

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 13  
REPEAT-UNTIL**



**Disusun oleh:**

**Andromeda Alika Ramadhani**

**109082500156**

**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, i int

    var stop bool

    fmt.Scan(&kata, &jumlah)

    i = 0

    for stop = false; !stop; {

        fmt.Println(kata)

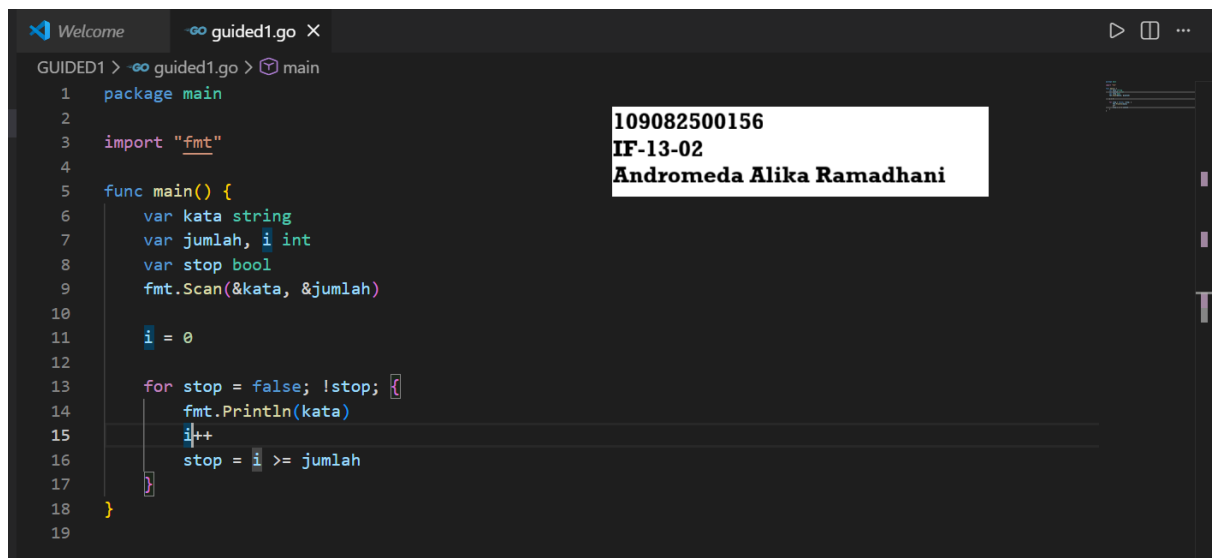
        i++

        stop = i >= jumlah

    }

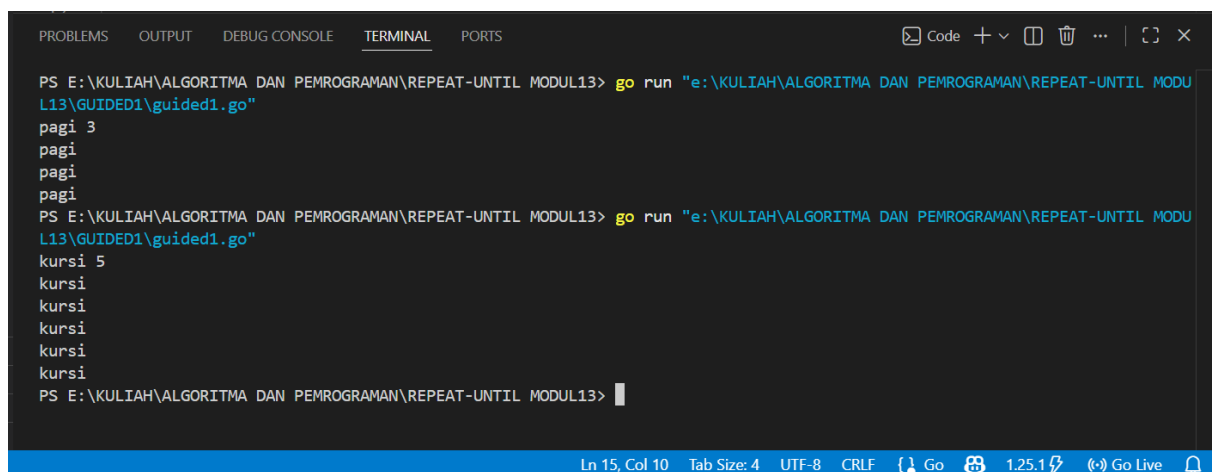
}
```

**Screenshoot program**



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kata string
7     var jumlah, i int
8     var stop bool
9     fmt.Scan(&kata, &jumlah)
10
11     i = 0
12
13     for stop = false; !stop; {
14         fmt.Println(kata)
15         i++
16         stop = i >= jumlah
17     }
18 }
19
```

109082500156  
IF-13-02  
Andromeda Alike Ramadhani



```
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13\GUIDED1\guided1.go"
pagi 3
pagi
pagi
pagi
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13\GUIDED1\guided1.go"
kursi 5
kursi
kursi
kursi
kursi
kursi
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13>
```

## Deskripsi program

Pada program diatas, kita diminta untuk membuat program untuk menampilkan sebuah kata sebanyak jumlah yang diinputkan:

1. Deklarasikan variabel yang akan kita gunakan. Variabel yang kita gunakan adalah variabel kata dengan tipe data string untuk menyimpan input teks yang akan ditampilkan, jumlah dan i dengan tipe data integer untuk menghitung perulangan, dan variabel stop dengan tipe data boolean untuk perulangan.
2. Gunakan "fmt.Scan(&kata, &jumlah)" untuk menginputkan kata dan jumlah.
3. Inisiasi variabel i = 0. Hal ini berarti program akan memulai perulangan dari angka 0.
4. Selanjutnya kita gunakan perulangan for yang akan terus berjalan selama variabel stop bernilai salah. Setiap kali perulangan berjalan, program akan menampilkan kata yang diinputkan menggunakan "fmt.Println". Setelah itu nilai i akan ditambah 1. Kemudian, program akan memeriksa kondisi stop = i >= jumlah dimana jika i sudah sama atau lebih dari jumlah, maka perulangan akan berhenti.

## 2. Guided 2

## Source Code

```
package main

import "fmt"

func main() {

    var bil int

    var stop bool

    bil = 0

    for stop = false; !stop; {

        fmt.Scan(&bil)

        stop = bil > 0

    }

    fmt.Println(bil, "adalah bilangan bulat positif")

}
```

## Screenshoot program



The screenshot shows a terminal window with a dark background. The top bar displays the VS Code interface with tabs for 'Welcome', 'guided1.go', and 'guided2.go'. The terminal prompt is 'GUIDED1 > GUIDED2 > guided2.go > main'. The code being executed is the same Go program as shown in the source code block. The output of the program is displayed in a white box on the right side of the terminal, showing the input '109082500156' and the output '109082500156 IF-13-02 Andromeda Alika Ramadhani'.

```
GUIDED1 > GUIDED2 > guided2.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var bil int
7      var stop bool
8
9      bil = 0
10     for stop = false; !stop; {
11         fmt.Scan(&bil)
12         stop = bil > 0
13     }
14
15     fmt.Println(bil, "adalah bilangan bulat positif")
16
17 }
```

109082500156  
IF-13-02  
Andromeda Alika Ramadhani

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\GUIDED1\GUIDED2\guided2.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\GUIDED1\GUIDED2\guided2.go"
17
17 adalah bilangan bulat positif
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13>
```

### Deskripsi program

Pada program diatas, kita diminta untuk membuat program untuk meminta pengguna menginputkan bilangan sampai pengguna menginputkan bilangan positif:

1. Deklarasikan variabel yang akan kita gunakan. Disini kita menggunakan variabel bil dengan tipe data integer untuk menyimpan input bilangan dan variabel stop dengan tipe data boolean untuk perulangan.
2. Inisiasi variabel bil = 0.
3. Selanjutnya gunakan perulangan for, dimana selama variabel stop masih bernilai salah, maka perulangan akan terus berjalan dan akan terus meminta input dari pengguna menggunakan "fmt.Scan(&bil)". Selanjutnya perulangan akan mengecek kondisi (stop = bil > 0), apakah bilangan yang diinputkan sudah lebih dari 0. Jika benar, maka perulangan akan berhenti.
4. Setelah perulangan selesai, program akan menampilkan pesan bahwa bilangan yang diinputkan adalah bilangan bulat positif.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    var selesai bool

    fmt.Scan(&x, &y)

    for selesai = false; !selesai; {
```

```

        x = x - y

        fmt.Println(x)

        selesai = x <= 0

    }

    fmt.Println(x == 0)

}

```

## Screenshoot program

The screenshot shows a code editor with three tabs: 'Welcome', 'guided1.go', and 'guided2.go'. The active tab is 'guided3.go', which contains the following Go code:

```

1  package main
2
3  import "fmt"
4
5  func main() {
6      var x, y int
7      var selesai bool
8      fmt.Scan(&x, &y)
9
10     for selesai = false; !selesai; {
11         x = x - y
12         fmt.Println(x)
13         selesai = x <= 0
14     }
15     fmt.Println(x == 0)
16 }
17

```

Overlaid on the right side of the code editor is a white box containing the following text:

```

109082500156
IF-13-02
Andromeda Alika Ramadhani

```

The screenshot shows a terminal window with the following output:

```

PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\GUIDED1\GUIDED2\GUIDED3\guided3.go"
5 2
3
1
-1
false
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\GUIDED1\GUIDED2\GUIDED3\guided3.go"
15 3
12
9
6
3
0
true
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\GUIDED1\GUIDED2\GUIDED3\guided3.go"
25 5
20
15
10
5
0
true
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13>

```

The terminal window has a status bar at the bottom showing 'Ln 17, Col 1', 'Tab Size: 4', 'UTF-8', 'CRLF', and icons for 'Go', '1.25.1', 'Go Live', and a bell icon.

### Deskripsi program

Pada program diatas, kita diminta untuk membuat program untuk mengecek apakah suatu bilangan merupakan kelipatan dari bilangan lainnya:

1. Deklarasikan variabel yang akan kita gunakan. Disini kita menggunakan variabel x dan y dengan tipe data integer untuk menyimpan input dua bilangan dan variabel selesai dengan tipe data boolean untuk perulangan.
2. Gunakan "fmt.Scan(&x, &y)" untuk menginputkan dua buah bilangan yang akan kita hitung.
3. Selanjutnya kita gunakan perulangan for dimana dimana selama variabel selesai masih bernilai salah, maka perulangan akan terus berjalan. Selama perulangan masih berjalan, nilai x akan dikurangi dengan nilai y. Selanjutnya program akan menampilkan hasil pengurangan tersebut menggunakan "fmt.Println(x)". Setelah itu program akan mengecek kondisi (selesai = x <=), apakah nilai x sudah kurang dari atau sama dengan x. Jika sudah maka perulangan akan berhenti.
4. Setelah itu program akan menampilkan nilai true/false, (apakah bilangan tersebut merupakan kelipatan dari bilangan lainnya atau bukan. Jika ya maka akan bernilai true dan jika tidak maka akan bernilai false.

### TUGAS

#### 1. Tugas 1

##### Source code

```
package main

import "fmt"

func main() {

    var bilangan, hitung, temp int

    fmt.Scan(&bilangan)

    hitung = 0

    temp = bilangan

    for {

        temp = temp / 10
```

```

        hitung++

        if temp == 0 {

            break

        }

    }

    fmt.Println(hitung)

}

```

### Screenshoot program

```

SOAL1 > go soal1.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var bilangan, hitung, temp int
7      fmt.Scan(&bilangan)
8
9      hitung = 0
10     temp = bilangan
11
12     for {
13         temp = temp / 10
14         hitung++
15         if temp == 0 {
16             break
17         }
18     }
19     fmt.Println(hitung)
20 }
21

```

109082500156  
IF-13-02  
Andromeda Alika Ramadhani

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\SOAL1\soal1.go"
5
1
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\SOAL1\soal1.go"
234
3
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\SOAL1\soal1.go"
78787
5
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\SOAL1\soal1.go"
1894256
7
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13>

```

Ln 19, Col 23 Tab Size: 4 UTF-8 CRLF 1.25.1 Go 1.25.1 Go Live

### Deskripsi program

Pada program diatas, kita diminta untuk membuat program untuk menghitung banyaknya digit dari suatu bilangan:



1. Deklarasikan variabel yang akan kita gunakan. Disini kita menggunakan variabel bilangan dengan tipe data integer untuk menyimpan input bilangan, variabel hitung dengan tipe data integer untuk menghitung jumlah digit, dan variabel temp untuk menyimpan proses perhitungan.
2. Gunakan “fmt.Scan(&bilangan)” untuk menginputkan bilangan.
3. Inisiasikan variabel hitung = 0 (perhitungan dimulai dari 0) dan variabel temp = bilangan (menyalin nilai input ke variabel sementara untuk proses pembagian tanpa mengubah nilai bilangan).
4. Selanjutnya, gunakan perulangan for. Di dalam perulangan for, nilai temp akan dibagi dengan 10 untuk menghilangkan satu digit terakhir, kemudian variabel hitung akan terus bertambah. Perulangan akan berhenti sampai nilai temp = 0. Hal ini berarti bahwa seluruh digit pada bilangan telah habis dibagi.
5. Setelah perulangan selesai, program akan menampilkan total jumlah dari suatu bilangan yang diinputkan menggunakan “fmt.Println(hitung)”

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {

    var desimal float64

    var batas, hitung int

    fmt.Scan(&desimal)

    hitung = int(desimal * 10)

    batas = (hitung / 10) * 10

    if hitung%10 != 0 {

        batas += 10

    }
```

```

        hitung += 1

        for {

            fmt.Printf("%.1f\n", float64(hitung)/10)

            if hitung >= batas {

                break

            }

            hitung++

        }

    }
}

```

## Screenshoot program

```

SOAL1 > SOAL2 > -go soal2.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var desimal float64
7      var batas, hitung int
8      fmt.Scan(&desimal)
9
10     hitung = int(desimal * 10)
11     batas = (hitung / 10) * 10
12     if hitung%10 != 0 {
13         batas += 10
14     }
15
16     hitung += 1
17     for {
18         fmt.Printf("%.1f\n", float64(hitung)/10)
19         if hitung >= batas {
20             break
21         }
22         hitung++
23     }
24 }
25

```

109082500156  
IF-13-02  
Andromeda Alike Ramadhani

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\SOAL1\SOAL2\soal2.go"
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\SOAL1\SOAL2\soal2.go"
2.7
2.8
2.9
3.0
Ln 16, Col 16 Tab Size: 4 UTF-8 CRLF Go 1.25.1 Go Live
```

### Deskripsi program

Pada program diatas, kita diminta untuk membuat program untuk mendapatkan bilangan bulat optimal dari bilangan yang diinputkan:

1. Deklarasikan variabel yang akan kita gunakan. Disini kita menggunakan variabel desimal dengan tipe data float untuk menyimpan input bilangan desimal, variabel batas dan hitung dengan tipe data integer untuk proses perhitungan.
2. Gunakan "fmt.Scan(&desimal)" untuk menginputkan bilangan desimal yang akan dihitung.
3. Selanjutnya nilai desimal diubah menjadi bilangan bulat dan dikalikan dengan 10 untuk memudahkan proses perhitungan angka di belakang koma. Kemudian variabel batas digunakan untuk melakukan pembulatan ke atas.
4. Selanjutnya gunakan logika if, dimana jika angka dibelakang koma tidak sama dengan 0, maka batas += 10 supaya mencapai kelipatannya berikutnya.
5. Setelah itu, nilai hitung akan ditambah 1, kemudian program akan melakukan perulangan for untuk menampilkan bilangan desimal secara berurutan. Pada setiap perulangan, nilai akan dibagi 10 dan menampilkan satu angka di belakang koma. Perulangan akan berhenti jika nilai hitung sudah mencapai lebih dari sama dengan nilai batas.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {
    var target, donasi, total, jmlDonatur int
    fmt.Scan(&target)

    for {
```

```

        fmt.Scan(&donasi)
        jmlDonatur++
        total += donasi

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

        if total >= target {
            break
        }
    }

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

```

### Screenshoot program

The screenshot shows a Go IDE with the following code in the editor:

```

1  package main
2
3  import "fmt"
4
5  func main() {
6      var target, donasi, total, jmlDonatur int
7      fmt.Scan(&target)
8
9      for {
10         fmt.Scan(&donasi)
11         jmlDonatur++
12         total += donasi
13
14         fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n", jmlDonatur, donasi, total)
15
16         if total >= target {
17             break
18         }
19     }
20     fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n", total, jmlDonatur)
21 }
22

```

On the right side of the IDE, a white box contains the following text:

109082500156  
IF-13-02  
Andromeda Alike Ramadhani

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\SOAL1\SOAL2\SOAL3\soal3.go"
300
100
Donatur 1: Menyumbang 100. Total terkumpul: 100
50
Donatur 2: Menyumbang 50. Total terkumpul: 150
200
Donatur 3: Menyumbang 200. Total terkumpul: 350
Target tercapai! Total donasi: 350 dari 3 donatur.
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\SOAL1\SOAL2\SOAL3\soal3.go"
500
150
Donatur 1: Menyumbang 150. Total terkumpul: 150
100
Donatur 2: Menyumbang 100. Total terkumpul: 250
50
Donatur 3: Menyumbang 50. Total terkumpul: 300
300
Donatur 4: Menyumbang 300. Total terkumpul: 600
Target tercapai! Total donasi: 600 dari 4 donatur.
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13> go run "e:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODU
L13\SOAL1\SOAL2\SOAL3\soal3.go"
200
300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
PS E:\KULIAH\ALGORITMA DAN PEMROGRAMAN\REPEAT-UNTIL MODUL13>
```

## Deskripsi program

Pada program diatas, kita diminta untuk membuat program untuk menghitung dan menampilkan total donasi hingga mencapai target yang ditentukan:

1. Deklarasikan variabel yang akan kita gunakan. Variabel yang kita gunakan adalah variabel target untuk menyimpan input target donasi, variabel donasi untuk menyimpan jumlah sumbangan dari donatur, variabel total untuk menyimpan total semua donasi yang berhasil dikumpulkan, dan variabel jmlDonatur untuk menghitung jumlah donatur.
2. Gunakan "fmt.Scan(&target)" untuk menginputkan nilai target donasi.
3. Selanjutnya gunakan perulangan for. Di dalam perulangan ini program akan terus meminta input donasi, jumlah donatur akan selalu bertambah 1 dan nilai donasi akan dijumlahkan ke dalam total. Program kemudian akan menampilkan informasi donatur ke berapa, jumlah sumbangan yang diberikan dan total semua sumbangan yang sudah diberikan.
4. Jika total sudah mencapai target atau lebih dari target, maka perulangan akan berhenti dan menampilkan informasi bahwa target telah tercapai, serta menampilkan total donasi yang terkumpul dan jumlah donatur yang berpartisipasi.