

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 13
REPEAT-UNTIL**



Disusun oleh:

RAFFI YULIANSYAH

109082500101

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 num int

    var word string

    var kondisi bool

    fmt.Scan(&word)

    fmt.Scan(&num)

    for kondisi = false; !kondisi; {

        fmt.Println(word)

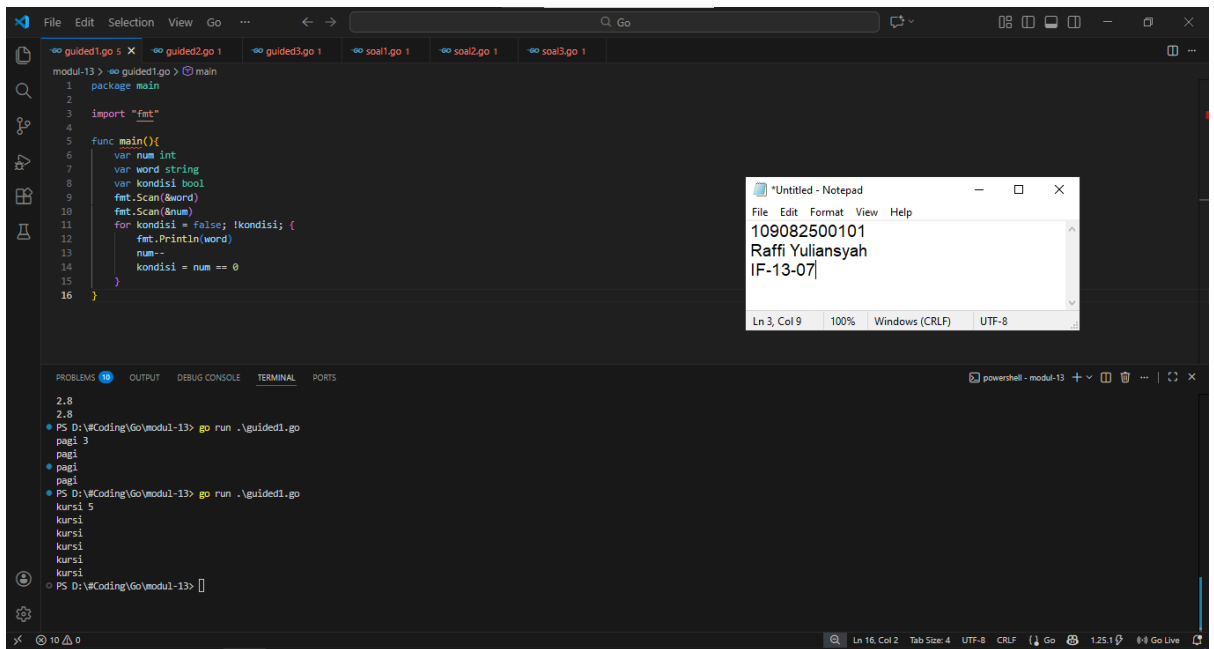
        num--

        kondisi = num == 0

    }

}
```

Screenshoot program



Deskripsi program

Program ini adalah program untuk menampilkan sebuah kata secara berulang kali sesuai dengan jumlah yang diinputkan. Program dimulai dengan mendeklarasikan tiga variabel, yakni num yang bertipe data bilangan bulat (int), word yang bertipe data teks (string), dan kondisi yang bertipe data boolean (bool). Selanjutnya, program akan diisi input, di mana input kata disimpan ke variabel word dan input angka disimpan ke variabel num, (fmt.Scan(&word) dan fmt.Scan(&num)). Setelah itu, program menjalankan struktur perulangan (for) yang diinisialisasi dengan kondisi bernilai false dan akan terus berjalan selama kondisi bernilai salah karena (!kondisi). Di dalam perulangan ini, program menampilkan isi variabel word (fmt.Println(word)), lalu mengurangi nilai num sebesar satu (num--), dan memperbarui nilai variabel kondisi menjadi hasil perbandingan apakah num sudah sama dengan 0 (kondisi = num == 0). Dan ketika num mencapai 0, kondisi menjadi true, (!kondisi) yang menyebabkan perulangan berhenti.

2. Guided 2

Source Code

```
package main

import "fmt"

func main(){

    var num int
```

```

        for kondisi := false; !kondisi;{

            fmt.Scan(&num)

            if num > 0 {

                fmt.Print(num, " Adalah bilangan bulat
positif")

                kondisi = true

            }

        }

    }
}

```

Screenshoot program

```

1 package main
2
3 import "fmt"
4
5 func main(){
6     var num int
7     for kondisi := false; !kondisi;{
8         fmt.Scan(&num)
9         if num > 0 {
10             fmt.Print(num, " Adalah bilangan bulat positif")
11             kondisi = true
12         }
13     }
14 }
15

```

Terminal Output:

```

PS D:\Coding\Go\modul-13> go run .\guided2.go
-2
-5
0
-17
4
4 Adalah bilangan bulat positif
PS D:\Coding\Go\modul-13> go run .\guided2.go
19
19 Adalah bilangan bulat positif
PS D:\Coding\Go\modul-13>

```

Deskripsi program

Program ini adalah program untuk memvalidasi input agar pengguna memasukkan bilangan bulat positif. Program dimulai dengan mendeklarasikan satu variabel, yakni `num`, yang bertipe data bilangan bulat (`int`). Selanjutnya, program menjalankan struktur perulangan (`for`) yang menginisialisasi variabel bantu `kondisi` dengan nilai `false` dan akan terus berproses selama kondisi tersebut bernilai salah (`!kondisi`). Di dalam perulangan ini, program akan meminta input angka yang disimpan ke dalam variabel `num` (`fmt.Scan(&num)`). Setelah itu, program memeriksa input tersebut menggunakan percabangan (`if`); jika nilai `num` lebih besar dari 0, program akan menampilkan pesan bahwa angka tersebut adalah bilangan bulat positif menggunakan `fmt.Print()` dan mengubah nilai `kondisi` menjadi `true`. Dan tindakan pengubahan nilai `kondisi` menjadi `true` inilah yang akan menghentikan perulangan,

sedangkan jika input bukan positif, program akan terus mengulang proses permintaan input.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    fmt.Scan(&x)

    fmt.Scan(&y)

    for kondisi := false; !kondisi; {

        x = x - y

        fmt.Println(x)

        if x <= 0 {

            kondisi = x == 0

            fmt.Print(kondisi)

            kondisi = true

        }

    }

}
```

Screenshoot program

The screenshot shows a Go IDE with a file named `guided3.go` open. The code defines a `main` function that takes two integers `x` and `y` as input. It enters a `for` loop that continues as long as `x` is greater than or equal to 0. Inside the loop, `x` is decremented by `y` (`x = x - y`), and the current value of `x` is printed. After the loop, the program checks if `x` is equal to 0. If true, it prints `true`; otherwise, it prints `false`.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x, y int
7     fmt.Scan(&x)
8     fmt.Scan(&y)
9     for kondisi := false; !kondisi; {
10         x = x - y
11         fmt.Println(x)
12         if x <= 0 {
13             kondisi = x == 0
14             fmt.Print(kondisi)
15         }
16         kondisi = true
17     }
18 }
19
```

The terminal output shows the execution of the program. It starts with `-1` and `false`, followed by a series of values for `x` (15, 12, 9, 6, 3, 0) and the boolean result `true`. The status bar at the bottom indicates the current cursor position is at line 14, column 31.

Deskripsi program

Program ini adalah program untuk melakukan pengurangan berulang nilai `x` dengan `y` serta menentukan apakah hasil akhirnya mencapai tepat angka nol atau bernilai negatif. Program dimulai dengan mendeklarasikan dua variabel, yakni `x` dan `y`, yang bertipe data bilangan bulat (`int`). Selanjutnya, program akan diisikan input, di mana angka pertama disimpan ke dalam variabel `x` dan angka kedua ke dalam variabel `y` (`fmt.Scan(&x)` dan `fmt.Scan(&y)`). Bunda itu, program menjalankan struktur perulangan (`for`) yang diinisialisasi dengan variabel `kondisi` bernilai `false` dan terus berproses selama `kondisi` bernilai salah. Di dalam perulangan ini, nilai `x` akan terus dikurangi dengan `y` (`x = x - y`) dan hasilnya ditampilkan baris demi baris. Jika nilai `x` sudah kurang dari atau sama dengan 0, program akan mengevaluasi apakah `x` tepat bernilai 0 (`kondisi = x == 0`) dan menampilkan hasil evaluasi boolean tersebut (mencetak `true` jika habis dibagi, atau `false` jika hasilnya negatif). Dan terakhir, variabel `kondisi` dipaksa menjadi `true` agar perulangan berhenti setelah hasil akhir tercapai.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var n, digit int

    digit = 0

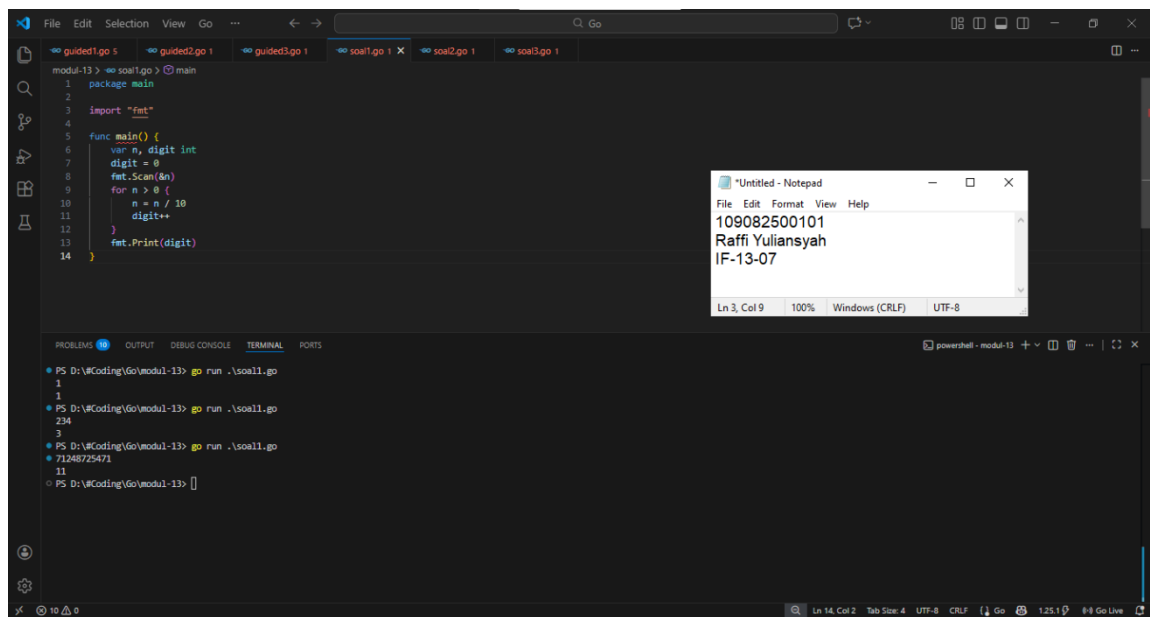
    fmt.Scan(&n)

    for n > 0 {
        n = n / 10

        digit++
    }

    fmt.Print(digit)
}
```

Screenshoot program



Deskripsi program

Program ini adalah program untuk menghitung jumlah digit yang terdapat dalam sebuah bilangan bulat positif. Program dimulai dengan mendeklarasikan dua variabel, yakni *n* dan *digit*, yang bertipe data bilangan bulat (*int*), di mana variabel *digit* diinisialisasi dengan nilai 0. Selanjutnya, program akan diisikan input dengan satu angka, yang mana angka tersebut akan disimpan ke dalam variabel *n* (*fmt.Scan(&n)*). Setelah itu, program akan memproses angka tersebut menggunakan struktur perulangan (*for*) yang terus berjalan selama nilai *n* lebih besar dari 0. Di dalam setiap perulangan, nilai *n* akan dibagi dengan 10 (*n = n / 10*) untuk menghilangkan digit terakhir dari bilangan tersebut, lalu variabel *digit* ditambahkan satu (*digit++*) sebagai penghitung. Dan setelah perulangan selesai (saat nilai *n* sudah habis atau menjadi 0), program akan menampilkan hasil akhir berupa jumlah digit yang tersimpan dalam variabel *digit* dengan *fmt.Print()*.

2. Tugas 2

Source code

```
package main

import (

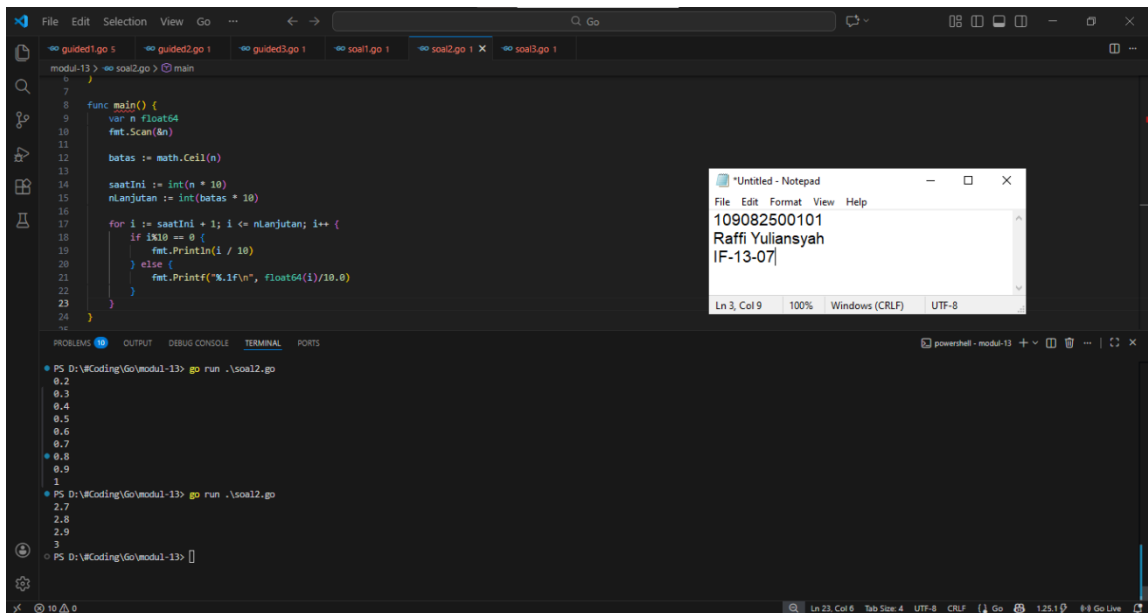
    "fmt"

    "math"

)
```

```
func main() {  
  
    var n float64  
  
    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 adalah program untuk menampilkan urutan bilangan desimal dengan kenaikan 0.1, dimulai dari setelah angka input hingga mencapai bilangan bulat terdekat di atasnya (*ceiling*). Program dimulai dengan mendeklarasikan satu variabel, yakni *n*, yang bertipe data bilangan desimal (*float64*). Selanjutnya, program akan diisi input, yang mana angka tersebut akan disimpan ke dalam variabel *n* (*fmt.Scan(&n)*). Setelah itu, program menghitung batas atas bilangan bulat menggunakan fungsi *math.Ceil(n)* dan mengonversi nilai input serta batas atas tersebut menjadi bilangan bulat dengan cara dikalikan 10 (disimpan dalam variabel *saatIni* dan *nLanjutan*) untuk mempermudah iterasi. Program kemudian menjalankan struktur perulangan (*for*) yang dimulai dari *saatIni + 1* hingga mencapai *nLanjutan*. Di dalam perulangan ini, program memeriksa apakah nilai iterasi saat ini habis dibagi 10 (*i%10 == 0*); jika ya, program menampilkan hasil pembagian sebagai bilangan bulat, namun jika tidak, program mengonversinya kembali menjadi desimal dan menampilkannya dengan format satu angka di belakang koma (*%.1f*).

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

The screenshot shows a Go program running in a terminal window. The program prompts for a target donation amount and then for individual donation amounts until the total reaches the target. The output shows four donations of 500, 255, 123, and 124, reaching a total of 1002.

```

PS D:\Coding\Go\modul-13> go run .\soal3.go
Masukkan target donasi: 1000
Masukkan jumlah uang untuk di donasikan:
500
Donatur 1: menyumbang 500. Total terkumpul: 500
255
Donatur 2: menyumbang 255. Total terkumpul: 755
123
Donatur 3: menyumbang 123. Total terkumpul: 878
124
Donatur 4: menyumbang 124. Total terkumpul: 1002
Target tercapai! total donasi: 1002 dari 4 donatur.

```

Deskripsi program

Program ini adalah program untuk menghitung proses penggalangan dana hingga jumlah total donasi mencapai atau melebihi target yang ditentukan. Program dimulai dengan mendeklarasikan tiga variabel utama, yakni target, total, dan donatur, yang bertipe data bilangan bulat (int). Selanjutnya, program akan meminta input awal untuk menetapkan sasaran dana yang akan disimpan ke dalam variabel target (`fmt.Scan(&target)`), serta menginisialisasi variabel total dan donatur dengan nilai 0. Setelah itu, program akan menjalankan proses penerimaan donasi menggunakan struktur perulangan (for) yang terus berjalan selama nilai total masih kurang dari atau sama dengan nilai target. Di dalam setiap perulangan ini, program memproses input nominal donasi baru (disimpan dalam variabel lokal donasi), lalu menambahkan jumlah donatur (`donatur += 1`), mengakumulasi donasi tersebut ke dalam total terkumpul (`total += donasi`), serta menampilkan status donasi saat ini ke layar. Dan ketika total donasi akhirnya sudah melampaui target (perulangan berhenti), program akan menampilkan pesan keberhasilan beserta ringkasan akhir total dana dan jumlah donatur yang berpartisipasi.