

LAPORAN PRA KTIKUM

Algoritma Pemrograman

MODUL 13

Repeat-until



Disusun oleh:

MUHAMMAD TETUKO KEMAL PASHA

109082500181

S1IF-13-04

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 word string
    var repetitions int
    fmt.Scan(&word, &repetitions)
    counter := 0
    for done := false; !done; {
        fmt.Println(word)
        counter++
        done = (counter >= repetitions)
    }
}
```

Screenshot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var word string
7     var repetitions int
8     fmt.Scan(&word, &repetitions)
9     counter := 0
10    for done := false; !done; {
11        fmt.Println(word)
12        counter++
13        done = (counter >= repetitions)
14    }
15 }
```

Ask about your code
AI responses may be inaccurate.

name:muhhammad tetuko Kemal pasha
nim :10908250181
kelas:S1-IF13-04

File Edit View ^A File Edit View ^A

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13> go run "c:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13\guided1.go"
pag1
pag1
pag1
pag1
pag1
PS C:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13> go run "c:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13\guided1.go"
pag1
pag1
pag1
pag1
pag1
PS C:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13>

Ln 2, Col 20 70 character Plain t | 100% Wind UTF-8

guided1.go X Explore and understand your code
Ask Auto

Lightning nearby 79°F 10 12.5.1 8:48 PM 12/14/2025

Deskripsi program

Berdasarkan gambar tangkapan layar, program Go yang sedang ditampilkan adalah *Guided 1* dari Laporan Praktikum Modul 13. Program ini dirancang untuk melakukan perulangan bertipe *repeat-until* yang diimplementasikan menggunakan *loop for* dengan kondisi berhenti yang dikontrol oleh variabel boolean *done*. Program ini menerima dua input dari pengguna: sebuah kata (*word*) dan jumlah pengulangan (*repetitions*). Di dalam *loop*, kata tersebut dicetak (*fmt.Println(word)*) , penghitung (*counter*) dinaikkan , dan variabel *done* diatur menjadi true ketika *counter* mencapai atau melebihi *repetitions* (*done = (counter >= repetitions)*), memastikan kata tersebut dicetak sebanyak jumlah yang ditentukan.

2. guided 2

Source code

```
package main

import "fmt"

func main() {
    var number int
    var continueLoop bool
    for continueLoop = true; continueLoop; {
        fmt.Scan(&number)
        continueLoop = number <= 0
    }
    fmt.Printf("%d adalah bilangan bulat positif\n", number)
}
```

Screenshot program

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The main area displays a Go file named `guided2.go` with the following code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var number int
7     var continueLoop bool
8     for continueLoop = true; continueLoop; {
9         fmt.Scan(&number)
10        continueLoop = number <= 0
11    }
12    fmt.Printf("%d adalah bilangan bulat positif\n", number)
13}
```

The terminal below shows the output of running the program with various inputs:

```
PS C:\Users\tetuk\OneDrive\文件\semester 1\LAPRAK\laprak mod 13> go run "c:\Users\tetuk\OneDrive\文件\semester 1\LAPRAK\laprak mod 13\guided2.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS C:\Users\tetuk\OneDrive\文件\semester 1\LAPRAK\laprak mod 13> go run "c:\Users\tetuk\OneDrive\文件\semester 1\LAPRAK\laprak mod 13\guided2.go"
17
17 adalah bilangan bulat positif
PS C:\Users\tetuk\OneDrive\文件\semester 1\LAPRAK\laprak mod 13>
```

The status bar at the bottom right indicates the date and time as 12/14/2025, 8:55 PM.

Deskripsi program

Berdasarkan gambar tangkapan layar, program Go yang sedang ditampilkan adalah *Guided 1* dari Laporan Praktikum Modul 13. Program ini dirancang untuk melakukan perulangan bertipe *repeat-until* yang diimplementasikan menggunakan *loop for* dengan kondisi berhenti yang dikontrol oleh variabel boolean `done`. Program ini menerima dua input dari pengguna: sebuah kata (word) dan jumlah pengulangan (repetitions). Di dalam *loop*, kata tersebut dicetak (`fmt.Println(word)`) , penghitung (counter) dinaikkan , dan variabel `done` diatur menjadi `true` ketika counter mencapai atau melebihi repetitions (`done = (counter >= repetitions)`), memastikan kata tersebut dicetak sebanyak jumlah yang ditentukan.

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)
}
```

Screenshot program

```
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)
}
```

name:muhhammad tetuko Kemal pasha
nim :109082500181
kelas:S1-IF13-04

Deskripsi program

Berdasarkan gambar tangkapan layar, ketiga program Go yang ditampilkan (Guided 1, Guided 2, dan Guided 3) semuanya mengilustrasikan implementasi perulangan REPEAT-UNTIL menggunakan struktur for pada bahasa pemrograman Go. Guided 1 mencetak sebuah kata berulang kali sampai jumlah cetakan mencapai target yang ditentukan, menggunakan variabel boolean done untuk mengontrol kondisi berhenti. Guided 2 terus-menerus meminta input angka dari pengguna dan akan mengulang permintaan selama angka yang dimasukkan kurang dari atau sama dengan nol, baru berhenti setelah mendapatkan bilangan bulat positif. Sementara itu, Guided 3 melakukan pengurangan berulang pada variabel \$x\$ oleh variabel \$y\$ hingga \$x \leq 0\$, yang kondisi berhentinya juga diatur oleh variabel boolean selesai. Secara keseluruhan, ketiga contoh ini menunjukkan bagaimana konsep *repeat-until* — di mana aksi dieksekusi minimal satu kali dan perulangan berlanjut *sampai* kondisi terpenuhi — diimplementasikan dalam Go.

1. tugas 1

Source code

```
package main

import (
    "fmt"
)

func main() {
    var bilangan int
    fmt.Scan(&bilangan)
    jumlahDigit := len(fmt.Sprintf("%d", bilangan))
    fmt.Println("Jumlah digit:", jumlahDigit)
}
```

Screenshot program

The screenshot shows a Go development environment with the following details:

- Code Editor:** The main window displays the source code for `tugas1.go`. The code reads an integer from standard input, converts it to a string, and then calculates its length to determine the number of digits.
- Terminal:** The bottom-right panel shows the terminal output. It includes a prompt, command execution, and the resulting output: "Jumlah digit: 1".
- Status Bar:** The bottom status bar provides system information such as battery level (79%), weather (Mostly cloudy), and system time (12:04 PM, 12/14/2025).

Deskripsi program

Berdasarkan kode dan *screenshot* yang tersedia (Guided 1, Guided 2, Guided 3, dan Tugas 1), program-program ini mengilustrasikan berbagai implementasi perulangan dalam bahasa Go. Guided 1 menunjukkan perulangan *repeat-until* yang mencetak sebuah kata berulang kali hingga mencapai jumlah pengulangan yang ditentukan, dikontrol oleh variabel boolean done. Guided 2 mengimplementasikan perulangan yang akan terus meminta input angka dari pengguna dan hanya akan berhenti ketika pengguna memasukkan bilangan bulat positif, menggunakan variabel continueLoop sebagai kondisi terminasi. Guided 3 menggunakan perulangan untuk melakukan pengurangan berulang ($x = x - y$) dan berhenti saat $x \leq 0$, yang menguji apakah x habis dibagi oleh y . Sementara itu, Tugas 1 menunjukkan cara menghitung jumlah digit sebuah bilangan bulat dengan mengubah bilangan tersebut menjadi *string* lalu menghitung panjangnya, tanpa menggunakan konsep perulangan *repeat-until* yang merupakan fokus modul ini. Secara keseluruhan, program-program ini berfungsi sebagai latihan dasar untuk memahami mekanisme kontrol alur perulangan.

2. tugas 2

Source code

```
package main

import "fmt"

func main() {

    var x float64

    fmt.Scan(&x)

    batasAtas := float64(int(x)) + 1

    for v := x + 0.1; v <= batasAtas; v += 0.1 {

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

    }

}
```

Screenshot program

```
package main

import "fmt"

func main() {
    var x float64
    fmt.Scan(&x)
    batasAtas := float64(int(x)) + 1
    for v := x + 0.1; v <= batasAtas; v += 0.1 {
        fmt.Printf("%.1f\n", v)
    }
}
```

name:muhammad tetuko Kemal pasha
nim :109082500181
kelas:S1-IF13-04

Deskripsi program

Kelima tangkapan layar ini menampilkan kode program Go dan hasil eksekusinya dari Modul 13 tentang perulangan. Program-program ini meliputi Guided 1 (mencetak kata 'pagi' berulang kali dengan jumlah pengulangan 5), Guided 2 (terus meminta input angka hingga mendapatkan bilangan bulat positif, seperti 5, -1, 0, lalu 17), Guided 3 (melakukan pengurangan berulang ($x = x - y$) dan menampilkan hasil $x==0$ untuk menguji pembagian), Tugas 1 (menghitung jumlah digit sebuah bilangan dengan mengubahnya menjadi *string*), dan Tugas 2 (mencetak deretan angka desimal mulai dari $x + 0.1$ hingga batas atas terdekat dengan kenaikan 0.1). Secara keseluruhan, file-file ini mendokumentasikan praktik implementasi perulangan, khususnya pola *repeat-until*, dan manipulasi data dasar menggunakan bahasa pemrograman Go.

3. tugas 3

Source code

```
package main

import "fmt"

func main() {
    var target int
    fmt.Scan(&target)
    total := 0
    donatur := 1
    for total < target {
        var donasi int
        fmt.Scan(&donasi)
        total += donasi
        fmt.Printf ("Donatur %d: menyumbang %d. Total terkumpul: %d\n", donatur, donasi, total)
        donatur++
    }
    fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n", total, donatur-1)
}
```

Screenshot program

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The top bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The Explorer sidebar shows a folder named 'LAPRAK MOD 13' containing several Go files: guided1.go, guided2.go, guided3.go, tempCodeRunne..., tugas1.go, tugas2.go, and tugas3.go. The main editor tab is 'tugas3.go' (line 17). The code in tugas3.go is:

```
func main() {
    var target int
    fmt.Scan(&target)
    total := 0
    donatur := 1
    for total < target {
        var donasi int
        fmt.Scan(&donasi)
        total += donasi
        fmt.Printf ("Donatur %d: menyumbang %d. Total terkumpul: %d\n", donatur, donasi, total)
        donatur++
    }
    fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n", total, donatur-1)
}
```

The terminal tab shows the output of running the code:

```
PS C:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13> go run "c:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13\tempCodeR
unnerFile.go"
300
100
Donatur 1: menyumbang 100. Total terkumpul: 100
50
Donatur 2: menyumbang 50. Total terkumpul: 150
200
Donatur 3: menyumbang 200. Total terkumpul: 350
Target tercapai! Total donasi: 350 dari 3 donatur.
PS C:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13> go run "c:\Users\tetuk\OneDrive\文档\semester 1\LAPRAK\laprak mod 13\tempCodeR
unnerFile.go"
500
150
Donatur 1: menyumbang 150. Total terkumpul: 150
```

A floating AI assistant window titled 'Ask about your code' is visible on the right side of the interface.

Deskripsi program

Berdasarkan kumpulan *screenshot* kode Go dan hasil eksekusinya, program-program ini mengilustrasikan berbagai penggunaan perulangan. Guided 1 dan Guided 2 menunjukkan implementasi pola *repeat-until* menggunakan perulangan *for* di Go; Guided 1 mengulang pencetakan kata, sementara Guided 2 mengulang permintaan input hingga mendapatkan bilangan bulat positif. Guided 3 menggunakan perulangan untuk melakukan pengurangan berulang ($x = x - y$) dan berhenti ketika $y \leq 0$, yang pada dasarnya menguji sisa pembagian. Untuk bagian Tugas, Tugas 1 menghitung jumlah digit dengan mengubah bilangan menjadi *string* dan mengukur panjangnya, Tugas 2 mencetak deretan angka *float* dengan kenaikan 0.1 dari $x + 0.1$ hingga batas atas, dan Tugas 3 mensimulasikan pengumpulan donasi berulang hingga total mencapai atau melebihi target. Secara keseluruhan, semua kode ini berfokus pada kontrol alur dan manipulasi data dasar menggunakan perulangan *for* di Go.