TIDAK MEMENUHI PERSYARATAN")
    }
}
```

Screenshots 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 the following structure:

```
17 func faktorial(n int) int {  
18     if n == 0 {  
19         return 1  
20     }  
21     return n * faktorial(n-1)  
22 }  
23  
24 func main() {  
25     var a, b, c, d int  
26     fmt.Println("input angka : ")  
27     fmt.Scanln(&a, &b, &c, &d)  
28     if a >= c && b >= d {  
29         fmt.Println(permutasi(a, c), kombinasi(a, c))  
30         fmt.Println(permutasi(b, d), kombinasi(b, d))  
31     } else {  
32         fmt.Println("MAAF ANDA TIDAK MEMENUHI PERSYARATAN")  
33     }  
34 }
```

The terminal at the bottom shows the execution of the program with the following output:

```
raisaum@Tokyo-Mac Modul 4 % go run "/Users/raisaum/Documents/S1 Teknik Informatika ITTP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 4/unguided1.go"  
input angka : 8 8 2 8  
56 28  
1 1  
raisaum@Tokyo-Mac Modul 4 %
```

Deskripsi:

Program ini adalah program untuk menghitung permutasi dan kombinasi dari dua pasang bilangan. Fungsi faktorial menghitung nilai faktorial dari bilangan yang diberikan. Fungsi permutasi menghitung permutasi $P(n, r)$ menggunakan rumus $P(n, r) = n! / (n - r)!$, sementara fungsi kombinasi menghitung kombinasi $C(n, r)$ dengan rumus $C(n, r) = n! / (r!(n - r)!)$. Di dalam fungsi main(), program menerima empat input bilangan dari pengguna (a, b, c, dan d). Jika $a \geq c$ dan $b \geq d$, maka program akan menghitung dan menampilkan permutasi serta kombinasi untuk kedua pasangan bilangan tersebut. Jika tidak memenuhi syarat, program akan menampilkan pesan “MAAF ANDA TIDAK MEMENUHI PERSYARATAN”.

Unguided 2

```
package main

import "fmt"

func hitungSkor(waktu [8]int) (soal int, total int) {
    for i := 0; i < 8; i++ {
        if waktu[i] <= 300 {
            soal = soal + 1
            total = total + waktu[i]
        }
    }
    return soal, total
}

func main() {
    var nama string
    var waktu [8]int
    var pemenangNama string
    var pemenangSoal int = 0
    var pemenangWaktu int = 0

    for {
        fmt.Scanln(&nama, &waktu[0], &waktu[1], &waktu[2],
        &waktu[3], &waktu[4], &waktu[5], &waktu[6], &waktu[7])

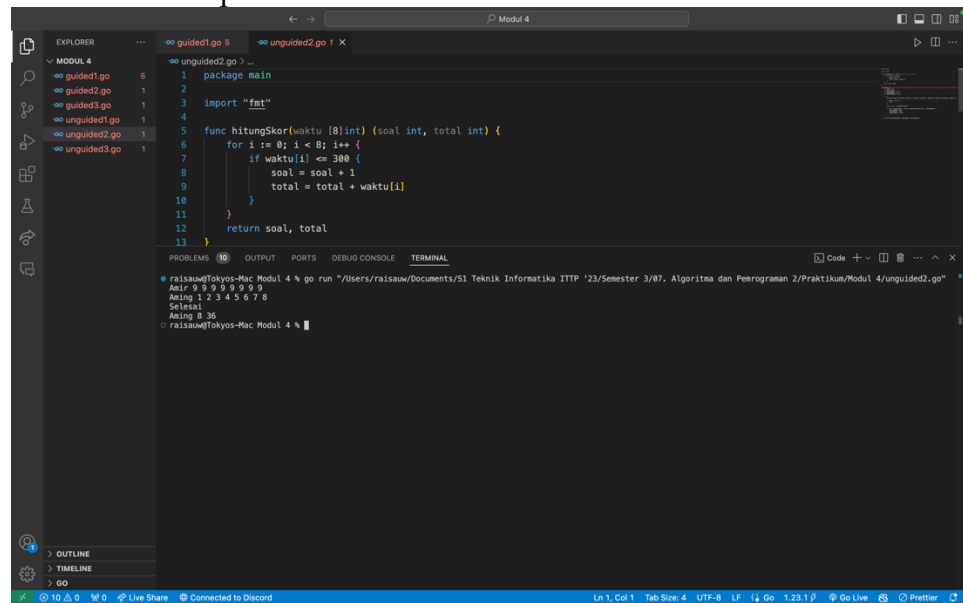
        if nama == "Selesai" {
            break
        }

        soal, total := hitungSkor(waktu)

        if soal > pemenangSoal || (soal == pemenangSoal &&
total < pemenangWaktu) {
            pemenangNama = nama
            pemenangSoal = soal
            pemenangWaktu = total
        }
    }

    fmt.Println(pemenangNama, pemenangSoal, pemenangWaktu)
}
```

Screenshots Output



```
1 package main
2
3 import "fmt"
4
5 func hitungSkor(waktu []int) (soal int, total int) {
6     for i := 0; i < 8; i++ {
7         if waktu[i] <= 300 {
8             soal = soal + 1
9             total = total + waktu[i]
10        }
11    }
12    return soal, total
13 }
```

Terminal Output:

```
raisaum@Tokyo-Mac Modul 4 % go run "/Users/raisaum/Documents/S1 Teknik Informatika ITP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 4/unguided2.go"
Masuk 9 9 9 9 9 9 9
Akhir 1 2 3 4 5 6 7 8
Selesai
Akhir 0 36
raisaum@Tokyo-Mac Modul 4 %
```

Deskripsi:

Program ini adalah program untuk menghitung dan menentukan pemenang berdasarkan skor dari beberapa peserta dalam sebuah kompetisi. Fungsi `hitungSkor` menerima array waktu yang berisi waktu untuk menyelesaikan 8 soal. Fungsi ini menghitung jumlah soal yang diselesaikan dalam waktu 300 detik atau kurang, serta total waktu yang dihabiskan untuk soal-soal tersebut. Di dalam fungsi `main()`, program terus meminta input dari pengguna berupa nama peserta dan waktu untuk setiap soal sampai pengguna memasukkan "Selesai". Setelah setiap input, program memeriksa apakah peserta baru tersebut memiliki jumlah soal yang diselesaikan lebih banyak atau jika sama, total waktu yang lebih sedikit dibandingkan dengan pemenang sebelumnya. Jika ya, peserta tersebut menjadi pemenang baru. Setelah semua input selesai, program mencetak nama pemenang, jumlah soal yang diselesaikan, dan total waktu yang dihabiskan.

Unguided 3

```
package main

import "fmt"

func hitungSukuBerikutnya(n int) int {
    if n%2 == 0 {
        return n / 2
    }
    return 3*n + 1
}

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

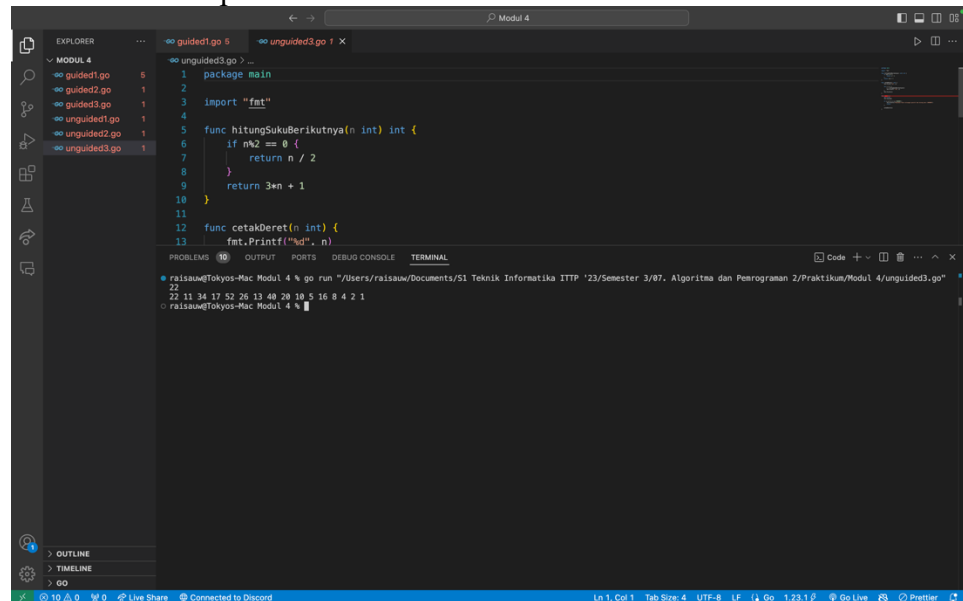
    for n != 1 {
        n = hitungSukuBerikutnya(n)
        fmt.Printf(" %d", n)
    }
    fmt.Println()
}

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

    if n <= 0 || n >= 1000000 {
        fmt.Println("Masukkan harus bilangan positif dan kurang
dari 1000000")
        return
    }

    cetakDeret(n)
}
```

Screenshots Output



```
1 package main
2
3 import "fmt"
4
5 func hitungSukuBerikutnya(n int) int {
6     if n%2 == 0 {
7         return n / 2
8     }
9     return 3*n + 1
10 }
11
12 func cetakDeret(n int) {
13     fmt.Printf("%d", n)
14 }
```

PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

```
raisaum@Tokyo-Mac Modul 4 % go run "/Users/raisaum/Documents/SI Teknik Informatika ITTP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 4/unguided3.go"
22
22 11 34 17 52 26 13 48 20 10 5 16 8 4 2 1
raisaum@Tokyo-Mac Modul 4 %
```

Deskripsi:

Program ini adalah program yang mencetak deret bilangan. Fungsi `hitungSukuBerikutnya` menerima bilangan bulat n dan menentukan suku berikutnya dalam deret: jika n genap, suku berikutnya adalah $n / 2$; jika n ganjil, suku berikutnya adalah $3n + 1$. Fungsi `cetakDeret` mencetak bilangan awal n , lalu terus menghitung dan mencetak suku-suku berikutnya hingga mencapai 1. Di dalam fungsi `main()`, program meminta pengguna untuk memasukkan bilangan awal, memastikan bahwa bilangan tersebut positif dan kurang dari 1.000.000. Jika input valid, program akan memanggil fungsi `cetakDeret` untuk mencetak deret dari bilangan tersebut.