

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**MODUL 13**  
**REPEAT-UNTIL**



**Disusun oleh:**  
**Ichsan Maulana Muhammad**  
**109082500093**  
**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

The screenshot shows a Visual Studio Code interface with the following details:

- EXPLORER:** Shows a tree view of files and folders, including 'MODUL 3', 'MODUL 4', 'MODUL 5 DAN 6', 'MODUL 7', 'MODUL 9', 'MODUL 10', 'MODUL 11', 'MODUL 12', and 'MODUL 13'. Under 'MODUL 13', there are several Go files: 'ALPRO TEORI 1.go', 'ALPRO TEORI 2.go', 'ALPRO TEORI 3.go', 'ALPRO TEORI 4.go', 'ALPRO TEORI 5.go', 'Guided 1.go', 'Guided 2.go', 'Guided 3.go', 'pemanasan1.go', 'pemanasan2.go', 'pemanasan3.go', 'pemanasan4.go', 'pemanasan5.go', and 'pemanasan6.go'. There is also a file named 'Bantulukulan 5.go'.
- CODE EDITOR:** Displays the content of 'guided 1.go'. The code defines a package 'main' that imports 'fmt'. It contains a main function that reads a word and a repetition count from standard input, then prints the word that many times using a for loop. The code is as follows:

```
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 }
```

- OUTPUT:** Shows the terminal output of running the program. The user runs 'go run "c:\Users\ASUS\yy\MODUL 13\guided 1.go"'. The program outputs the word 'kursi' five times.
- RIGHT SIDE:** A separate window titled 'logUp' displays student information: NAMA: ICHSAN MAULANA M, NIM: 109082500093, and KELAS: IF-13-04.

## Deskripsi program

1. Program diawali dengan deklarasi package main dan import "fmt" yang digunakan untuk menangani proses input dan output. Seluruh proses program dijalankan di dalam fungsi main().
2. Di dalam fungsi main(), dideklarasikan dua variabel:
  - word bertipe string untuk menyimpan kata yang diinputkan oleh pengguna.
  - repetitions bertipe int untuk menyimpan jumlah pengulangan yang diinginkan pengguna.
3. Program membaca dua buah input dari pengguna menggunakan `fmt.Scan(&word, &repetitions)`, yaitu sebuah kata dan sebuah bilangan bulat yang menunjukkan jumlah pengulangan.
4. Program mendeklarasikan variabel counter bertipe int dan menginisialisasinya dengan nilai 0, yang berfungsi untuk menghitung jumlah pengulangan yang telah dilakukan.
5. Program menggunakan perulangan for dengan variabel kontrol done bertipe bool yang awalnya bernilai false, sehingga perulangan akan terus berjalan selama done bernilai false.
6. Di dalam perulangan, program mencetak kata yang diinputkan menggunakan `fmt.Println(word)` dan kemudian menambah nilai counter sebanyak satu setiap kali perulangan dijalankan.
7. Setelah itu, program memeriksa apakah nilai counter sudah mencapai atau melebihi jumlah pengulangan yang diminta (`counter >= repetitions`). Jika kondisi tersebut terpenuhi, maka done diubah menjadi true dan perulangan berhenti.

8. Output program berupa kata yang diinputkan pengguna, yang dicetak ke layar sebanyak jumlah pengulangan yang telah 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 a Go development environment with the following details:

- Code Editor:** The main window displays the file `int.go` containing the following Go 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)
}
```
- Terminal:** The bottom terminal window shows the command `go run "c:\Users\ASUS\yy\MODUL 13\guided 2.go"` being run, followed by the output:

```
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 13\guided 2.go"
17
17 adalah bilangan bulat positif
PS C:\Users\ASUS\yy>
```
- Log:** A separate window titled "logUp" shows the user's information:

```
NAMA: ICHSAN MAULANA M
NIM: 109082500093
KELAS: IF-13-04
```

## Deskripsi program

1. Program diawali dengan deklarasi package main dan import "fmt" untuk menggunakan fungsi input dan output. Seluruh proses dijalankan di dalam fungsi main() sebagai fungsi utama program.
2. Di dalam fungsi main(), dideklarasikan dua variabel:
  - number bertipe int untuk menyimpan bilangan yang diinputkan oleh pengguna.
  - continueLoop bertipe bool yang berfungsi sebagai pengontrol perulangan.
3. Program menggunakan perulangan for dengan kondisi continueLoop bernilai true, sehingga program akan terus meminta input selama kondisi tersebut terpenuhi.
4. Di dalam perulangan, program membaca masukan bilangan bulat dari pengguna menggunakan `fmt.Scan(&number)`.
5. Setelah menerima input, nilai continueLoop diubah berdasarkan kondisi `number <= 0`:
  - Jika bilangan kurang dari atau sama dengan 0, maka continueLoop bernilai true dan perulangan akan terus berjalan.
  - Jika bilangan lebih besar dari 0, maka continueLoop bernilai false dan perulangan berhenti.
6. Dengan mekanisme ini, program memastikan bahwa input yang diterima adalah bilangan bulat positif sebelum keluar dari perulangan.

7. Setelah perulangan selesai, program menampilkan hasil menggunakan `fmt.Printf`, yang menyatakan bahwa bilangan yang dimasukkan adalah bilangan bulat positif.
8. Output program berupa satu baris teks dengan format:  
`n` adalah bilangan bulat positif,  
di mana `n` adalah bilangan bulat positif yang berhasil dimasukkan oleh pengguna.

### 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

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows a tree view of files and folders, including "MODUL 13", "guided 3.go", and several "ALPRO TEORI" and "guided" files.
- Editor:** Displays a Go program named "in1.go". The code defines a package "main" with a "main()" function that reads two integers from the user, calculates their sum, and prints it. The code is annotated with line numbers (1-17).
- Terminal:** Shows the command "go run" being executed in the directory "c:\Users\ASUS\yy\MODUL 13". The output of the program is displayed, showing the sum of 3 and 9 as 12.
- Status Bar:** Shows the file path "c:\Users\ASUS\yy\in1.go", the current line (Ln 3), column (Col 16), character count (56), and encoding (UTF-16 LE).

## Deskripsi program

1. Program diawali dengan deklarasi package main dan import "fmt" yang digunakan untuk menangani proses input dan output. Seluruh proses dijalankan di dalam fungsi main().
  2. Di dalam fungsi main(), dideklarasikan tiga variabel:
    - x bertipe int untuk menyimpan bilangan awal.
    - y bertipe int untuk menyimpan bilangan pengurang.
    - selesai bertipe bool yang digunakan sebagai pengontrol perulangan.
  3. Program membaca dua buah input bilangan bulat dari pengguna menggunakan fmt.Scan(&x, &y).
  4. Program menggunakan perulangan for dengan kondisi !selesai, sehingga perulangan akan terus berjalan selama variabel selesai bernilai false.
  5. Di dalam perulangan, nilai x dikurangi dengan y ( $x = x - y$ ) dan hasil pengurangan tersebut langsung dicetak ke layar menggunakan fmt.Println(x).
  6. Setelah mencetak nilai x, program memeriksa kondisi  $x \leq 0$ . Jika kondisi ini terpenuhi, maka variabel selesai diubah menjadi true sehingga perulangan berhenti.
  7. Setelah perulangan selesai, program mencetak hasil pengecekan kondisi  $x == 0$  menggunakan fmt.Println( $x == 0$ ).
  8. Output program berupa deretan hasil pengurangan x dengan y yang ditampilkan setiap iterasi, diakhiri dengan nilai boolean true atau false yang menunjukkan apakah nilai akhir x sama dengan nol.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var n, digit int
    var selesai bool

    fmt.Scan(&n)
    digit = 0

    for selesai = false; !selesai; {
        n = n / 10
        digit++
        selesai = n == 0
    }

    fmt.Println(digit)
}
```

## Screenshot program

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

- File Explorer:** Shows files like `main.go`, `tugas 1.go`, and `ALPRO TEORI 1.go`.
- Code Editor:** Displays the `main.go` file with the following code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n, digit int
7     var selesai bool
8
9     fmt.Scan(&n)
10    digit = 0
11
12    for selesai = false; !selesai; {
13        n = n / 10
14        digit++
15        selesai = n == 0
16    }
17
18    fmt.Println(digit)
19}
20
```
- Terminal:** Shows command-line history for running the program.
- Output:** Displays the student's name, NIM, and class information.

## **Deskripsi program**

1. Program diawali dengan deklarasi package main dan import "fmt" untuk menangani proses input dan output. Seluruh proses dijalankan di dalam fungsi main().
  2. Di dalam fungsi main(), dideklarasikan tiga variabel
    - n bertipe int untuk menyimpan bilangan bulat positif yang diinputkan.
    - digit bertipe int untuk menyimpan jumlah digit dari bilangan tersebut.
    - selesai bertipe bool sebagai kondisi berhenti pada perulangan repeat-until.
  3. Program membaca satu buah bilangan bulat positif dari pengguna menggunakan fmt.Scan(&n).
  4. Variabel digit diinisialisasi dengan nilai 0 sebelum perulangan dimulai.
  5. Program menggunakan perulangan for dengan kondisi !selesai sebagai bentuk implementasi repeat-until, sehingga proses perhitungan dijalankan minimal satu kali.
  6. Di dalam perulangan, bilangan n dibagi dengan 10 dan nilai digit ditambah satu untuk setiap pembagian yang dilakukan.
  7. Setelah aksi dijalankan, program memeriksa kondisi berhenti, yaitu  $n == 0$ . Jika kondisi ini terpenuhi, maka variabel selesai bernilai true dan perulangan dihentikan.
  8. Setelah perulangan selesai, program menampilkan nilai digit sebagai banyaknya digit dari bilangan yang diinputkan.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var bil float64
    var nilai, batas int
    var selesai bool

    fmt.Scan(&bil)
    nilai = int(bil * 10)
    batas = (int(bil) + 1) * 10

    for selesai = false; !selesai; {
        nilai = nilai + 1
        selesai = nilai > batas
        if !selesai {
            fmt.Printf("%.1f\n", float64(nilai)/10)
        }
    }
}
```

## Screenshot program

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

- EXPLORER:** Shows project structure with files like MODUL 3, MODUL 4, MODUL 5 DAN 6, MODUL 7, MODUL 9, MODUL 10, MODUL 11, MODUL 12, MODUL 13, guided 1.go, guided 2.go, guided 3.go, tugas 1.go, and tugas 2.go.
- CODE EDITOR:** Displays the content of `tugas 2.go` with the following code:

```
package main
import "fmt"

func main() {
    var bil float64
    var nilai, batas int
    var selesai bool

    fmt.Scan(&bil)
    nilai = int(bil * 10)
    batas = (int(bil) + 1) * 10

    for selesai = false; !selesai; {
        nilai = nilai + 1
        selesai = nilai > batas
        if !selesai {
            fmt.Printf("%.1f\n", float64(nilai)/10)
        }
    }
}
```
- TERMINAL:** Shows command-line output:

```
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 13\tugas 2.go"
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
```

```
PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 13\tugas 2.go"
2.7
2.8
```
- LOG:** A separate window titled "logUp" displays:

```
NAMA: ICHSAN MAULANA M
NIM: 109082500093
KELAS: IF-13-04
```

## Deskripsi program

1. Program diawali dengan deklarasi package main dan import "fmt" untuk proses input dan output, dengan fungsi main() sebagai fungsi utama.
2. Di dalam fungsi main(), dideklarasikan lima variabel:
  - bil bertipe float64 untuk menyimpan bilangan desimal yang diinputkan pengguna.
  - nilai bertipe int untuk menyimpan nilai bilangan dalam bentuk skala per sepuluh.
  - batas bertipe int untuk menyimpan nilai pembulatan ke atas sebagai kondisi berhenti.
  - selesai bertipe bool sebagai penanda berhentinya perulangan repeat-until.
3. Program membaca satu buah bilangan desimal dari pengguna menggunakan `fmt.Scan(&bil)`.
4. Nilai nilai diinisialisasi dengan hasil konversi input  $\times 10$  ke bilangan bulat, sedangkan batas ditentukan dari hasil pembulatan ke atas bilangan input yang dikalikan 10.
5. Program menggunakan perulangan for dengan kondisi !selesai sebagai implementasi repeat-until, sehingga proses penjumlahan pasti dijalankan minimal satu kali.
6. Di dalam perulangan, nilai nilai ditambah satu pada setiap iterasi, yang merepresentasikan penambahan sebesar 0.1 pada bilangan desimal.
7. Setelah penambahan dilakukan, program memeriksa kondisi berhenti, yaitu ketika nilai telah melebihi batas. Selama kondisi tersebut belum terpenuhi, nilai hasil penjumlahan ditampilkan ke layar.

8. Program berhenti ketika nilai pembulatan ke atas telah tercapai, dan seluruh hasil penjumlahan ditampilkan secara berurutan sesuai dengan ketentuan soal.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

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

    fmt.Scan(&target)

    total = 0
    donatur = 0

    for selesai = false; !selesai; {
        fmt.Scan(&donasi)

        donatur++
        total = total + donasi

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

        selesai = total >= target
    }

    fmt.Printf("Target tercapai! Total donasi: %d dari %d
donatur.\n", total, donatur,
}
```

## Screenshot program

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

- EXPLORER:** Shows project structure with files like `MODUL 13`, `tugas 3.go`, `main.go`, etc.
- CODE EDITOR:** Displays the `main.go` file content:1 package main
2
3 import "fmt"
4
5 func main() {
6 var target, donasi, total, donatur int
7 var selesai bool
8
9 fmt.Scan(&target)
10
11 total = 0
12 donatur = 0
13
14 for selesai = false; !selesai; {
15 fmt.Scan(&donasi)
16
17 donatur++
18 total = total + donasi
19
20 fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n", donatur, donasi, total)
21
22 selesai = total >= target
23 }
24
25 fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n", total, donatur)
26 }
- TERMINAL:** Shows command-line output of the program running:PS C:\Users\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 13\tugas 3.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\ASUS\yy> go run "c:\Users\ASUS\yy\MODUL 13\tugas 3.go"
500
150
Donatur 1: Menyumbang 150. Total terkumpul: 150
- LOG:** Shows a log file with student information:NAMA: ICHSAN MAULANA M
NIM: 109082500093
KELAS: IF-13-04

## Deskripsi program

1. Program diawali dengan deklarasi `package main` dan `import "fmt"` yang digunakan untuk proses input dan output, dengan fungsi `main()` sebagai fungsi utama program.
2. Di dalam fungsi `main()`, dideklarasikan beberapa variabel, yaitu:
  - `target` bertipe `int` untuk menyimpan target donasi yang harus dicapai.
  - `donasi` bertipe `int` untuk menyimpan jumlah donasi dari setiap donatur.
  - `total` bertipe `int` untuk menyimpan total akumulasi donasi.
  - `donatur` bertipe `int` untuk menyimpan jumlah donatur yang telah menyumbang.
  - `selesai` bertipe `bool` sebagai penanda berhentinya perulangan `repeat-until`.
3. Program membaca satu buah input bilangan bulat dari pengguna menggunakan `fmt.Scan(&target)` sebagai target donasi yang harus dicapai.
4. Variabel `total` dan `donatur` diinisialisasi dengan nilai awal 0 sebelum perulangan dimulai.
5. Program menggunakan struktur perulangan `for` dengan kondisi `!selesai` sebagai implementasi `repeat-until`, sehingga proses perulangan pasti dijalankan minimal satu kali.
6. Di dalam perulangan, program membaca input donasi dari pengguna menggunakan `fmt.Scan(&donasi)`, kemudian nilai donasi tersebut ditambahkan ke variabel `total`, dan jumlah donatur ditambah satu setiap iterasi.

7. Setelah proses penambahan dilakukan, program menampilkan informasi donatur ke layar yang berisi nomor donatur, jumlah donasi yang diberikan, serta total donasi yang telah terkumpul.
8. Kondisi berhenti perulangan diperiksa, yaitu ketika total donasi telah mencapai atau melebihi target ( $\text{total} \geq \text{target}$ ). Jika kondisi ini terpenuhi, maka variabel selesai bernilai true dan perulangan dihentikan.
9. Setelah perulangan selesai, program menampilkan pesan akhir yang menyatakan bahwa target donasi telah tercapai, beserta total donasi yang terkumpul dan jumlah donatur.