

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 12
WHILE-LOOP**



Disusun oleh:

Manggala Patra Raditya

109082500179

S1IF-13-02

Asisten Praktikum

Adithana dharma putra

Alfin Ilham Berlianto

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

2025

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {

    var n int

    fmt.Print("Masukan: ")

    fmt.Scan(&n)

    if n == 0 {

        fmt.Println(1)

        return

    }

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

        if i == 1 {

            fmt.Print(i)

        } else {

            fmt.Print(i, " x ")

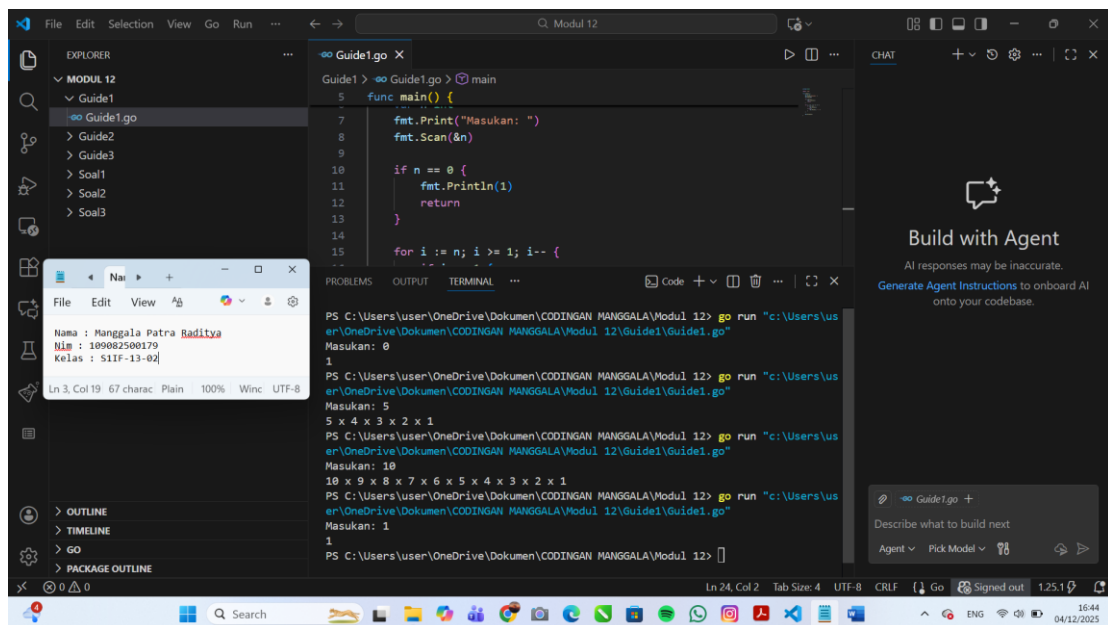
        }

    }

    fmt.Println()

}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor. The Explorer pane on the left shows a project structure with 'MODUL 12' containing 'Guide1.go', 'Guide2', 'Guide3', 'Soal1', 'Soal2', and 'Soal3'. The main editor displays the code for 'Guide1.go':

```
Guide1.go X
Guide1 > Guide1.go > main
5 func main() {
7     fmt.Print("Masukan: ")
8     fmt.Scan(&n)
9
10    if n == 0 {
11        fmt.Println(1)
12        return
13    }
14
15    for i := n; i >= 1; i-- {
16        // ...
17    }
18 }
```

The terminal at the bottom shows the execution of the program with the following output:

```
PS C:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 12\Guide1\Guide1.go"
Masukan: 0
1
PS C:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 12\Guide1\Guide1.go"
Masukan: 5
5 x 4 x 3 x 2 x 1
PS C:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 12\Guide1\Guide1.go"
Masukan: 10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
PS C:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 12\Guide1\Guide1.go"
Masukan: 1
1
PS C:\Users\user\OneDrive\Documents\CODINGAN MANGGALA\Modul 12>
```

A small window in the foreground shows a form with the following text:

```
File Edit View
Nama : Manggala Patra Raditya
NIM : 109082500179
Kelas : S1IF-13-02
```

Deskripsi program

Program yang dibuat bertujuan untuk menampilkan **deret faktorial** dari suatu bilangan bulat non-negatif yang dimasukkan oleh pengguna. Faktorial merupakan operasi matematika yang menyatakan hasil perkalian berurutan dari suatu bilangan dengan bilangan-bilangan di bawahnya hingga angka 1. Misalnya, faktorial dari 5 dituliskan sebagai $5 \times 4 \times 3 \times 2 \times 1$.

Pada program ini, pengguna diminta memasukkan sebuah nilai berupa bilangan bulat. Program kemudian melakukan pengecekan terhadap input tersebut. Apabila nilai yang dimasukkan adalah **0**, maka program langsung menampilkan hasil **1**, sesuai dengan definisi bahwa $0! = 1$.

Jika nilai yang dimasukkan lebih besar dari 0, program akan menggunakan perulangan **mundur** (descending loop) untuk mencetak deret faktorial mulai dari nilai input hingga angka 1. Setiap angka dicetak dengan format perkalian, dan operator \times hanya diberikan di antara angka-angka sehingga tampilan akhir menjadi lebih rapi. Setelah seluruh deret selesai dicetak, program menutup keluaran dengan baris baru.

Secara keseluruhan, program ini dibuat untuk melatih pemahaman mengenai input-output dasar, penggunaan kondisi (if), serta pemanfaatan perulangan dalam bahasa Go.

2. Guided 2

Source Code

```
package main

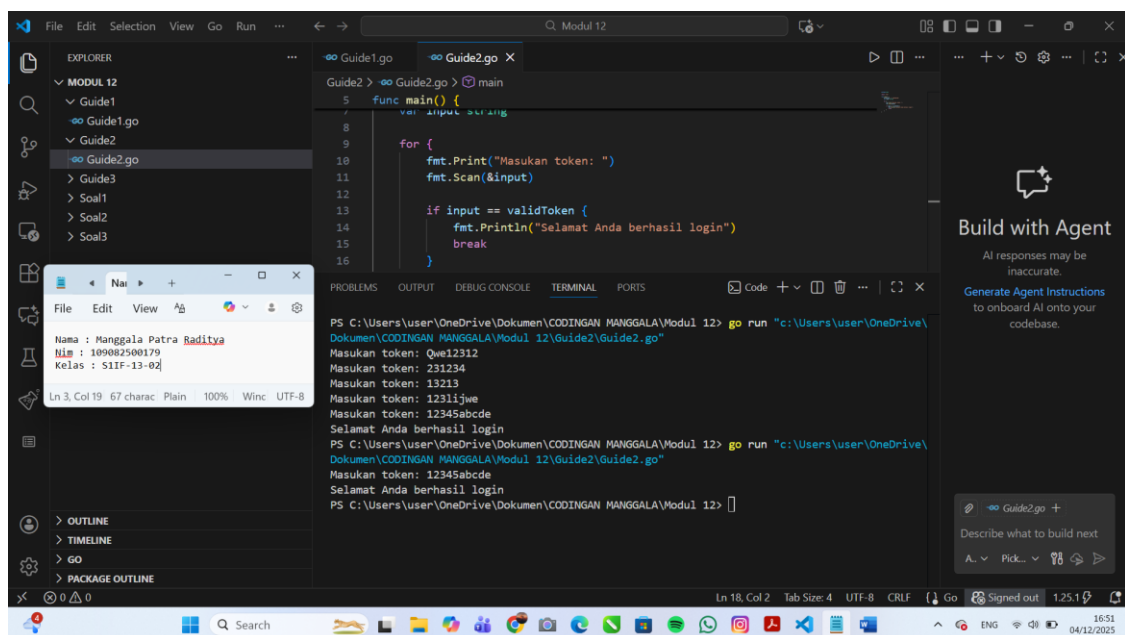
import "fmt"

func main() {
    validToken := "12345abcde"
    var input string

    for {
        fmt.Print("Masukan token: ")
        fmt.Scan(&input)

        if input == validToken {
            fmt.Println("Selamat Anda berhasil login")
            break
        }
    }
}
```

Screenshoot program



Deskripsi program

Program ini dibuat untuk mensimulasikan proses login sederhana menggunakan sebuah token. Pada dasarnya, pengguna diminta untuk memasukkan sebuah token, lalu program akan mengecek apakah token tersebut sudah benar atau belum. Token yang dianggap valid dalam program ini adalah **"12345abcde"**.

Saat program dijalankan, pengguna diminta memasukkan token pertama kali. Jika token yang dimasukkan ternyata salah, program tidak langsung berhenti, tetapi akan meminta input lagi secara berulang. Hal ini dilakukan menggunakan perulangan yang terus berjalan sampai pengguna memasukkan token yang benar.

Begitu token yang dimasukkan sama dengan token valid, program akan menampilkan pesan **"Selamat Anda berhasil login"** dan proses langsung dihentikan. Dengan begitu, program memastikan bahwa hanya input yang tepat yang dapat melewati proses login.

Program ini sederhana, tapi sudah menggambarkan konsep dasar login, pengecekan kondisi, dan penggunaan perulangan dalam bahasa Go.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var n int

    fmt.Print("Masukan: ")

    fmt.Scan(&n)


    a := 0

    b := 1


    fmt.Print(a, " ", b)


    for i := 3; i <= n; i++ {

        c := a + b

        fmt.Print(" ", c)

        a = b

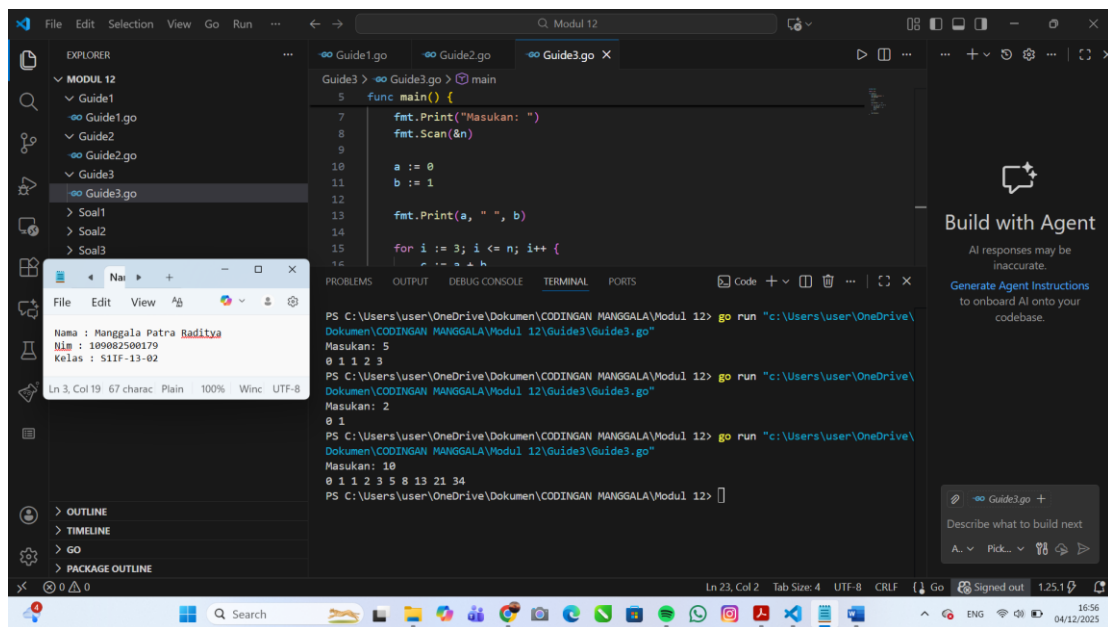
        b = c

    }


    fmt.Println()

}
```

Screenshoot program



The screenshot shows a Visual Studio Code editor with a Go project named 'MODUL 12'. The file explorer on the left shows a directory structure with 'Guide1.go', 'Guide2.go', and 'Guide3.go'. The main editor window displays the code for 'Guide3.go', which implements a Fibonacci sequence generator. The code prompts the user to input a value 'n' and then prints the first 'n' numbers of the Fibonacci sequence. The terminal window at the bottom shows the execution of the program, with the user inputting '5' and '2', and the program outputting the corresponding Fibonacci sequences: '0 1 1 2 3' for '5' and '0 1' for '2'.

```
func main() {  
    fmt.Println("Masukan: ")  
    fmt.Scan(&n)  
  
    a := 0  
    b := 1  
  
    fmt.Println(a, " ", b)  
  
    for i := 3; i <= n; i++ {  
        c := a + b  
        a = b  
        b = c  
        fmt.Print(c, " ")  
    }  
    fmt.Println()  
}
```

Terminal Output:

```
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\  
Dokumen\CODINGAN MANGGALA\Modul 12\Guide3\Guide3.go"  
Masukan: 5  
0 1 1 2 3  
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\  
Dokumen\CODINGAN MANGGALA\Modul 12\Guide3\Guide3.go"  
Masukan: 2  
0 1  
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\  
Dokumen\CODINGAN MANGGALA\Modul 12\Guide3\Guide3.go"  
Masukan: 10  
0 1 1 2 3 5 8 13 21 34  
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12>
```

Deskripsi program

Program ini digunakan untuk menampilkan N bilangan pertama dari deret Fibonacci. Dua angka awal deret ini adalah 0 dan 1, lalu setiap angka berikutnya merupakan hasil penjumlahan dua angka sebelumnya. Pengguna memasukkan nilai N (minimal 2), kemudian program langsung mencetak 0 dan 1, lalu menghitung angka-angka berikutnya menggunakan perulangan sampai jumlah yang diminta terpenuhi. Hasil akhirnya adalah deret Fibonacci sebanyak N angka yang ditampilkan secara berurutan.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    usernameBenar := "Admin"

    passwordBenar := "Admin"

    var u, p string

    gagal := 0

    for {

        fmt.Print("Masukan username dan password: ")

        fmt.Scan(&u, &p)

        if u == usernameBenar && p == passwordBenar {

            fmt.Println(gagal, "percobaan gagal login")

            break

        } else {

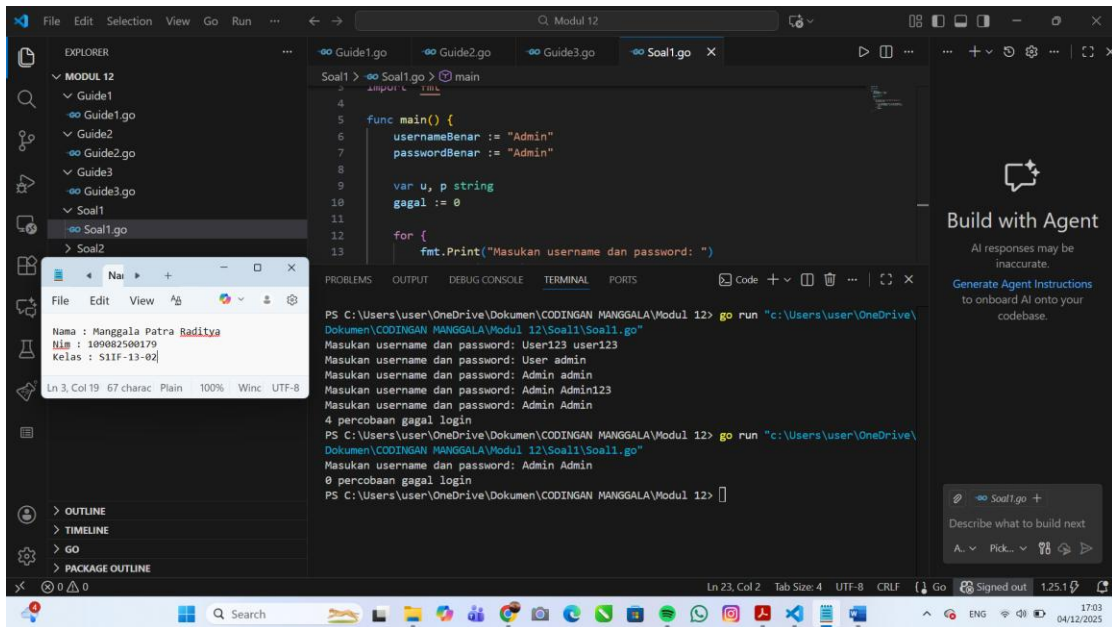
            gagal++

        }

    }

}
```


Screenshoot program



```
Soal1.go:5: func main() {
6:     usernameBenar := "Admin"
7:     passwordBenar := "Admin"
8:
9:     var u, p string
10:    gagal := 0
11:
12:    for {
13:        fmt.Print("Masukan username dan password: ")
14:        u = bufio.ReadString('\n')
15:        p = bufio.ReadString('\n')
16:        if u == "Admin" && p == "Admin" {
17:            fmt.Println("Login Berhasil")
18:            return
19:        } else {
20:            gagal++
21:            fmt.Println("Salah Username atau Password")
22:        }
23:    }
24:    fmt.Println("Gagal login", gagal, "kali")
25:}
```

```
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\
Dokumen\CODINGAN MANGGALA\Modul 12\Soal1\Soal1.go"
Masukan username dan password: User123 user123
Masukan username dan password: User admin
Masukan username dan password: Admin admin
Masukan username dan password: Admin Admin123
Masukan username dan password: Admin Admin
4 percobaan gagal login
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\
Dokumen\CODINGAN MANGGALA\Modul 12\Soal1\Soal1.go"
Masukan username dan password: Admin Admin
0 percobaan gagal login
PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12>
```

Deskripsi program

Program ini dibuat untuk menghitung berapa kali seseorang gagal login karena salah memasukkan username dan password. Pada awalnya, program menyiapkan username dan password yang benar, yaitu **"Admin"** untuk keduanya. Setelah itu, pengguna diminta memasukkan username dan password secara bersamaan. Jika input yang dimasukkan masih salah, program tidak langsung berhenti, tetapi terus meminta input ulang sambil menambahkan jumlah percobaan gagal. Proses ini berlangsung berulang-ulang sampai pengguna akhirnya memasukkan username dan password yang benar. Ketika login berhasil, program menampilkan berapa kali percobaan login sebelumnya yang gagal. Dengan cara ini, program dapat menunjukkan jumlah kesalahan yang terjadi sebelum pengguna benar-benar berhasil login.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var n int

    fmt.Scan(&n)

    for n > 0 {

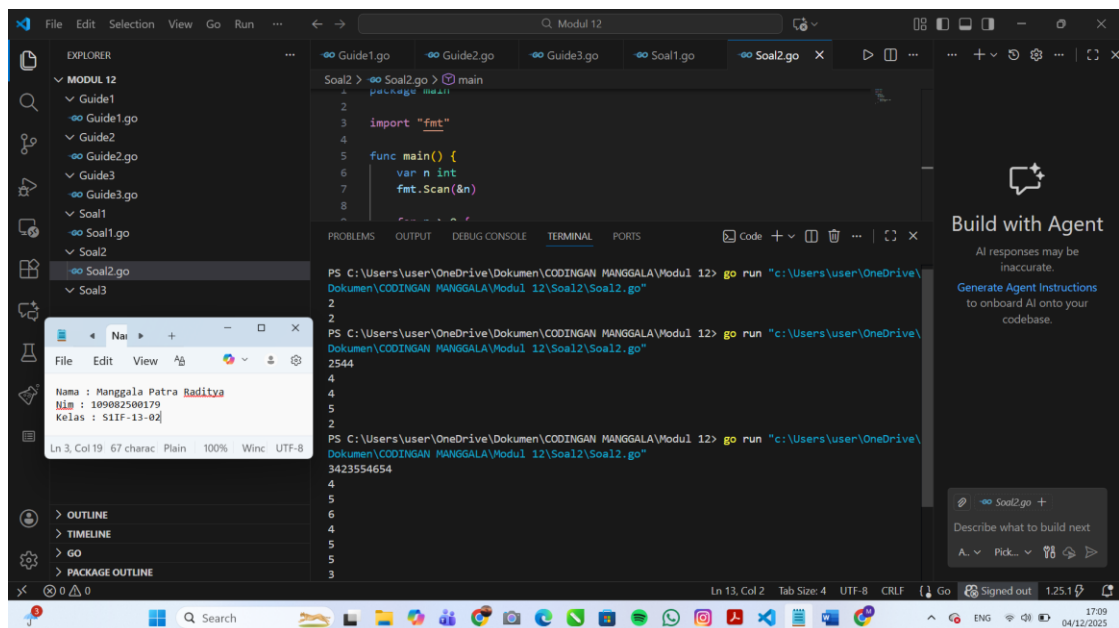
        fmt.Println(n % 10)

        n = n / 10

    }

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk mencacah dan menampilkan setiap digit yang terdapat di dalam suatu bilangan bulat positif. Bilangan yang dimasukkan akan diproses dari digit paling kanan (digit terakhir) menuju digit paling kiri (digit pertama).

Pertama, program membaca sebuah bilangan bulat positif dari input. Setelah itu, program melakukan proses pengambilan digit paling kanan dengan menggunakan operasi modulus ($n \% 10$). Digit tersebut kemudian langsung ditampilkan ke layar.

Selanjutnya, program membuang digit yang sudah diambil dengan cara membagi bilangan tersebut dengan 10 ($n = n / 10$). Proses ini dilakukan secara berulang sampai seluruh digit pada bilangan habis.

Dengan cara tersebut, setiap digit akan ditampilkan mulai dari digit terakhir sampai digit pertama, sesuai contoh keluaran yang diberikan.

3. Tugas 3

Source code

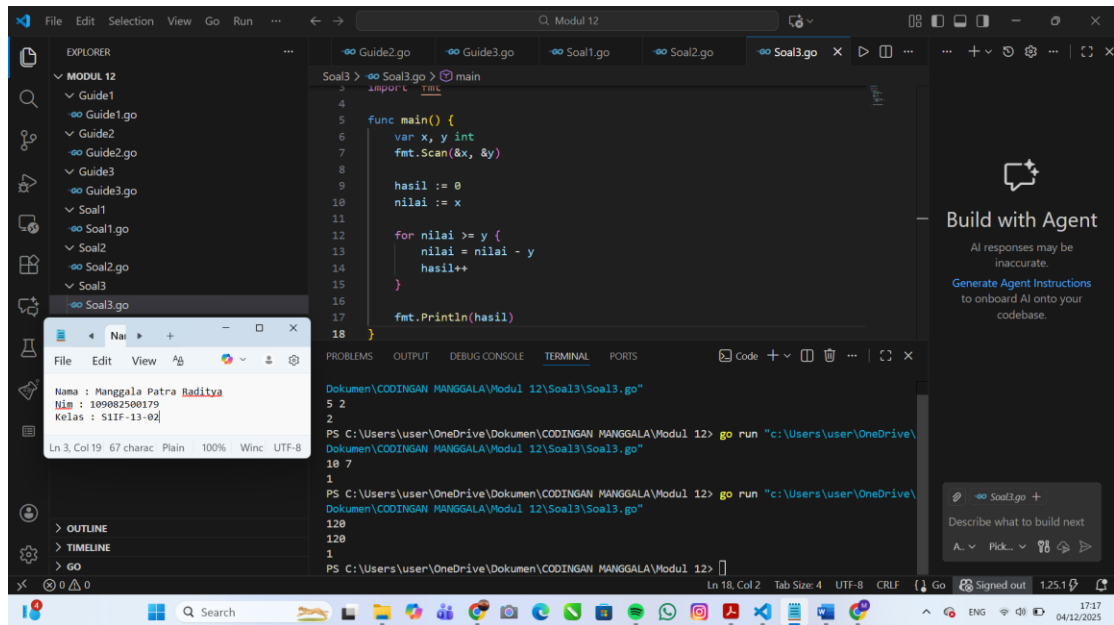
```
package main
import "fmt"
func main() {
    var x, y int
    fmt.Scan(&x, &y)

    hasil := 0
    nilai := x

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

    fmt.Println(hasil)
}
```

Screenshoot program



```
Soal3.go:1:1 main
4
5 func main() {
6     var x, y int
7     fmt.Scan(&x, &y)
8
9     hasil := 0
10    nilai := x
11
12    for nilai >= y {
13        nilai = nilai - y
14        hasil++
15    }
16
17    fmt.Println(hasil)
18 }
```

Build with Agent
AI responses may be inaccurate.
[Generate Agent Instructions](#)
to onboard AI onto your codebase.

PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12\Soal3\Soal3.go"

5 2

2

PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12\Soal3\Soal3.go"

10 7

1

PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12> go run "c:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12\Soal3\Soal3.go"

120

120

1

PS C:\Users\user\OneDrive\Dokumen\CODINGAN MANGGALA\Modul 12>

Deskripsi program

Program ini menghitung hasil *integer division* dari dua bilangan bulat positif tanpa memakai operator pembagian. Program membaca dua bilangan, yaitu x sebagai nilai yang dibagi dan y sebagai pembagi. Proses pembagian dilakukan dengan cara mengurangi x dengan y secara berulang. Setiap kali pengurangan terjadi, penghitung hasil ditambah satu. Ketika x tidak lagi cukup untuk dikurangi oleh y, perulangan berhenti, dan nilai penghitung itulah yang menjadi hasil pembagian. Hasil tersebut kemudian ditampilkan sebagai keluaran program.