

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 12  
WHILE-LOOP**



**Disusun oleh:**

**FAREL TRI JULIAN**

**109082500163**

**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

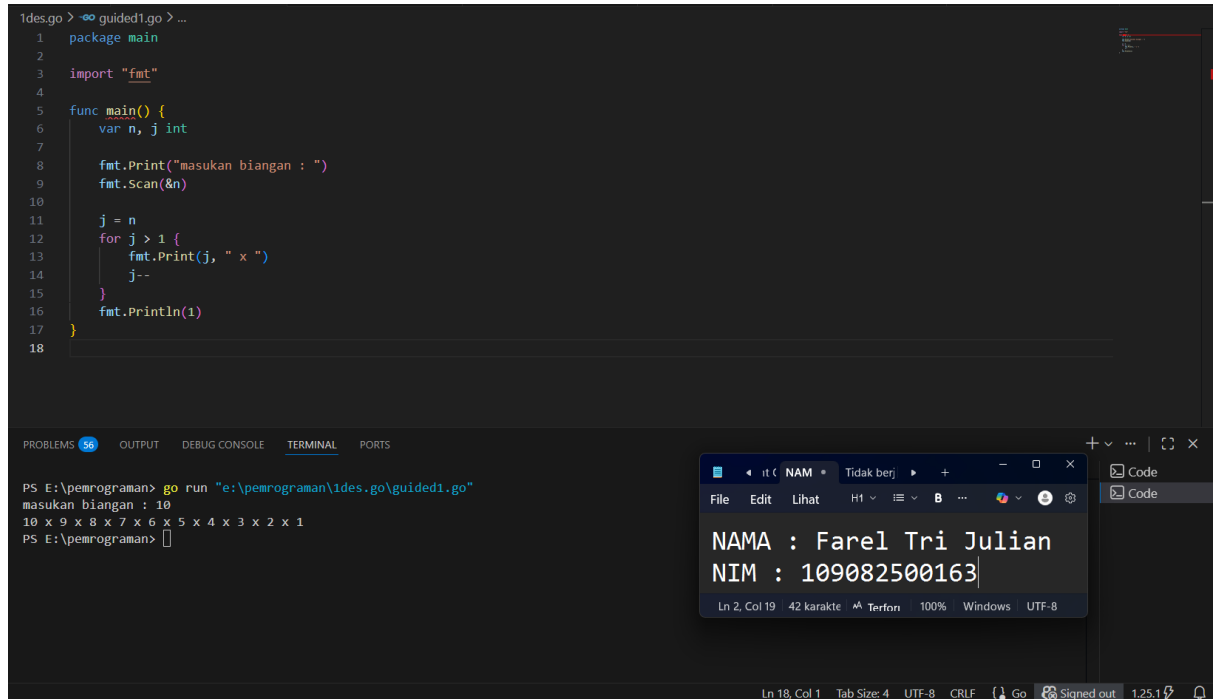
import "fmt"

func main() {
    var n, j int

    fmt.Print("masukan biangan : ")
    fmt.Scan(&n)

    j = n
    for j > 1 {
        fmt.Print(j, " x ")
        j--
    }
    fmt.Println(1)
}
```

## Screenshoot program



```
1des.go > go guided1.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n, j int
7
8     fmt.Print("masukan biangan : ")
9     fmt.Scan(&n)
10
11     j = n
12     for j > 1 {
13         fmt.Print(j, " x ")
14         j--
15     }
16     fmt.Println(1)
17 }
18
```

```
PS E:\pemrograman> go run "e:\pemrograman\ides.go\guided1.go"
masukan biangan : 10
10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1
PS E:\pemrograman>
```

NAMA : Farel Tri Julian  
NIM : 109082500163

## Deskripsi program

Program ini bertujuan untuk menampilkan urutan perkalian bilangan mundur angka yang diinputkan pengguna hingga angka 1, menyerupai representasi visual operasi faktorial. Program dimulai dengan membaca input bilangan bulat  $n$  ke dalam variabel  $j$ , kemudian menjalankan perulangan `for` (yang berfungsi sebagai `while-loop`) selama nilai  $j$  lebih besar dari 1; di dalam perulangan tersebut, program mencetak angka saat ini diikuti tanda " x " dan kemudian mengurangi nilai  $j$  sebesar satu (`decrement`). Setelah perulangan selesai, program mencetak angka 1 sebagai penutup baris, sehingga jika pengguna memasukkan angka 5, hasilnya akan tertulis sebagai 5 x 4 x 3 x 2 x 1.

## 2. Guided 2

### Source Code

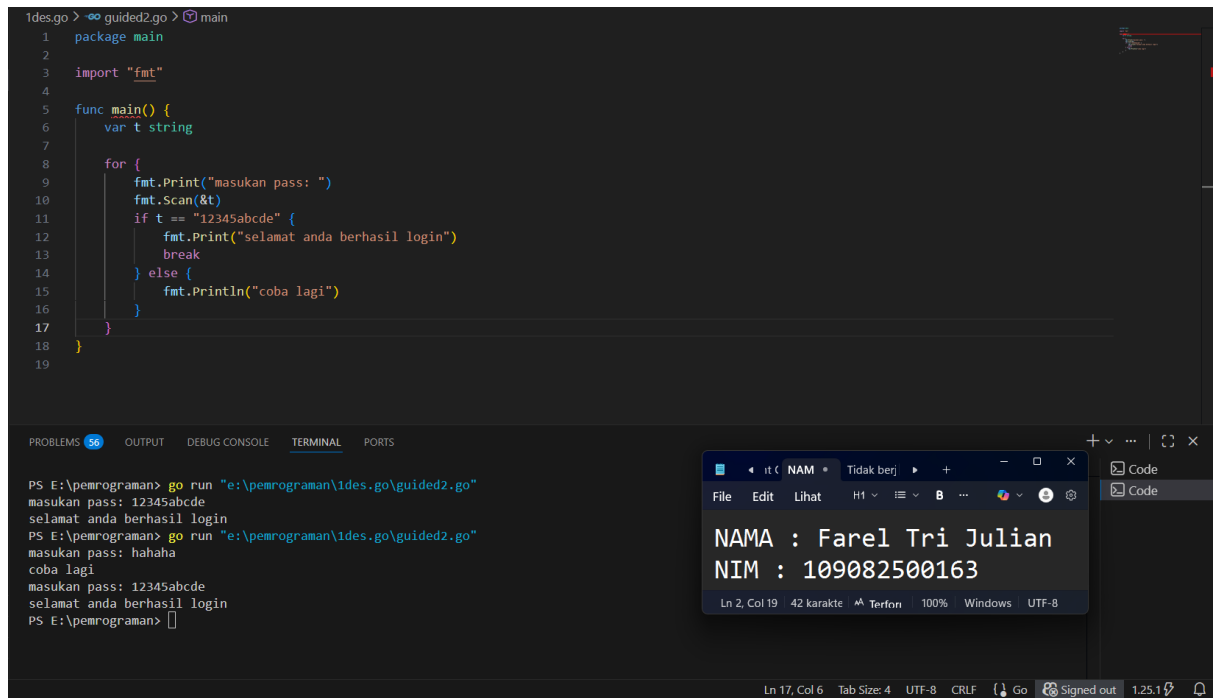
```
package main

import "fmt"

func main() {
    var t string

    for {
        fmt.Print("masukan pass: ")
        fmt.Scan(&t)
        if t == "12345abcde" {
            fmt.Print("selamat anda berhasil login")
            break
        } else {
            fmt.Println("coba lagi")
        }
    }
}
```

## Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The program is a simple login system using a loop to repeatedly prompt for a password until it matches the correct one.

```
1des.go > go guided2.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var t string
7
8     for {
9         fmt.Print("masukan pass: ")
10        fmt.Scan(&t)
11        if t == "12345abcde" {
12            fmt.Print("selamat anda berhasil login")
13            break
14        } else {
15            fmt.Println("coba lagi")
16        }
17    }
18 }
19
```

The terminal output shows the program being run and the user's input:

```
PS E:\pemrograman> go run "e:\pemrograman\1des.go\guided2.go"
masukan pass: 12345abcde
selamat anda berhasil login
PS E:\pemrograman> go run "e:\pemrograman\1des.go\guided2.go"
masukan pass: hahaha
coba lagi
masukan pass: 12345abcde
selamat anda berhasil login
PS E:\pemrograman>
```

A small window titled "NAM" is also visible, displaying the user's name and NIM:

```
NAMA : Farel Tri Julian
NIM : 109082500163
```

## Deskripsi program

Program tersebut mengimplementasikan sistem autentikasi sederhana menggunakan struktur perulangan tak terbatas untuk memvalidasi kata sandi. Di dalam blok for, program terus-menerus meminta input pengguna ke dalam variabel string t dan membandingkannya dengan nilai kunci "12345abcde"; jika input cocok, program mencetak pesan keberhasilan dan menggunakan perintah break untuk menghentikan perulangan, namun jika input tidak cocok, program akan mencetak pesan "coba lagi" dan kembali meminta input hingga kata sandi yang benar dimasukkan.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("masukan nilai: ")
    fmt.Scan(&n)

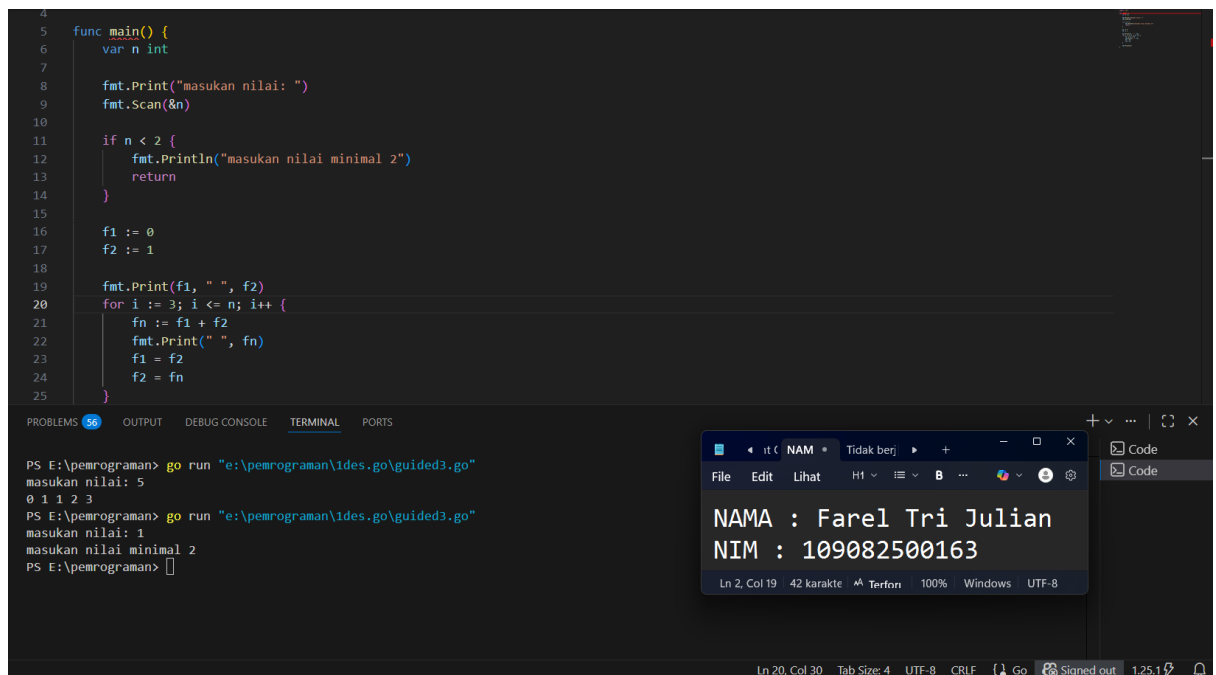
    if n < 2 {
        fmt.Println("masukan nilai minimal 2")
        return
    }

    f1 := 0
    f2 := 1

    fmt.Print(f1, " ", f2)
    for i := 3; i <= n; i++ {
        fn := f1 + f2
        fmt.Print(" ", fn)
        f1 = f2
        f2 = fn
    }

    fmt.Println()
}
```

## Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The program is a Fibonacci sequence generator. It prompts the user for a value 'n'. If 'n' is less than 2, it prints an error message and returns. Otherwise, it prints the first two numbers of the sequence (0 and 1) and then enters a loop from 3 to 'n' to calculate and print the subsequent Fibonacci numbers.

```
4
5 func main() {
6     var n int
7
8     fmt.Print("masukan nilai: ")
9     fmt.Scan(&n)
10
11     if n < 2 {
12         fmt.Println("masukan nilai minimal 2")
13         return
14     }
15
16     f1 := 0
17     f2 := 1
18
19     fmt.Print(f1, " ", f2)
20     for i := 3; i <= n; i++ {
21         fn := f1 + f2
22         fmt.Print(" ", fn)
23         f1 = f2
24         f2 = fn
25     }
```

The terminal output shows the program being run with 'n' as 5, resulting in the sequence '0 1 1 2 3'. A second run with 'n' as 1 shows the error message 'masukan nilai minimal 2'.

```
PS E:\pemrograman> go run "e:\pemrograman\ides.go\guided3.go"
masukan nilai: 5
0 1 1 2 3
PS E:\pemrograman> go run "e:\pemrograman\ides.go\guided3.go"
masukan nilai: 1
masukan nilai minimal 2
PS E:\pemrograman>
```

Overlaid on the terminal is a small window titled 'NAMA' containing the text: 'NAMA : Farel Tri Julian' and 'NIM : 109082500163'.

## Deskripsi program

Program ini dirancang untuk menghasilkan dan menampilkan deret bilangan Fibonacci sebanyak  $n$  suku berdasarkan input pengguna. Setelah memvalidasi bahwa input minimal bernilai 2, program menginisialisasi dan mencetak dua angka pertama dari deret tersebut (0 dan 1), kemudian menggunakan perulangan for mulai dari suku ketiga hingga suku ke- $n$  untuk menghitung angka berikutnya dengan menjumlahkan dua angka sebelumnya ( $f1 + f2$ ), lalu mencetak hasilnya secara berurutan sembari memperbarui nilai variabel untuk iterasi selanjutnya.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    correctuser := "Admin"
    correctpass := "Admin"
    var ipass, iuss string
    failed := 0

    for {
        fmt.Print("masukan ussername dan pass : ")
        fmt.Scan(&ipass, &iuss)

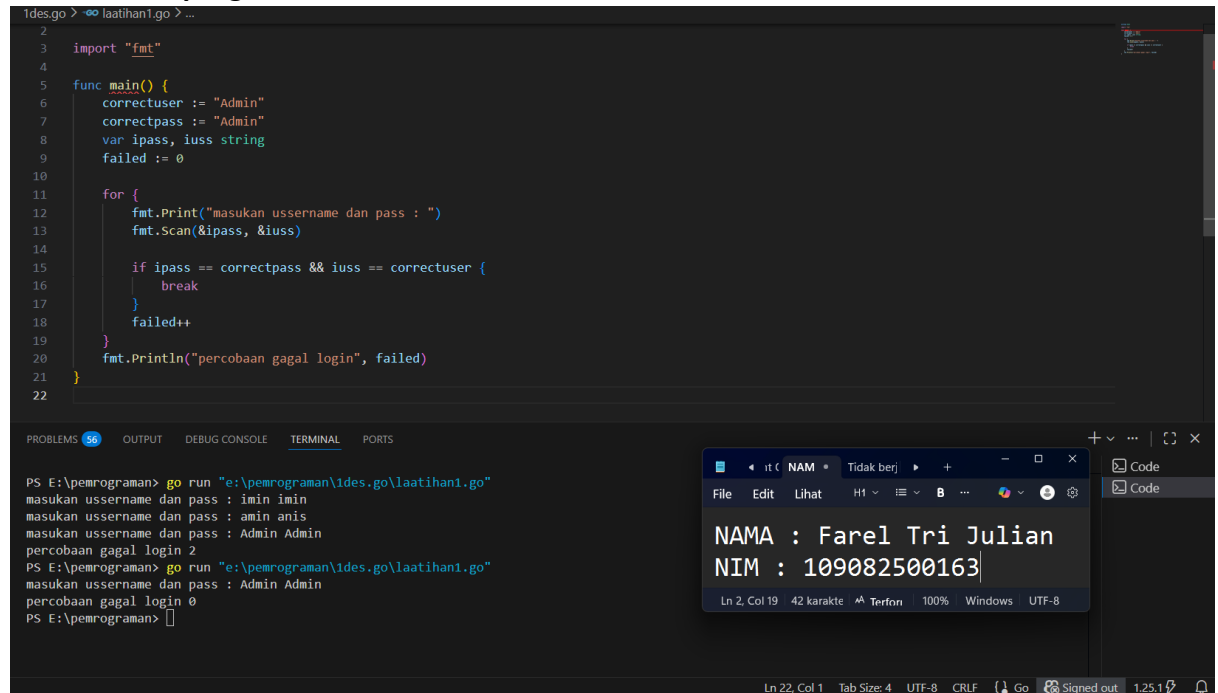
        if ipass == correctpass && iuss == correctuser {
            break
        }

        failed++
    }

    fmt.Println("percobaan gagal login", failed)
}
```



## Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The program is a simple login simulation that prompts the user for a username and password. It has a hardcoded correct username and password of "Admin". It uses a loop to allow multiple login attempts, counting the number of failed attempts. If the user enters the correct credentials, the loop breaks. Otherwise, it increments the failed count. After the loop, it prints the total number of failed login attempts.

```
1des.go > laatihan1.go > ...
2
3 import "fmt"
4
5 func main() {
6     correctuser := "Admin"
7     correctpass := "Admin"
8     var ipass, iuss string
9     failed := 0
10
11     for {
12         fmt.Print("masukan ussername dan pass : ")
13         fmt.Scan(&ipass, &iuss)
14
15         if ipass == correctpass && iuss == correctuser {
16             break
17         }
18         failed++
19     }
20     fmt.Println("percobaan gagal login", failed)
21 }
22
```

The terminal output shows the program being run from a PowerShell prompt. The user enters "imin imin" and "amin anis", both of which result in "percobaan gagal login 2". Then, the user enters "Admin Admin", which results in "percobaan gagal login 0".

```
PS E:\pemrograman> go run "e:\pemrograman\ides.go\laatihan1.go"
masukan ussername dan pass : imin imin
masukan ussername dan pass : amin anis
masukan ussername dan pass : Admin Admin
percobaan gagal login 2
PS E:\pemrograman> go run "e:\pemrograman\ides.go\laatihan1.go"
masukan ussername dan pass : Admin Admin
percobaan gagal login 0
PS E:\pemrograman>
```

## Deskripsi program

Program ini berfungsi sebagai simulasi login sederhana yang melacak jumlah kegagalan autentikasi pengguna sebelum akhirnya berhasil masuk. Program menginisialisasi kredensial yang valid (username dan password "Admin") dan variabel penghitung kegagalan, kemudian menjalankan perulangan tak terbatas yang meminta pengguna memasukkan dua data, jika kedua input tersebut cocok dengan kredensial yang benar, perintah break akan menghentikan perulangan, namun jika tidak, variabel penghitung failed akan bertambah satu di setiap iterasi. Setelah loop berhenti (login berhasil), program menampilkan total angka percobaan login yang gagal ke layar.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Masukan: ")
    fmt.Scan(&n)

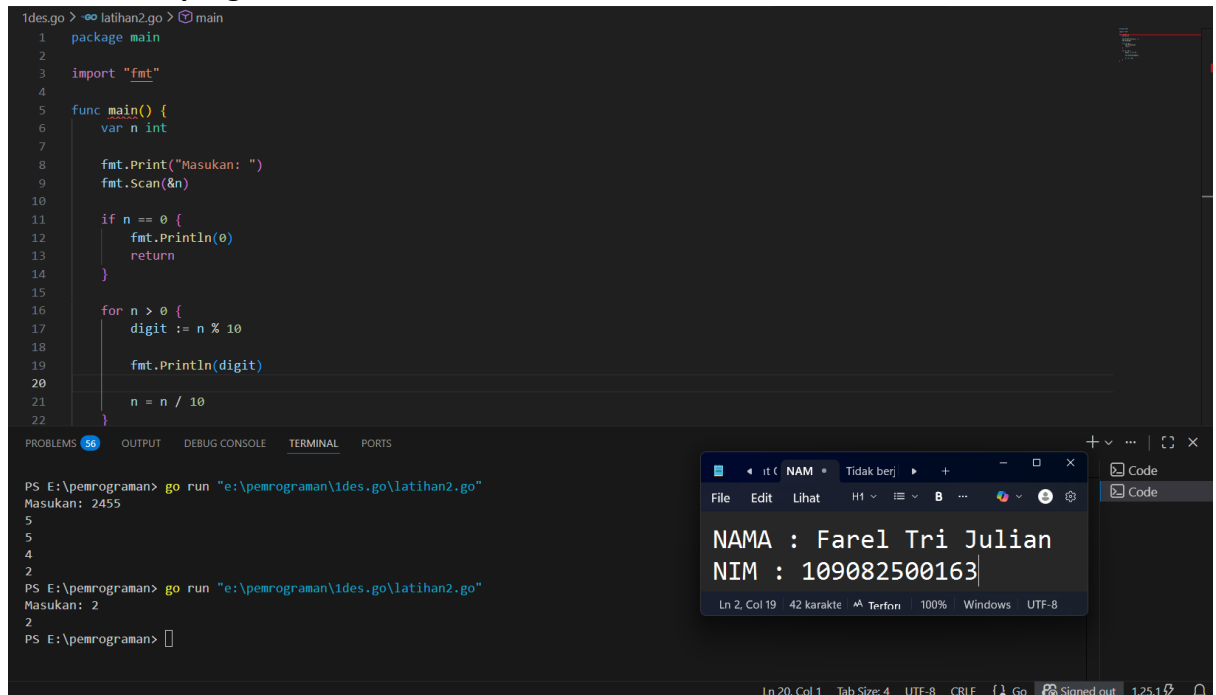
    if n == 0 {
        fmt.Println(0)
        return
    }

    for n > 0 {
        digit := n % 10

        fmt.Println(digit)

        n = n / 10
    }
}
```

## Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The program is designed to reverse a number by extracting its digits one by one using a loop.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7
8     fmt.Print("Masukan: ")
9     fmt.Scan(&n)
10
11     if n == 0 {
12         fmt.Println(0)
13         return
14     }
15
16     for n > 0 {
17         digit := n % 10
18
19         fmt.Println(digit)
20
21         n = n / 10
22     }
```

The terminal shows the program being run with the input 2455, which outputs the digits 5, 4, 2, and 5 in reverse order. A second run with input 2 outputs 2.

```
PS E:\pemrograman> go run "e:\pemrograman\ides.go\latihan2.go"
Masukan: 2455
5
4
2
5
PS E:\pemrograman> go run "e:\pemrograman\ides.go\latihan2.go"
Masukan: 2
2
PS E:\pemrograman>
```

On the right, a small window displays the user's name and NIM: NAMA : Farel Tri Julian, NIM : 109082500163.

## Deskripsi program

Program ini bertujuan untuk memecah dan menampilkan setiap digit dari sebuah bilangan bulat positif secara terbalik menggunakan operasi matematika dalam sebuah perulangan. Setelah menerima input bilangan  $n$  dan menangani kasus khusus untuk angka 0, program memasuki blok perulangan `for` (yang berfungsi sebagai `while-loop`) yang berjalan selama nilai  $n$  positif di setiap iterasi, program mengekstraksi digit terakhir menggunakan operasi modulus ( $n \% 10$ ), mencetaknya ke baris baru, dan kemudian memangkas digit tersebut dari bilangan utama menggunakan pembagian integer ( $n / 10$ ) hingga seluruh digit selesai diproses.

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

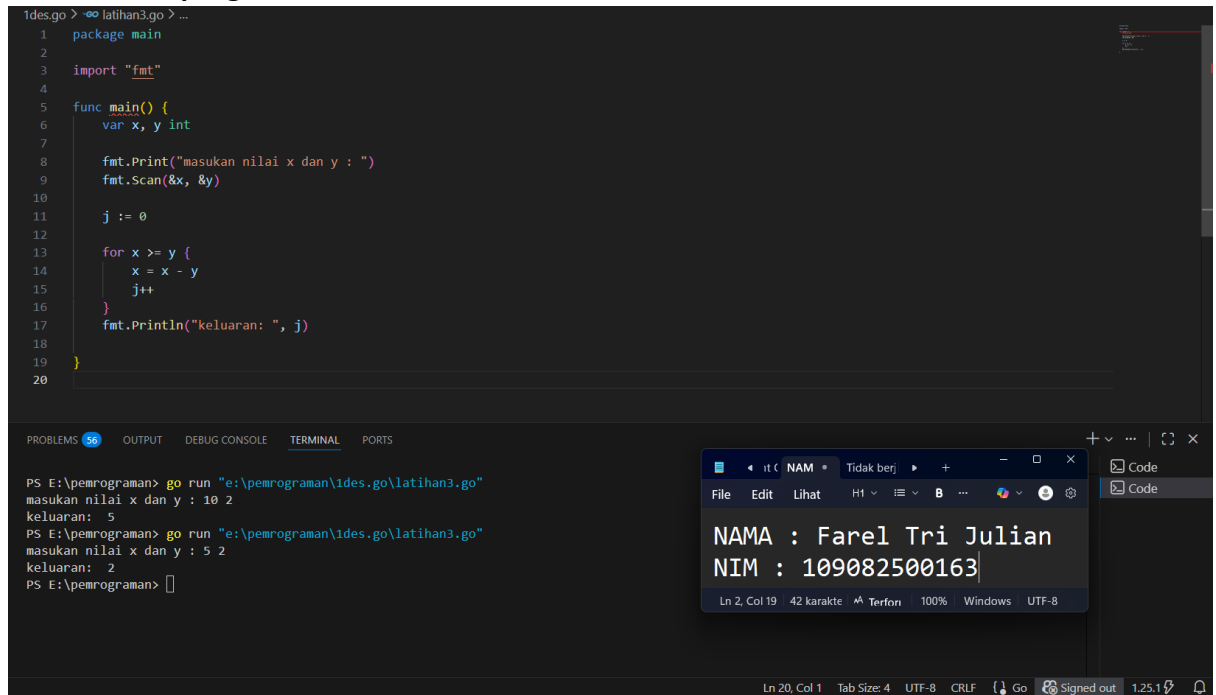
func main() {
    var x, y int

    fmt.Print("masukan nilai x dan y : ")
    fmt.Scan(&x, &y)

    j := 0

    for x >= y {
        x = x - y
        j++
    }
    fmt.Println("keluaran: ", j)
}
```

## Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x, y int
7
8     fmt.Print("masukan nilai x dan y : ")
9     fmt.Scan(&x, &y)
10
11     j := 0
12
13     for x >= y {
14         x = x - y
15         j++
16     }
17     fmt.Println("keluaran: ", j)
18 }
19
20
```

PROBLEMS 56 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\pemrograman> go run "e:\pemrograman\ides.go\latihan3.go"
masukan nilai x dan y : 10 2
keluaran: 5
PS E:\pemrograman> go run "e:\pemrograman\ides.go\latihan3.go"
masukan nilai x dan y : 5 2
keluaran: 2
PS E:\pemrograman>
```

NAMA : Farel Tri Julian  
NIM : 109082500163

Ln 2, Col 19 42 karakter Terfor 100% Windows UTF-8 Signed out 1.25.1

## Deskripsi program

Program ini berfungsi untuk menghitung hasil pembagian bilangan bulat dari dua angka input, x dan y, dengan menerapkan metode pengurangan berulang sebagai pengganti operator pembagian standar. Setelah membaca kedua nilai input, program menjalankan perulangan for yang terus mengurangi nilai x dengan y selama x masih lebih besar atau sama dengan y, sembari mencatat berapa kali pengurangan tersebut terjadi menggunakan variabel penghitung j. Ketika perulangan berhenti (karena sisa x sudah kurang dari y), program menampilkan nilai j yang merepresentasikan hasil bagi dari operasi tersebut.