

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 13
REPEAT-UNTIL**



Disusun oleh:

Fasa Yuwan Rabbani

109082500081

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 done := false; !done; {
        fmt.Println(word)
        counter++
        done = (counter >= repetitions)
    }
}
```

Screenshoot program

The screenshot shows a Go IDE with several tabs. The active tab is 'Guide1.go'. The code in the editor 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 }
16
```

A small window titled 'NIM' is overlaid on the code, displaying the following information:

```
NIM : 109082500081
Kelas : S1F-13-07
Nama : Fasa Yuwan Rabbani
```

The terminal at the bottom shows the command `go run Guide1.go` being executed, and the output is:

```
pagi
3
pagi
pagi
pagi
PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13\Guide>
```

Deskripsi program

Program ini membaca sebuah kata dan angka dari pengguna, lalu mencetak kata tersebut sebanyak angka yang dimasukkan. Misalnya, jika pengguna mengetik “pagi” dan angka 3, maka kata “pagi” akan ditampilkan tiga kali. Program menggunakan perulangan dengan kondisi logis untuk memastikan jumlah cetakan sesuai dengan angka yang diminta. Cocok untuk latihan dasar input dan perulangan dalam bahasa Go.

2. Guided 2

Source Code

```
package main

import "fmt"

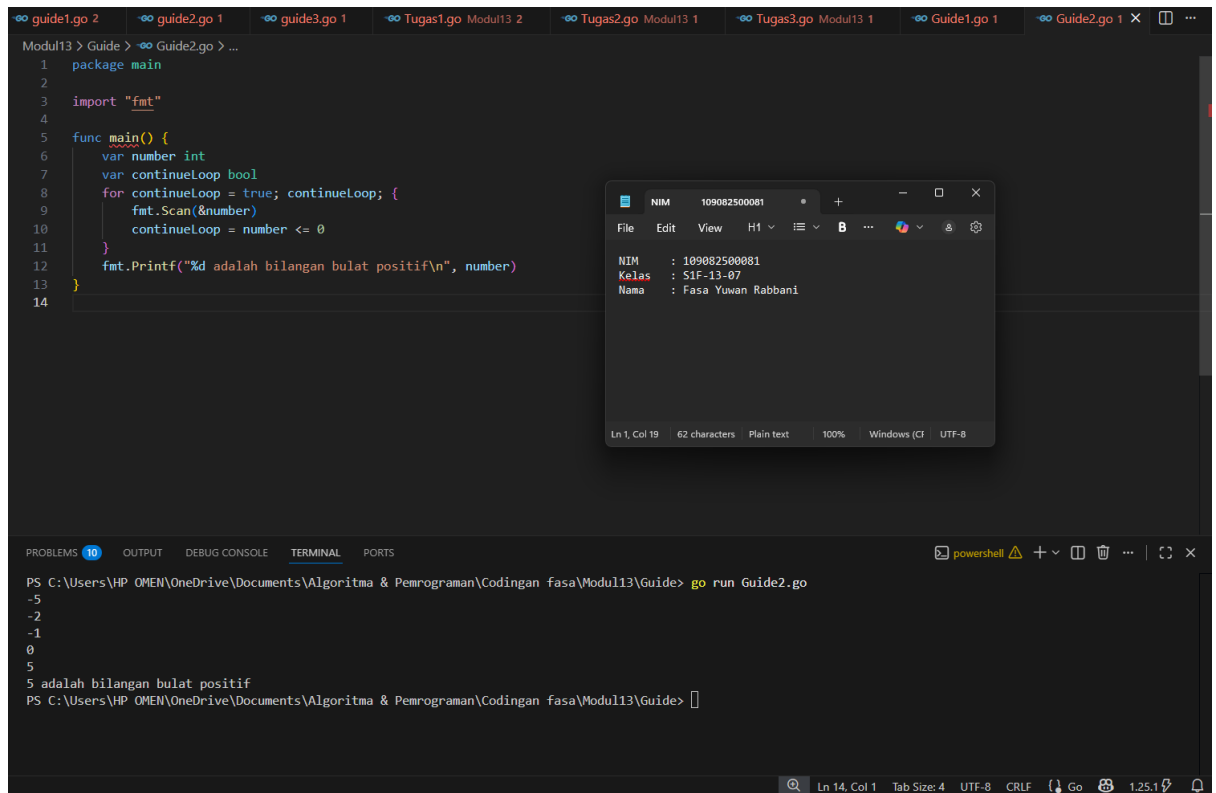
func main() {
    var number int
    var continueLoop bool
    for continueLoop = true; continueLoop; {
        fmt.Scan(&number)
        continueLoop = number <= 0
    }
}
```

```

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

```

Screenshoot program



Deskripsi program

Program ini bertujuan untuk memastikan pengguna memasukkan bilangan bulat positif. Kalau yang dimasukkan masih nol atau negatif, program akan terus meminta input sampai pengguna memberikan angka yang benar. Begitu angka positif dimasukkan, program langsung mencetak pesan konfirmasi bahwa angka tersebut adalah bilangan bulat positif. Cocok untuk latihan validasi input dan penggunaan perulangan dalam Go.

3. Guided 3

Source Code

```

package main

import "fmt"

func main() {
    var a int
}

```

```

        fmt.Print("Masukkan sebuah bilangan untuk di
        jadikan fibonanci : ")

        fmt.Scan(&a)

        n1, n2 := 0, 1

        fmt.Printf("Deret Fibonacci hingga %d adalah: \n",
        a)

        for n1 <= a {

            fmt.Printf("%d ", n1)

            n1, n2 = n2, n1+n2

        }

        fmt.Println()

    }

```

Screenshoot program

The screenshot shows a Go IDE with the following code in `Guide3.go`:

```

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

```

The terminal output shows the program execution:

```

PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13\Guide> go run Guide3.go
5
2
3
1
-1
false
PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13\Guide>

```

Deskripsi program

Program ini membaca dua angka dari pengguna, lalu mengurangi angka pertama dengan angka kedua secara berulang. Setiap hasil pengurangan ditampilkan satu per satu. Proses ini terus berlangsung sampai hasilnya nol atau negatif. Di akhir, program

mencetak true jika hasil akhirnya tepat nol, atau false jika hasilnya kurang dari nol. Program ini cocok untuk latihan logika perulangan dan kondisi dalam bahasa Go.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var a, digit int

    fmt.Print("Masukkan sebuah angka bilangan bulat : ")

    fmt.Scan(&a)

    for a > 0 {
        a = a / 10
        digit++
    }

    fmt.Printf("Jumlah digit dari bilangan tersebut adalah: %d\n",
digit)
}
```

Screenshoot program

```
Modul13 > Tug1.go > main
1 package main
2 import "fmt"
3
4 func main() {
5     var a, digit int
6     fmt.Print("Masukkan sebuah angka bilangan bulat : ")
7     fmt.Scan(&a)
8
9     for a > 0 {
10        a = a / 10
11        digit++
12    }
13    fmt.Printf("Jumlah digit dari bilangan tersebut adalah: %d\n", digit)
14 }
```

```
PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13> go run Tug1.go
Masukkan sebuah angka bilangan bulat : 5
Jumlah digit dari bilangan tersebut adalah: 1
PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13> go run Tug1.go
Masukkan sebuah angka bilangan bulat : 234
Jumlah digit dari bilangan tersebut adalah: 3
PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13>
```

Deskripsi program

Program ini ditulis dalam bahasa Go dan berfungsi untuk menghitung jumlah digit dari sebuah bilangan bulat positif yang dimasukkan oleh pengguna. Setelah menerima input melalui `fmt.Scan`, program menjalankan perulangan `for` yang terus membagi angka dengan 10 hingga nilainya menjadi nol. Setiap kali angka dibagi, variabel penghitung digit akan bertambah satu, mencerminkan jumlah digit yang telah dihapus. Hasil akhir ditampilkan menggunakan `fmt.Printf`, menunjukkan berapa banyak digit yang dimiliki angka tersebut. Program ini sederhana namun efektif dalam mendemonstrasikan penggunaan perulangan dan operasi aritmatika dasar dalam Go.

2. Tugas 2

Source code

```
package main

import (
    "fmt"
    "math"
)
```

```
func main() {  
    var a float64  
    fmt.Print("Masukkan sebuah angka desimal : ")  
    fmt.Scan(&a)  
  
    batas := math.Ceil(a)  
    saatini := int(a * 10)  
    nLanjutan := int(batas * 10)  
  
    for i := saatini; i <= nLanjutan; i++ {  
        fmt.Printf("%.1f ", float64(i)/10)  
    }  
    fmt.Println()  
  
}
```

Screenshoot program

The screenshot shows a Go IDE with the following components:

- Source Code (Tugas2.go):**

```
4  "fmt"
5  "math"
6  )
7
8  func main() {
9      var a float64
10     fmt.Print("Masukkan sebuah angka desimal : ")
11     fmt.Scan(&a)
12
13     batas := math.Ceil(a)
14     saatini := int(a * 10)
15     nLanjutan := int(batas * 10)
16
17     for i := saatini; i <= nLanjutan; i++ {
18         fmt.Printf("%.1f ", float64(i)/10)
19     }
20     fmt.Println()
21
22 }
23
24
```
- Terminal Output:**

```
PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13> go run Tugas2.go
Masukkan sebuah angka desimal : 0.2
0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13> go run Tugas2.go
Masukkan sebuah angka desimal : 2.7
2.7 2.8 2.9 3.0
PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13>
```

Deskripsi program

Program Go ini digunakan untuk menghasilkan deretan angka desimal yang dimulai dari nilai input hingga mencapai pembulatan ke atas dari nilai tersebut. Setelah pengguna memasukkan sebuah angka desimal, program menghitung batas atas menggunakan `math.Ceil`, lalu mengalikan angka input dan batas tersebut dengan 10 agar bisa diolah sebagai bilangan bulat. Dengan perulangan `for`, program mencetak setiap nilai dari titik awal hingga batas akhir, kemudian membagi kembali dengan 10 agar hasilnya kembali dalam format desimal satu angka di belakang koma. Hasil akhirnya adalah deretan angka yang meningkat secara bertahap hingga mencapai pembulatan ke atas dari input awal.

3. Tugas 3

Source code

```
package main
import "fmt"

func main() {
    var a int
    var b int
    fmt.Print("Masukkan Angka Pertama : ")
    fmt.Scan(&a)
    fmt.Print("Masukkan Angka Kedua : ")
    fmt.Scan(&b)
```

```

        if b <= 0 {
            fmt.Println("Bilangan b harus lebih dari 0.")
            return
        }
        hasil := 0
        total := b

        for total <= a {
            hasil++
            total += b
        }

        fmt.Println(hasil)
    }
}

```

Screenshoot program

The screenshot shows a Go program in VS Code and its execution output in the terminal. The program is located in `Tugas3.go` and implements a loop to collect donations until a target is reached.

```

4 func main() {
5     var a, t, d int
6     fmt.Print("Masukkan sebuah angka : ")
7     fmt.Scan(&a)
8
9     t = 0
10    d = 0
11
12    fmt.Println("Masukkan nominal untuk didonasikan : ")
13
14    for t <= a {
15        var donasi int
16        fmt.Scan(&donasi)
17        d += 1
18
19        t += donasi
20        fmt.Printf("Donatur %d: menyumbang : %d. Total Terkumpul : %d\n", t, donasi, t)
21    }
22    fmt.Printf("Target Tercapai! total donasi %d dari %d Donatur. \n", t, d)
23
24 }

```

The terminal output shows the program's execution with the following input and output:

```

PS C:\Users\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13> go run Tugas3.go
Masukkan sebuah angka : 300
Masukkan nominal untuk didonasikan :
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\HP OMEN\OneDrive\Documents\Algoritma & Pemrograman\Codingan fasa\Modul13>

```

Deskripsi program

Program ini dibuat untuk mengumpulkan donasi hingga mencapai target yang ditentukan oleh pengguna. Setelah pengguna memasukkan angka target, program akan meminta nominal donasi secara berulang dari setiap donatur. Setiap kali donasi dimasukkan, program mencatat jumlahnya, menambahkan ke total yang sudah terkumpul, dan mencetak informasi siapa donatur beberapa serta total donasi saat

itu. Proses ini terus berlangsung sampai total donasi melebihi atau sama dengan target. Di akhir, program menampilkan pesan bahwa target telah tercapai, lengkap dengan jumlah total donasi dan jumlah donatur yang berpartisipasi. Program ini cocok untuk simulasi penggalangan dana sederhana dan melatih logika perulangan serta input dinamis.