

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 13
REPEAT-UNTIL**



Disusun oleh:

NAYAKA FARROS RIANDRA

109082500055

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

1. Guided 1 Source Code

```
package main

import "fmt"

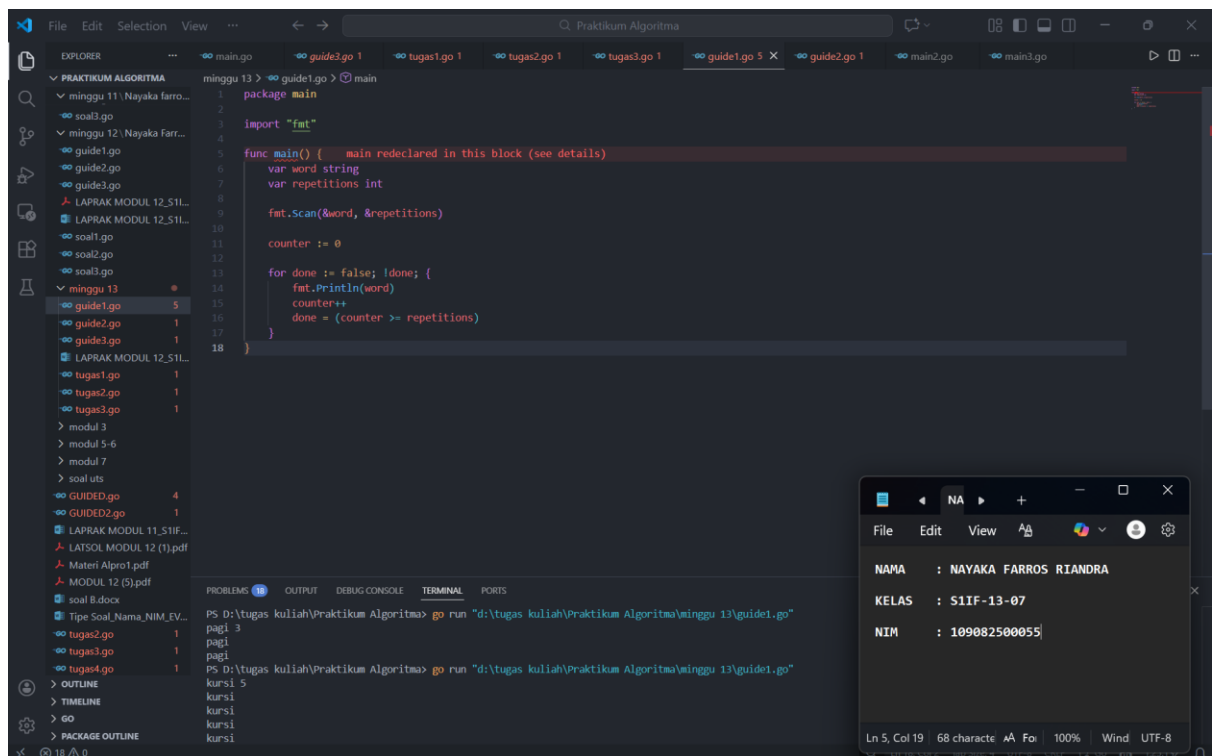
func main() {
    var word string
    var repetitions int

    fmt.Scan(&word, &repetitions)

    counter := 0

    for done := false; !done; {
        fmt.Println(word)
        counter++
        done = (counter >= repetitions)
    }
}
```

Screenshoot program



Deskripsi program

Program ini merupakan aplikasi sederhana yang ditulis menggunakan bahasa pemrograman Go untuk mencetak sebuah kata secara berulang-ulang sesuai dengan jumlah yang diinginkan oleh pengguna. Prosesnya dimulai dengan mendeklarasikan variabel bertipe string untuk menyimpan kata dan variabel integer untuk jumlah repetisi, kemudian program meminta input dari pengguna melalui terminal.

Setelah mendapatkan input, program mengeksekusi struktur perulangan for yang dikendalikan oleh logika boolean dan sebuah variabel penghitung (counter) yang dimulai dari nol. Di dalam setiap putaran perulangan tersebut, program menampilkan kata ke layar lalu menambahkan nilai penghitung satu per satu. Proses ini akan terus berjalan dan baru berhenti secara otomatis ketika nilai penghitung sudah mencapai jumlah repetisi yang ditentukan sebelumnya.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var number int

    var continueLoop bool

    for continueLoop = true; continueLoop; {

        fmt.Scan(&number)

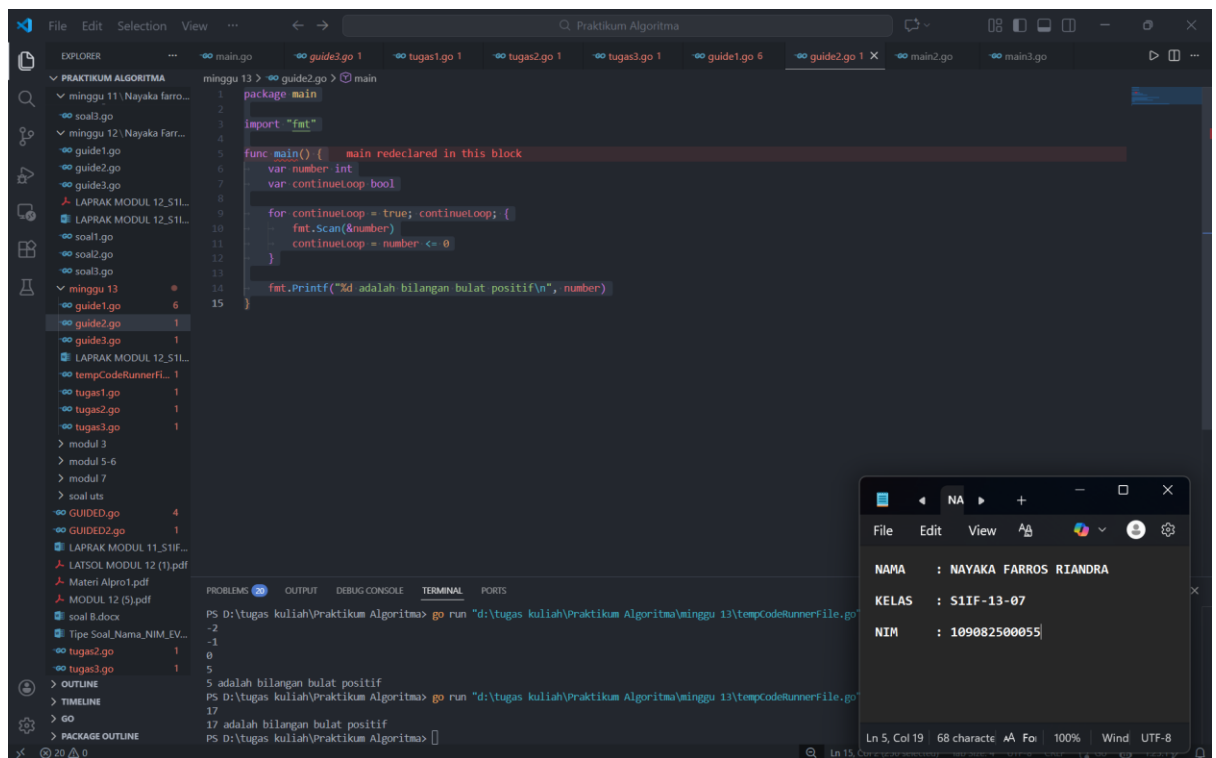
        continueLoop = number <= 0

    }

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

}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor. The file explorer on the left shows a project named 'PRAKTIKUM ALGORITMA' with various files. The main editor displays the code for 'main.go' in the 'main' package. The code defines a 'main' function that uses a loop to prompt the user for input. The loop continues as long as the input is not a positive integer. Once a positive integer is entered, the program prints a confirmation message.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     // main redeclared in this block
7     var number int
8     var continueLoop bool
9
10    for continueLoop = true; continueLoop; {
11        fmt.Scan(&number)
12        continueLoop = number <= 0
13    }
14    fmt.Printf("%d adalah bilangan bulat positif\n", number)
15 }
```

The terminal output window in the bottom right corner shows the following information:

```
File Edit View
NAMA : NAYAKA FARROS RIANDRA
KELAS : S1IF-13-07
NIM : 109082500055
```

Deskripsi program

Program ini dirancang untuk memvalidasi input pengguna agar hanya menerima bilangan bulat positif menggunakan mekanisme perulangan yang dikontrol oleh variabel boolean. Pada awalnya, program mendeklarasikan variabel integer untuk menampung angka dan variabel boolean sebagai kondisi penentu berjalannya loop. Selanjutnya, program masuk ke dalam struktur perulangan yang akan terus meminta input dari pengguna dan mengevaluasi apakah angka tersebut bernilai nol atau negatif. Selama pengguna memasukkan angka nol atau bilangan negatif, status variabel kondisi akan tetap bernilai benar sehingga program terus mengulang permintaan input. Perulangan baru akan berhenti secara otomatis ketika pengguna memasukkan angka positif, yang kemudian diikuti dengan pencetakan pesan konfirmasi bahwa angka tersebut adalah bilangan bulat positif yang valid.

3. Guided 3

Source Code

```
package main

import "fmt"

func main(){

    var a, b int

    fmt.Scan(&a)

    fmt.Scan(&b)


    for kondisi := false; !kondisi;{

        a=a-b

        fmt.Println(a)


        if a<=0{

            kondisi= a == 0

            fmt.Println(kondisi)

            kondisi=true

        }

    }

}
```

Screenshoot program

The screenshot shows a Go program in a VS Code editor. The program is located in `minggu 13 > guide3.go` and contains the following code:

```
1 package main
2
3 import "fmt"
4
5 func main(){ main redeclared in this block
6     var a, b int
7     fmt.Scan(&a)
8     fmt.Scan(&b)
9
10    for kondisi := false; !kondisi;{
11        a=a-b
12        fmt.Println(a)
13
14        if a<=0{
15            kondisi= a == 0
16            fmt.Println(kondisi)
17
18            kondisi=true
19        }
20    }
21 }
22 }
```

The terminal output shows the execution of the program:

```
PS D:\tugas kuliah\Praktikum Algoritma> go run "d:\tugas kuliah\Praktikum Algoritma\minggu 13\guide3.go"
5
2
3
1
-1
false
PS D:\tugas kuliah\Praktikum Algoritma> go run "d:\tugas kuliah\Praktikum Algoritma\minggu 13\guide3.go"
15
3
```

A small window titled "NA" is open in the bottom right corner, displaying student information:

NAMA	: NAYAKA FARROS RIANDRA
KELAS	: S1IF-13-07
NIM	: 109082500055

Deskripsi program

Algoritma pada kode ini berfokus pada manipulasi aritmatika dua bilangan bulat melalui mekanisme pengurangan bertingkat. Variabel pertama yang diinputkan pengguna akan terus-menerus dikikis nilainya sebesar variabel kedua di dalam sebuah blok perulangan, kemudian sisa hasil setiap operasi tersebut langsung ditampilkan secara berurutan. Siklus operasi ini diatur untuk terus berjalan selama hasilnya masih positif, dan sistem baru akan memutus mata rantai perulangan secara otomatis tepat saat mendeteksi bahwa nilai akhir telah menyentuh angka nol atau menjadi negatif melalui pemeriksaan kondisi boolean di dalam blok if.

TUGAS

1. Tugas 1

Source Code

```
package main

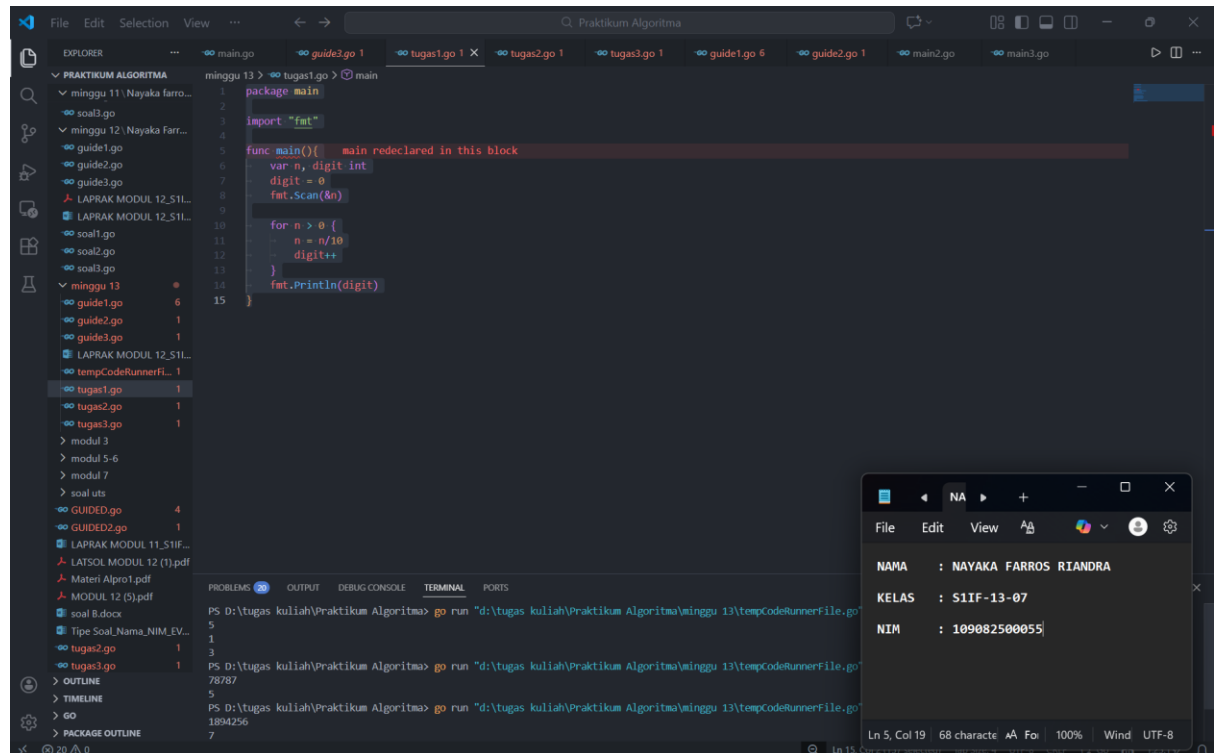
import "fmt"

func main(){
    var n, digit int
    digit = 0
    fmt.Scan(&n)

    for n > 0 {
        n = n/10
        digit++
    }

    fmt.Println(digit)
}
```


Screenshoot program



Deskripsi program

Program ini berfungsi untuk menghitung jumlah digit yang terdapat dalam sebuah bilangan bulat positif yang diinputkan oleh pengguna. Setelah mendeklarasikan variabel untuk menampung angka input dan variabel penghitung yang dimulai dari nol, program menjalankan struktur perulangan yang terus beroperasi selama nilai angka tersebut lebih besar dari nol. Di dalam setiap iterasi perulangan, angka input dibagi dengan sepuluh untuk menghilangkan digit terakhirnya secara bertahap, dan secara bersamaan variabel penghitung dinaikkan nilainya satu per satu. Proses ini akan berhenti ketika angka tersebut habis atau bernilai nol, lalu program mencetak total jumlah digit yang telah dihitung ke layar terminal.

2. Tugas 2

Source code

```
package main

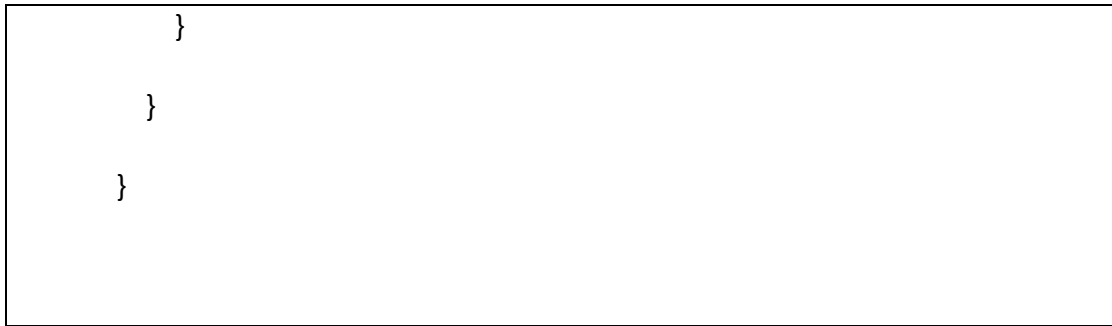
import (
    "fmt"
    "math"
)

func main(){
    var n float64
    fmt.Scan(&n)

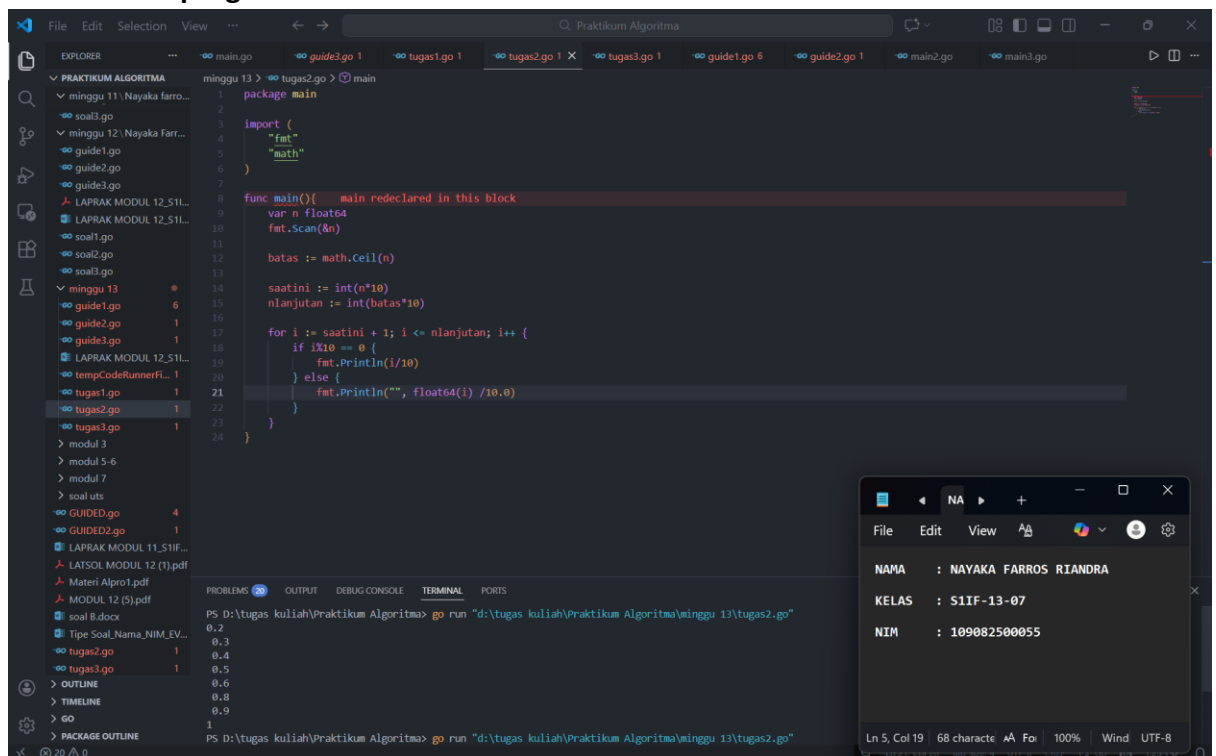
    batas := math.Ceil(n)

    saatini := int(n*10)
    nlanjutan := int(batas*10)

    for i := saatini + 1; i <= nlanjutan; i++ {
        if i%10 == 0 {
            fmt.Println(i/10)
        } else {
            fmt.Println("%.1f\n", float64(i)/10.0)
        }
    }
}
```



Screenshoot program



Deskripsi program

Program ini dirancang untuk menampilkan urutan bilangan desimal dengan kenaikan sebesar nol koma satu, dimulai dari angka setelah input pengguna hingga mencapai bilangan bulat pembulatan ke atas terdekat. Setelah menerima input bertipe data *floating-point*, program menghitung batas akhir perulangan menggunakan fungsi matematika *ceiling*. Uniknya, untuk menjaga presisi perhitungan dan menghindari *floating-point error*, logika program mengonversi semua nilai menjadi bilangan bulat dengan mengalikannya dengan sepuluh sebelum masuk ke blok perulangan. Loop kemudian berjalan menelusuri rentang angka tersebut, lalu membagi kembali nilainya dengan sepuluh saat menampilkan *output* ke layar, sembari melakukan pengecekan kondisi untuk menentukan apakah angka tersebut harus dicetak sebagai bilangan bulat murni atau tetap dalam format desimal.

3. Tugas 3

Source code

```
package main

import "fmt"

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

    total = 0
    donatur = 0

    fmt.Println("masukkan jumlah uang untuk didonasikan : ")

    for total < target {
        var donasi int
        fmt.Scan(&donasi)
        donatur+=1
        total += donasi

        fmt.Printf("Donatur %d: menyumbang %d. total terkumpul: %d\n",
donatur, donasi, total)
    }

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

Screenshoot program

The screenshot shows a Go IDE with a project named 'PRAKTIKUM ALGORITMA'. The file explorer on the left shows a directory structure with files like 'minggu 11', 'minggu 12', 'LAPRAK MODUL 12_S11L', 'soal1.go', 'soal2.go', 'soal3.go', 'minggu 13', 'guide1.go', 'guide2.go', 'guide3.go', 'LAPRAK MODUL 12_S11L', 'tempCodeRunnerFi...', 'tugas1.go', 'tugas2.go', 'tugas3.go', 'modul 3', 'csmdu3.go', 'guided3.go', 'modul 5 6', 'modul 7', 'soal uts', 'GUIDED.go', 'GUIDED2.go', 'LAPRAK MODUL 11_S11F...', 'LATSOI MODUL 12 (1).pdf', 'Materi Alpro1.pdf', 'MODUL 12 (5).pdf', 'soal 8.docx', 'Tipe Soal_Nama_NIM_EV...', 'OUTLINE', 'TIMELINE', 'GO', and 'PACKAGE OUTLINE'. The main editor shows the code for 'tugas3.go' with the following content:

```
func main() {  
    // main redeclared in this block  
    var target, total, donatur int  
    fmt.Println("masukkan target donasi : ")  
    fmt.Scan(&target)  
  
    total = 0  
    donatur = 0  
  
    fmt.Println("masukkan jumlah uang untuk didonasikan : ")  
  
    for total < target {  
        var donasi int  
        fmt.Scan(&donasi)  
        donatur++  
        total += donasi  
  
        fmt.Printf("Donatur %d: menyumbang %d. total terkumpul: %d\n", donatur, donasi, total)  
    }  
  
    fmt.Printf("target tercapai total donasi: %d dari %d donatur. \n", total, donatur)  
}
```

The terminal output shows the execution of the program:

```
PS D:\tugas kuliah\Praktikum Algoritma> go run "d:\tugas kuliah\Praktikum Algoritma\minggu 13\tugas3.go"  
masukkan target donasi :  
300  
masukkan jumlah uang untuk didonasikan :  
100  
Donatur 1: menyumbang 100. total terkumpul: 100  
50  
Donatur 2: menyumbang 50. total terkumpul: 150  
200  
Donatur 3: menyumbang 200. total terkumpul: 350
```

Deskripsi program

Program ini dirancang sebagai sistem pencatat donasi sederhana yang akan terus menerima masukan dana hingga target nominal tertentu terpenuhi. Pada tahap awal, pengguna diminta untuk menetapkan angka target donasi yang ingin dicapai, sementara variabel untuk total dana dan jumlah donatur diinisialisasi ke angka nol. Selanjutnya, program memasuki struktur perulangan yang akan terus berjalan secara otomatis selama total dana yang terkumpul masih berada di bawah target yang ditetapkan. Di setiap putaran donasi, program meminta input nominal sumbangan, menambahkan jumlah donatur, mengakumulasi dana ke dalam total saldo, serta menampilkan laporan progres terkini ke layar. Sistem baru akan berhenti meminta input ketika total donasi sudah menyamai atau melampaui target, yang kemudian ditutup dengan menampilkan ringkasan akhir berisi total dana yang berhasil dihimpun beserta jumlah donatur yang berpartisipasi.