

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

**MODUL 13**

**Repeat-until**



**Disusun oleh:**

**Cofa Xavier Marvel**

**109082500001**

**S1IF-13-04**

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## 1. Guided 1

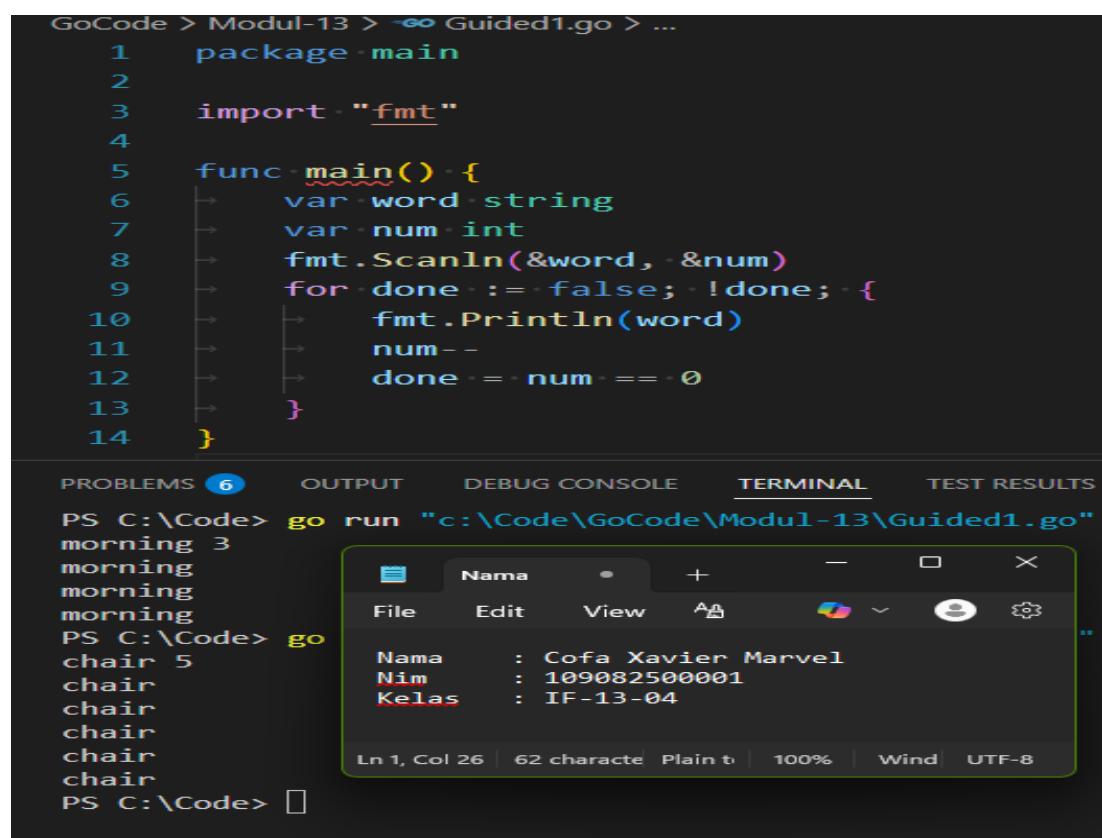
### Source Code

```
package main

import "fmt"

func main() {
    var word string
    var num int
    fmt.Scanln(&word, &num)
    for done := false; !done; {
        fmt.Println(word)
        num--
        done = num == 0
    }
}
```

### Screenshot program



The screenshot shows a terminal window and a text editor side-by-side.

**Terminal Output:**

```
GoCode > Modul-13 > go Guided1.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var word string
7     var num int
8     fmt.Scanln(&word, &num)
9     for done := false; !done; {
10         fmt.Println(word)
11         num--
12         done = num == 0
13     }
14 }
```

```
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Guided1.go"
morning 3
morning
morning
morning
PS C:\Code> go
chair 5
chair
chair
chair
chair
chair
chair
PS C:\Code> 
```

**Text Editor Screenshot:**

A screenshot of a text editor window titled "Guided1.go". The code is identical to the one shown in the terminal. Below the editor, a terminal window shows the execution of the program. The output in the terminal is:

```
morning 3
morning
morning
morning
```

The text editor window shows the following output:

| Nama  | : | Cofa Xavier Marvel |
|-------|---|--------------------|
| Nim   | : | 109082500001       |
| Kelas | : | IF-13-04           |

Below the table, the status bar of the text editor displays: Ln 1, Col 26 | 62 character Plain te | 100% | Wind | UTF-8

### **Deskripsi program**

**This program repeatedly prints word, and decreases num by 1 until num is equal to 0.**

### **2. Guided 2**

#### **Source Code**

```
package main

import "fmt"

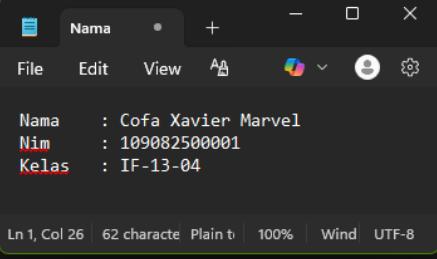
func main() {
    var num int
    for posi := false; !posi; {
        fmt.Print("enter an int:")
        fmt.Scan(&num)
        posi = num > 0
    }
    fmt.Println(num, " is a positive int")
}
```

#### **Screenshot program**

```
GoCode > Modul-13 > Guided2.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var num int
7     for posi := false; !posi; {
8         fmt.Print("enter an int:")
9         fmt.Scan(&num)
10        posi = num > 0
11    }
12    fmt.Println(num, " is a positive int")
13 }
```

PROBLEMS 8    OUTPUT    DEBUG CONSOLE    **TERMINAL**    TEST RESULTS    PORTS

```
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Guided2.go"
enter an int:-5
enter an int:-2
enter an int:-2
enter an int:0
enter an int:5
5 is a positive int
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Guided2.go"
enter an int:17
17 is a positive int
PS C:\Code>
```



The terminal window also shows the command PS C:\Code> go run "c:\Code\GoCode\Modul-13\Guided2.go" being run twice. The first run prints the three negative numbers, and the second run prints the positive number 17.

### Deskripsi program

This program repeats until `posi` is true, `posi` is true if the `num` given is greater than zero.

### Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var num1, num2 int
    fmt.Scan(&num1, &num2)

    for num1 > 0 {
        num1 -= num2
        fmt.Println(num1)
    }

    fmt.Println(num1 == 0)
}
```

## Screenshot program

The screenshot shows a GoCode IDE interface. The code editor displays a Go program named `Guided3.go` with the following content:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var num1, num2 int
7     fmt.Scan(&num1, &num2)
8
9     for num1 > 0 {
10        num1 -= num2
11        fmt.Println(num1)
12    }
13
14    fmt.Println(num1 == 0)
15 }
```

The terminal window below shows the execution of the program:

```
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Guided3.go"
15 3
12
9
6
3
0
true
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Guided3.go"
5 2
3
1
-1
false
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Guided3.go"
25 5
20
15
10
5
0
true
```

A modal dialog box titled "Nama" is open over the terminal, displaying student information:

| Field | Value              |
|-------|--------------------|
| Nama  | Cofa Xavier Marvel |
| Nim   | 109082500001       |
| Kelas | IF-13-04           |

At the bottom of the terminal window, status bars show "Ln 1, Col 26 | 62 characters | Plain text | 100% | Wind | UTF-8".

### **Deskripsi program**

**this program finds if num1 is a multiple of num2, it does this through repeatedly decreasing num1 by num2 and if num1 is zero then num1 is a multiple of num2 if it is not zero then it is not a multiple of num2.**

### **TUGAS**

#### **Tugas 1**

##### **Source code**

```
package main

import "fmt"

func main() {
    var len, num int
    fmt.Scan(&num)

    for num > 0 {
        len++
        num /= 10
    }
    fmt.Println(len)
}
```

##### **Screenshot program**

```
GoCode > Modul-13 > Tugas1.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var len, num int
7     fmt.Scan(&num)
8
9     for num > 0 {
10        len++
11        num /= 10
12    }
13    fmt.Println(len)
14 }
15
```

The screenshot shows a Go code editor interface. On the left is the source code, and on the right is a terminal window showing the output of the program. The terminal window has a title bar 'Nama' and a menu bar with 'File', 'Edit', 'View', 'Ab', and a settings icon. It displays three lines of input: 'Nama : Cofa Xavier Marvel', 'Nim : 109082500001', and 'Kelas : IF-13-04'. At the bottom of the terminal window, there is status information: 'Ln 1, Col 26 | 62 character Plain t | 100%'. Below the terminal window, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is underlined), 'TEST RESULTS', and 'PC'. The terminal window contains the following command-line session:

```
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Tugas1.go"
5
1
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Tugas1.go"
234
3
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Tugas1.go"
78787
5
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Tugas1.go"
1894256
7
PS C:\Code> []
```

### Deskripsi program

This program finds the length of a integer by repeatedly increasing len by 1 and dividing the integer by 10 until it is equal to 0.

### Tugas 2

#### Source code

```
package main

import (
```

```
"fmt"
"math"
)

func main() {
    var dec float64
    fmt.Scan(&dec)
    num := dec
    fmt.Printf("Ceiling = %.2f\n", math.Ceil(num))

    for i := int(dec * 10); i <= int(math.Ceil(num)*10); i++ {
        fmt.Printf("%.1f\n", dec)
        dec += 0.1
    }
}
```

### Screenshot program

```

GoCode > Modul-13 > Tugas2.go > ...
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var dec float64
10    fmt.Scan(&dec)
11    num := dec
12    fmt.Printf("Ceiling = %.2f\n", math.Ceil(num))
13
14    for i := int(dec * 10); i <= int(math.Ceil(num)*10); i++ {
15        fmt.Printf("%.1f\n", dec)
16        dec += 0.1
17    }
18 }

```

PROBLEMS 10

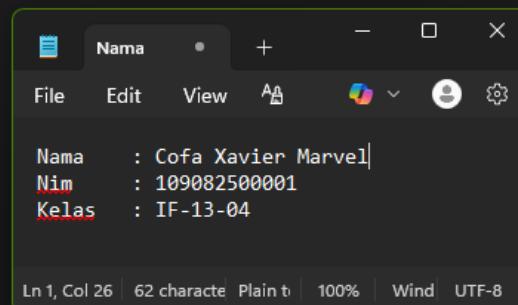
OUTPUT

DEBUG CONSOLE

TERMINAL

TEST RESULTS

PORTS



```

0.2
Ceiling = 1.00
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Tugas2.go"
2.7
Ceiling = 3.00
2.7
2.8
2.9
3.0
PS C:\Code>

```

### Deskripsi program

This program increments a decimal by one until it is equal to the rounded value. It does this through converting the dec and the ceiling of dec into ints and multiplying them by ten, this process is used to better accuracy in the repeat until loop, then the program repeats the process dec print dec, dec plus 0.1, and i plus 1 until i is greater than the int multiple of ten ceiling of dec.

Tugas 3

## Source code

```
package main

import "fmt"

func main() {
    var Donors, Donated, TotCollected, Goal int

    fmt.Scan(&Goal)

    for ; Goal > 0; Goal -= Donated {
        fmt.Scan(&Donated)
        Donors++
        TotCollected += Donated
        fmt.Printf("Donor %d: Donated %d. Total collected: %d\n", Donors, Donated,
TotCollected)
    }
    if Goal < 0 {
        fmt.Printf("\nTarget reached! Total donation: %d from %d donors.", TotCollected, Donors)
    }
}
```

## Screenshot program

The screenshot shows a GoCode IDE interface. At the top, there's a navigation bar with 'GoCode > Modul-13 > Tugas3.go > ...'. Below it is the code editor with the following Go code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var Donors, Donated, TotCollected, Goal int
7
8     fmt.Scan(&Goal)
9
10    for ; Goal > 0; Goal -= Donated {
11        fmt.Scan(&Donated)
12        Donors++
13        TotCollected += Donated
14        fmt.Printf("Donor %d: Donated %d. Total collected: %d\n", Donors, Donated, TotCollected)
15    }
16    if Goal < 0 {
17        fmt.Printf("\nTarget reached! Total donation: %d from %d donors.", TotCollected, Donors)
18    }
19 }
```

Below the code editor are tabs for 'PROBLEMS' (4), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (selected), 'TEST RESULTS', and 'PORTS'. The 'TERMINAL' tab shows the following command-line output:

```
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Tugas3.go"
200
300
Donor 1: Donated 300. Total collected: 300

Target reached! Total donation: 300 from 1 donors.
PS C:\Code> go run "c:\Code\GoCode\Modul-13\Tugas3.go"
300
100
Donor 1: Donated 100. Total collected: 100
50
Donor 2: Donated 50. Total collected: 150
200
Donor 3: Donated 200. Total collected: 350

Target reached! Total donation: 350 from 3 donors.
```

To the right of the terminal, a modal window titled 'Nama' is displayed, containing student information:

| Nama  | : | Cofa Xavier Marvel |
|-------|---|--------------------|
| Nim   | : | 109082500001       |
| Kelas | : | IF-13-04           |

At the bottom of the modal window, status bars show 'Ln 1, Col 26 | 62 character Plain text | 100% | Wind | UTF-8'.

## Deskripsi program

This program collects the amount donated and number of the donators until it reaches the goal then show the number of donators a total of donations.

It does this through first getting the donation goal then it repeatedly Scans for donations in the form of integers, increasing the number of donators by one, increasing the total collected by the donation given, and printing the donor number, amount donated, and total collected donations until the goal, which has been decreased by the donations, is lesser than zero.