

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 12  
WHILE-LOOP**



**Disusun oleh:**

**Nabyla Zahirah Ramadhani**

**109082500104**

**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 n, j int

    fmt.Scan(&n)

    j = n

    for j > 1 {

        fmt.Print(j, " x ")

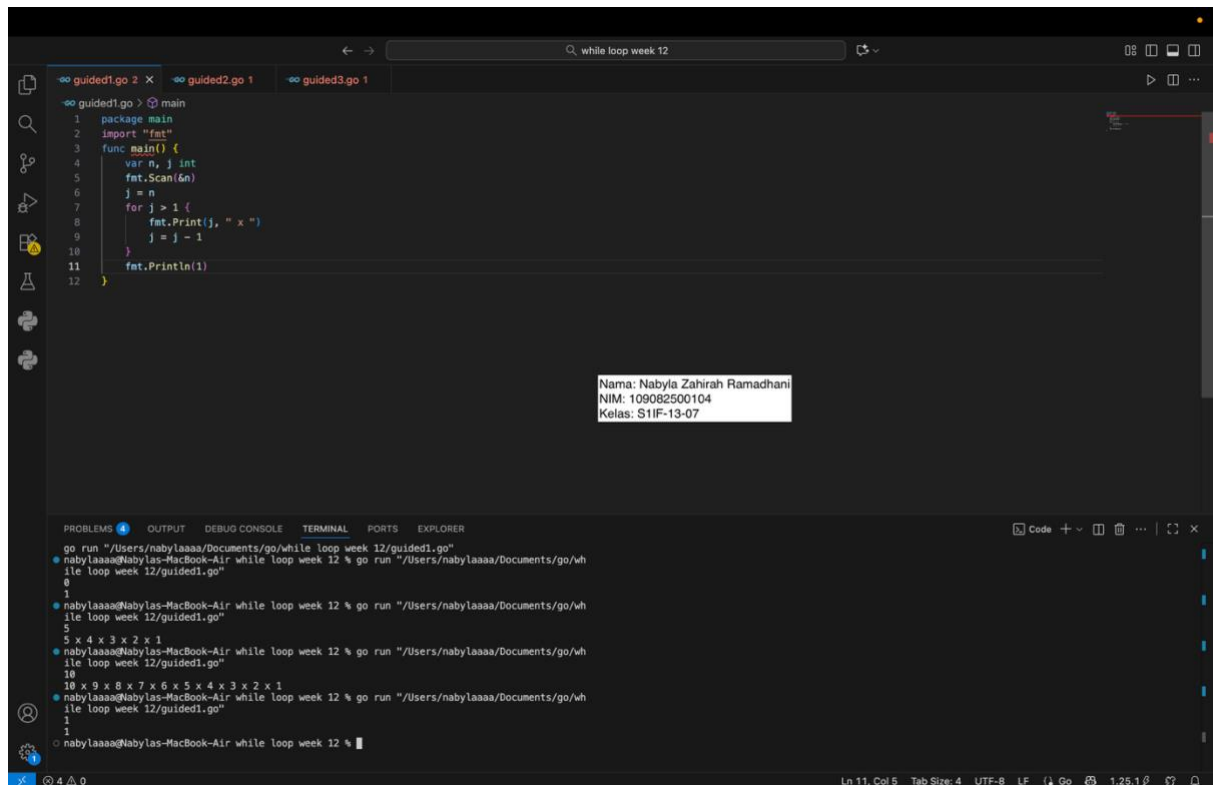
        j = j - 1

    }

    fmt.Println(1)

}
```

#### Screenshoot program



## Deskripsi program

Program ini untuk menampilkan deret bilangan Faktorial dari suatu bilangan, variabelnya `n` dan `j` yang bertipe integer. Pertama program akan membaca input angka dari user menggunakan `fmt.Scan(&n)`. `for j > 1` artinya selama `j` lebih besar dari 1 maka perulangan terus dijalankan, di dalam perulangan program menampilkan nilai `j` di ikuti oleh `x` menggunakan `fmt.Print(j, "x ")`. Setelah mencetak nilai tadi program mengurangi `j` sebanyak 1 menggunakan `j = j - 1` sehingga setiap putaran akan menampilkan angka yang lebih kecil. Ketika `j` tidak lebih besar dari 1, perulangan selesai. Sebagai penutup deret, program mencetak angka 1 menggunakan `fmt.Println(1)`.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main () {

    var token string

    fmt.Print("Masukkan token: ")

    fmt.Scan(&token)
```

```

for token != "12345abcde" {

    fmt.Scan(&token)

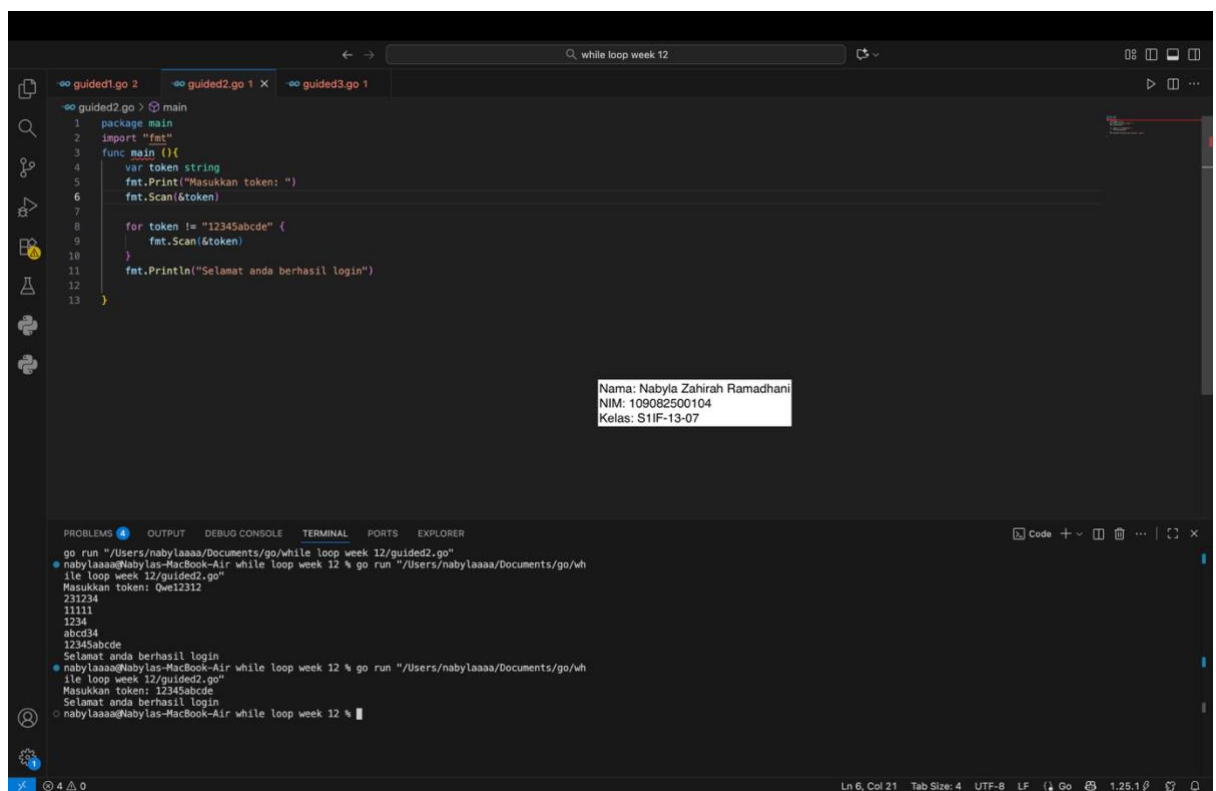
}

fmt.Println("Selamat anda berhasil login")

}

```

### Screenshoot program



### Deskripsi program

Program ini meminta pengguna memasukkan token, mengecek apakah token sudah benar, dan terus meminta ulang sampai pengguna mengetik token yang tepat. Variabelnya token yang bertipe string kemudian program meminta input token menggunakan `fmt.Print("Masukkan token: ")`, dan akan dibaca menggunakan `fmt.Scan(&token)`. Setelah itu, program menjalankan sebuah perulangan menggunakan `for token != "12345abcde"`. Kondisi ini berarti selama nilai token tidak sama dengan "12345abcde", perulangan akan terus berjalan. Jika pengguna salah memasukkan token, program tidak menampilkan pesan dan langsung meminta input lagi lewat `fmt.Scan(&token)`. Perulangan baru akan berhenti ketika user mengetik token yang benar, yaitu "12345abcde". Ketika token sudah cocok, barulah program

akan menampilkan *fmt.Println("Selamat anda berhasil login")* sebagai tanda bahwa proses login berhasil.

### 3. 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 = j + 1

    }

}
```

**Screenshoot program**

The screenshot shows a Go IDE with a file named `guided3.go` open. The code is a Go program that calculates the first `n` Fibonacci numbers. It uses `fmt.Scan(&n)` to read input, initializes `s1 = 0`, `s2 = 1`, and `j = 0`. A `for` loop runs from `j = 0` to `j < n`. Inside the loop, it prints `s1`, calculates the next Fibonacci number `temp = s1 + s2`, shifts the values (`s1 = s2`, `s2 = temp`), and increments `j`. The terminal output shows the program being run multiple times with different inputs, producing the corresponding Fibonacci sequences.

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

Terminal Output:

```
go run "/Users/nabylaaa/Documents/go/while loop week 12/guided3.go"
nabylaaa@Nabylas-MacBook-Air:~/Documents/go/while loop week 12$ go run "/Users/nabylaaa/Documents/go/while loop week 12/guided3.go"
5
0 1 1 2 3
nabylaaa@Nabylas-MacBook-Air:~/Documents/go/while loop week 12$ go run "/Users/nabylaaa/Documents/go/while loop week 12/guided3.go"
2
0 1
nabylaaa@Nabylas-MacBook-Air:~/Documents/go/while loop week 12$ go run "/Users/nabylaaa/Documents/go/while loop week 12/guided3.go"
10
0 1 1 2 3 5 8 13 21 34
nabylaaa@Nabylas-MacBook-Air:~/Documents/go/while loop week 12$
```

Nama: Nabyla Zahirah Ramadhani  
NIM: 109082500104  
Kelas: S1IF-13-07

## Deskripsi program

Program ini mencetak `N` bilangan pertama dalam deret Fibonacci, variabelnya `n`, `s1`, `s2`, `j` dan `temp` yang bertipe `int`. Pertama program akan membaca angka yang di input oleh user menggunakan `fmt.Scan(&n)`, lalu program akan mengisi `s1 = 0` dan `s2 = 1`, karena dua angka pertama deret Fibonacci memang dimulai dari 0 dan 1, untuk `j = 0` agar proses perulangan bisa dimulai dari 0. Program kemudian masuk ke dalam perulangan `for j < n`, yang berarti selama jumlah bilangan Fibonacci yang sudah dicetak masih kurang dari `n`, perulangan akan terus berjalan. Di dalam perulangan ini, program mencetak nilai `s1` dengan `fmt.Print(s1, " ")`, sehingga setiap angka Fibonacci muncul berurutan dengan spasi di belakangnya. Setelah mencetak angka tersebut, program menghitung angka Fibonacci berikutnya dengan menjumlahkan dua angka sebelumnya, yaitu `temp = s1 + s2`. Setelah itu, program menggeser posisi kedua angka tadi nilai `s1` diubah menjadi `s2` lewat `s1 = s2`, dan nilai `s2` diganti dengan hasil penjumlahan tadi lewat `s2 = temp`. Akhirnya, penghitung `j` ditambah 1 melalui `j = j + 1`, sehingga suatu saat nilai `j` akan mencapai `n`, dan perulangan akan berhenti.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main (){

    var usn, pw string

    gagal := 0

    fmt.Print("Masukkan username & password: ")

    fmt.Scan(&usn, &pw)

    for usn != "Admin" || pw != "Admin"{

        gagal++

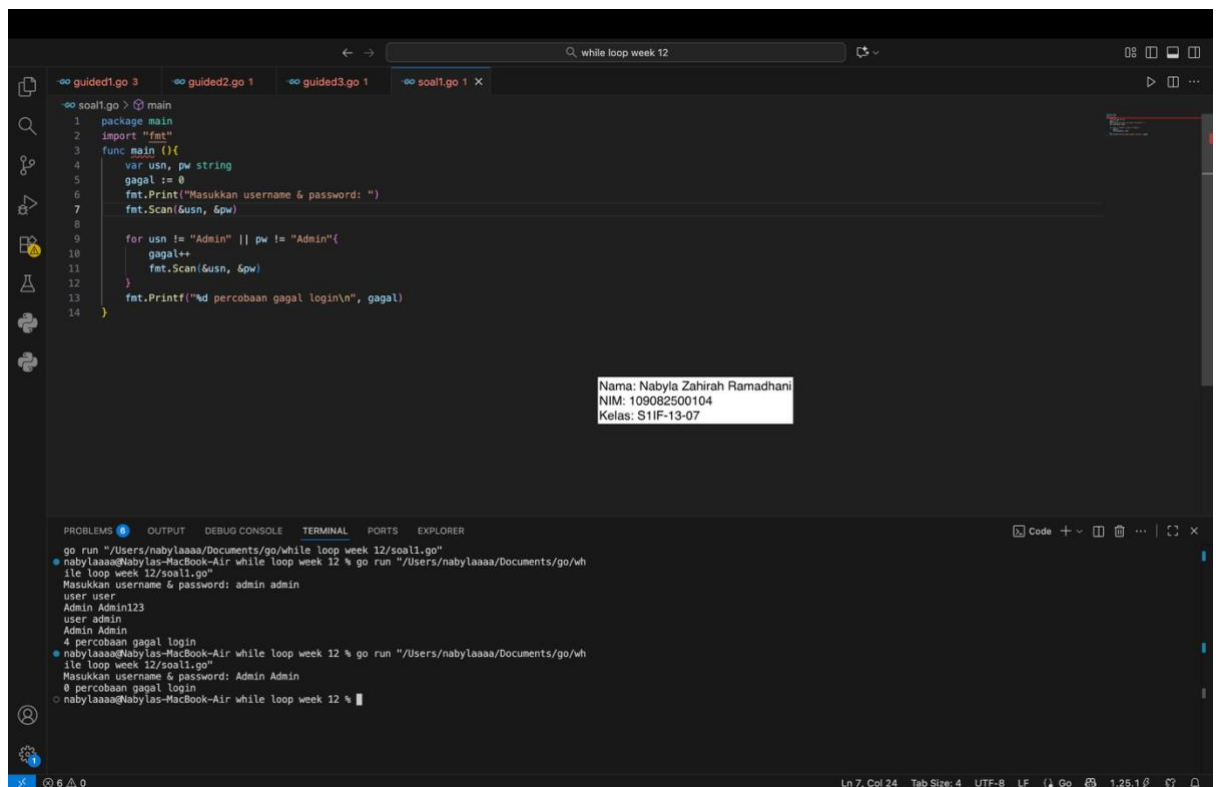
        fmt.Scan(&usn, &pw)

    }

    fmt.Printf("%d percobaan gagal login\n", gagal)

}
```

#### Screenshoot program



```
1 package main
2 import "fmt"
3 func main() {
4     var usn, pw string
5     gagal := 0
6     fmt.Print("Masukkan username & password: ")
7     fmt.Scan(&usn, &pw)
8
9     for usn != "Admin" || pw != "Admin" {
10         gagal++
11         fmt.Scan(&usn, &pw)
12     }
13     fmt.Printf("%d percobaan gagal login\n", gagal)
14 }
```

go run "/Users/nabylaaaa/Documents/go/while loop week 12/soal1.go"  
nabylaaaa@Nabylas-MacBook-Air: while loop week 12 % go run "/Users/nabylaaaa/Documents/go/wh  
ile loop week 12/soal1.go"  
Masukkan username & password: admin admin  
user user  
Admin Admin123  
user admin  
Admin Admin  
4 percobaan gagal login  
nabylaaaa@Nabylas-MacBook-Air: while loop week 12 % go run "/Users/nabylaaaa/Documents/go/wh  
ile loop week 12/soal1.go"  
Masukkan username & password: Admin Admin  
0 percobaan gagal login  
nabylaaaa@Nabylas-MacBook-Air: while loop week 12 %

## Deskripsi program

Program ini untuk menghitung berapa banyak seseorang pengguna gagal melakukan login, karena kesalahan memberikan username dan password. Variabelnya `usn` dan `pw` yang bertipe string dan ada variabel `gagal := 0`, yang sejak awal nilainya 0 supaya bisa menghitung berapa kali pengguna salah memasukkan data login. Program akan meminta input `usn` dan `pw` dari user menggunakan `fmt.Print("Masukkan username & password: ")`, dan akan dibaca dua input sekaligus melalui `fmt.Scan(&usn, &pw)`. Setelah itu program masuk ke sebuah perulangan dengan kondisi `for usn != "Admin" || pw != "Admin"`, ini artinya selama username salah atau password salah, perulangan akan terus berjalan. Jika salah satu saja tidak cocok, kondisi tetap dianggap salah sehingga loop berlanjut. Setiap kali input salah, program menambah jumlah kegagalan dengan `gagal++`, lalu kembali membaca username dan password baru menggunakan `fmt.Scan(&usn, &pw)`. Program akan terus meminta input sampai pengguna memasukkan yang benar. Perulangan akan berhenti hanya jika kedua input sudah tepat, yaitu username "Admin" dan password "Admin". Begitu kondisi terpenuhi, program keluar dari loop dan mencetak jumlah percobaan yang gagal lewat `fmt.Printf("%d percobaan gagal login\n", gagal)`.



## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main () {

    var n int

    fmt.Print("Masukkan bilangan: ")

    fmt.Scan(&n)


    for n > 0 {

        digit := n % 10

        fmt.Println(digit)

        n = n / 10

    }

}
```

### Screenshoot program

The screenshot shows a Go IDE with a project named "while loop week 12". The Explorer panel on the left shows a directory structure with files: `guided1.go`, `guided2.go`, `guided3.go`, `soal1.go`, `soal2.go`, and `soal3.go`. The main editor displays the code for `soal2.go`:

```
1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Print("Masukkan bilangan: ")
6     fmt.Scan(&n)
7
8     for n > 0 {
9         digit := n % 10
10        fmt.Println(digit)
11        n = n / 10
12    }
13 }
```

Below the code editor, a white box contains the student's information:

Nama: Nabyla Zahirah Ramadhani  
NIM: 109082500104  
Kelas: S1IF-13-07

The bottom panel shows the terminal output of the program:

```
nabylaaaa@Nabylas-MacBook-Air while loop week 12 % go run "/Users/nabylaaaa/Documents/go/while loop week 12/soal2.go"
4
5
2
nabylaaaa@Nabylas-MacBook-Air while loop week 12 % go run "/Users/nabylaaaa/Documents/go/while loop week 12/soal2.go"
Masukkan bilangan: 3423554654
4
5
6
4
5
5
3
2
4
3
```

## Deskripsi program

Program ini untuk mencacah setiap digit yang terdapat di dalam suatu bilangan bulat positif, variabelnya `n` yang bertipe `int`. Program ini dimulai dengan meminta pengguna memasukkan sebuah bilangan menggunakan `fmt.Print("Masukkan bilangan: ")` dan akan dibaca menggunakan `fmt.Scan(&n)`. Setelah itu, program masuk ke sebuah perulangan `for n > 0`, yang berarti loop akan terus berjalan selama nilai `n` masih lebih dari nol. Di dalam loop, program mengambil digit paling belakang dari bilangan tersebut menggunakan `digit:=n % 10` (bilangan `n` di modulus 10, sisa bagi). Hasil itu disimpan sebagai digit, lalu langsung ditampilkan dengan `fmt.Println(digit)`. Setelah digit terakhir dicetak, nilai `n` diperkecil dengan `n = n / 10`. Operasi pembagian ini membuang digit terakhir, sehingga pada iterasi berikutnya digit sebelumnya bisa diambil.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main () {

    var x, y int

    fmt.Print("Masukkan bilangan x dan y: ")
    fmt.Scan(&x, &y)

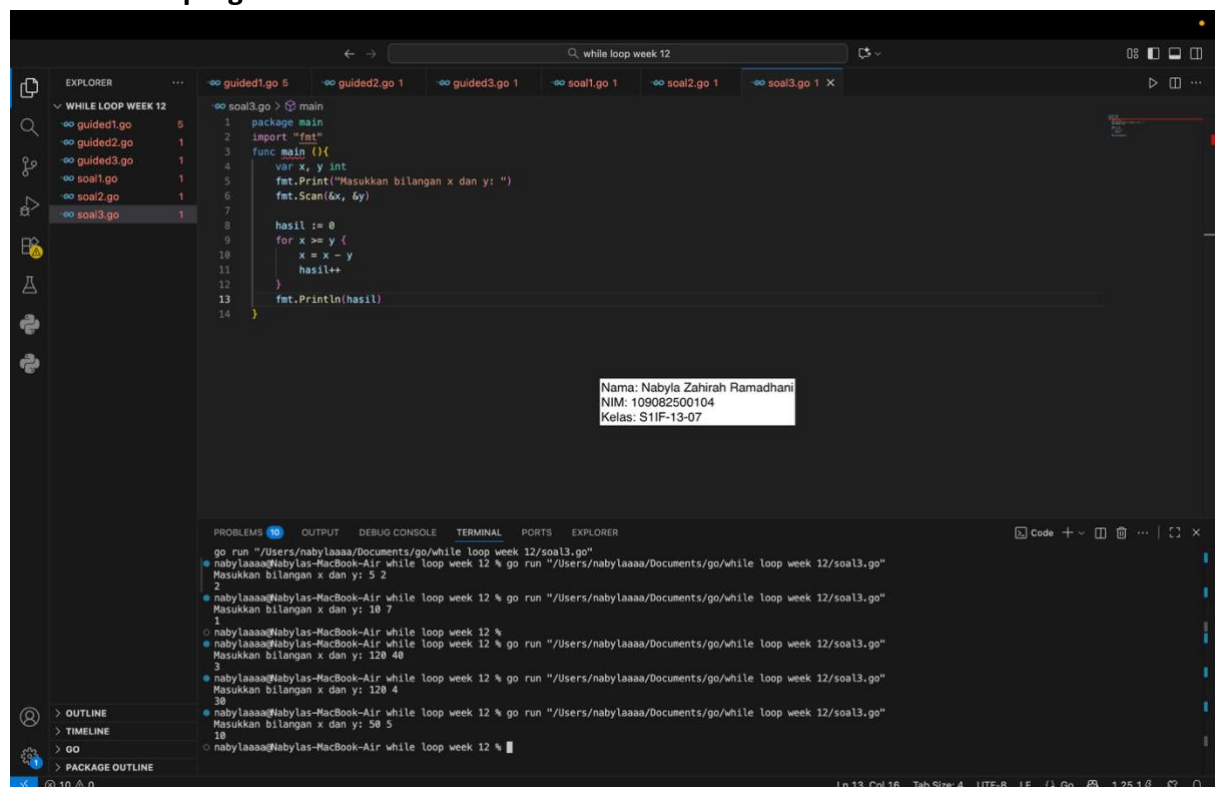
    hasil := 0

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

    fmt.Println(hasil)

}
```

#### Screenshoot program



#### Deskripsi program

Program ini untuk mencari hasil integer division dari dua bilangan, variabelnya x dan y yang bertipe int. Pertama program meminta input bilangan x dan y dari user menggunakan `fmt.Print("Masukkan bilangan x dan y: ")` dan akan dibaca

menggunakan *fmt.Scan(&x, &y)*. Setelah itu, program membuat variabel *hasil:=0* yang awalnya diset ke 0. Variabel inilah yang nantinya dipakai untuk menghitung berapa kali operasi bisa dilakukan. Program ini membuat perulangan *for x >= y*, yang artinya akan terus berjalan selama nilai *x* masih lebih besar atau sama dengan *y*. Di dalam loop, program akan menghitung  $x = x - y$ , lalu setiap kali pengurangan terjadi, hasil ditambah satu. Ketika akhirnya nilai *x* sudah lebih kecil dari *y*, *perulangannya berhenti*, dan program mencetak nilai hasil menggunakan *fmt.Println(hasil)*.