

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 13
REPEAT-UNTIL**



Disusun oleh:

FARID HERDIYANTO VITASANDI

109082500123

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 banyak_kata int

    fmt.Print("Masukkan kata: ")

    fmt.Scan(&kata)

    fmt.Print("Masukkan banyak perulangan: ")

    fmt.Scan(&banyak_kata)


    i := 0

    for{

        fmt.Println(kata)

        i++

        if i == banyak_kata{

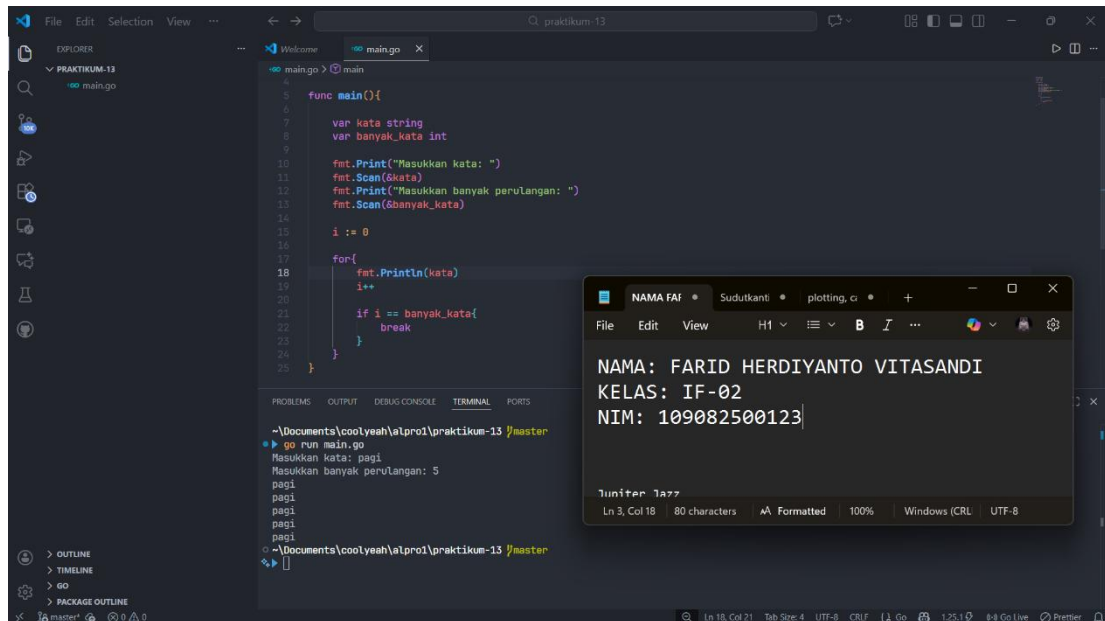
            break

        }

    }

}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor and its execution output in a terminal window. The program is located in `main.go` and is part of a project named `praktikum-13`.

```
func main(){
    var kata string
    var banyak_kata int

    fmt.Println("Masukkan kata: ")
    fmt.Scan(&kata)
    fmt.Println("Masukkan banyak pengulangan: ")
    fmt.Scan(&banyak_kata)

    i := 0
    for{
        fmt.Println(kata)
        i++

        if i == banyak_kata{
            break
        }
    }
}
```

The terminal window shows the output of the program:

```
~\Documents\coolyeah\alpro1\praktikum-13 /master
$ go run main.go
Masukkan kata: pagi
Masukkan banyak pengulangan: 5
pagi
pagi
pagi
pagi
pagi
```

Deskripsi program

Program di atas meminta User untuk memasukkan sebuah kata dan jumlah pengulangan yang diinginkan. Variabel `i` digunakan sebagai penghitung jumlah pengulangan yang telah terjadi, dimulai dari nilai 0. Pada setiap iterasi, program mencetak kata yang diberikan lalu menambah nilai `i` satu per satu. Setelah nilai `i` mencapai jumlah pengulangan yang ditentukan pengguna, program menghentikan pengulangan dengan perintah `break`, sehingga proses pencetakan tidak berlanjut melebihi batas yang ditetapkan.

Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var bilangan int

    var continueLoop bool

    for continueLoop = true; continueLoop; {

        fmt.Print("Masukkan bilangan bulat positif: ")

        fmt.Scan(&bilangan)

        continueLoop = bilangan <= 0

        if bilangan <= 0 {

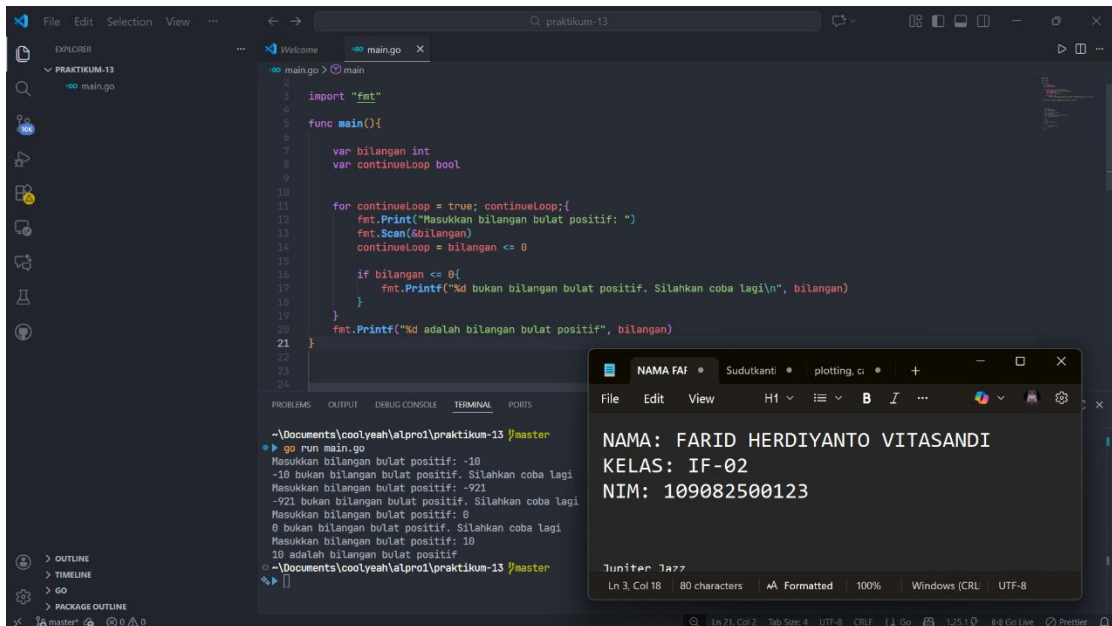
            fmt.Printf("%d bukan bilangan bulat positif.\n", bilangan)

            continue
        }

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

    }
}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor and its execution output in the terminal. The program is located in `main.go` and uses a `for` loop with a `continueLoop` boolean variable to repeatedly prompt the user for a positive integer until a valid input is received.

```
1  package main
2
3  import "fmt"
4
5  func main() {
6
7      var bilangan int
8      var continueLoop bool
9
10
11     for continueLoop = true; continueLoop; {
12         fmt.Print("Masukkan bilangan bulat positif: ")
13         fmt.Scan(&bilangan)
14         continueLoop = bilangan <= 0
15
16         if bilangan <= 0 {
17             fmt.Printf("%d bukan bilangan bulat positif. Silahkan coba lagi\n", bilangan)
18         }
19     }
20     fmt.Printf("%d adalah bilangan bulat positif", bilangan)
21 }
22
23
24
```

The terminal output shows the program's execution with the following steps:

```
~/Documents/coolyeah/latpro1/praktikum-13 /master
$ go run main.go
Masukkan bilangan bulat positif: -10
-10 bukan bilangan bulat positif. Silahkan coba lagi
Masukkan bilangan bulat positif: -921
-921 bukan bilangan bulat positif. Silahkan coba lagi
Masukkan bilangan bulat positif: 0
0 bukan bilangan bulat positif. Silahkan coba lagi
Masukkan bilangan bulat positif: 10
10 adalah bilangan bulat positif
~/Documents/coolyeah/latpro1/praktikum-13 /master
```

Deskripsi program

Program di atas adalah program yang dibuat untuk mengecek apakah suatu bilangan itu bilangan bulat positif atau bukan. Program menggunakan variabel `continueLoop` sebagai pengendali perulangan untuk memastikan bahwa User memasukkan bilangan bulat positif. Pada awal perulangan, nilai `continueLoop` diinisialisasi sebagai `true`, sehingga loop akan langsung berjalan. Setiap iterasi, program akan meminta input berupa bilangan dan kemudian memperbarui nilai `continueLoop` berdasarkan kondisi `bilangan <= 0`. Jika bilangan yang dimasukkan kurang dari atau sama dengan nol, program menampilkan pesan bahwa input tidak valid dan perulangan akan berlanjut. Namun, jika bilangan lebih besar dari nol, nilai `continueLoop` menjadi `false`, sehingga loop berhenti. Setelah perulangan selesai, program menampilkan bahwa bilangan tersebut merupakan bilangan bulat positif.

2. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    var isKelipatan bool

    fmt.Print("Masukkan bilangan (x): ")

    fmt.Scan(&x)

    fmt.Print("Masukkan bilangan (y): ")

    fmt.Scan(&y)

    for isKelipatan = false; !isKelipatan; {

        x -= y

        fmt.Println(x)

        if x == 0 {

            isKelipatan = true

            break

        }

        if x < 0 {

            isKelipatan = false

            break

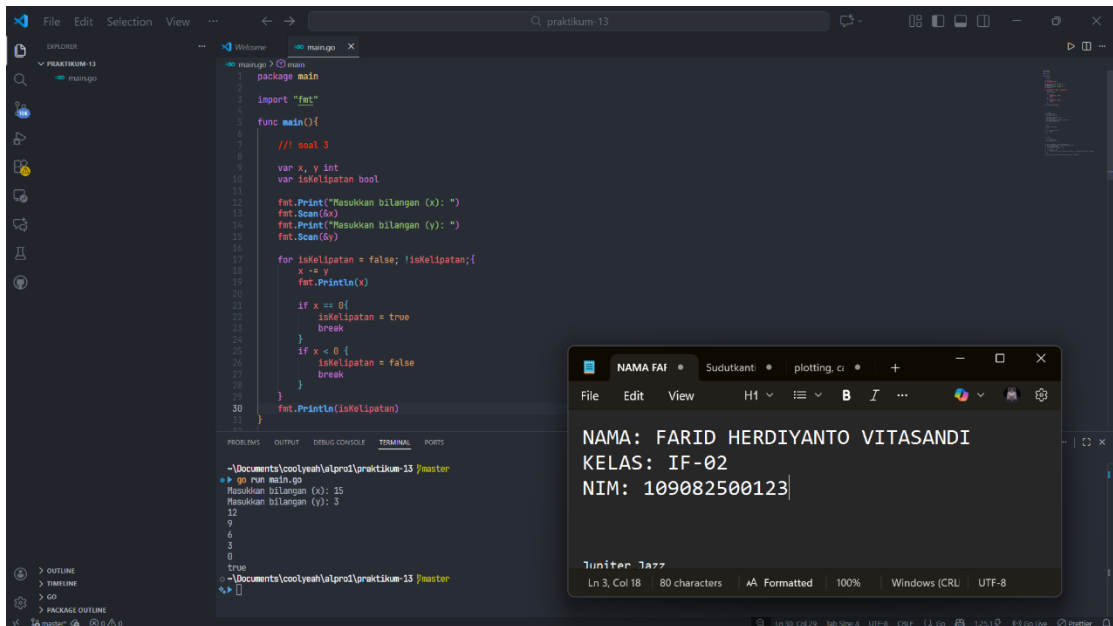
        }

    }

}
```

```
    }  
  
    }  
  
    fmt.Println(isKelipatan)  
}
```

Screenshoot program



Deskripsi program

Program di atas akan membaca dua bilangan, yaitu x dan y , kemudian menggunakan perulangan untuk menentukan apakah x merupakan kelipatan dari y dengan cara mengurangi nilai x secara berulang. Variabel `isKelipatan` digunakan sebagai penanda hasil pengecekan. Selama nilai `isKelipatan` masih *false*, program mengurangi x dengan y dan menampilkan nilai x setelah pengurangan. Jika hasil pengurangan membuat x bernilai 0, program menetapkan `isKelipatan` menjadi *true* karena hal tersebut menunjukkan bahwa x merupakan kelipatan dari y , lalu menghentikan perulangan. Namun, jika nilai x menjadi negatif, program menghentikan perulangan karena hal itu menandakan bahwa x tidak dapat dikurangi lagi secara tepat oleh y , sehingga tidak merupakan kelipatan. Setelah perulangan selesai, program menampilkan nilai akhir dari `isKelipatan` sebagai hasil pengecekan.

TUGAS

1. Soal 1

Source Code

```
package main

import "fmt"

func main() {

    var bilangan int

    fmt.Println("Masukkan bilangan bulat positif: ")

    fmt.Scan(&bilangan)

    count := 0

    for{

        count++

        bilangan = bilangan / 10

        if bilangan == 0{

            break

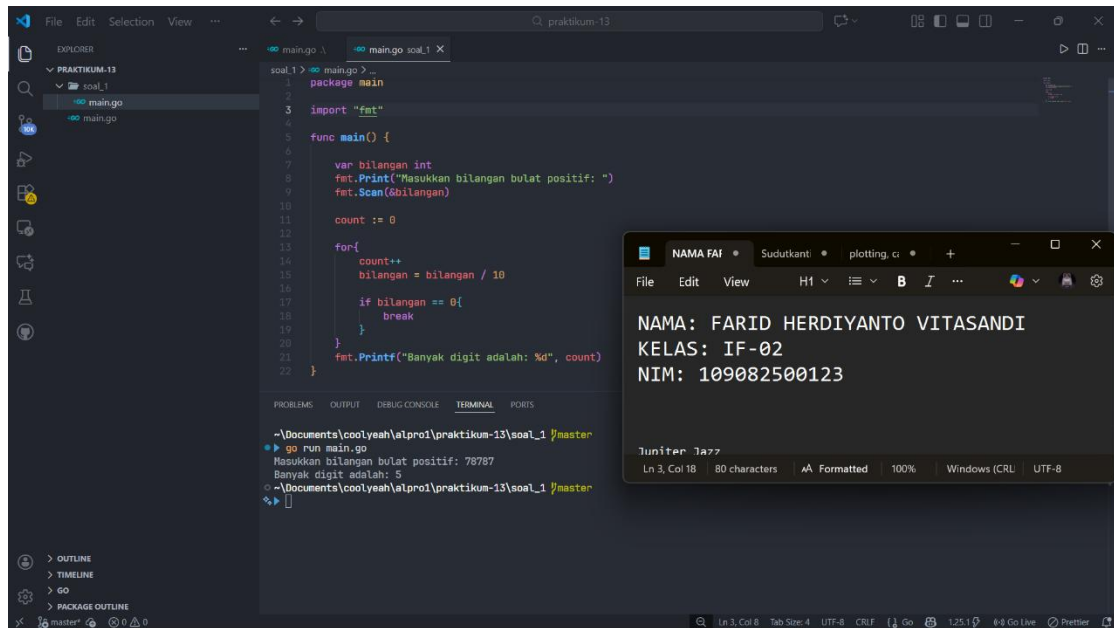
        }

    }

    fmt.Printf("Banyak digit adalah: %d", count)

}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor and its execution output in a terminal window. The program is located in `soal1.go` and uses the `fmt` package. It prompts the user to enter a positive integer, then enters a loop that divides the number by 10 and increments a counter until the number becomes 0. The terminal output shows the user inputting 78787 and the program outputting 5, which is the number of digits in the input.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     var bilangan int
8     fmt.Print("Masukkan bilangan bulat positif: ")
9     fmt.Scan(&bilangan)
10
11     count := 0
12
13     for{
14         count++
15         bilangan = bilangan / 10
16
17         if bilangan == 0{
18             break
19         }
20     }
21     fmt.Printf("Banyak digit adalah: %d", count)
22 }
```

Terminal Output:

```
~\Documents\coolyeah\alpro1\praktikum-13\soal_1\nmaster
go run main.go
Masukkan bilangan bulat positif: 78787
Banyak digit adalah: 5
~\Documents\coolyeah\alpro1\praktikum-13\soal_1\nmaster
```

Deskripsi program

Program di atas adalah program yang digunakan untuk menghitung jumlah digit dari bilangan bulat positif yang dimasukkan oleh User. Setelah membaca input, program akan menginisialisasi variabel `count` sebagai penghitung digit. Pada setiap iterasi, nilai `count` ditambah satu, kemudian bilangan dibagi 10 untuk menghapus digit terakhirnya. Perulangan terus berlangsung hingga nilai bilangan menjadi 0, yang menandakan bahwa seluruh digit telah diproses. Setelah perulangan berhenti, program menampilkan jumlah digit yang telah dihitung melalui variabel `count`.

2. Soal 2

Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {

    var bilangan_desimal float64

    fmt.Print("Masukkan bilangan desimal: ")
    fmt.Scan(&bilangan_desimal)

    sum := bilangan_desimal
    target := math.Ceil(bilangan_desimal)

    for{

        sum += 0.1

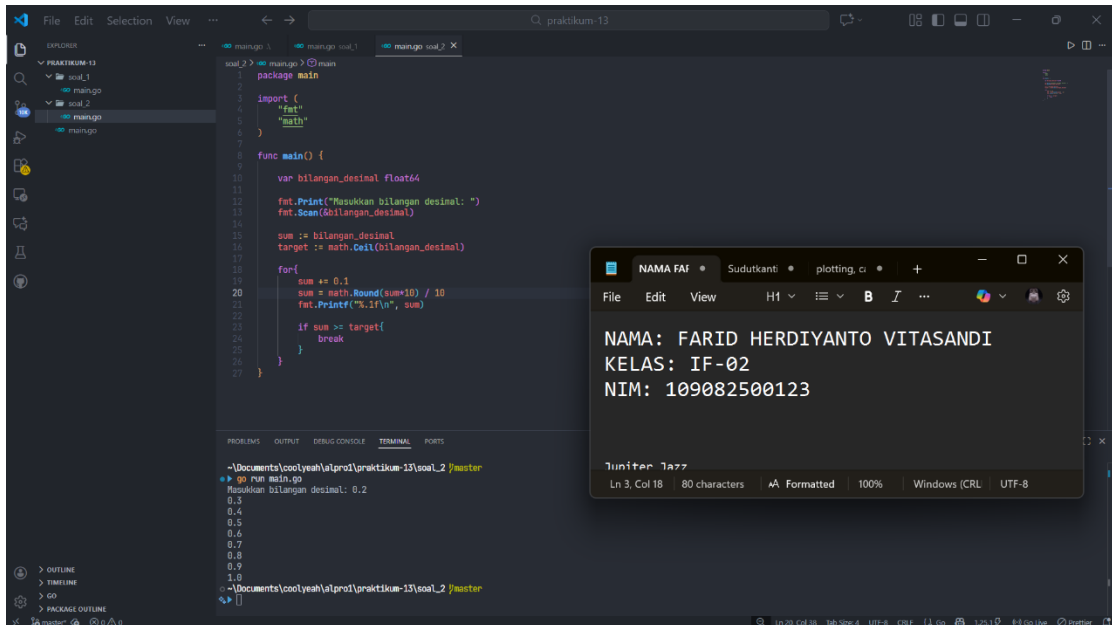
        sum = math.Round(sum*10) / 10

        fmt.Printf("%.1f\n", sum)

        if sum >= target{
```

```
        break
    }
}
}
```

Screenshoot program



```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9
10     var bilangan_desimal float64
11     fmt.Println("Masukkan bilangan desimal: ")
12     fmt.Scan(&bilangan_desimal)
13
14     sum := bilangan_desimal
15     target := math.Ceil(bilangan_desimal)
16
17     for{
18         sum += 0.1
19         sum = math.Round(sum*10) / 10
20         fmt.Printf("%.1f\n", sum)
21
22         if sum >= target{
23             break
24         }
25     }
26 }
```

```
~\Documents\cool\yah\alpro1\praktikum-13\soal_2 /master
> go run main.go
Masukkan bilangan desimal: 0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
~\Documents\cool\yah\alpro1\praktikum-13\soal_2 /master
```

NAMA: FARID HERDIYANTO VITASANDI
KELAS: IF-02
NIM: 109082500123

7unitar 1azp
Ln 3, Col 18 80 characters AA Formatted 100% Windows (CRLF) UTF-8

Deskripsi program

Program di atas akan membaca sebuah bilangan desimal dari yang dimasukkan oleh User, kemudian program akan melakukan proses penjumlahan bertahap hingga mencapai nilai pembulatan atas dari bilangan tersebut. Nilai awal variabel sum diisi dengan bilangan desimal yang dimasukkan, sedangkan variabel target berisi hasil pembulatan ke atas menggunakan `math.Ceil`. Pada saat iterasi, program akan menambahkan 0.1 pada sum, lalu membulatkannya ke satu angka di belakang koma menggunakan operasi `math.Round(sum*10) / 10` agar hasil tetap presisi. Setiap nilai yang dihasilkan ditampilkan ke layar. Perulangan akan berhenti ketika nilai sum telah mencapai atau melampaui nilai target.

3. Soal 3

Source Code

```
package main

import "fmt"

func main(){

    var target_donasi int

    fmt.Print("Masukkan target donasi: ")

    fmt.Scan(&target_donasi)

    total_donasi := 0

    donatur := 0

    for{

        var donasi int

        donatur++

        fmt.Printf("Donatur ke-%d: Menyumbang ", donatur)

        fmt.Scan(&donasi)

        total_donasi += donasi

        if total_donasi >= target_donasi{

            break
        }
    }
}
```

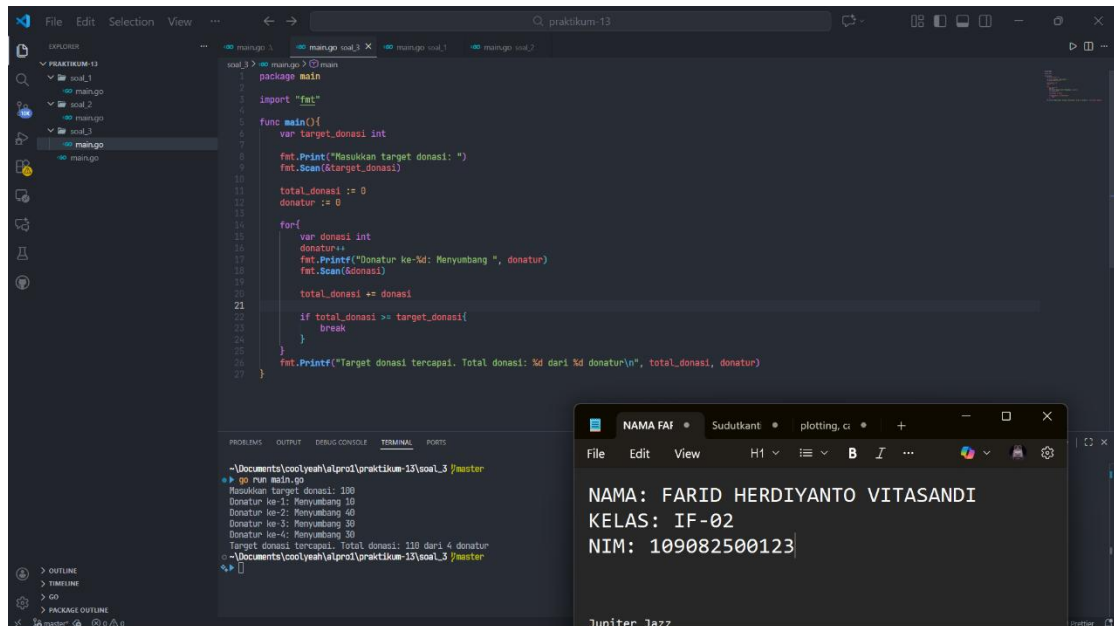
```
        }

    }

    fmt.Printf("Target donasi tercapai. Total donasi: %d dari
%d donatur\n", total_donasi, donatur)

}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor and its execution output in a terminal window.

Go Program Code:

```
1 package main
2 import "fmt"
3
4 func main(){
5     var target_donasi int
6
7     fmt.Print("Masukkan target donasi: ")
8     fmt.Scan(&target_donasi)
9
10    total_donasi := 0
11    donatur := 0
12
13    for{
14        var donasi int
15        donatur++
16        fmt.Printf("Donatur ke-%d: Menyumbang ", donatur)
17        fmt.Scan(&donasi)
18
19        total_donasi += donasi
20
21        if total_donasi >= target_donasi{
22            break
23        }
24    }
25    fmt.Printf("Target donasi tercapai. Total donasi: %d dari %d donatur\n", total_donasi, donatur)
26 }
```

Terminal Output:

```
~/Documents/coolayah/alpro1/praktikum-13/soal_3 /master
$ go run main.go
Masukkan target donasi: 100
Donatur ke-1: Menyumbang 10
Donatur ke-2: Menyumbang 40
Donatur ke-3: Menyumbang 30
Donatur ke-4: Menyumbang 30
Target donasi tercapai. Total donasi: 110 dari 4 donatur
```

Deskripsi program

Program di atas akan meminta User untuk memasukkan target donasi, lalu program akan menjalankan iterasi untuk mengumpulkan donasi dari setiap donatur hingga target terpenuhi. Variabel `total_donasi` digunakan untuk menyimpan akumulasi jumlah donasi, sementara `donatur` menghitung jumlah orang yang telah menyumbang. Pada setiap iterasi, program menambah hitungan donatur, meminta input jumlah donasi dari donatur tersebut, lalu menambahkan nilainya ke `total_donasi`. Perulangan terus berjalan hingga total donasi mencapai atau melampaui target yang telah ditentukan. Setelah target terpenuhi, program menghentikan perulangan dan menampilkan total donasi yang terkumpul serta jumlah donatur yang berpartisipasi.