

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 13
REPEAT-UNTIL**



Disusun oleh:

NADIFA AZKHIA SYARIF

109082530002

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 int

    fmt.Scan(&kata)

    fmt.Scan(&jumlah)

    counter := 0

    for {

        fmt.Println(kata)

        counter++

        if counter == jumlah {

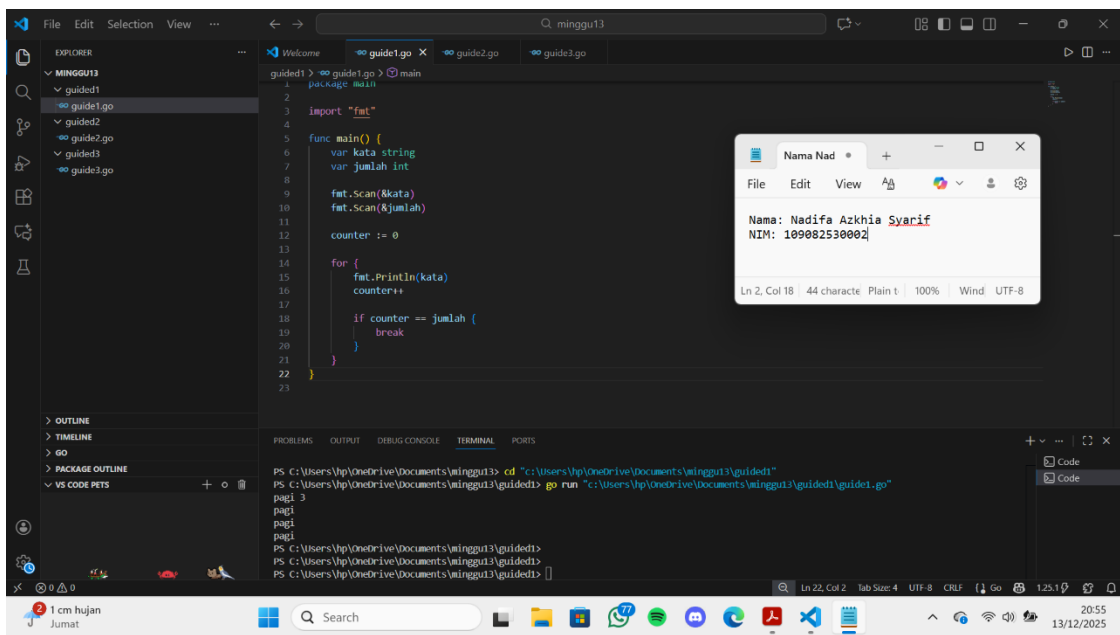
            break

        }

    }

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menampilkan sebuah kata sebanyak jumlah yang ditentukan. Program menerima satu masukan berupa kata dan satu masukan berupa bilangan bulat sebagai jumlah pengulangan. Setelah itu, program menjalankan perulangan tanpa kondisi yang dikendalikan oleh sebuah variabel penghitung. Pada setiap perulangan, kata akan ditampilkan dan nilai penghitung bertambah satu. Perulangan akan dihentikan ketika nilai penghitung sudah sama dengan jumlah pengulangan yang ditentukan, sehingga kata tersebut ditampilkan sesuai dengan jumlah yang diinginkan.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var n int
```

```

    for {

        fmt.Scan(&n)

        if n > 0 {

            break

        }

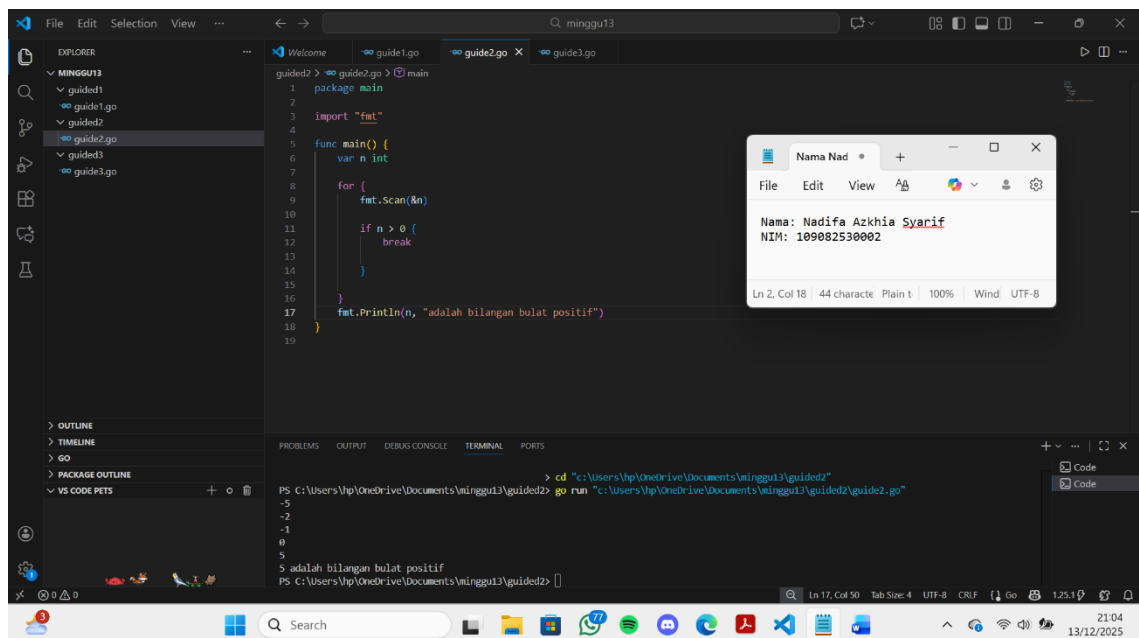
    }

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

}

```

Screenshoot program



Deskripsi program

Program ini digunakan untuk memastikan bahwa nilai yang diproses adalah bilangan bulat positif. Program akan terus membaca masukan berupa bilangan bulat di dalam sebuah perulangan tanpa kondisi. Setiap kali sebuah bilangan dimasukkan, program akan mengecek apakah nilainya lebih besar dari nol. Jika bilangan tersebut belum

positif (nol atau negatif), maka program tidak akan berhenti dan akan kembali meminta masukan. Perulangan baru akan dihentikan ketika bilangan yang dimasukkan bernilai positif. Setelah keluar dari perulangan, program menampilkan sebuah kalimat yang menyatakan bahwa bilangan tersebut adalah bilangan bulat positif.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    fmt.Scan(&x)

    fmt.Scan(&y)

    for x > 0 {

        fmt.Println(x)

        x = x - y

    }

    fmt.Println(x)

    if x == 0 {

        fmt.Println(true)

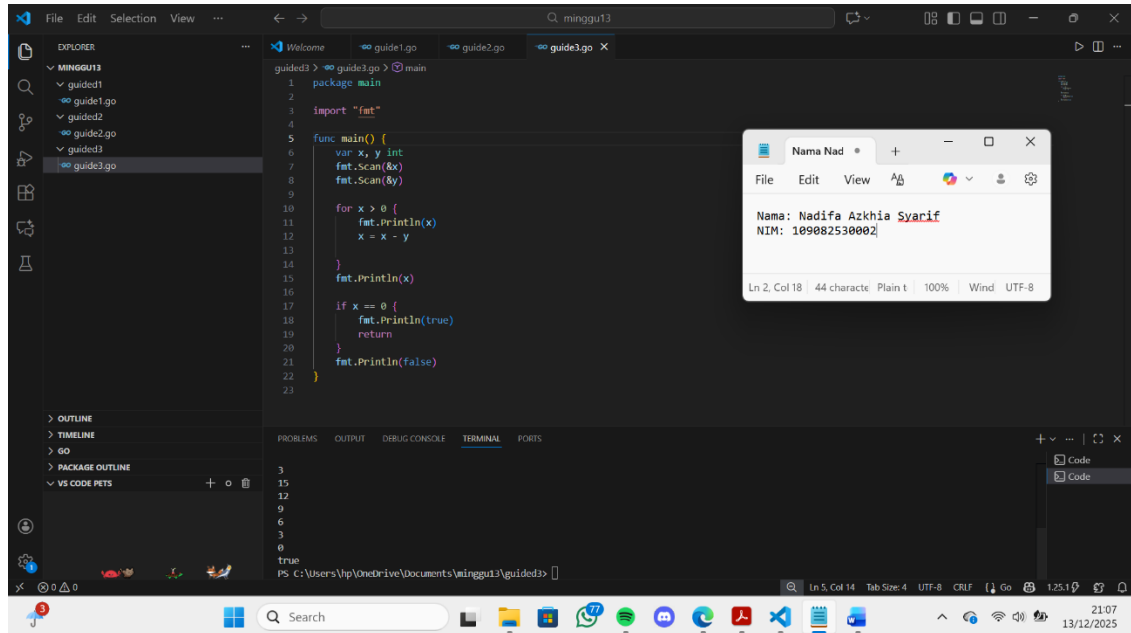
        return

    }

    fmt.Println(false)

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk mengecek apakah suatu bilangan merupakan kelipatan dari bilangan lainnya dengan cara pengurangan berulang. Program membaca dua bilangan bulat, yaitu x dan y . Nilai x kemudian dikurangi dengan y secara terus-menerus di dalam perulangan selama x masih bernilai positif. Setiap nilai x sebelum dikurangi akan ditampilkan ke layar. Setelah perulangan berhenti, program menampilkan nilai x terakhir. Jika nilai tersebut sama dengan nol, berarti x adalah kelipatan dari y dan program menampilkan nilai true. Jika nilai akhirnya negatif, berarti x bukan kelipatan dari y dan program menampilkan nilai false.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var n int
```

```
    fmt.Scan(&n)

    count := 0

    for {

        count++

        n = n / 10

        if n == 0 {

            break

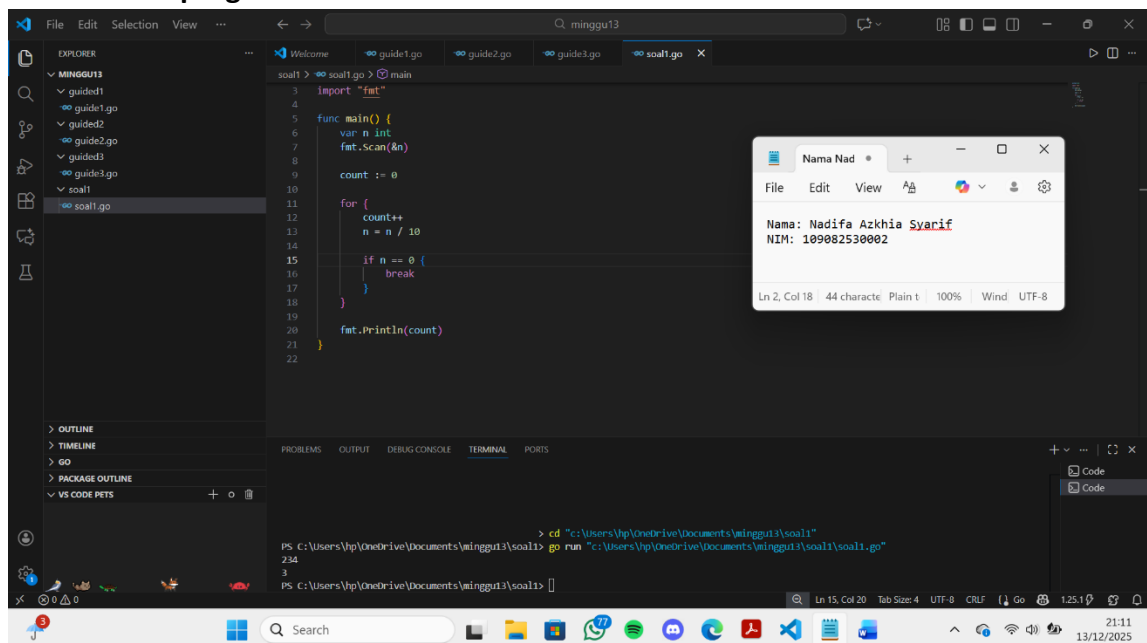
        }

    }

    fmt.Println(count)

}
```

Screenshoot program



Deskripsi program

Program menerima satu masukan berupa bilangan bulat positif. Program kemudian membagi bilangan tersebut dengan 10 secara berulang untuk menghilangkan satu digit setiap kali perulangan berjalan. Setiap perulangan menambah nilai penghitung digit. Proses ini terus dilakukan sampai nilai bilangan menjadi 0. Jumlah perulangan yang terjadi menunjukkan banyaknya digit dari bilangan tersebut, lalu hasilnya ditampilkan sebagai keluaran.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var x float64

    fmt.Scan(&x)

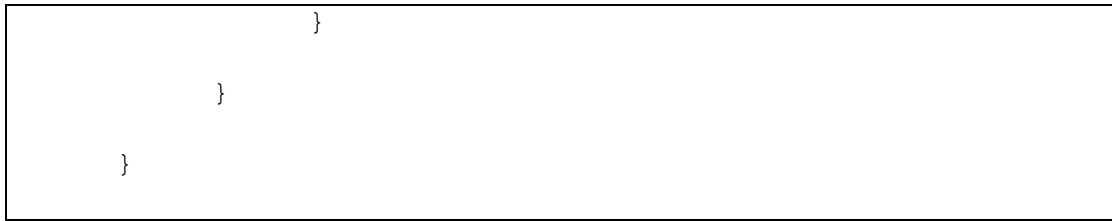
    atas := int(x)

    if x > float64(atas) {
        atas = atas + 1
    }

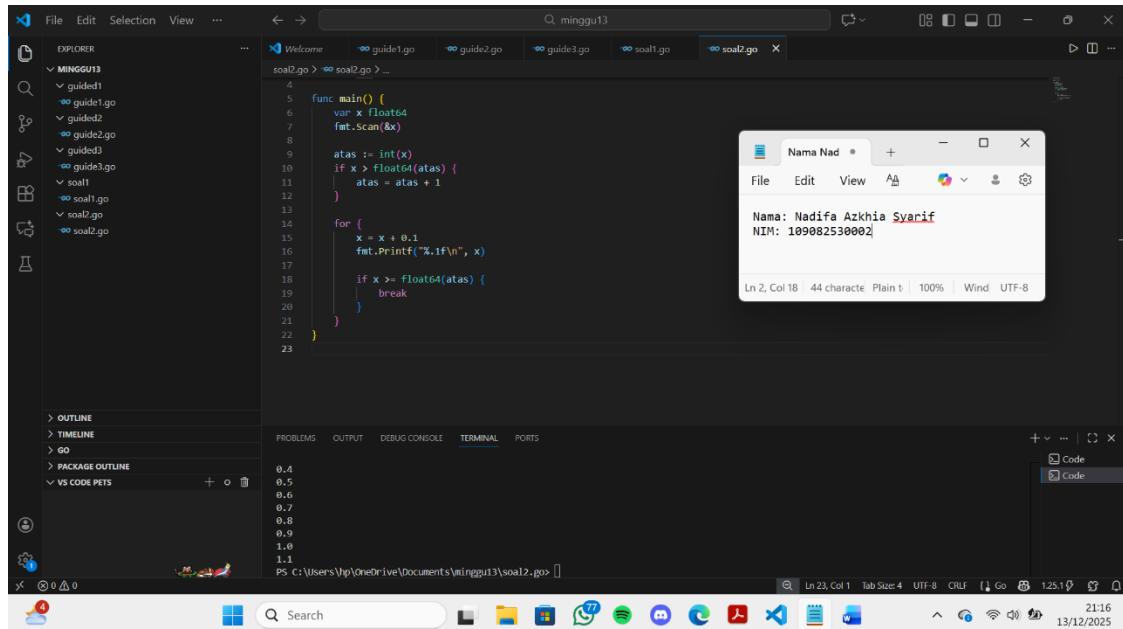
    for {
        x = x + 0.1

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

        if x >= float64(atas) {
            break
        }
    }
}
```

Screenshoot program



Deskripsi program

Program ini menerima sebuah bilangan desimal sebagai masukan. Program kemudian menentukan pembulatan ke atas dari bilangan tersebut dengan mengubahnya ke bilangan bulat dan menambah satu jika masih memiliki bagian desimal. Setelah itu, program melakukan perulangan dengan menambahkan nilai 0.1 setiap kali dan menampilkan hasilnya. Perulangan akan berhenti ketika nilai hasil penjumlahan sudah mencapai atau melewati hasil pembulatan ke atas dari bilangan awal.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var target int
    fmt.Scan(&target)

    total := 0
    donatur := 0
```

```

        for {
            var donasi int
            fmt.Scan(&donasi)

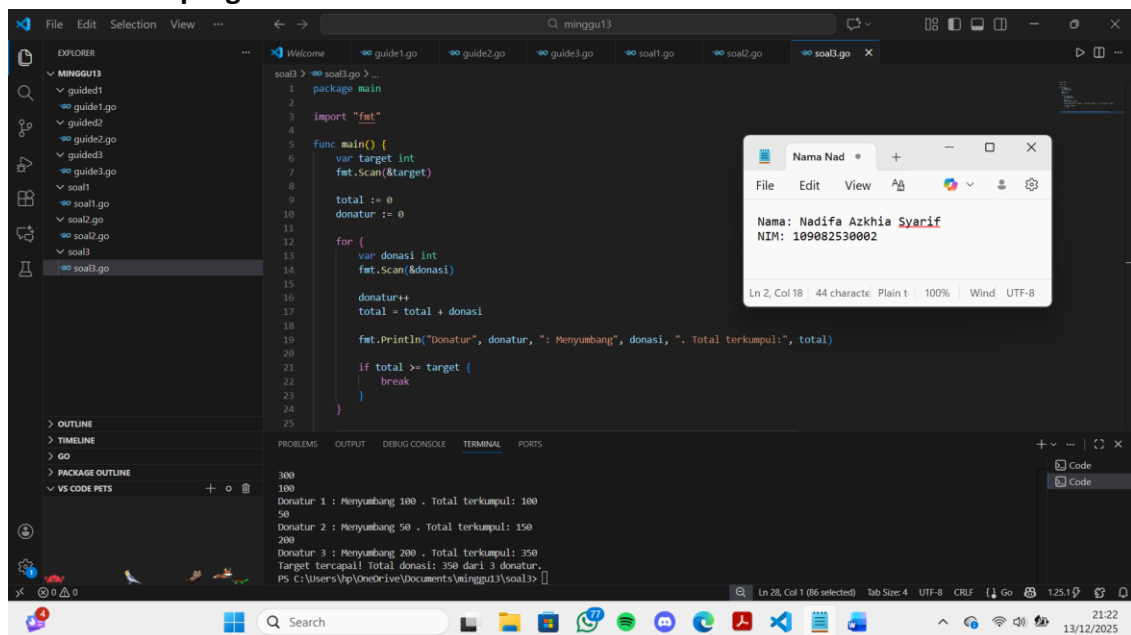
            donatur++
            total = total + donasi

            fmt.Println("Donatur", donatur, ": Menyumbang",
donasi, ". Total terkumpul:", total)

            if total >= target {
                break
            }
        }
        fmt.Println("Target tercapai! Total donasi:", total,
"dari", donatur, "donatur.")
    }

```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menghitung total donasi sampai mencapai atau melebihi target yang telah ditentukan. Program pertama-tama membaca sebuah bilangan bulat sebagai target donasi. Setelah itu, program akan terus menerima masukan berupa jumlah donasi dari setiap donatur. Setiap donasi yang masuk akan ditambahkan ke total donasi dan jumlah donatur juga akan dihitung. Pada setiap perulangan, program menampilkan informasi donatur ke berapa, jumlah sumbangan, dan total donasi sementara. Perulangan akan berhenti ketika total donasi sudah mencapai atau melebihi target. Setelah itu, program menampilkan pesan bahwa target telah tercapai beserta total donasi dan jumlah donatur.