

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 13

REPEAT - UNTIL



Disusun oleh:

ELSA DWI RIZQIYANTI

109082500090

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 kata string
    var jumlah int

    fmt.Scan(&kata, &jumlah)

    counter := 0

    for {
        fmt.Println(kata)
        counter++

        if counter == jumlah {
            break
        }
    }
}
```

Screenshot program

The screenshot shows a Windows desktop environment with a Visual Studio Code window open. The code editor displays a file named 'guide1.go' with the following content:

```
guide1.go
package main
import "fmt"
func main() {
    var kata string
    var jumlah int
    fmt.Scan(&kata, &jumlah)
    counter := 0
    for {
        fmt.Println(kata)
        counter++
        if counter == jumlah {
            break
        }
    }
}
```

The terminal window below shows the command `go run guide1.go` being run, followed by the output of the program which prints the word 'pagi' 3 times and 'kursi' 5 times.

A small window titled 'Elsa Dwi Rizqiyanti' is also visible, showing her name and student ID.

Deskripsi program

Isi Perulangan

`fmt.Println(kata)`

`counter++`

`fmt.Println(kata)` mencetak isi variabel kata ke layar.

`counter++` menambah nilai counter sebanyak 1 setiap kali loop berjalan.

Kondisi Berhenti (Until)

`if counter == jumlah {`

`break`

`}`

Jika counter sudah sama dengan jumlah, maka:

`break` menghentikan perulangan.

Artinya, kata akan dicetak sebanyak jumlah kali.

Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var n int

    for {
        fmt.Scan(&n)

        if n > 0 {
            break
        }
    }

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

Screenshot program

The screenshot shows a Windows desktop environment with a Visual Studio Code window open. The code editor displays two files: `guide1.go` and `guide2.go`. The `guide2.go` file contains the following Go code:

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

The terminal window shows the command `go run guide2.go` being run, followed by the output: `-5 -2 -1 0 5` and `5 adalah bilangan bulat positif`.

Deskripsi program

Kondisi Berhenti (Until)

```
if n > 0 {
```

```
    break
```

```
}
```

Program memeriksa nilai n:

Jika $n > 0$ (bilangan bulat positif),

maka perulangan dihentikan dengan break.

Jika $n \leq 0$, perulangan berlanjut dan program meminta input ulang.

Output Hasil

```
fmt.Println(n, "adalah bilangan bulat positif")
```

Setelah keluar dari perulangan, program menampilkan pesan bahwa

nilai n adalah bilangan bulat positif.

Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Scan(&x, &y)

    sisa := x

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

        if sisa <= 0 {
            break
        }
    }

    if sisa == 0 {
        fmt.Println(true)
    } else {
        fmt.Println(false)
    }
}
```

Screenshot program

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

- EXPLORER:** Shows files in the current module, including `guide1.go`, `guide2.go`, and `guide3.go`.
- TERMINAL:** Displays the command `go run guide3.go` being run multiple times, showing the output of the program.
- OUTPUT:** Shows the standard output of the program, which prints the name "Elsa Dwi Rizqiyanti" and the ID "109082500090".
- DEBUG CONSOLE:** Shows the command `go run guide3.go` again.
- PROBLEMS:** Shows 4 errors.
- TERMINAL:** Shows the command `go run guide3.go` again.
- PORTS:** Shows port usage information.
- APPLICATION:** A window titled "Elsa Dwi R" displays the output "Elsa Dwi Rizqiyanti" and "109082500090".
- STATUS BAR:** Shows the file path "C:\praktikum alpro\modul 13", tab size, encoding, and date/time.

Deskripsi program

Inisialisasi Variabel sisa

`sisa := x`

Variabel sisa menyimpan nilai awal x.

Digunakan untuk menampung hasil pengurangan berulang.

Proses Pengurangan dan Output

`sisa = sisa - y`

`fmt.Println(sisa)`

Setiap iterasi, nilai sisa dikurangi y.

Hasil sementara langsung dicetak ke layar.

Kondisi Berhenti

`if sisa <= 0 {`

`break`

`}`

Perulangan dihentikan jika:

sisa kurang dari atau sama dengan 0

Artinya, pengurangan berhenti ketika nilai sudah habis atau negatif.

Pemeriksaan Hasil Akhir

`if sisa == 0 {`

```
        fmt.Println(true)
} else {
    fmt.Println(false)
}
Jika sisa == 0:
Program mencetak true
Artinya x habis dibagi y (pengurangan tepat)
Jika sisa ≠ 0:
Program mencetak false
Artinya x tidak habis dibagi y
```

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var n int

    for {
        fmt.Println("Masukkan bilangan bulat positif: ")

        fmt.Scan(&n)

        if n > 0 {
            break
        }

        fmt.Println("Input harus bilangan positif!")
    }

    count := 0

    tmp := n

    for tmp > 0 {
        tmp /= 10
        count++
    }

    fmt.Println("Jumlah angka pada bilangan", n, "adalah", count)
}
```

```

        count++

    }

    fmt.Printf("Banyaknya digit: %d\n", count)

}

```

Screenshot program

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

- Explorer View:** Shows files in the "MODUL 13" folder, including guide1.go, guide2.go, guide3.go, and nomer1.go.
- Code Editor:** Displays the content of nomer1.go. The code uses a for loop to read integers until a non-positive one is entered, then prints the count of digits.
- Terminal:** Shows the command "PS C:\praktikum\alpro\modul 13> go run nomer1.go" followed by several test runs where the user inputs different numbers and the program outputs their digit counts.
- Output Window:** Shows the output of the program's execution, including the user input and the calculated digit counts.
- Status Bar:** Provides information about the file path, line, column, tab size, encoding, and other system details.

Deskripsi program

Perulangan Validasi Input (Repeat–Until):

```

for {
    fmt.Print("Masukkan bilangan bulat positif: ")
    fmt.Scan(&n)

    if n > 0 {
        break
    }

    fmt.Println("Input harus bilangan positif!")
}

```

Program meminta input terus-menerus sampai pengguna memasukkan bilangan positif.

Jika input $n \leq 0$:

Program menampilkan pesan kesalahan

Mengulang input kembali

Jika $n > 0$:

Perulangan dihentikan dengan break

Inisialisasi Variabel Penghitung:

count := 0

tmp := n

count digunakan untuk menghitung jumlah digit.

tmp adalah variabel sementara agar nilai n tidak berubah

Perulangan Menghitung Digit:

```
for tmp > 0 {  
    tmp /= 10  
    count++  
}
```

Setiap iterasi:

tmp dibagi 10 (menghapus satu digit dari belakang)

count bertambah 1

Proses berhenti saat tmp menjadi 0

Jumlah iterasi = jumlah digit bilangan

Menampilkan Hasil:

```
fmt.Printf("Banyaknya digit: %d\n", count)
```

Program menampilkan jumlah digit dari bilangan yang dimasukkan.

2. Tugas 2

Source code

```
package main  
  
import "fmt"  
  
func main() {  
    var x float64  
    fmt.Scan(&x)  
  
    atas := int(x)  
    if float64(atas) != x {  
        atas++  
    }  
    fmt.Println(atas)  
}
```

```

}

angka := x

for {
    angka = angka + 0.1
    fmt.Printf("%.1f\n", angka)

    if angka >= float64(atas) {
        break
    }
}

}

```

Screenshot program

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

- File Explorer:** Shows files in the "MODUL 13" directory, including "guide 1.png", "guide 2.png", "guide 1.go" (4 lines), "guide2.go" (1 line), "guide3.go" (1 line), "guides 3.png", "nomer 1.png", "nomer1.go" (1 line), "nomer2.go" (11 lines), and "nomer3.go" (1 line).
- Code Editor:** Displays the content of "nomer2.go". The code defines a main function that reads a float64 value from standard input, converts it to an integer using `int(x)`, and then increments it by 1. It then prints the original float64 value followed by its integer part using `fmt.Sprintf("%.1f\n", angka)`. A loop adds 0.1 to the variable `angka` until it reaches or exceeds the ceiling value (`atas`). The code editor shows line numbers 5 through 25.
- Terminal:** Shows the command `go run nomer2.go` being run in the terminal. The output is a sequence of floating-point numbers starting from 0.2 up to 3.0, with each number followed by its integer part separated by a decimal point (e.g., 0.2, 0.3, ..., 3.0).
- Output Window:** Shows the output of the application window titled "Elsa Dwi Rizqiyanti". The window displays the text "Elsa Dwi Rizqiyanti" and the ID "109082500090".
- PowerShell Taskbar:** Shows multiple PowerShell windows open in the taskbar.

Deskripsi program

Menentukan Batas Atas (Ceiling):

```

atas := int(x)
if float64(atas) != x {
    atas++
}

```

Nilai `x` diubah menjadi bilangan bulat dengan `int(x)` (pembulatan ke bawah).

Jika `x` bukan bilangan bulat, maka:

atas ditambah 1

Dengan demikian, atas adalah pembulatan ke atas (ceiling) dari x.

Contoh:

$x = 3.0 \rightarrow \text{atas} = 3$

$x = 3.2 \rightarrow \text{atas} = 4$

Inisialisasi Variabel Perulangan:

angka := x

angka digunakan sebagai nilai awal perulangan.

Dimulai dari nilai input x.

Proses Penambahan dan Output:

angka = angka + 0.1

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

Setiap iterasi:

angka ditambah 0.1

Dicetak dengan 1 angka di belakang koma

Kondisi Berhenti:

```
if angka >= float64(atas) {
```

```
    break
```

```
}
```

Perulangan dihentikan jika:

nilai angka sudah mencapai atau melebihi batas atas

Ini merupakan bagian until dari repeat-until.

3. Tugas 3

Source code

```
package main

import "fmt"

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

    total := 0
    donatur := 0
```

```

for {

    var donasi int

    fmt.Scan(&donasi)

    donatur++

    total += donasi

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

    if total >= target {

        break
    }
}

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

```

Screenshot program

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

- File Explorer:** Shows a folder named "MODULE 13" containing several files: "guide1.go", "guide2.go", "guide3.go", "nomer1.go", "nomer2.go", and "nomer3.go".
- Code Editor:** Displays the content of the "nomer3.go" file. The code is identical to the one shown in the text block above.
- Terminal:** Shows the output of running the program with the command "go run nomer3.go". The output displays the total amount collected by each donor and the final total when the target is reached.

```

PS C:\praktikum alpro\modul 13> go run nomer3.go
300 100 50 200
Donatur 1: Menyumbang 300. Total terkumpul: 300
Donatur 2: Menyumbang 100. Total terkumpul: 400
Donatur 3: Menyumbang 50. Total terkumpul: 450
Donatur 4: Menyumbang 200. Total terkumpul: 350
Target tercapai! Total donasi: 350 dari 4 donatur.
PS C:\praktikum alpro\modul 13> go run nomer3.go
500 150 100 50 300
Donatur 1: Menyumbang 500. Total terkumpul: 500
Donatur 2: Menyumbang 150. Total terkumpul: 650
Donatur 3: Menyumbang 100. Total terkumpul: 750
Donatur 4: Menyumbang 50. Total terkumpul: 800
Donatur 5: Menyumbang 300. Total terkumpul: 1100
Target tercapai! Total donasi: 1100 dari 5 donatur.
PS C:\praktikum alpro\modul 13> go run nomer3.go
200 300
Donatur 1: Menyumbang 200. Total terkumpul: 200
Target tercapai! Total donasi: 200 dari 1 donatur.
PS C:\praktikum alpro\modul 13>

```

Deskripsi program

Proses Penjumlahan dan Output:

donatur++

total += donasi

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

Jumlah donatur bertambah satu setiap ada input donasi.

Nilai donasi ditambahkan ke total.

Program menampilkan:

nomor donatur

jumlah donasi

total donasi sementara

Kondisi Berhenti

```
if total >= target {
    break
}
```

Perulangan dihentikan jika total donasi sudah mencapai atau melebihi target.

Ini merupakan bagian until dari konsep repeat–until.

Output Akhir

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

Setelah perulangan berhenti, program menampilkan:

total donasi akhir

jumlah donatur