

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 12  
WHILE-LOOP**



**Disusun oleh:**

**FARID HERDIYANTO VITASANDI**

**109082500123**

**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 bilangan int

    fmt.Print("Masukkan bilangan: ")

    fmt.Scan(&bilangan)

    if bilangan < 0 {

        fmt.Print("Bilangan tidak boleh negatif")

        return

    }

    if bilangan == 0 {

        fmt.Println(1)

    }else{

        for i := bilangan; i >= 1; i--{

            fmt.Print(i)

            if i != 1 {

                fmt.Print(" x ")

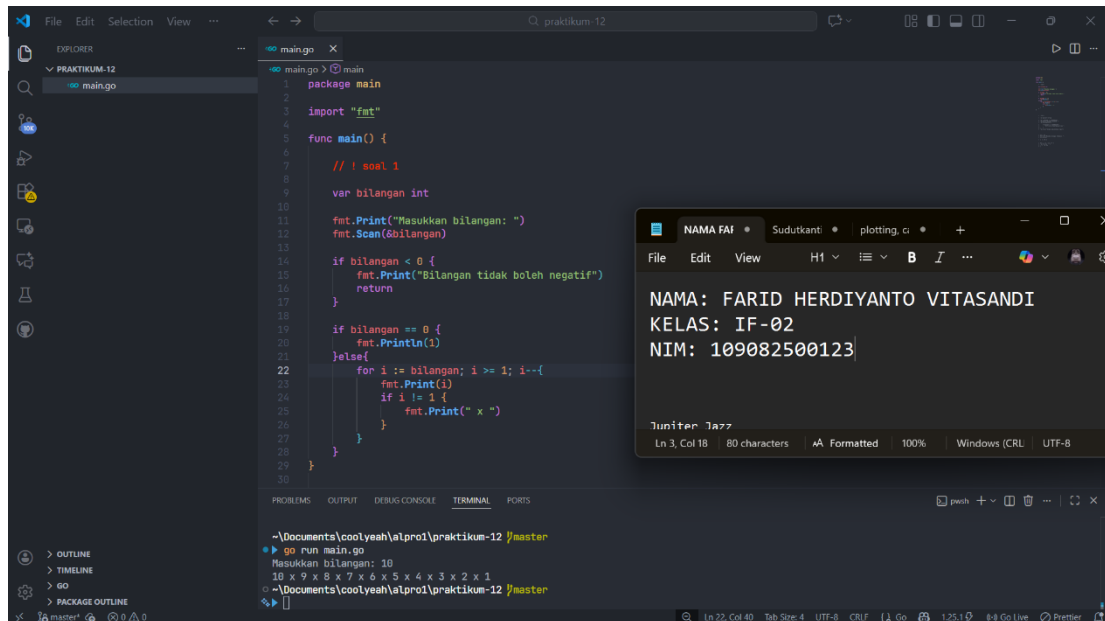
            }

        }

    }

}
```

## Screenshoot program



The screenshot shows a Go program in a text editor and its execution output in a terminal window.

**Go Program Code:**

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     // ! soal 1
8
9     var bilangan int
10
11     fmt.Print("Masukkan bilangan: ")
12     fmt.Scan(&bilangan)
13
14     if bilangan < 0 {
15         fmt.Print("Bilangan tidak boleh negatif")
16         return
17     }
18
19     if bilangan == 0 {
20         fmt.Println(1)
21     } else {
22         for i := bilangan; i >= 1; i-- {
23             fmt.Print(i)
24             if i != 1 {
25                 fmt.Print(" x ")
26             }
27         }
28     }
29 }
30
```

**Terminal Output:**

```
~\Documents\coolyeah\alpro1\praktikum-12 #master
$ go run main.go
Masukkan bilangan: 10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
~\Documents\coolyeah\alpro1\praktikum-12 #master
```

The terminal window also displays student information:

```
NAMA: FARID HERDIYANTO VITASANDI
KELAS: IF-02
NIM: 109082500123
```

Uniten Dayz

## Deskripsi program

Program di atas meminta User untuk memasukkan sebuah bilangan, kemudian melakukan pengecekan awal untuk memastikan bahwa nilai yang diberikan tidak bernilai negatif. Apabila bilangan kurang dari nol, program langsung menghentikan eksekusi dan menampilkan pesan bahwa input tidak boleh negatif. Jika bilangan bernilai 0, program mencetak angka 1 sebagai hasil. Namun, apabila bilangan lebih besar dari 0, program menampilkan deret perkalian mundur mulai dari bilangan tersebut hingga 1.

## Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var password string

    const token = "12345abcde"

    for password != token {

        fmt.Print("Masukkan password: ")

        fmt.Scan(&password)

        if password != token {

            fmt.Println("Password salah, silahkan coba lagi.")

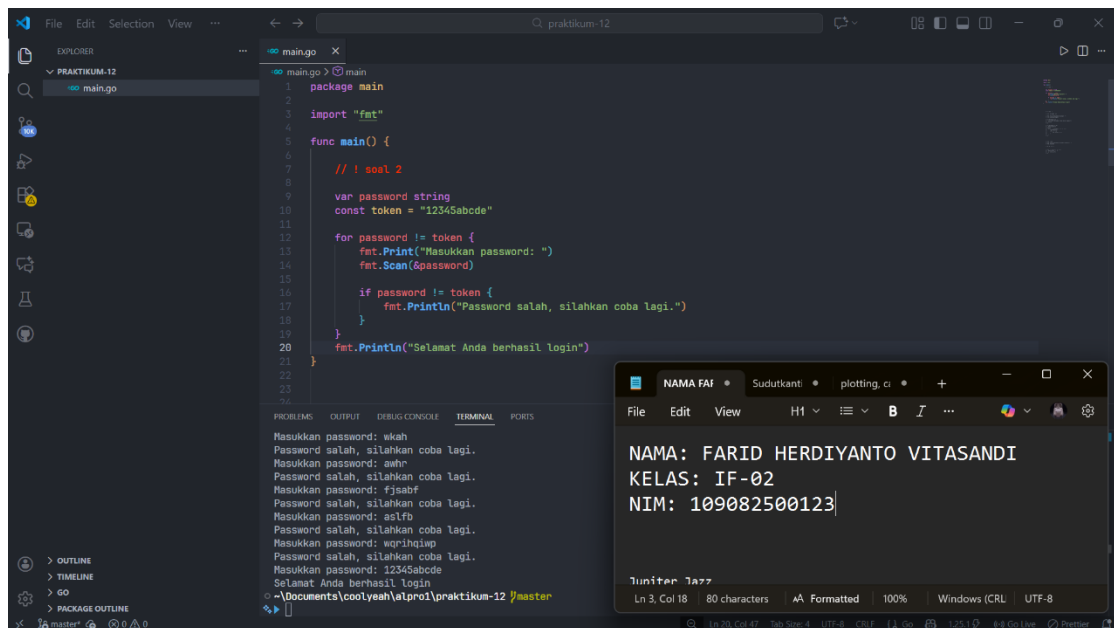
        }

    }

    fmt.Println("Selamat Anda berhasil login")

}
```

## Screenshoot program



The screenshot shows a Go program in a VS Code editor. The program is located in `main.go` under the `main` package. It defines a `main` function that prompts the user for a password and checks it against a constant `token` value of `"12345abcde"`. If the password matches, it prints "Selamat Anda berhasil login". Otherwise, it prints "Password salah, silahkan coba lagi." and loops back to the password prompt.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     // ! soal 2
8
9     var password string
10    const token = "12345abcde"
11
12    for password != token {
13        fmt.Print("Masukkan password: ")
14        password = fmt.Scanln()
15
16        if password != token {
17            fmt.Println("Password salah, silahkan coba lagi.")
18        }
19    }
20    fmt.Println("Selamat Anda berhasil login")
21 }
22
23
24
```

The terminal output shows several failed login attempts with the message "Masukkan password: " followed by the user input and the error message "Password salah, silahkan coba lagi.". The final output is "Selamat Anda berhasil login".

Below the terminal, there is a text box with the following text:

```
NAMA: FARID HERDIYANTO VITASANDI
KELAS: IF-02
NIM: 109082500123
```

## Deskripsi program

Program di atas dibuat untuk membaca input berupa nama tanaman yang dimasukkan User, kemudian memeriksanya menggunakan struktur *switch* untuk menentukan kategorinya. Jika User memasukkan “nepenthes” atau “drosera”, program akan menampilkan bahwa tanaman tersebut termasuk tanaman karnivora dan berasal dari Indonesia. Apabila input berupa “venus” atau “sarracenia”, program tetap mengelompokkan tanaman tersebut sebagai tanaman karnivora, tetapi menegaskan bahwa keduanya buka. Jika nama tanaman yang dimasukkan tidak termasuk dalam daftar tersebut, program akan menampilkan bahwa tanaman tersebut bukan bagian dari kelompok tanaman karnivora.

## 2. Guided 3

### Source Code

```
package main

import "fmt"

func main() {

    var N, s1, s2, j, temp int

    fmt.Scan(&N)

    s1 = 0

    s2 = 1

    j = 0

    for j < N{

        fmt.Print(s1, " ")

        temp = s1 + s2

        s1 = s2

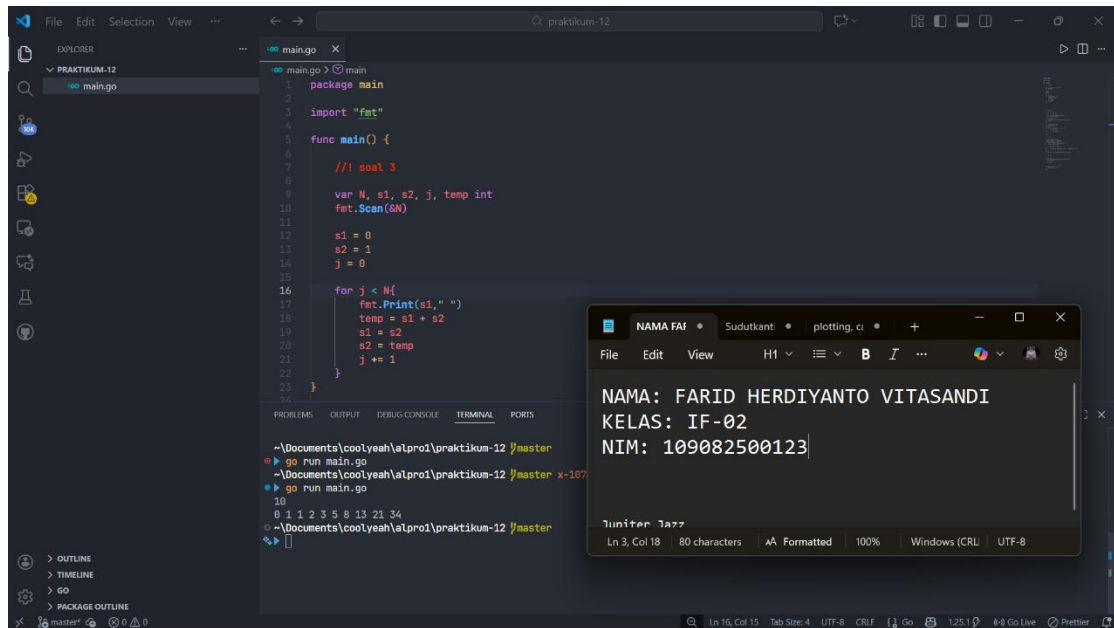
        s2 = temp

        j += 1

    }

}
```

## Screenshoot program



The screenshot shows a Go program in a file named `main.go` within a project named `praktikum-12`. The program is designed to generate a Fibonacci sequence of length `N`. It uses the `fmt` package for input and output. The `main` function reads an integer `N` from the user, initializes `s1` to 0 and `s2` to 1, and then enters a loop where it prints `s1`, calculates the next Fibonacci number by adding `s1` and `s2` to `temp`, and updates `s1` and `s2` accordingly. The loop continues until `j` reaches `N`.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     //! soal 3
8
9     var N, s1, s2, j, temp int
10    fmt.Scan(&N)
11
12    s1 = 0
13    s2 = 1
14    j = 0
15
16    for j < N{
17        fmt.Print(s1, " ")
18        temp = s1 + s2
19        s1 = s2
20        s2 = temp
21        j += 1
22    }
23 }
```

The terminal window shows the execution of the program. The user inputs `10` for `N`, and the program outputs the sequence: `0 1 1 2 3 5 8 13 21 34`.

```
~\Documents\coolyeah\alpro1\praktikum-12 /master
$ go run main.go
10
0 1 1 2 3 5 8 13 21 34
$
```

## Deskripsi program

Kode di atas dibuat untuk membaca nilai `N` yang di-input oleh User, kemudian menghasilkan deret Fibonacci sebanyak `N` kali. Nilai awal dari deret Fibonacci ditentukan melalui variable `s1` yang bernilai 0 dan variable `s2` yang bernilai 1, sementara variabel `j` digunakan sebagai penghitung iterasi. Selama `j` masih lebih kecil dari `N`, program akan mencetak nilai `s1` sebagai elemen deret, lalu menghitung nilai berikutnya dengan menjumlahkan `s1` dan `s2` lalu menyimpannya pada variable `temp`. Setelah itu, nilai `s1` akan diperbarui menjadi nilai `s2`, dan nilai `s2` akan diperbarui menjadi `temp`, sehingga pola Fibonacci akan terus berlanjut. Pada setiap iterasi, nilai `j` ditambah 1 hingga jumlah elemen yang dicetak mencapai nilai variable `N`.

## TUGAS

### 1. Soal 1

#### Source Code

```
package main

import "fmt"

func main() {

    const correctUsername = "Admin"

    const correctPassword = "Admin"

    percobaanGagal := 0

    isLoggedIn := false

    var username, password string

    for !isLoggedIn{

        fmt.Print("Masukkan Username: ")

        fmt.Scan(&username)

        fmt.Print("Masukkan Password: ")

        fmt.Scan(&password)

        if username == correctUsername && password ==
correctPassword{

            isLoggedIn = true
```



```
        fmt.Println("Login berhasil!")

        fmt.Printf("Jumlah percobaan login yang gagal:
%d", percobaanGagal)

    }else{

        percobaanGagal++

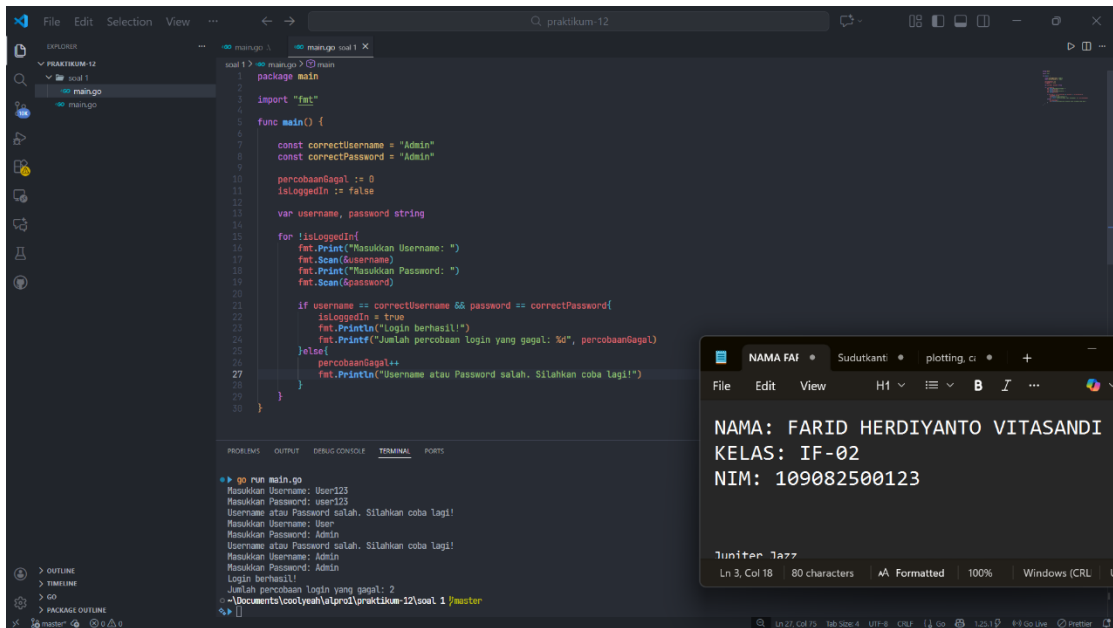
        fmt.Println("Username atau Password salah.
Silahkan coba lagi!")

    }

}

}
```

## Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     const correctUsername = "Admin"
8     const correctPassword = "Admin"
9
10    percobaanGagal := 0
11    isLoggedIn := false
12
13    var username, password string
14
15    for !isLoggedIn{
16        fmt.Print("Masukkan Username: ")
17        username = fmt.Scanln()
18        fmt.Print("Masukkan Password: ")
19        password = fmt.Scanln()
20
21        if username == correctUsername && password == correctPassword{
22            isLoggedIn = true
23            fmt.Println("Login berhasil!")
24            fmt.Printf("Jumlah percobaan login yang gagal: %d", percobaanGagal)
25        }else{
26            percobaanGagal++
27            fmt.Println("Username atau Password salah. Silahkan coba lagi!")
28        }
29    }
30 }
```

run main.go

```
Masukkan Username: user123
Masukkan Password: user123
Username atau Password salah. Silahkan coba lagi!
Masukkan Username: user
Masukkan Password: Admin
Username atau Password salah. Silahkan coba lagi!
Masukkan Username: Admin
Masukkan Password: Admin
Login berhasil!
Jumlah percobaan login yang gagal: 2
```

NAMA: FARID HERDIYANTO VITASANDI  
KELAS: IF-02  
NIM: 109082500123

## Deskripsi program

Program di atas adalah program yang dibuat untuk melakukan proses autentikasi dengan membandingkan input pengguna terhadap username dan password yang telah ditentukan sebelumnya. Dua konstanta yaitu `correctUsername` dan `correctPassword` digunakan sebagai acuan untuk validasi, sementara variable `percobaanGagal` akan mencatat jumlah kesalahan login dan variable `isLoggedIn` digunakan sebagai indikator untuk keberhasilan login. Selama nilai `isLoggedIn` masih bernilai `false`, program akan terus meminta User untuk memasukkan username dan password. Jika keduanya sesuai dengan nilai yang benar, maka program akan mengubah nilai `isLoggedIn` menjadi `true`, menampilkan pesan bahwa login berhasil, serta mencatat jumlah percobaan login yang gagal sebelumnya. Namun, jika data yang dimasukkan tidak cocok, program akan menambah nilai `percobaanGagal` dan menampilkan pesan bahwa username atau password salah, sebelum nantinya akan mengulangi proses input username dan password.

## 2. Soal 2

### Source Code

```
package main

import "fmt"

func main() {

    var bilangan int

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

    for bilangan < 0 {

        fmt.Println("Bilangan harus positif!")

        fmt.Print("Masukkan bilangan bulat positif: ")

        fmt.Scan(&bilangan)

    }

    hitungDigit := 0

    for bilangan > 0 {

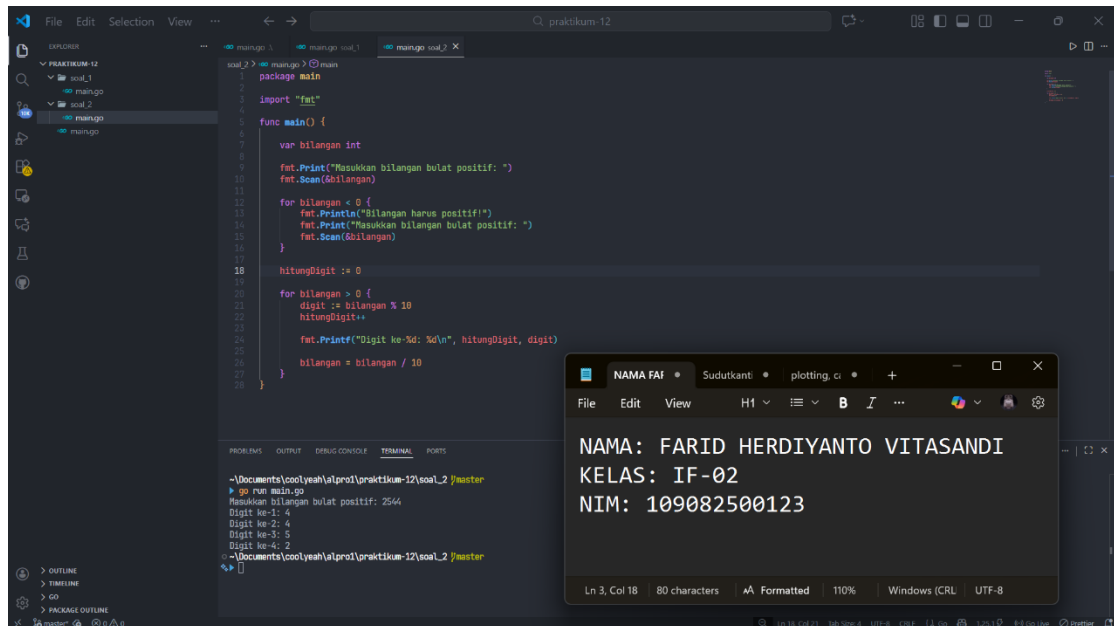
        digit := bilangan % 10

        hitungDigit++

        fmt.Printf("Digit ke-%d: %d\n", hitungDigit, digit)
```

```
        bilangan = bilangan / 10
    }
}
```

## Screenshoot program



The screenshot shows a Go program in a VS Code editor and its execution output in a terminal window. The program is located in `main.go` and implements a function to reverse a positive integer by extracting its digits from right to left.

```
1 package main
2 import "fmt"
3
4 func main() {
5     var bilangan int
6     fmt.Print("Masukkan bilangan bulat positif: ")
7     fmt.Scan(&bilangan)
8
9     for bilangan < 0 {
10        fmt.Println("Bilangan harus positif!")
11        fmt.Print("Masukkan bilangan bulat positif: ")
12        fmt.Scan(&bilangan)
13    }
14
15    hitungDigit := 0
16
17    for bilangan > 0 {
18        digit := bilangan % 10
19        hitungDigit++
20
21        fmt.Printf("Digit ke-%d: %d\n", hitungDigit, digit)
22
23        bilangan = bilangan / 10
24    }
25 }
```

The terminal window shows the program's output for the input 2544:

```
~\Documents\cool\goh\praktikum-12\soal_2 $ go run main.go
Masukkan bilangan bulat positif: 2544
Digit ke-1: 4
Digit ke-2: 4
Digit ke-3: 5
Digit ke-4: 2
```

## Deskripsi program

Program di atas meminta User untuk memasukkan sebuah bilangan bulat positif, kemudian melakukan pengecekan menggunakan perulangan *for* untuk memastikan bahwa nilai yang diberikan tidak bernilai negatif. Jika nilai yang dimasukkan masih kurang dari nol, maka program akan terus meminta input ulang hingga User memberikan bilangan yang valid. Setelah input yang valid diterima, program akan memulai proses penghitungan digit dengan menginisialisasi variabel `hitungDigit` sebagai penghitung urutan digit. Melalui iterasi berikutnya, program akan mengambil digit paling belakang dari bilangan menggunakan operasi modulus 10, menampilkan digit tersebut beserta urutannya, kemudian membagi bilangan dengan 10 untuk menghilangkan digit yang sudah diproses. Iterasi ini akan terus berjalan hingga seluruh digit dalam bilangan selesai dihitung dan ditampilkan.

### 3. Soal 3

#### Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    fmt.Print("Masukkan x: ")

    fmt.Scan(&x)

    fmt.Print("Masukkan y: ")

    fmt.Scan(&y)

    for y == 0 || x < y {

        if y == 0 {

            fmt.Println("y tidak boleh 0!")

            fmt.Print("Masukkan y: ")

            fmt.Scan(&y)

        }

        if x < y {

            fmt.Println("x harus lebih besar dari y!")

            fmt.Print("Masukkan x: ")

            fmt.Scan(&x)

        }

    }

}
```

```
}

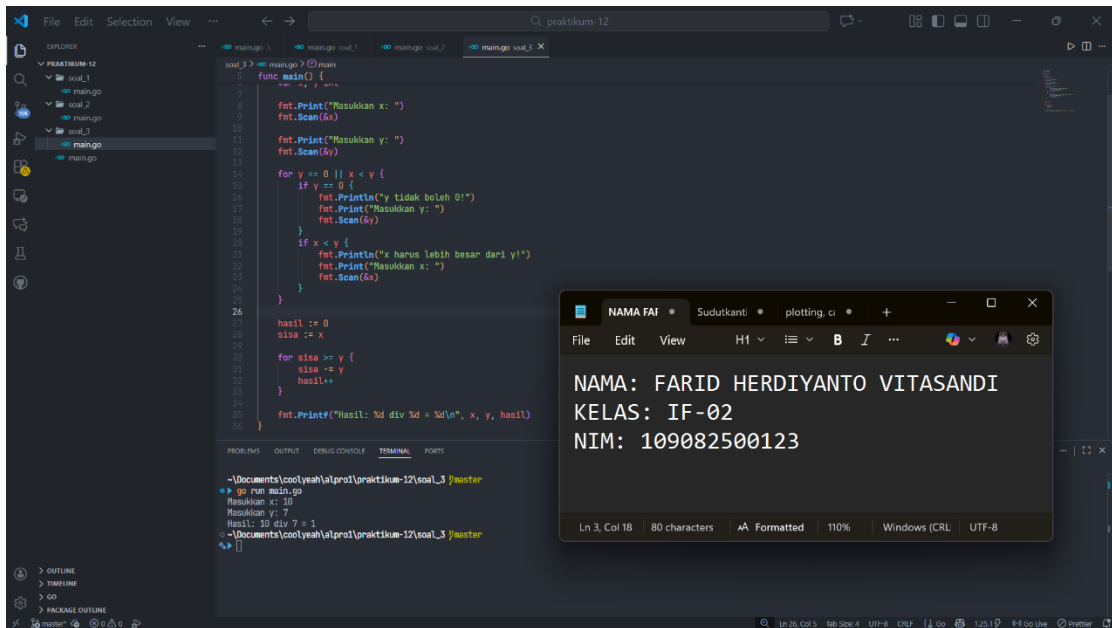
hasil := 0

sisal := x

for sisa >= y {
    sisa -= y
    hasil++
}

fmt.Printf("Hasil: %d div %d = %d\n", x, y, hasil)
}
```

## Screenshoot program



```
1 func main() {
2     // ...
3     fmt.Println("Masukkan x: ")
4     fmt.Scan(&x)
5
6     fmt.Println("Masukkan y: ")
7     fmt.Scan(&y)
8
9     for y == 0 || x < y {
10        if y == 0 {
11            fmt.Println("y tidak boleh 0!")
12            fmt.Print("Masukkan y: ")
13            fmt.Scan(&y)
14        }
15        if x < y {
16            fmt.Println("x harus lebih besar dari y!")
17            fmt.Print("Masukkan x: ")
18            fmt.Scan(&x)
19        }
20    }
21
22    hasil := 0
23    sisa := x
24
25    for sisa >= y {
26        sisa -= y
27        hasil++
28    }
29
30    fmt.Printf("Hasil: %d div %d = %d\n", x, y, hasil)
31 }
```

```
~\Documents\cool\yah\alpro\praktikum-12\soal_3 $master
$ go run main.go
Masukkan x: 10
Masukkan y: 7
Hasil: 10 div 7 = 1
~\Documents\cool\yah\alpro\praktikum-12\soal_3 $master
```

NAMA: FARID HERDIYANTO VITASANDI  
KELAS: IF-02  
NIM: 109082500123

## Deskripsi program

Program ini akan menerima dua input dari User, yaitu x dan y. Kemudian program akan melakukan validasi untuk memastikan bahwa nilai y tidak bernilai nol dan x tidak lebih kecil dari y. Validasi ini dilakukan di dalam perulangan *for* yang nantinya akan terus meminta input ulang selama salah satu kondisi belum terpenuhi. Setelah kedua nilai yang dimasukkan valid, program akan menghitung hasil pembagian bilangan bulat x dengan y tanpa menggunakan operator pembagian. Proses ini dilakukan dengan mengurangi y secara berulang dari x menggunakan variabel sisa, sambil menambah variabel hasil setiap kali pengurangan terjadi. Perulangan berhenti ketika sisa menjadi lebih kecil dari y, dan nilai hasil yang diperoleh merupakan hasil pembagian bilangan bulat.