

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 13
REPEAT-UNTIL**



Disusun oleh:
MUHAMMAD FIRDAUS ARDIANSYAH
109082500126
S1IF-13-07

Asisten Praktikum
Adithana dharma putra
Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

LATIHAN KELAS – GUIDED

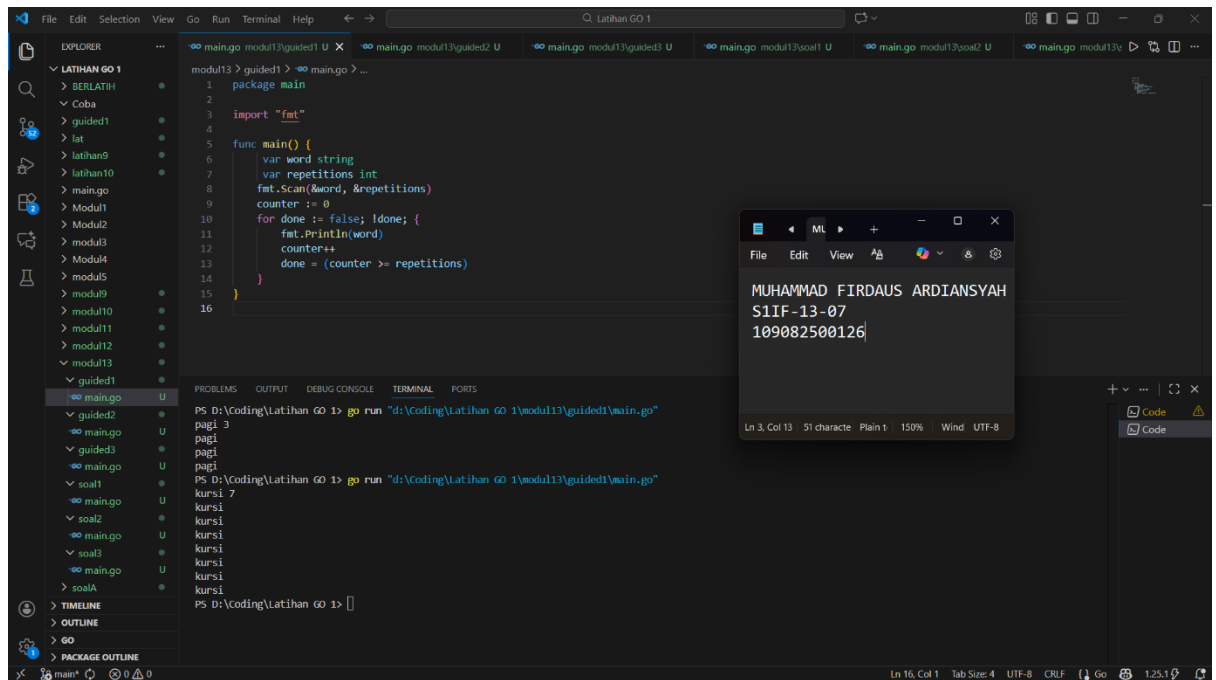
1. Guided 1 Source Code

```
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 langsung memproses dua input dari user, yaitu sebuah kata (string) dan bilangan bulat (integer), menggunakan fungsi `fmt.Scan`. Kode kemudian mengeksekusi perintah untuk mencetak kata tersebut secara berulang ke bawah sebanyak angka yang dimasukkan. Hal ini terlihat jelas pada output terminal, di mana input "pagi 3" menghasilkan cetakan kata "pagi" sebanyak tiga baris, dan "kursi 7" menghasilkan tujuh baris.

Mekanisme perulangan yang digunakan mengandalkan logika variabel boolean `done` sebagai penanda henti dan variabel `counter` sebagai penghitung. Dalam setiap iterasi loop, program mencetak kata dan menaikkan nilai `counter`, lalu mengevaluasi apakah jumlahnya sudah mencapai target. Jika kondisi `counter >= repetitions` terpenuhi, variabel `done` berubah menjadi `true`, yang secara otomatis menghentikan proses perulangan.

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

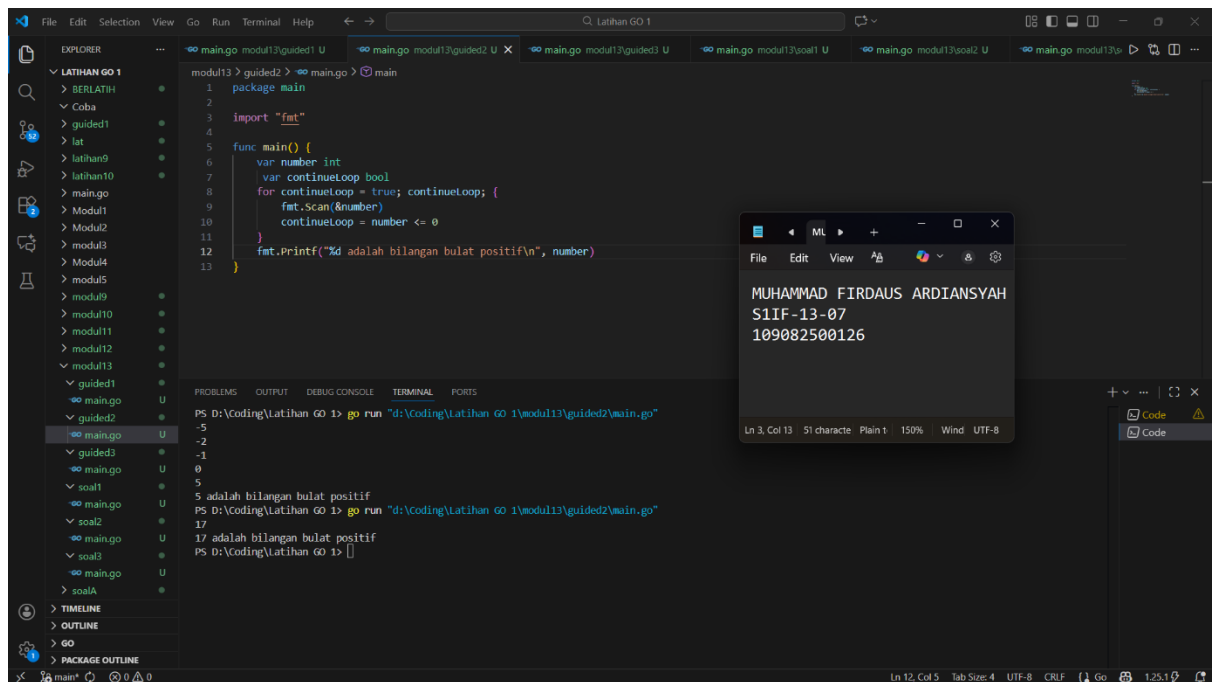
    }

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

}

```

Screenshoot program



Deskripsi program

Program ini dirancang untuk memvalidasi input dari user dengan terus meminta masukan bilangan bulat (integer) hingga nilai positif diberikan. Menggunakan fungsi `fmt.Scan` di dalam blok perulangan, kode terus membaca angka yang diketikkan. Seperti terlihat pada terminal, input angka non-positif seperti -5, -2, atau 0 akan membuat program terus berjalan, namun input positif seperti 5 atau 17 akan menghentikan loop dan mencetak pesan konfirmasi bahwa angka tersebut adalah bilangan bulat positif.

Mekanisme kontrol alur program ini menggunakan `for` loop yang dikendalikan oleh variabel boolean `continueLoop`. Logika utamanya terletak pada baris `continueLoop = number <= 0`, yang secara dinamis memperbarui status perulangan setelah setiap input. Selama user memasukkan angka kurang dari atau sama dengan nol, variabel kondisi tetap bernilai `true` sehingga input terus diminta berulang kali, dan loop baru berhenti otomatis saat kondisi bernilai `false` (angka positif).

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var x int
    var y int
    var selesai bool

    fmt.Scan(&x, &y)

    for selesai = false; !selesai; {
        x = x - y

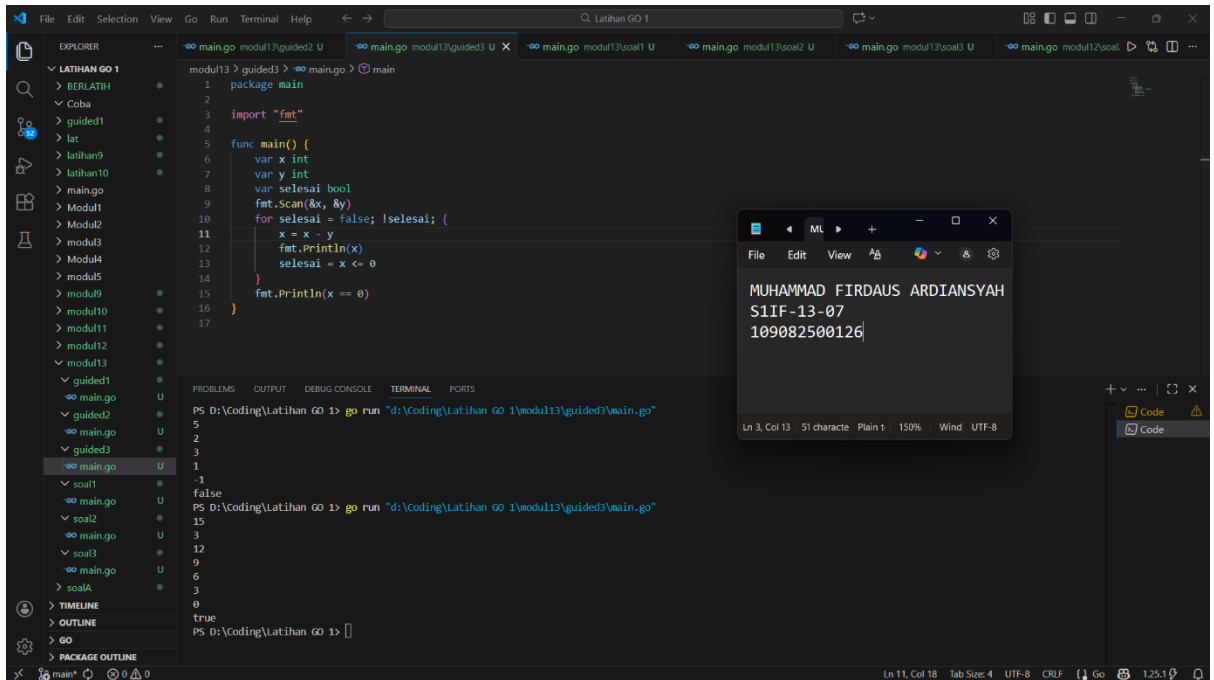
        fmt.Println(x)

        selesai = x <= 0
    }

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



Screenshoot program



Deskripsi program

Program ini menerima dua input bilangan bulat dari user (variabel x dan y) dan melakukan operasi pengurangan berulang terhadap nilai x dengan y. Di dalam blok perulangan, nilai x terus dikurangi dan hasilnya dicetak baris demi baris hingga nilainya menjadi nol atau negatif. Hal ini terlihat pada terminal di mana input "5 2" menghasilkan sisa -1, sedangkan "15 3" berakhir tepat di angka 0.

Secara teknis, program menggunakan loop yang bergantung pada variabel boolean selesai sebagai kondisi berhenti. Loop terus berjalan selama !selesai, dan variabel selesai baru akan bernilai true ketika $x \leq 0$. Setelah loop berakhir, baris terakhir kode mengevaluasi dan mencetak apakah $x == 0$, yang memberikan output true jika bilangan pertama habis dibagi bilangan kedua, atau false jika terdapat sisa pengurangan (negatif).

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var n, digit int

    digit = 0

    fmt.Print("Masukkan n: ")

    fmt.Scan(&n)

    for n > 0 {

        n = n / 10

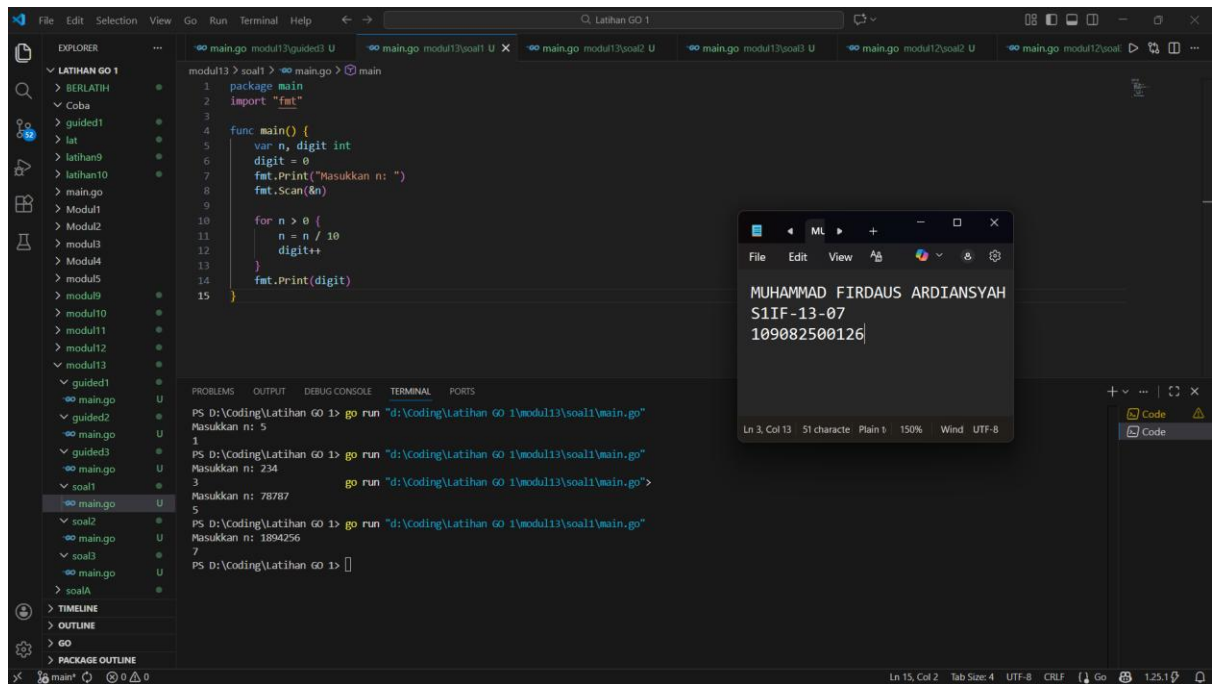
        digit++

    }

    fmt.Print(digit)

}
```

Screenshoot program



Deskripsi program

Program ini bertujuan untuk menghitung jumlah digit yang terdapat dalam sebuah bilangan bulat yang diinputkan oleh user. Setelah membaca input angka menggunakan `fmt.Scan`, program langsung memprosesnya dan menampilkan total digit dari bilangan tersebut. Seperti yang terlihat pada terminal, input "5" terhitung sebagai 1 digit, "234" menjadi 3 digit, hingga angka "1894256" yang teridentifikasi memiliki 7 digit.

Mekanisme penghitungan dilakukan menggunakan `for` loop yang terus berjalan selama nilai `n` lebih besar dari 0. Di dalam blok perulangan, program melakukan operasi pembagian integer (`n = n / 10`) untuk memangkas digit terakhir angka secara bertahap, sambil menaikkan nilai penghitung digit sebesar satu di setiap putaran. Proses ini berhenti otomatis ketika seluruh angka telah dibagi habis (menjadi 0), menyisakan variabel `digit` sebagai hasil akhir jumlah digit.

2. Tugas 2

Source code

```
package main

import (

    "fmt"
```



```
"math"

)

func main() {

    var n float64

    fmt.Print("Masukkan n: ")

    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 {

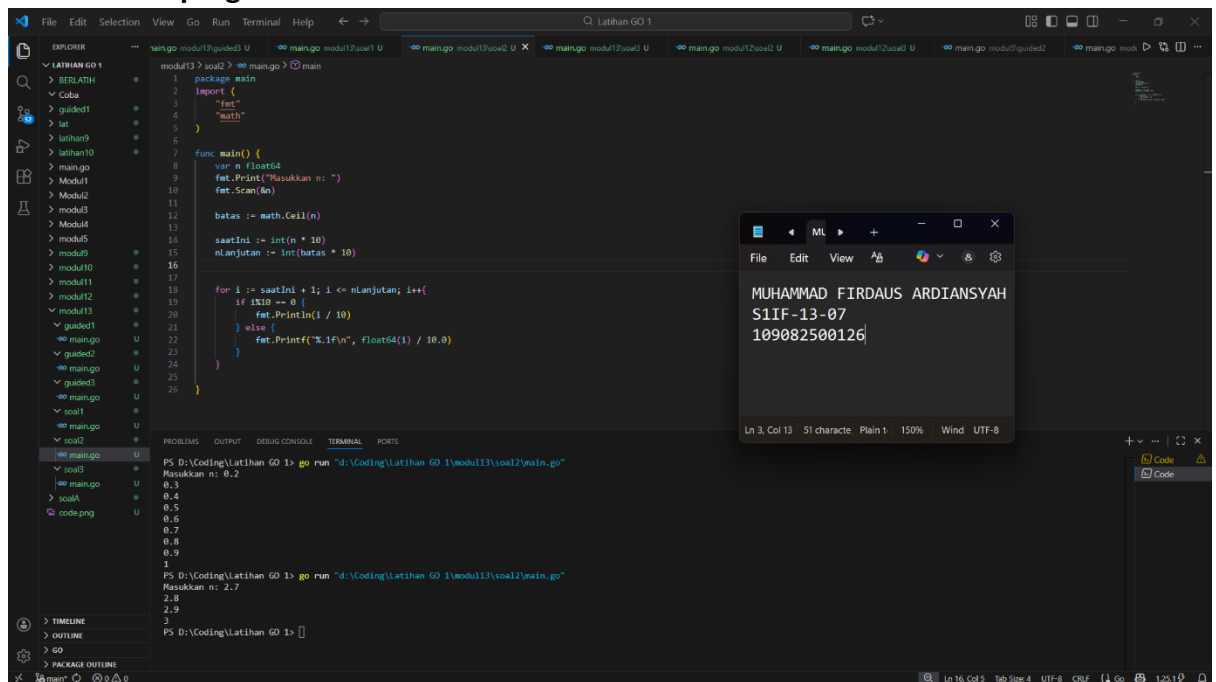
            fmt.Printf("%.1f\n", float64(i) / 10.0)

        }

    }

}
```

Screenshoot program



Deskripsi program

Program ini meminta input berupa bilangan desimal (float) dari user, kemudian menampilkan urutan angka dengan kenaikan 0.1 mulai dari nilai setelah input hingga mencapai bilangan bulat terdekat di atasnya (ceiling). Seperti terlihat pada terminal, input "0.2" menghasilkan deret angka dari 0.3 hingga 1, sedangkan input "2.7" mencetak angka 2.8, 2.9, dan berakhir di 3.

Secara teknis, program menggunakan pendekatan aritmatika integer untuk menangani perulangan desimal guna menjaga presisi. Input awal dikalikan 10 dan dibulatkan ke atas menggunakan `math.Ceil` untuk menentukan batas akhir iterasi (`nLanjutan`). Loop `for` kemudian berjalan menelusuri nilai-nilai tersebut, di mana setiap angka dibagi kembali dengan 10 saat dicetak. Terdapat logika `if-else` di dalam loop yang memformat output agar bilangan bulat ditampilkan tanpa koma (seperti "1"), sementara angka pecahan ditampilkan dengan satu desimal (seperti "0.3").

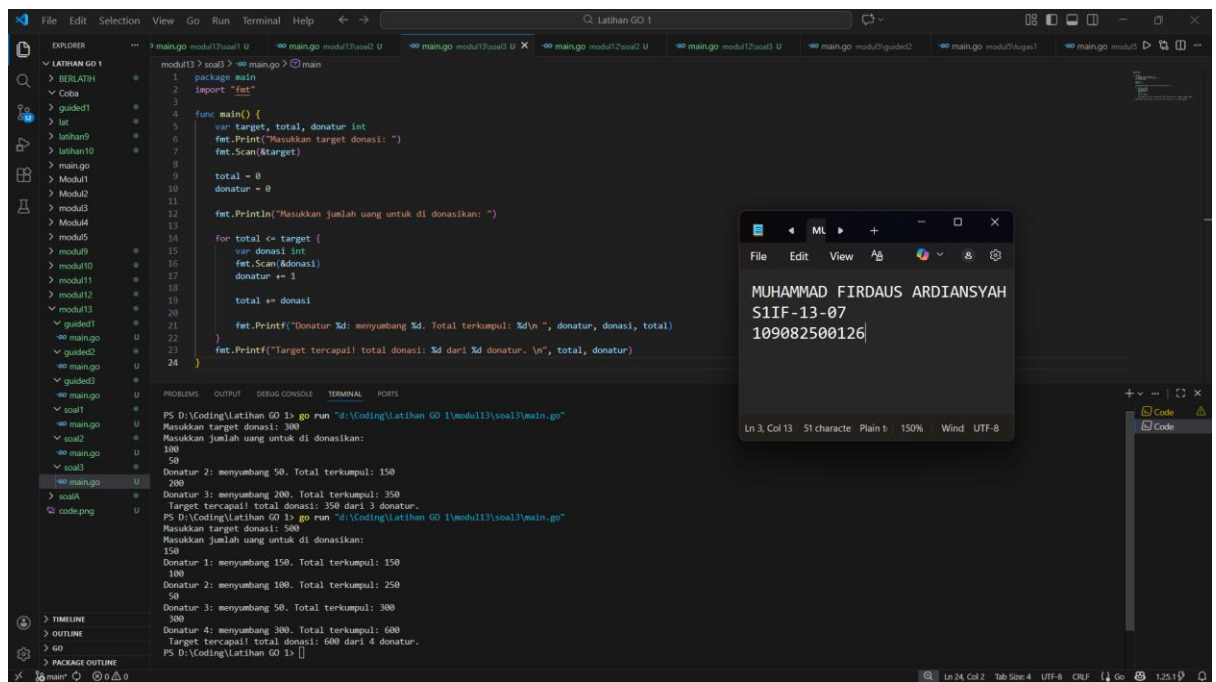
3. Tugas 3

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  
  
    fmt.Println("Masukkan jumlah uang untuk di  
donasikan: ")  
  
    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



Deskripsi program

Program ini mensimulasikan sistem penggalangan dana sederhana yang meminta user untuk menetapkan target nominal donasi terlebih dahulu. Setelah target ditentukan, program secara berulang meminta input jumlah uang dari para donatur dan menjumlahkannya secara *real-time*. Pada contoh di terminal, saat target diatur ke 300, program terus menerima input donasi (100, 50, dan 200) hingga total dana terkumpul mencapai 350, yang secara otomatis menghentikan proses penginputan karena target telah terlampaui.

Secara teknis, mekanisme program ini mengandalkan for loop dengan kondisi $\text{total} \leq \text{target}$. Selama total donasi yang terkumpul masih di bawah atau sama dengan angka target, program akan terus meminta input tambahan, menghitung jumlah donatur ($\text{donatur}++$), dan mengakumulasi total dana ($\text{total} += \text{donasi}$). Loop ini baru akan berhenti ketika kondisi menjadi false, yaitu saat variabel total sudah bernilai lebih besar dari target, yang kemudian menampilkan rekapitulasi akhir jumlah donasi dan total donatur.