

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 13
REPEAT UNTIL**



Disusun oleh:

Salsadilla Hanny Azizah

109082500014

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

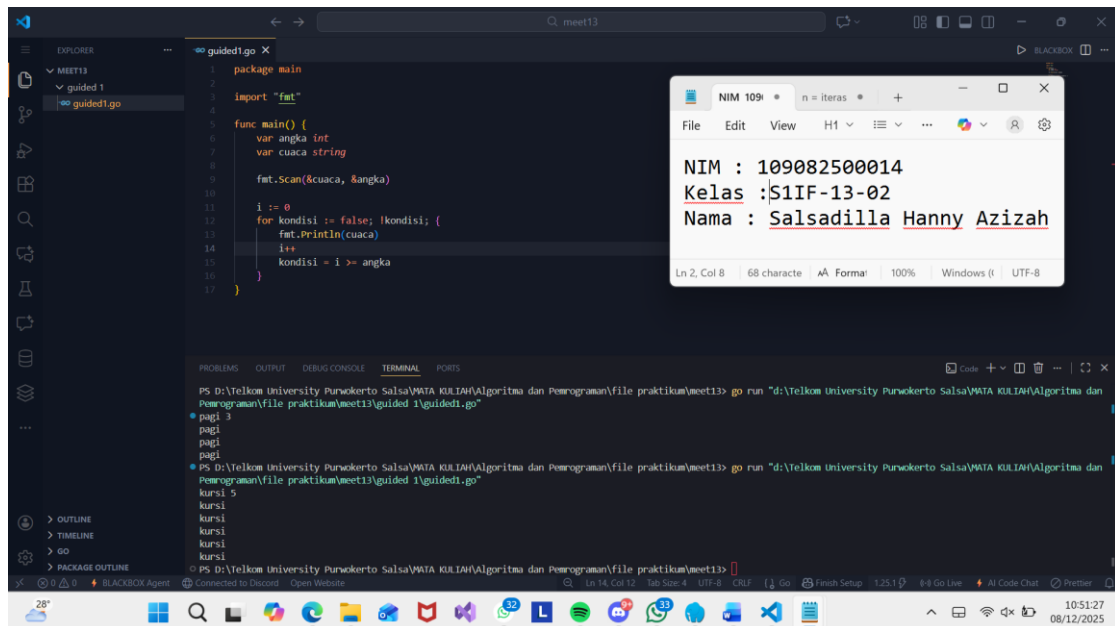
import "fmt"

func main() {
    var angka int
    var cuaca string

    fmt.Scan(&cuaca, &angka)

    i := 0
    for kondisi := false; !kondisi; {
        fmt.Println(cuaca)
        i++
        kondisi = i >= angka
    }
}
```

Screenshoot program



Deskripsi program :

Program ini digunakan untuk mencetak sebuah kata sebanyak jumlah yang dimasukkan oleh pengguna. Program menerima dua input, yaitu sebuah kata dan sebuah angka. Kata tersebut akan ditampilkan berulang, sedangkan angka menentukan berapa kali kata itu harus dicetak. Perulangan yang digunakan bekerja seperti konsep *repeat-until*, yaitu program langsung menampilkan kata minimal satu kali, lalu terus mengulang sampai jumlah cetak sudah sesuai dengan angka yang diberikan. Setelah jumlahnya terpenuhi, perulangan berhenti. Dengan cara ini, jika pengguna memasukkan "pagi 3", program akan mencetak kata "pagi" sebanyak tiga kali. Begitu juga jika pengguna memasukkan "kursi 5", program akan mencetak "kursi" sebanyak lima kali.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var angka int

    for {
```

```

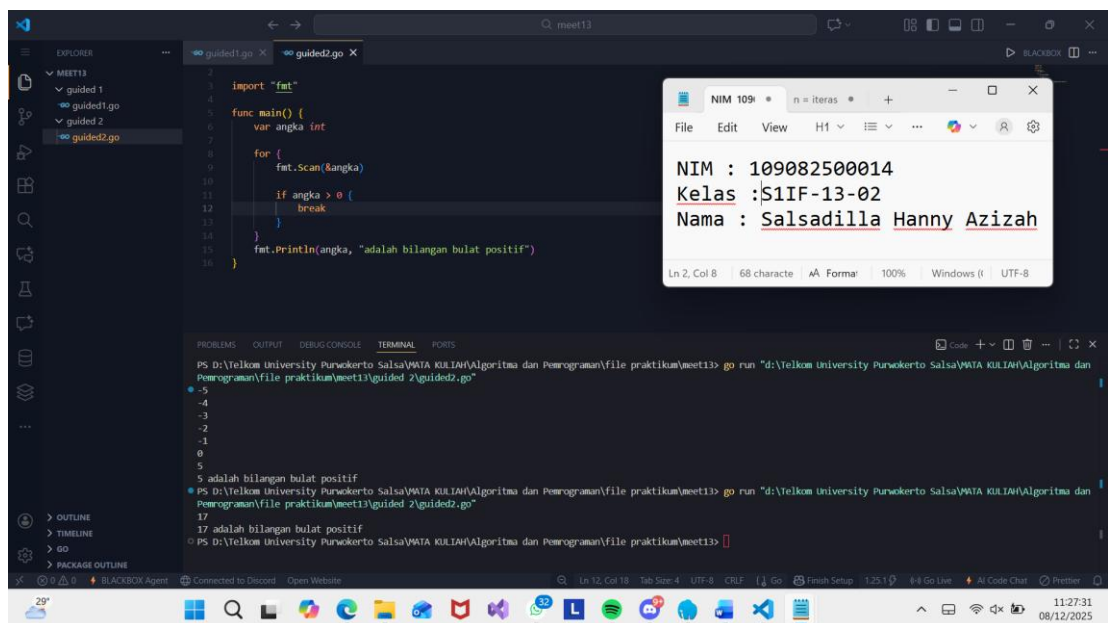
        fmt.Scan(&angka)

        if angka > 0 {
            break
        }
    }

    fmt.Println(angka, "adalah bilangan bulat positif")
}

```

Screenshoot program



Deskripsi program :

Program ini digunakan untuk memastikan bahwa pengguna memasukkan sebuah bilangan bulat positif. Program membaca satu input angka, tetapi jika angka yang diberikan tidak lebih dari nol, program akan terus meminta input baru. Perulangan yang digunakan adalah perulangan tanpa syarat berhenti (*infinite loop*), dan hanya akan dihentikan jika kondisi tertentu terpenuhi. Pada setiap iterasi, program memeriksa apakah angka yang dimasukkan lebih besar dari nol. Jika iya, perulangan dihentikan dengan `break`, dan program menampilkan pesan bahwa angka tersebut adalah bilangan bulat positif. Jika tidak, program kembali meminta input sampai pengguna memasukkan angka yang benar. Dengan cara ini, program memastikan bahwa output hanya muncul ketika pengguna memberikan bilangan bulat 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

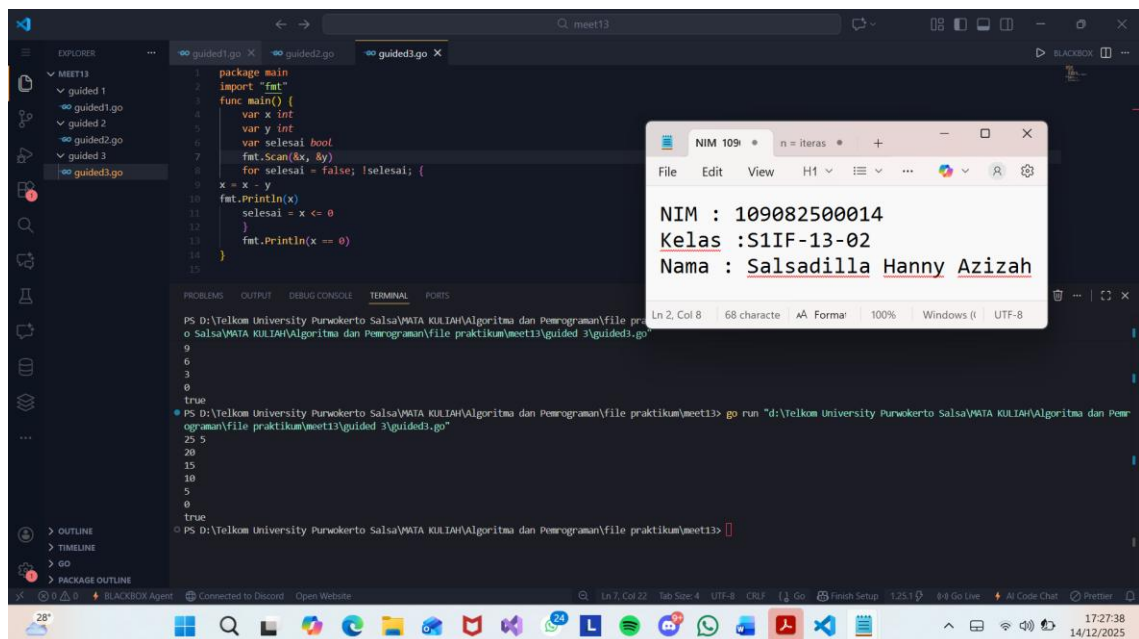
    }

    fmt.Println(x == 0)

}

```

Screenshoot program



Deskripsi program:

Program ini bertujuan untuk menentukan apakah suatu bilangan merupakan kelipatan dari bilangan lain. Program menerima dua bilangan bulat positif sebagai input, yaitu X dan Y. Setelah itu, nilai X dikurangi dengan Y secara berulang, dan setiap hasil pengurangan ditampilkan. Pengurangan dilakukan terus sampai nilai X menjadi negatif. Dari proses ini dapat diketahui apakah X habis dikurangi oleh Y atau tidak. Jika selama proses pengurangan nilai X mencapai nol, maka X merupakan kelipatan dari Y.

Jika tidak pernah mencapai nol, maka X bukan kelipatan dari Y. Dengan cara ini, program dapat menentukan kelipatan tanpa menggunakan operasi pembagian.

TUGAS

1. Tugas 1

Source code

```
package main

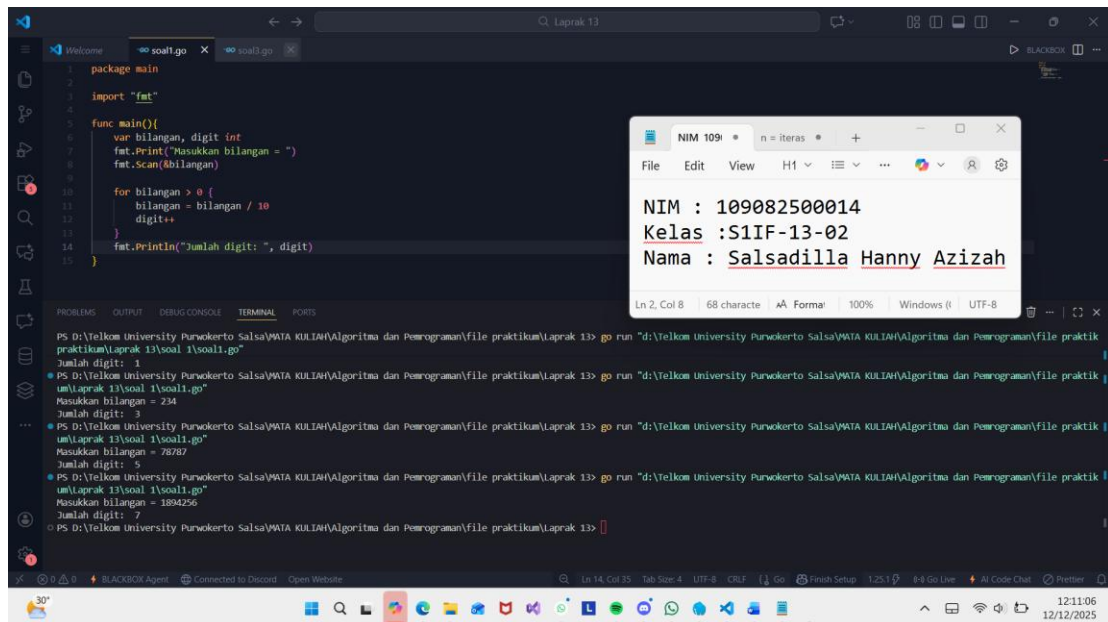
import "fmt"

func main() {
    var bilangan, digit int
    fmt.Print("Masukkan bilangan = ")
    fmt.Scan(&bilangan)

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

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

Screenshoot program



Deskripsi program:

Program ini dibuat untuk menghitung berapa banyak digit yang dimiliki sebuah bilangan. Pengguna cukup memasukkan satu bilangan bulat positif, lalu program akan memproses angka tersebut dengan cara membaginya terus-menerus dengan 10. Setiap kali dibagi 10, satu digit paling belakang dari angka itu akan hilang. Dari proses ini, program bisa mengetahui berapa kali pembagian terjadi, dan jumlah pembagian tersebut menampilkan banyak digit yang dimiliki bilangan tersebut. Setelah angka habis atau menjadi nol, program menampilkan hasil akhirnya, yaitu total digit dari angka yang dimasukkan. Dengan ini, program dapat menghitung jumlah digit untuk bilangan berapa pun, baik pendek maupun panjang.

2. Tugas 2

Source code

```
package main

import "fmt"

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

    x := int(input * 10)
    batas := (int(input) + 1) * 10
```

```

        for x < batas {
            x += 1

            if x == batas {
                fmt.Println(x / 10)
            } else {
                fmt.Printf("%.1f\n", float64(x)/10)
            }
        }
    }
}

```

Screenshoot program

The screenshot shows a Go program in VS Code. The source code in `soal2.go` is as follows:

```

package main

import "fmt"

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

    x := int(input * 10)
    batas := (int(input) + 1) * 10

    for x < batas {
        x += 1

        if x == batas {
            fmt.Println(x / 10)
        } else {
            fmt.Printf("%.1f\n", float64(x)/10)
        }
    }
}

```

The terminal output shows the program running and printing the sequence of numbers from 0.4 to 2.9, with the final output being 3 (when x reaches the ceiling of 3.0).

```

PS D:\Telkom University Purwokerto Salsa\WATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laparak 13> go run "d:\Telkom University Purwokerto Salsa\WATA KULIAH\Algoritma dan Pemrograman\file praktikum\Laparak 13\soal 2\soal2.go"
0.4
0.5
0.6
0.7
0.8
0.9
1
2.7
2.8
2.9
3

```

Deskripsi program:

Program ini digunakan untuk menampilkan urutan bilangan dari hasil penjumlahan secara bertahap sampai mencapai pembulatan ke atas (ceil) dari bilangan yang dimasukkan pengguna. Masukan yang diterima berupa sebuah bilangan desimal, misalnya 0.2 atau 2.7. Setelah menerima input, program akan memulai proses dengan menaikkan nilai tersebut perlahan menggunakan penambahan 0.1 pada setiap perulangan. Setiap kali nilainya bertambah, program langsung menampilkan angka hasil penjumlahan tersebut. Proses ini berjalan terus sampai angkanya mencapai bilangan bulat terdekat di atasnya. Ketika nilai akhir sudah sama dengan bilangan bulat tersebut, program tetap menampilkannya sebagai angka bulat tanpa desimal. Dengan cara ini, program membantu menunjukkan bagaimana nilai desimal bergerak secara bertahap menuju pembulatan ke atasnya.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var target, donasi, total, donatur int

    fmt.Print("Masukkan Target Donasi = ")
    fmt.Scan(&target)

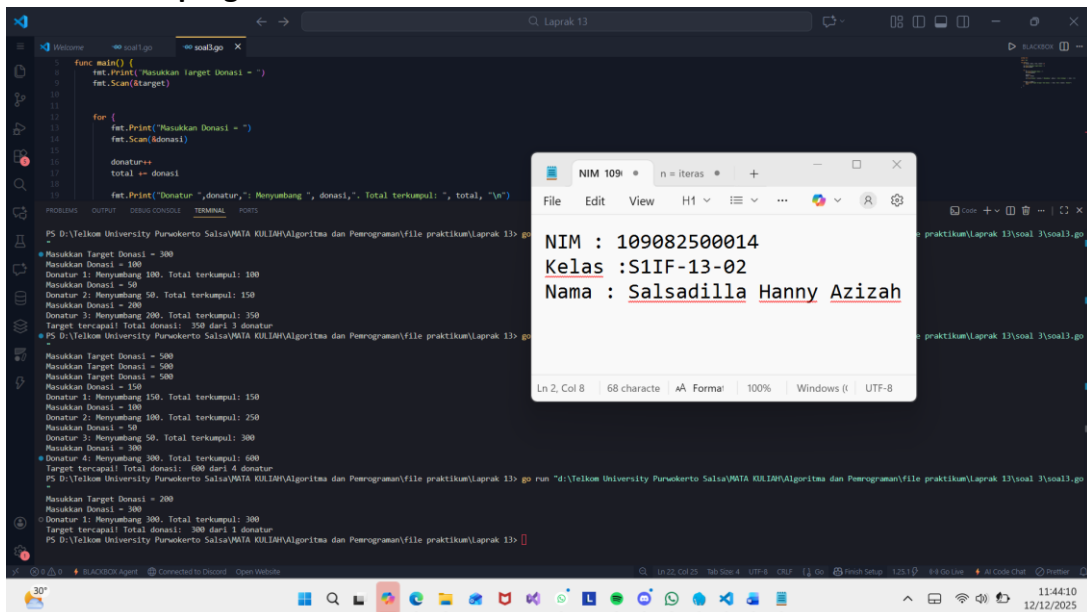
    for {
        fmt.Print("Masukkan Donasi = ")
        fmt.Scan(&donasi)

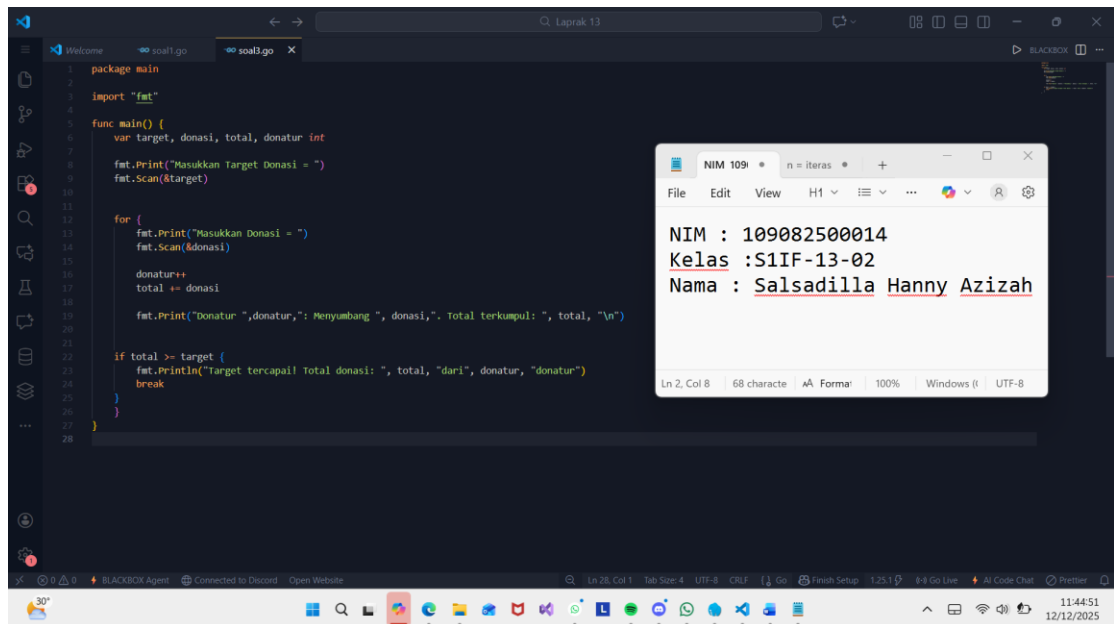
        donatur++
        total += donasi

        fmt.Print("Donatur ", donatur, ": Menyumbang ",
            donasi, ". Total terkumpul: ", total, "\n")

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

Screenshoot program





Deskripsi program :

Program ini dibuat untuk menghitung donasi sampai target yang sudah ditentukan tercapai. Pengguna diminta memasukkan target donasi yang ingin dicapai. Lalu, program mencatat donasi dari setiap donatur. Setiap pengguna memasukkan jumlah donasi, program akan menambahkannya ke total donasi dan mencatat ada satu donatur baru yang ikut menyumbang. Program langsung menampilkan informasi donatur keberapa, dan jumlah donasi yang diberikan, sampai total donasi yang sudah terkumpul. Proses ini berjalan terus selama total donasi belum mencapai target. Kalau total donasi sudah tercapai atau melebihi target, program akan menampilkan bahwa target donasi sudah tercapai, lalu memberi tahu total donasi yang terkumpul serta berapa banyak donatur yang ikut berkontribusi.