

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 13  
REPEAT UNTIL**



**Disusun oleh:**

**REZKY FARREL**

**109082500203**

**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

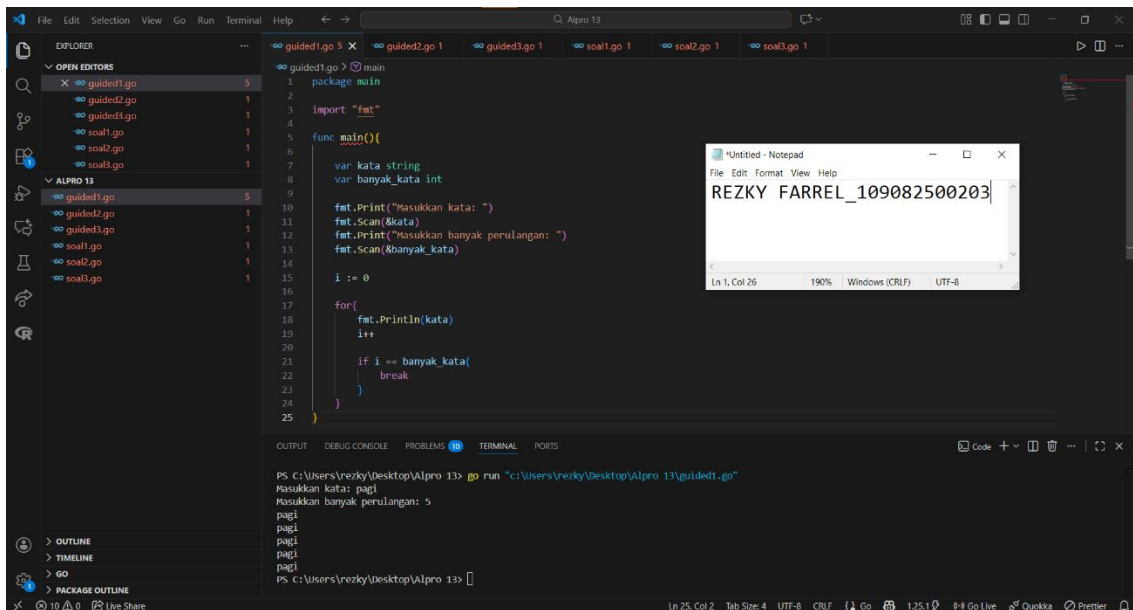
    fmt.Print("Masukkan kata: ")
    fmt.Scan(&kata)
    fmt.Print("Masukkan banyak perulangan: ")
    fmt.Scan(&banyak_kata)

    i := 0

    for{
        fmt.Println(kata)
        i++

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

Screenshoot program :



### Deskripsi program :

Program ini berfungsi untuk mencetak sebuah kata sebanyak jumlah pengulangan yang diminta pengguna. Program dimulai dengan mendeklarasikan dua variabel, yaitu kata untuk menyimpan input berupa teks, dan banyak\_kata untuk menyimpan jumlah pengulangan. Setelah itu, program meminta pengguna memasukkan sebuah kata dan angka yang menunjukkan berapa kali kata tersebut harus dicetak..

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main(){

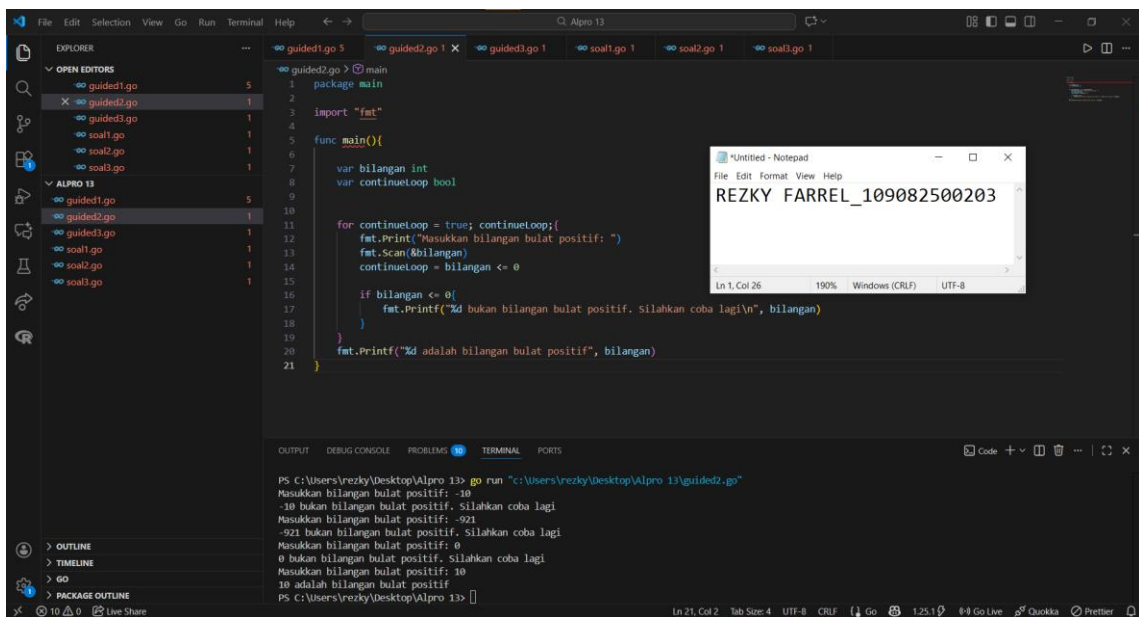
    var kata string
    var banyak_kata int

    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
        }
    }
}
```



```
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 }
```

OUTPUT

```
PS C:\Users\rezky\Desktop\Alpro 13> go run "c:\Users\rezky\Desktop\Alpro 13\guided2.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
PS C:\Users\rezky\Desktop\Alpro 13>
```

### Deskripsi program :

Program bertujuan untuk memvalidasi masukan pengguna agar hanya menerima bilangan bulat positif. Pada awalnya, program mendeklarasikan variabel integer untuk menyimpan angka dan variabel boolean bernama continueLoop untuk mengontrol jalannya perulangan. Di dalam blok perulangan for, program meminta input dari pengguna dan kemudian mengevaluasi angka tersebut; jika angka yang dimasukkan bernilai nol atau negatif, variabel continueLoop akan diset menjadi true dan pesan kesalahan akan ditampilkan, memaksa perulangan untuk kembali meminta input. Sebaliknya, jika pengguna memasukkan angka positif, continueLoop akan berubah menjadi false yang otomatis menghentikan perulangan,

lalu program akan mencetak pesan konfirmasi bahwa angka tersebut adalah bilangan bulat positif yang valid.

### Guided 3

#### Source Code

```
package main

import "fmt"

func main(){

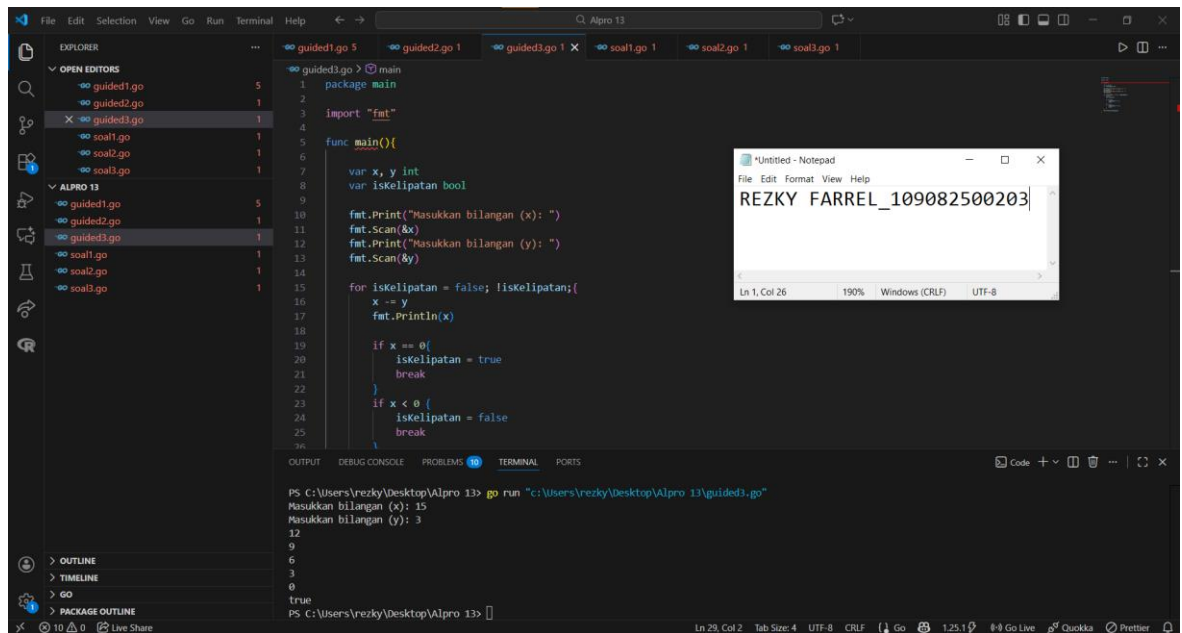
    var x, y int
    var isKelipatan bool

    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 ini bertujuan untuk menentukan apakah bilangan x merupakan kelipatan dari bilangan y dengan menggunakan metode pengurangan berulang, bukan operator modulus. Program dimulai dengan meminta pengguna memasukkan dua bilangan bulat, x dan y, serta menginisialisasi status isKelipatan sebagai false. Di dalam struktur perulangan, nilai x terus-menerus dikurangi dengan y pada setiap iterasi, dan sisa pengurangan tersebut ditampilkan ke layar. Perulangan ini akan terus berjalan hingga salah satu dari dua kondisi berhenti terpenuhi: jika nilai x mencapai tepat angka nol, status isKelipatan diubah menjadi true (menandakan x habis dibagi y) dan loop dihentikan; sebaliknya, jika nilai x menjadi negatif, berarti x tidak habis dibagi y, dan loop berhenti dengan status tetap false.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import (
    "fmt"
)

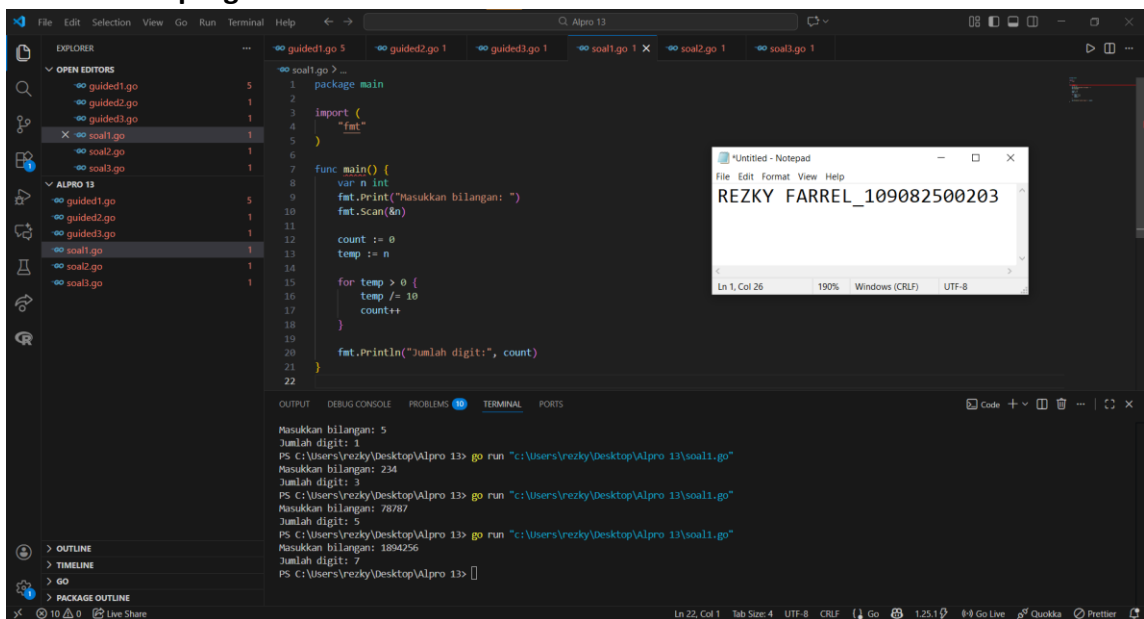
func main() {
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)

    count := 0
    temp := n

    for temp > 0 {
        temp /= 10
        count++
    }

    fmt.Println("Jumlah digit:", count)
}
```

#### Screenshoot program :



#### Deskripsi program :

Program ini dirancang untuk menghitung banyaknya digit dari sebuah bilangan bulat positif yang dimasukkan oleh pengguna. Proses dimulai dengan meminta input bilangan yang disimpan dalam variabel `n`, lalu menyalin nilainya ke variabel sementara bernama `temp` agar nilai asli tidak hilang, serta menyiapkan variabel `count` bernilai nol sebagai penghitung. Logika intinya menggunakan perulangan `for` yang berjalan selama nilai `temp` masih lebih besar dari nol. Di dalam setiap iterasi perulangan tersebut, nilai `temp` dibagi dengan 10 (pembagian integer) yang secara efektif membuang digit terakhir angka tersebut, dan secara bersamaan nilai `count` ditambahkan satu.

## Tugas 2

### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {

    var bilangan_desimal float64

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

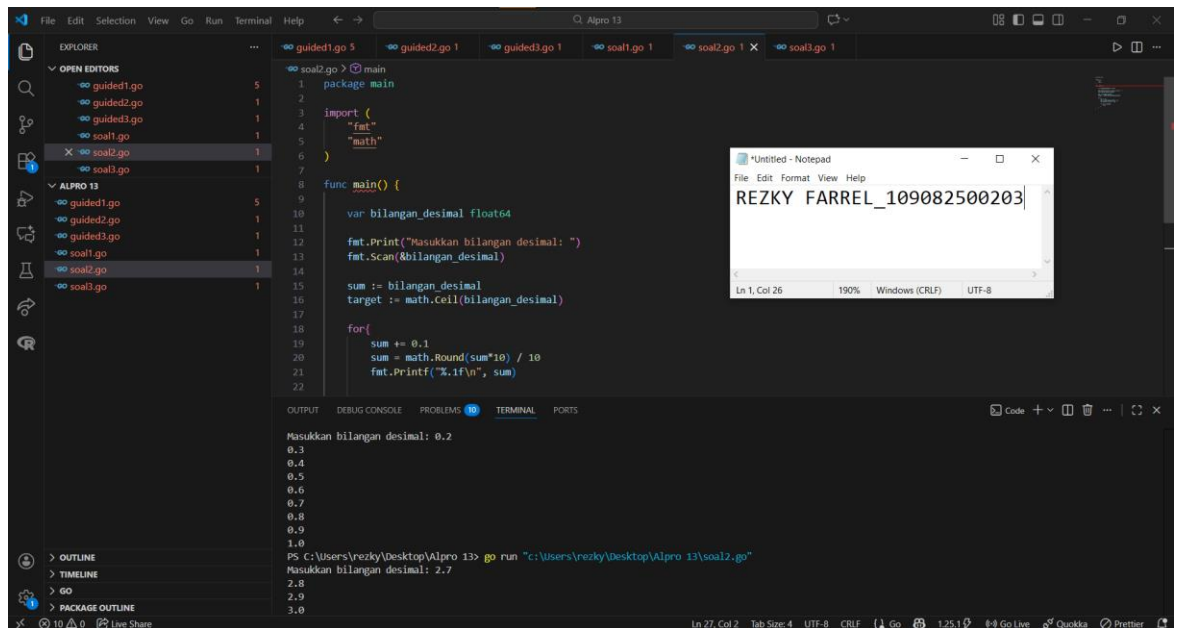
    sum := bilangan_desimal
    target := math.Ceil(bilangan_desimal)

    for{
        sum += 0.1
        sum = math.Round(sum*10) / 10
        fmt.Printf("%.1f\n", sum)

        if sum >= target{
            break
        }
    }
}
```

### Screenshoot program





### Deskripsi program

Program dirancang untuk memproses bilangan desimal masukan pengguna dengan cara menambahkan nilai 0.1 secara berulang hingga mencapai bilangan bulat terdekat di atasnya (ceiling). Setelah menerima input, program menentukan target pemberhentian menggunakan fungsi `math.Ceil` yang membulatkan angka desimal ke atas. Di dalam perulangan, nilai variabel terus ditambah dengan 0.1, namun program juga menerapkan rumus koreksi `math.Round(sum*10) / 10`.

### Tugas 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)
```

```

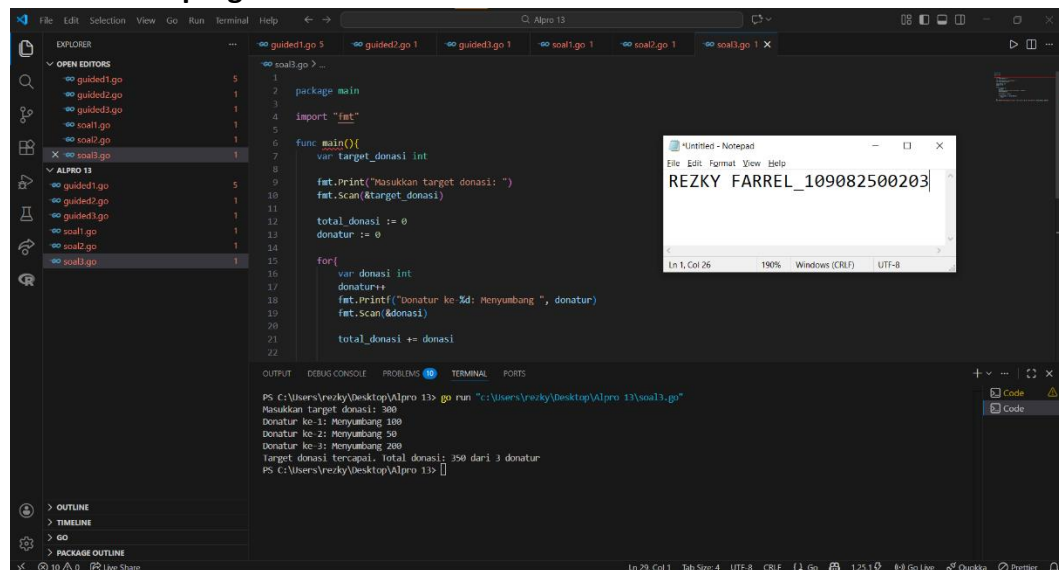
        total_donasi += donasi

        if total_donasi >= target_donasi{
            break
        }
    }

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

```

### Screenshoot program :



### Deskripsi program

Program ini merupakan simulasi aplikasi penggalangan dana yang bertujuan untuk mengumpulkan donasi hingga mencapai target tertentu. Proses dimulai dengan meminta pengguna memasukkan nominal target donasi, lalu program menginisialisasi variabel untuk mencatat total dana terkumpul dan jumlah donatur. Menggunakan struktur perulangan tanpa batas (for loop),