

**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 13**

**REPEAT-UNTIL**



**Disusun oleh:**

**NAUFAL BINTANG PRATAMA**

**109082500096**

**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 word string
    var repetitions int

    fmt.Scan(&word, &repetitions)

    counter := 0

    for {
        fmt.Println(word)
        counter++

        if counter >= repetitions {
            break
        }
    }
}
```

#### Screenshot program

The screenshot shows a Microsoft Visual Studio Code interface. On the left, there's a code editor with a file named 'guide1mod13.go' containing Go code. The code defines a package 'main' that reads a word and a repetition count from standard input, then prints the word repeatedly until the counter reaches or exceeds the repetition count. A terminal window on the right shows the execution of the program, where it takes 'pagi' and '3' as input, then prints 'pagi' three times. Another run shows 'kursi' and '5', resulting in five 'kursi' prints.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var word string
7     var repetitions int
8
9     fmt.Scan(&word, &repetitions)
10
11     counter := 0
12
13     for {
14         fmt.Println(word)
15         counter++
16
17         if counter >= repetitions {
18             break
19         }
20     }
21 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\go\modul13> go run .\guide1mod13\guide1mod13.go  
pagi 3  
pagi  
pagi  
pagi  
PS D:\go\modul13> go run .\guide1mod13\guide1mod13.go  
kursi 5  
kursi  
kursi  
kursi  
kursi

## Deskripsi program

Program ini digunakan untuk menampilkan sebuah kata atau teks ke layar secara berulang sebanyak jumlah yang ditentukan oleh pengguna.

Program menerima dua input dari pengguna, yaitu sebuah kata (string) dan jumlah pengulangan (integer). Setelah itu, program menggunakan perulangan tak hingga (for) yang dikendalikan oleh sebuah counter untuk mencetak kata tersebut ke layar. Proses pengulangan akan berhenti ketika jumlah cetakan sudah mencapai jumlah pengulangan yang dimasukkan oleh pengguna.

Singkatnya, program ini digunakan untuk mencetak sebuah kata berulang kali sesuai jumlah yang diinginkan oleh pengguna

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
```

```

var n int

for {
    fmt.Scan(&n)

    if n > 0 {
        break
    }
}

fmt.Printf("%d adalah bilangan bulat positif\n", n)
}

```

### Screenshot program

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

- Code Editor:** The main editor window displays the file `guide2mod.go` with the following content:

```

package main
import "fmt"
func main() {
    var n int
    for {
        fmt.Scan(&n)
        if n > 0 {
            break
        }
    }
    fmt.Printf("%d adalah bilangan bulat positif\n", n)
}

```

- Terminal:** The terminal window shows the command `go run .\Guide2mod13\guide2mod.go` being run, with the following output:
  - 5
  - 2
  - 1
  - 0
  - 5
  - 5 adalah bilangan bulat positif
- Modal Window:** A small modal window titled "NA" is open, displaying the student's information:
  - NAUFAL BINTANG PRATAMA
  - S11F-13-07
  - 109082500096
- Bottom Bar:** The taskbar shows various system icons and the date/time: 10/12/2025, 10:39.

### Deskripsi program

Program ini digunakan untuk melakukan proses validasi input berupa bilangan bulat yang dimasukkan oleh pengguna. Pada awal program, pengguna diminta untuk memasukkan sebuah bilangan bulat melalui input standar. Program kemudian menggunakan struktur perulangan untuk terus membaca dan memeriksa nilai yang dimasukkan. Apabila bilangan yang diberikan bernilai nol atau negatif, maka program akan mengulang proses input hingga diperoleh bilangan yang memenuhi

**syarat. Setelah pengguna memasukkan bilangan bulat yang bernilai positif, perulangan akan dihentikan. Selanjutnya, program menampilkan output berupa pesan yang menyatakan bahwa bilangan tersebut merupakan bilangan bulat positif.**

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var x, y int
    isKelipatan := true

    fmt.Scan(&x, &y)

    for {
        x -= y
        fmt.Println(x)

        if x < 0 {
            isKelipatan = false
            break
        }

        if x == 0 {
            break
        }
    }

    fmt.Println(isKelipatan)
}
```

#### Screenshot program

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, Run, etc. The title bar says "modul13". The left sidebar has icons for Welcome, guide1mod13.go, guide2mod.go, and guide3mod13.go (which is currently active). The main editor area contains the following Go code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x, y int
7     isKelipatan := true
8
9     fmt.Scan(&x, &y)
10
11    for {
12        x -= y
13        fmt.Println(x)
14
15        if x < 0 {
16            isKelipatan = false
17            break
18        }
19
20        if x == 0 {
21            break
22    }
23    }
24}
25
```

Below the editor, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing command-line output:

```
-1
false
PS D:\go\modul13> go run .\Guide3mod13\guide3mod13.go
15 3
12
9
6
3
0
true
PS D:\go\modul13>
```

The status bar at the bottom right shows Ln 25, Col 29, Tab Size: 4, UTF-8, CRLF, 1.25.1, Prettier, 16:32, and 16/12/2025.

### Deskripsi program

Program ini digunakan untuk menentukan apakah suatu bilangan merupakan kelipatan dari bilangan lainnya. Program menerima dua buah input bilangan bulat, yaitu x dan y. Selanjutnya, program melakukan proses pengurangan nilai x dengan y secara berulang menggunakan perulangan for. Setiap hasil pengurangan akan ditampilkan ke layar untuk memperlihatkan proses perhitungannya.

Apabila hasil pengurangan menghasilkan nilai negatif, maka program menyimpulkan bahwa x bukan merupakan kelipatan dari y dan variabel penanda akan diubah menjadi false. Sebaliknya, jika hasil pengurangan mencapai nilai nol, maka program menyimpulkan bahwa x adalah kelipatan dari y. Setelah perulangan selesai, program menampilkan hasil akhir berupa nilai boolean yang menunjukkan apakah bilangan tersebut merupakan kelipatan atau bukan.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
```

```

var n int

var digit int


fmt.Scan(&n)

for n != 0 {
    n = n / 10
    digit++
}

fmt.Println(digit)
}

```

## Screenshot program

The screenshot shows a Visual Studio Code interface for a Go project named 'modul13'. The left sidebar lists files: 'Welcome', 'guide1mod13.go', 'guide2mod.go', 'guide3mod13.go', 'tugas1mod13.go' (the active file), 'tugas2mod13.go', and 'tugas3mod13.go'. The main editor area contains the provided Go code. Below the editor are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab shows command-line output:

```

PS D:\go\modul13> go run .\Tugas1mod13\tugas1mod13.go
5
1
PS D:\go\modul13> go run .\Tugas1mod13\tugas1mod13.go
234
3
PS D:\go\modul13>

```

A floating terminal window displays the student's information:

```

NAUFAL BINTANG PRATAMA
S11F-13-07
109082500096

```

The bottom status bar shows the file path 'Ln 15, Col 1', character count '46 characters', mode 'Plain text', zoom '100%', and encoding 'UTF-8'. The system tray at the bottom includes icons for file, search, and various system services.

## Deskripsi program

Program ini digunakan untuk menghitung jumlah digit dari sebuah bilangan bulat yang dimasukkan oleh pengguna. Pada awal program, pengguna diminta memasukkan satu buah bilangan bulat melalui input standar. Setelah itu, program melakukan proses pengolahan data dengan menggunakan struktur perulangan. Di dalam perulangan tersebut, bilangan akan dibagi dengan 10 secara berulang untuk menghilangkan digit terakhir dari bilangan tersebut. Setiap kali proses pembagian dilakukan, program menghitung satu digit yang ada pada bilangan. Proses ini akan

terus dilakukan hingga nilai bilangan menjadi nol. Setelah perulangan selesai, program menampilkan output berupa jumlah digit dari bilangan bulat yang telah dimasukkan oleh pengguna.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

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

    start := int(x * 10)
    end := (start/10 + 1) * 10

    for i := start + 1; i <= end; i++ {
        if i == end {
            fmt.Println(i / 10)
        } else {
            fmt.Printf("%.1f\n", float64(i)/10)
        }
    }
}
```

### Screenshoot program

The screenshot shows a Windows desktop environment. In the center is a terminal window titled 'modul13' with the command 'PS D:\go\modul13>'. It displays the output of a Go program: 'NAUFAL BINTANG PRATAMA', 'SIIF-13-07', and '109082500096'. At the bottom of the terminal window, status information is shown: 'Ln 3, Col 13 | 46 characters | Plain text | 100% | Wind | UTF-8'. Below the terminal is a taskbar with various icons. To the left of the terminal is a code editor window for VS Code. The code editor has tabs for 'Welcome', 'guide1mod13.go', 'guide2mod.go', 'guide3mod13.go', 'tugas1mod13.go', and 'tugas2mod13.go'. The active file is 'tugas2mod13.go'. The code itself is a Go program that reads a float64 input from the user, multiplies it by 10, and then prints the result as a decimal number from 0.7 to 2.9. The code editor interface includes standard VS Code icons like file operations, search, and preview.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x float64
7     fmt.Scan(&x)
8
9     start := int(x * 10)
10    end := (start/10 + 1) * 10
11
12    for i := start + 1; i <= end; i++ {
13        if i == end {
14            fmt.Println(i / 10)
15        } else {
16            fmt.Printf("%.1f\n", float64(i)/10)
17        }
18    }
19 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

0.7  
0.8  
0.9  
1  
PS D:\go\modul13>  
PS D:\go\modul13> go run .\Tugas2mod13\tugas2mod13.go  
2.7  
2.8  
2.9  
3  
PS D:\go\modul13>

### Deskripsi program

Program ini digunakan untuk menampilkan deretan bilangan desimal berdasarkan nilai input yang diberikan oleh pengguna. Program menerima satu input berupa bilangan pecahan bertipe float64. Nilai tersebut kemudian dikalikan dengan 10 dan diubah ke bentuk bilangan bulat untuk memudahkan proses perhitungan batas awal dan batas akhir.

Program menentukan batas akhir dengan membulatkan nilai input ke satuan berikutnya. Selanjutnya, program menggunakan perulangan for untuk menampilkan nilai desimal secara bertahap dengan kenaikan 0,1. Setiap nilai yang dihasilkan akan ditampilkan ke layar dengan satu angka di belakang koma. Pada nilai terakhir, program menampilkan hasil tanpa format desimal khusus. Program ini membantu memahami penggunaan perulangan, konversi tipe data, dan manipulasi bilangan desimal dalam bahasa Go.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

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

```

total := 0
donatur := 0

for total < target {
    var donasi int
    fmt.Scan(&donasi)

    donatur++
    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,
)
}

```

## Screenshot program

The screenshot shows a code editor interface with several tabs open, including 'tugas3mod13.go' which contains the provided Go code. Below the editor is a terminal window displaying the execution of the program and its output. The terminal output shows the program reading a target of 350, then iteratively adding donations of 50, 150, 100, and 50 to reach a total of 350, printing each step.

```

package main

import "fmt"

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

    total := 0
    donatur := 0

    for total < target {
        var donasi int
        fmt.Scan(&donasi)

        donatur++
        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,
    )
}

```

```

NAUFAL BINTANG PRATAMA
S11F-13-07
10908250096

Ln 3, Col 13 | 46 character Plain text 100% Wind UTF-8

```

## Deskripsi program

**Program ini digunakan untuk mensimulasikan proses pengumpulan donasi hingga mencapai target yang telah ditentukan. Pada awal program, pengguna diminta memasukkan nilai target donasi yang ingin dicapai. Selanjutnya, program**

menggunakan struktur perulangan untuk menerima input berupa jumlah donasi dari setiap donatur secara bertahap.

Setiap kali pengguna memasukkan nilai donasi, program akan menambahkan jumlah tersebut ke total donasi yang terkumpul dan menambah jumlah donatur. Program juga menampilkan informasi setiap donatur, termasuk nomor donatur, jumlah sumbangan yang diberikan, serta total donasi sementara. Proses ini akan terus berlangsung selama total donasi belum mencapai target. Setelah target tercapai, program menampilkan pesan bahwa target donasi telah terpenuhi beserta total donasi dan jumlah donatur yang berpartisipasi.