

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 12

Repeat-Until



Disusun oleh:

Ahmad Malik Arrayyan

109082500116

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

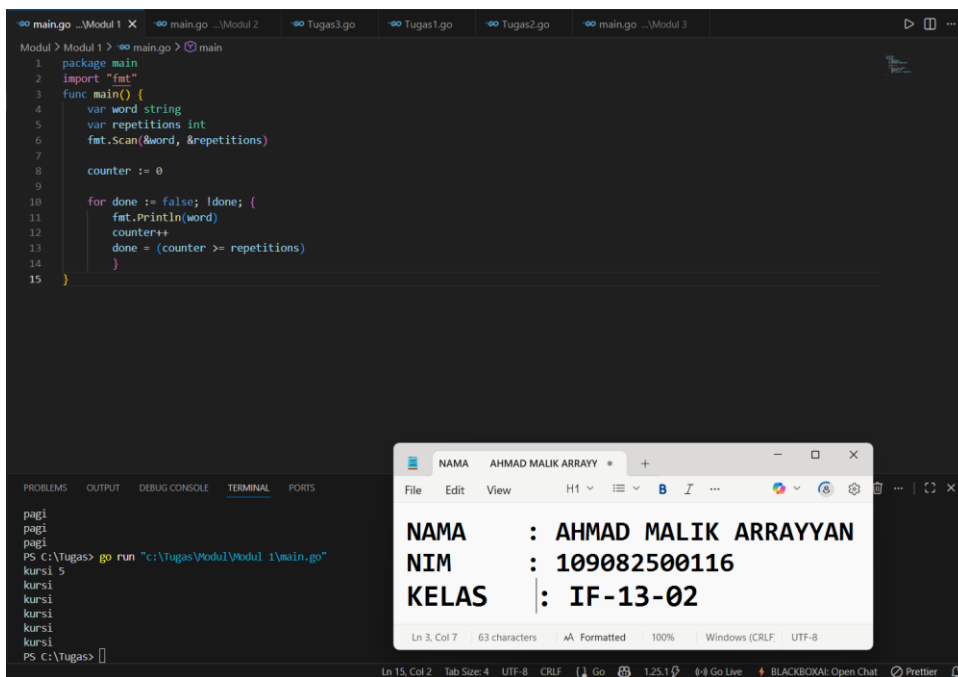
import "fmt"

func main() {
    var word string
    var repetitions int
    fmt.Scan(&word, &repetitions)

    counter := 0

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

Screenshoot program



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

```
PS C:\Tugas> go run "c:\Tugas\Modul 1\main.go"
pagi
pagi
pagi
kursi
kursi
kursi
kursi
kursi
PS C:\Tugas>
```

Deskripsi program

Program ini berfungsi untuk menerima sebuah kata dan jumlah pengulangan dari pengguna, lalu mencetak kata tersebut berulang kali sesuai jumlah yang diminta. Setelah input dibaca, program menggunakan sebuah penghitung untuk mencatat berapa kali kata sudah dicetak. Perulangan dibuat dengan bentuk *for* yang meniru konsep *repeat-until*, sehingga perintah mencetak kata selalu dijalankan terlebih dahulu sebelum dicek apakah harus berhenti. Setiap kali kata dicetak, penghitung bertambah satu, dan ketika jumlah cetakan sudah mencapai jumlah pengulangan yang diinginkan, perulangan dihentikan. Dengan cara ini, kata akan tampil tepat sesuai jumlah yang diminta pengguna.

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

The screenshot shows a Go program in a VS Code editor and its execution in a terminal. The program is located in `main.go` and defines a `main` function that uses a `for` loop to repeatedly scan for a positive integer. The terminal output shows the program running and printing the message "5 adalah bilangan bulat positif" twice, corresponding to the input values 5 and 17.

```

Modul > Modul 2 > main.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var number int
7     var continueLoop bool
8
9     for continueLoop = true; continueLoop; {
10         fmt.Scan(&number)
11         continueLoop = number <= 0
12     }
13
14     fmt.Printf("%d adalah bilangan bulat positif\n", number)
15 }

```

```

PS C:\Tugas> go run "c:\Tugas\Modul\Modul 2\main.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS C:\Tugas> go run "c:\Tugas\Modul\Modul 2\main.go"
17
17 adalah bilangan bulat positif
PS C:\Tugas>

```

Output window content:

```

NAMA      : AHMAD MALIK ARRAYYAN
NIM       : 109082500116
KELAS    : IF-13-02

```

Deskripsi program

Program ini digunakan untuk memastikan bahwa pengguna memasukkan **bilangan bulat positif**. Pada awalnya, program menyiapkan sebuah variabel untuk menampung angka yang dimasukkan serta sebuah variabel penanda untuk mengontrol perulangan. Program kemudian terus meminta input angka menggunakan perulangan for yang meniru konsep *repeat-until*. Selama angka yang dimasukkan masih kurang dari atau sama dengan nol, perulangan akan terus berjalan. Setelah pengguna memasukkan bilangan yang bernilai positif, perulangan dihentikan dan program menampilkan pesan bahwa angka tersebut adalah bilangan bulat positif.

3. Guided 3

Source Code

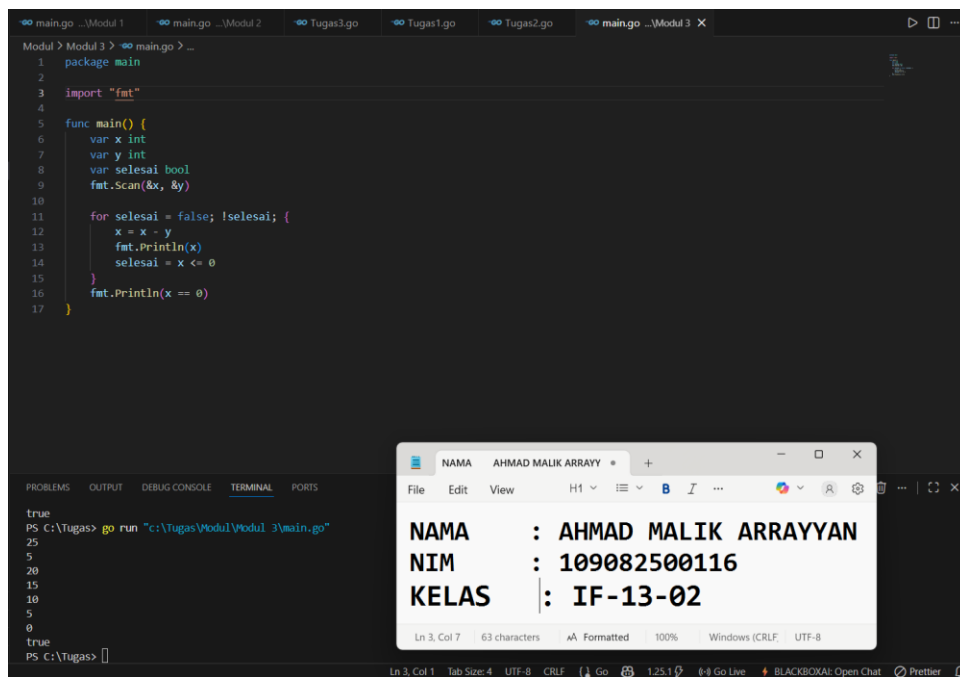
```
package main

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)
}
```

Screenshoot program



The screenshot shows a Go program in a code editor and its execution output in a terminal. The program is located in `Modul 3` and implements a loop that subtracts a value `y` from `x` until `x` is less than or equal to 0. The terminal shows the program being run with `25` and `5` as inputs, resulting in a series of values (20, 15, 10, 5, 0) and finally printing `true`.

```
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
11     for selesai = false; !selesai; {
12         x = x - y
13         fmt.Println(x)
14         selesai = x <= 0
15     }
16     fmt.Println(x == 0)
17 }
```

```
true
PS C:\Tugas> go run "c:\Tugas\Modul 3\main.go"
25
5
20
15
10
5
0
true
PS C:\Tugas>
```

Output window content:

```
NAMA      : AHMAD MALIK ARRAYYAN
NIM       : 109082500116
KELAS    : IF-13-02
```

Deskripsi program

Program ini sebenarnya melakukan simulasi pembagian dengan cara pengurangan berulang. Program meminta dua angka (dalam contoh ini 25 dan 5), lalu angka pertama (25) terus dikurangi dengan angka kedua (5) secara bertahap sampai nilainya habis atau menjadi negatif. Setiap hasil pengurangan (20, 15, 10, 5, 0) ditampilkan di layar, dan pada akhirnya program akan mencetak "true" karena angka tersebut habis dibagi (sisanya tepat nol).

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif: ")
    fmt.Scan(&n)

    count := 0

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

    fmt.Println(count)
}
```

Screenshoot program

The screenshot shows a Go program in a code editor and its execution output in a terminal. The program is a simple loop that divides a number by 10 until it reaches 0, counting the number of divisions. The output shows the program being run three times with different inputs: 234, 78787, and 1894256.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("Masukkan bilangan bulat positif: ")
8     fmt.Scan(&n)
9
10    count := 0
11
12    for n > 0 {
13        n = n / 10
14        count++
15    }
16
17    fmt.Println(count)
18 }
19
```

Terminal Output:

```
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 1\Tugas1.go"
Masukkan bilangan bulat positif: 234
3
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 1\Tugas1.go"
Masukkan bilangan bulat positif: 78787
5
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 1\Tugas1.go"
Masukkan bilangan bulat positif: 1894256
7
PS C:\Tugas>
```

Deskripsi program

Program menerima sebuah bilangan bulat positif sebagai input. Kemudian, program membagi angka tersebut dengan 10 secara berulang untuk menghilangkan satu digit dari belakang setiap kali perulangan berjalan. Setiap pembagian menambah satu hitungan sampai angkanya menjadi 0. Jumlah perulangan itulah yang menunjukkan banyaknya digit dari bilangan tersebut, lalu hasilnya ditampilkan.

2. Tugas 2

Source code

```
package main

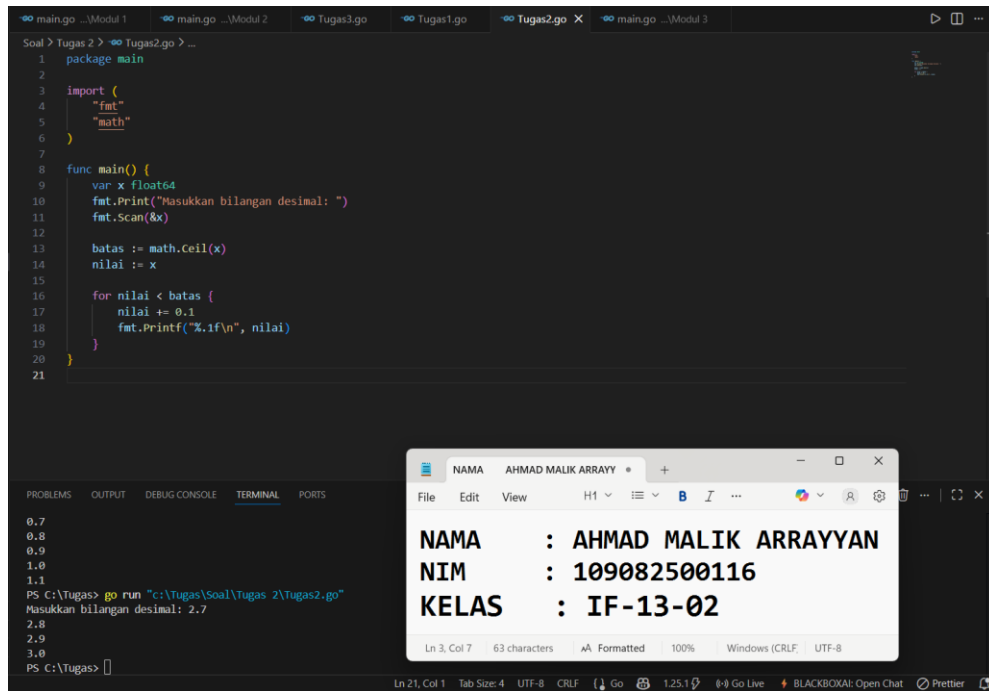
import (
    "fmt"
    "math"
)

func main() {
    var x float64
    fmt.Print("Masukkan bilangan desimal: ")
    fmt.Scan(&x)

    batas := math.Ceil(x)
    nilai := x

    for nilai < batas {
        nilai += 0.1
        fmt.Printf("%.1f\n", nilai)
    }
}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor and its execution output in a terminal window. The program, located in `Tugas2.go`, imports `fmt` and `math`. It defines a `main` function that reads a decimal number `x` from the user. It then calculates the ceiling of `x` using `math.Ceil(x)` and stores it in `batas`. A loop increments a variable `nilai` by 0.1 until it reaches `batas`, printing each value with the format `%.1f`. The terminal window shows the program being run, the input `2.7`, and the output values `0.7`, `0.8`, `0.9`, `1.0`, `1.1`, `2.8`, `2.9`, and `3.0`.

```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var x float64
10    fmt.Print("Masukkan bilangan desimal: ")
11    fmt.Scan(&x)
12
13    batas := math.Ceil(x)
14    nilai := x
15
16    for nilai < batas {
17        nilai += 0.1
18        fmt.Printf("%.1f\n", nilai)
19    }
20 }
21
```

Output:

```
0.7
0.8
0.9
1.0
1.1
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 2\Tugas2.go"
Masukkan bilangan desimal: 2.7
2.8
2.9
3.0
PS C:\Tugas>
```

Deskripsi program

Program membaca bilangan desimal, lalu menentukan batas akhirnya dengan pembulatan ke atas menggunakan `math.Ceil`. Nilai awal kemudian ditambah 0.1 setiap perulangan sampai mencapai batas tersebut. Untuk menghindari angka panjang seperti 0.30000004, output ditampilkan dengan format satu angka di belakang koma (`%.1f`), sehingga hasilnya tetap rapi tanpa perlu `math.Round`.

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,
    )
}
```

Screenshoot program

The screenshot shows a Go program in a code editor and its execution output in a terminal. The program is a simple loop that reads a target value and then repeatedly reads donation amounts until the total reaches or exceeds the target. It then prints the total and the number of donors.

```
5 func main() {
6     var target int
7     fmt.Scan(&target)
8
9     total := 0
10    donatur := 0
11
12    for total < target {
13        var donasi int
14        fmt.Scan(&donasi)
15
16        donatur++
17        total += donasi
18
19        fmt.Printf(
20            "Donatur: %d. Menyumbang %d. Total terkumpul: %d\n",
21            donatur, donasi, total,
22        )
23    }
24
25    fmt.Printf(
26        "Target tercapai! Total donasi: %d dari %d donatur.\n",
27        total, donatur,
28    )
29 }
```

The terminal output shows the program running and the user entering values for the target and donations. The output is as follows:

```
50
Donatur 3: Menyumbang 50. Total terkumpul: 250
300
Donatur 4: Menyumbang 300. Total terkumpul: 550
Target tercapai! Total donasi: 550 dari 4 donatur.
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 3\Tugas3.go"
200
300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
PS C:\Tugas> 
```

Deskripsi program

Program pertama-tama membaca target donasi. Setelah itu, program terus meminta input donasi dari setiap donatur menggunakan perulangan for. Setiap donasi akan ditambahkan ke total dan jumlah donatur dihitung satu per satu. Setelah setiap donasi masuk, program menampilkan informasi donatur, jumlah sumbangan, dan total dana yang terkumpul saat itu. Perulangan akan berhenti ketika total donasi sudah mencapai atau melebihi target, lalu program menampilkan pesan bahwa target telah tercapai beserta total donasi dan jumlah donatur.