

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 13**

**Repeat Until**



**Disusun oleh:**

**Farel Tri Julian**

**109082500163**

**S1IF-13-02**

**Asisten Praktikum**

Adithana dharma putra

Alfin Ilham Berlianto

**PROGRAM STUDI S1 INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## LATIHAN KELAS – GUIDED

### 1. Guided 1 Source Code

```
package main

import "fmt"

func main() {
    var kata string
    var jumlah int

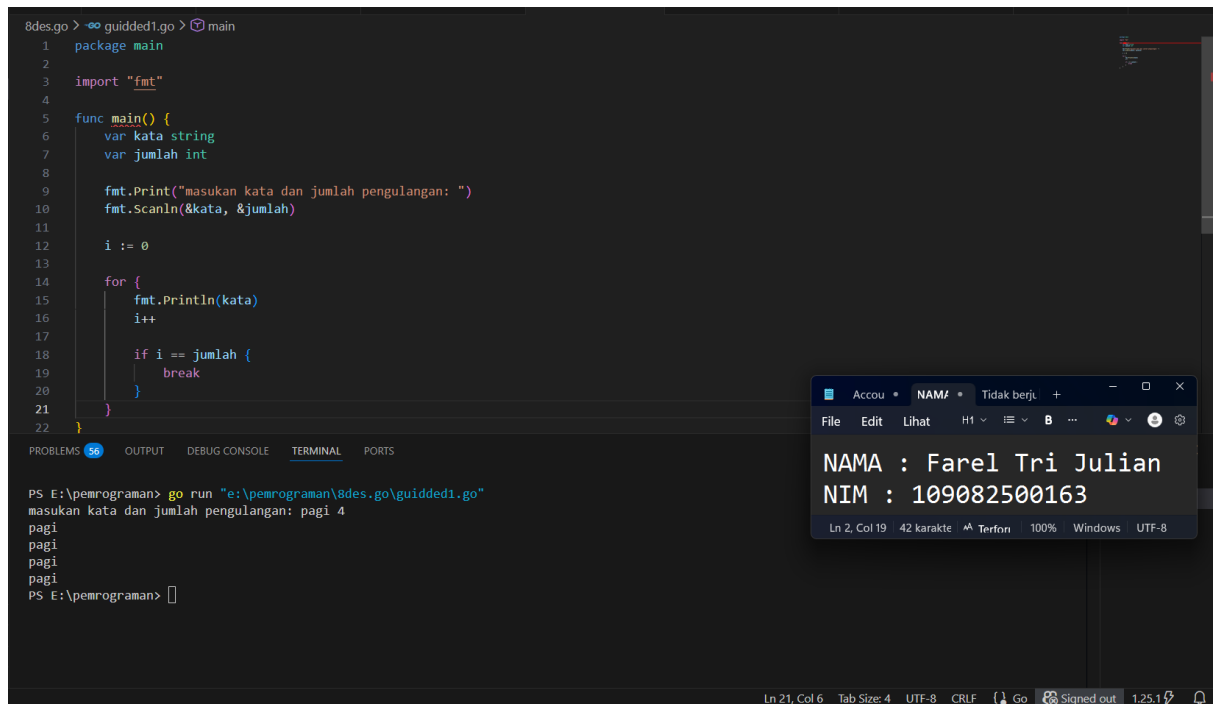
    fmt.Print("masukan kata dan jumlah pengulangan: ")
    fmt.Scanln(&kata, &jumlah)

    i := 0

    for {
        fmt.Println(kata)
        i++

        if i == jumlah {
            break
        }
    }
}
```

## Screenshoot program



The screenshot shows a Go program being executed. The code in the editor is as follows:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kata string
7     var jumlah int
8
9     fmt.Print("masukan kata dan jumlah pengulangan: ")
10    fmt.Scanln(&kata, &jumlah)
11
12    i := 0
13
14    for {
15        fmt.Println(kata)
16        i++
17
18        if i == jumlah {
19            break
20        }
21    }
22 }
```

The terminal output shows the program running and printing the word "pagi" four times, corresponding to the input value of 4. The text window displays the user's name and NIM:

```
NAMA : Farel Tri Julian
NIM : 109082500163
```

## Deskripsi program

Program Golang ini berfungsi untuk mencetak sebuah kata berulang kali ke layar sesuai dengan jumlah yang ditentukan oleh pengguna. Program dimulai dengan mendeklarasikan variabel kata untuk menyimpan teks dan jumlah untuk angka pengulangan, lalu menerima input pengguna melalui fungsi `fmt.Scanln`. Proses utama dijalankan menggunakan perulangan `for` tanpa kondisi (infinite loop) yang terus mencetak nilai kata dan menaikkan variabel penghitung `i`, di mana perulangan tersebut hanya akan berhenti ketika nilai `i` sudah sama dengan jumlah yang diinputkan.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
    var n int

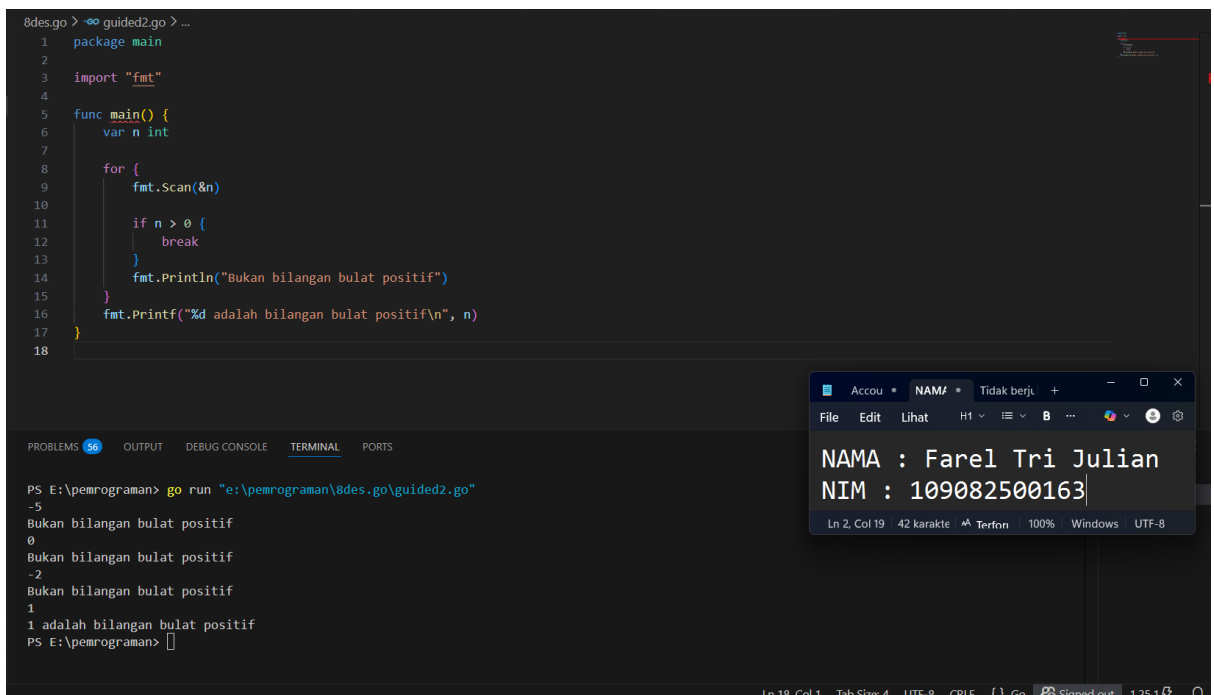
    for {
        fmt.Scan(&n)

        if n > 0 {
            break
        }

        fmt.Println("Bukan bilangan bulat positif")
    }

    fmt.Printf("%d adalah bilangan bulat positif\n", n)
}
```

### Screenshoot program



```
8des.go > -oo guided2.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7
8     for {
9         fmt.Scan(&n)
10
11         if n > 0 {
12             break
13         }
14         fmt.Println("Bukan bilangan bulat positif")
15     }
16     fmt.Printf("%d adalah bilangan bulat positif\n", n)
17 }
18
```

PROBLEMS 56 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\pemrograman> go run "e:\pemrograman\8des.go\guided2.go"
-5
Bukan bilangan bulat positif
0
Bukan bilangan bulat positif
-2
Bukan bilangan bulat positif
1
1 adalah bilangan bulat positif
PS E:\pemrograman>
```

NAMA : Farel Tri Julian  
NIM : 109082500163

Ln 2, Col 19 42 karakter Terfon 100% Windows UTF-8

Ln 18, Col 1 Tab Size: 4 UTF-8 CRLF Signed out 1.25.1

### Deskripsi program

program Golang ini berfungsi sebagai validasi input untuk memastikan pengguna memasukkan bilangan bulat positif. Menggunakan mekanisme perulangan tak terbatas, program akan terus-menerus membaca input ke dalam variabel `n` dan mengevaluasinya; jika nilai `n` kurang dari atau sama dengan nol, program akan menampilkan pesan peringatan "Bukan bilangan bulat positif" dan kembali meminta input. Perulangan hanya akan dihentikan secara paksa menggunakan perintah `break` ketika syarat `n > 0` terpenuhi, yang kemudian diakhiri dengan mencetak konfirmasi bahwa angka tersebut valid.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var x, y int

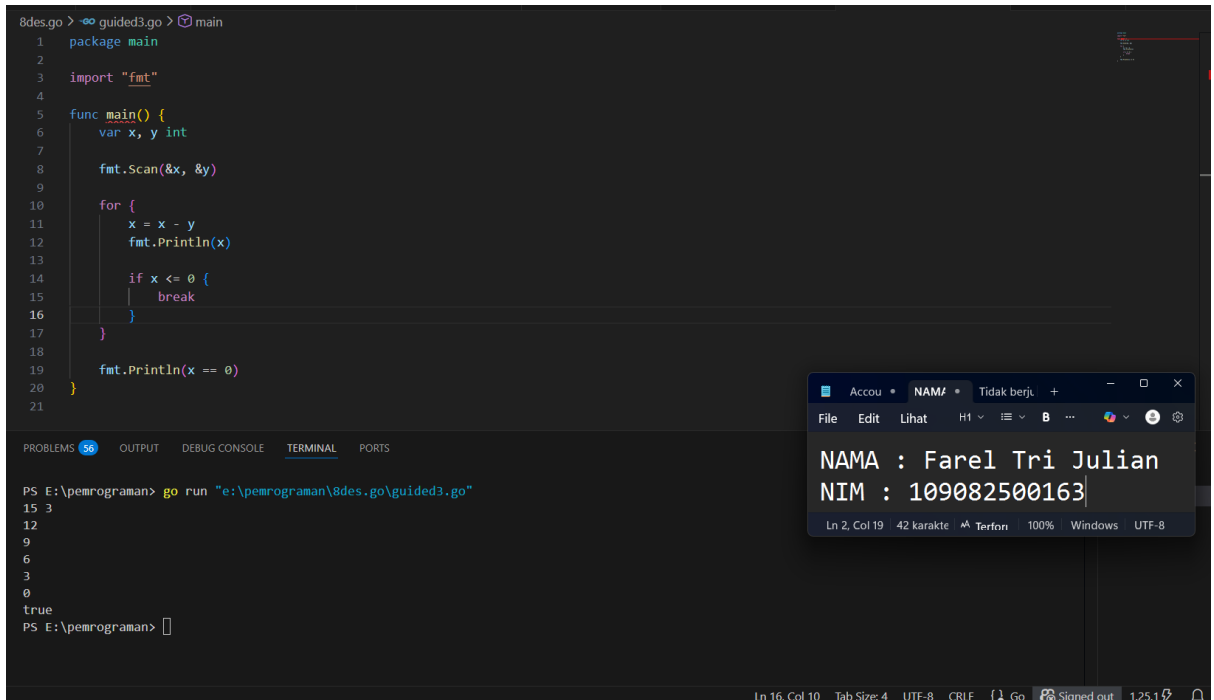
    fmt.Scan(&x, &y)

    for {
        x = x - y
        fmt.Println(x)

        if x <= 0 {
            break
        }
    }

    fmt.Println(x == 0)
}
```

## Screenshoot program



```
8des.go > guided3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x, y int
7
8     fmt.Scan(&x, &y)
9
10    for {
11        x = x - y
12        fmt.Println(x)
13
14        if x <= 0 {
15            break
16        }
17    }
18
19    fmt.Println(x == 0)
20 }
21
```

```
PS E:\pemrograman> go run "e:\pemrograman\8des.go\guided3.go"
15 3
12
9
6
3
0
true
PS E:\pemrograman>
```

## Deskripsi program

program Golang ini membaca dua input bilangan bulat (x dan y) lalu melakukan simulasi operasi pembagian menggunakan metode pengurangan berulang. Dalam sebuah perulangan, nilai x terus dikurangi dengan nilai y dan hasilnya ditampilkan langkah demi langkah hingga nilai x menjadi nol atau negatif. Setelah perulangan berhenti, program mencetak hasil evaluasi logika (true atau false) untuk menentukan apakah nilai x berakhir tepat di angka nol, yang menandakan bahwa nilai awal x habis dibagi oleh y

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)

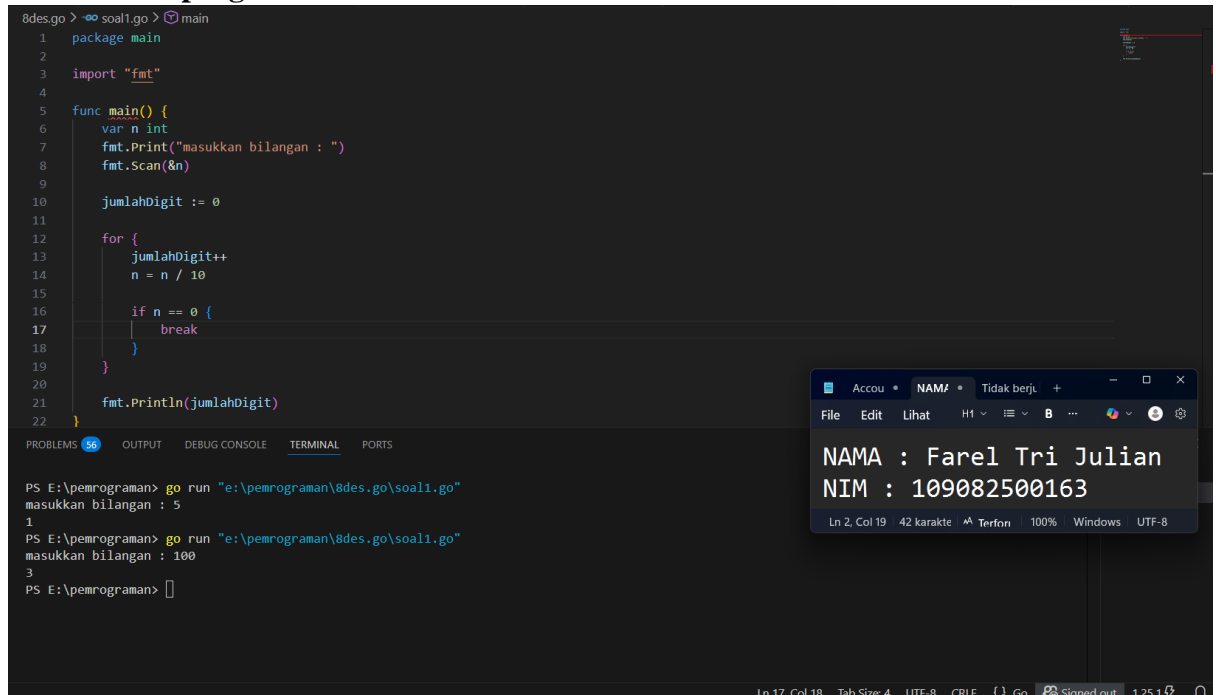
    jumlahDigit := 0

    for {
        jumlahDigit++
        n = n / 10

        if n == 0 {
            break
        }
    }

    fmt.Println(jumlahDigit)
}
```

## Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The program is a simple function that takes an integer input and calculates the number of digits. The terminal shows two test cases: input 5 resulting in 1 digit, and input 100 resulting in 3 digits.

```
8des.go > ~\soal1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("masukkan bilangan : ")
8     fmt.Scan(&n)
9
10    jumlahDigit := 0
11
12    for {
13        jumlahDigit++
14        n = n / 10
15
16        if n == 0 {
17            break
18        }
19    }
20
21    fmt.Println(jumlahDigit)
22 }
```

Terminal Output:

```
PS E:\pemrograman> go run "e:\pemrograman\8des.go\soal1.go"
masukkan bilangan : 5
1
PS E:\pemrograman> go run "e:\pemrograman\8des.go\soal1.go"
masukkan bilangan : 100
3
PS E:\pemrograman>
```

## Deskripsi program

program Golang ini dirancang untuk menghitung jumlah digit dari sebuah bilangan bulat yang dimasukkan oleh pengguna. Setelah menerima input angka ke dalam variabel `n`, program menjalankan perulangan tak terbatas yang melakukan dua hal: menaikkan variabel penghitung `jumlahDigit` dan memangkas angka tersebut digit demi digit menggunakan operasi pembagian integer dengan 10 ( $n = n / 10$ ). Perulangan ini akan terus berjalan sampai nilai `n` habis atau menjadi nol, kemudian program akan mencetak hasil akhir dari `jumlahDigit` yang merepresentasikan panjang angka tersebut.



## 2. Tugas 2

### Source code

```
package main

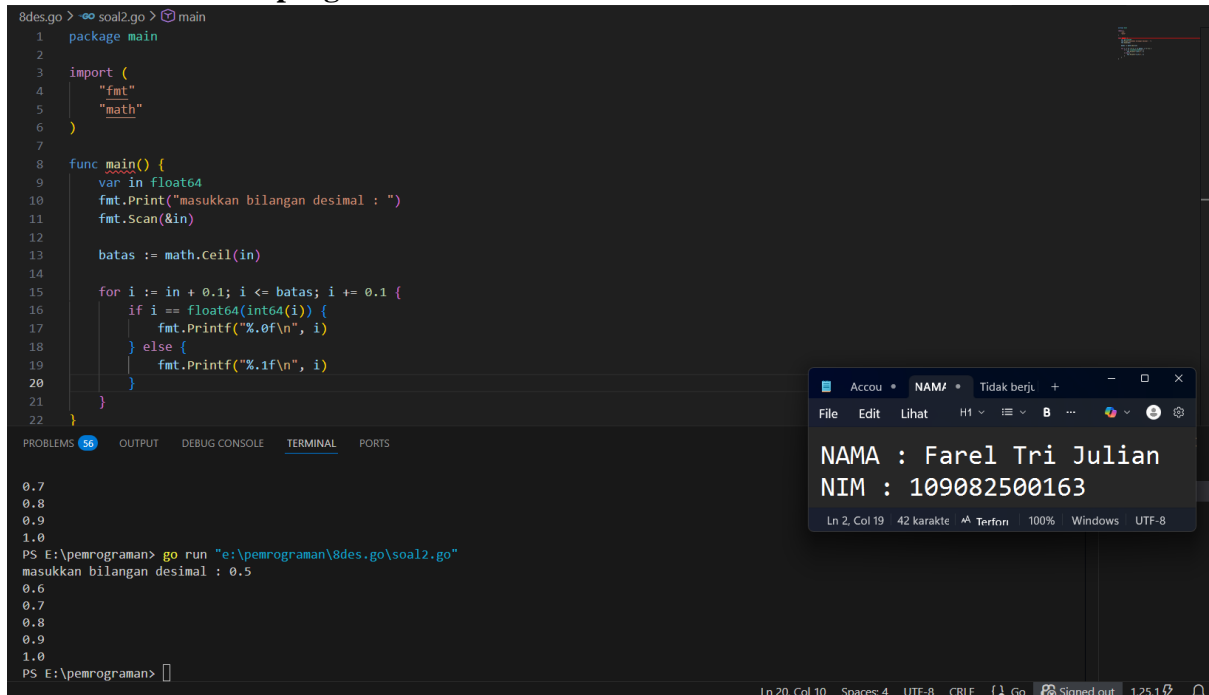
import (
    "fmt"
    "math"
)

func main() {
    var in float64
    fmt.Print("masukkan bilangan decimal : ")
    fmt.Scan(&in)

    batas := math.Ceil(input)

    for i := in + 0.1; i <= batas; i += 0.1 {
        if i == float64(int64(i)) {
            fmt.Printf("%.0f\n", i)
        } else {
            fmt.Printf("%.1f\n", i)
        }
    }
}
```

## Screenshoot program



```
8des.go > soal2.go > main
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var in float64
10    fmt.Print("masukkan bilangan desimal : ")
11    fmt.Scan(&in)
12
13    batas := math.Ceil(in)
14
15    for i := in + 0.1; i <= batas; i += 0.1 {
16        if i == float64(int64(i)) {
17            fmt.Printf("%.0f\n", i)
18        } else {
19            fmt.Printf("%.1f\n", i)
20        }
21    }
22 }
```

0.7  
0.8  
0.9  
1.0

PS E:\pemrograman> go run "e:\pemrograman\8des.go\soal2.go"

masukkan bilangan desimal : 0.5

0.6  
0.7  
0.8  
0.9  
1.0

PS E:\pemrograman>

NAMA : Farel Tri Julian  
NIM : 109082500163

## Deskripsi program

program Golang ini meminta pengguna memasukkan sebuah bilangan desimal, lalu mencetak urutan angka mulai dari nilai tersebut hingga mencapai bilangan bulat terdekat di atasnya. Program menggunakan fungsi `math.Ceil` untuk menentukan batas akhir, kemudian melakukan perulangan dengan kenaikan 0.1 di setiap langkahnya. Di dalam perulangan, terdapat logika kondisional untuk memformat output: jika angka saat itu merupakan bilangan bulat, maka dicetak tanpa desimal, sedangkan jika masih berupa pecahan, dicetak dengan satu angka di belakang koma.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {
    var target, total, donatur int
    fmt.Print("Masukkan target donasi: ")
    fmt.Scan(&target)

    total = 0
    donatur = 0

    for {
        var donasi int
        donatur++

        fmt.Printf("Donatur %d menyumbang: ", donatur)
        fmt.Scan(&donasi)

        total += donasi
        fmt.Printf("Donatur %d: Menyumbang %d. Total  
terkumpul: %d\n", donatur, donasi, total)

        if total >= target {
            break
        }
    }

    fmt.Printf("Target tercapai! Total donasi: %d dari %d  
donatur.\n", total, donatur)
}
```

## Screenshoot program

```
8des.go > cd soal3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var target, total, donatur int
7     fmt.Print("masukkan target donasi: ")
8     fmt.Scan(&target)
9
10    total = 0
11    donatur = 0
12
13    for {
14        var donasi int
15        donatur++
16
17        fmt.Printf("Donatur %d menyumbang: ", donatur)
18        fmt.Scan(&donasi)
19
20        total += donasi
21        fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n", donatur, donasi, total)
22    }
23 }
```

PROBLEMS 56 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\pemrograman> go run "e:\pemrograman\8des.go\soal3.go"
masukkan target donasi: 200
Donatur 1 menyumbang: 150
Donatur 1: Menyumbang 150. Total terkumpul: 150
Donatur 2 menyumbang: 20
Donatur 2: Menyumbang 20. Total terkumpul: 170
Donatur 3 menyumbang: 50
Donatur 3: Menyumbang 50. Total terkumpul: 220
Target tercapai! Total donasi: 220 dari 3 donatur.
PS E:\pemrograman>
```

Accou + NAM/ + Tidak berj. +  
File Edit Lihat H1 100% Windows UTF-8  
NAMA : Farel Tri Julian  
NIM : 109082500163  
Ln 2, Col 19 42 karakter 100% Windows UTF-8

## Deskripsi program

program Golang ini mensimulasikan proses penggalangan dana yang berjalan terus-menerus hingga mencapai target nominal tertentu. Setelah pengguna menetapkan angka target di awal, program memasuki perulangan tak terbatas untuk meminta input nominal sumbangan dari setiap donatur, sekaligus menghitung jumlah donatur dan mengakumulasi total dana yang masuk secara bertahap. Setelah setiap donasi, program menampilkan status terkini, dan perulangan hanya akan dihentikan secara otomatis menggunakan perintah break apabila total dana yang terkumpul sudah sama dengan atau melebihi target, yang kemudian ditutup dengan pesan keberhasilan.