

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 12
WHILE-LOOP**



Disusun oleh:

BAGUS IRSYAD KAMAL

109082500215

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 int

    fmt.Print("Masukan bilangan Bulat non Negatif :")

    fmt.Scan(&n)


    if n == 0 {

        fmt.Println("1")

        return

    }

    i := n

    for {

        fmt.Print(i)

        if i > 1 {

            fmt.Print(" X ")

        }

        i--

        if i == 0 {

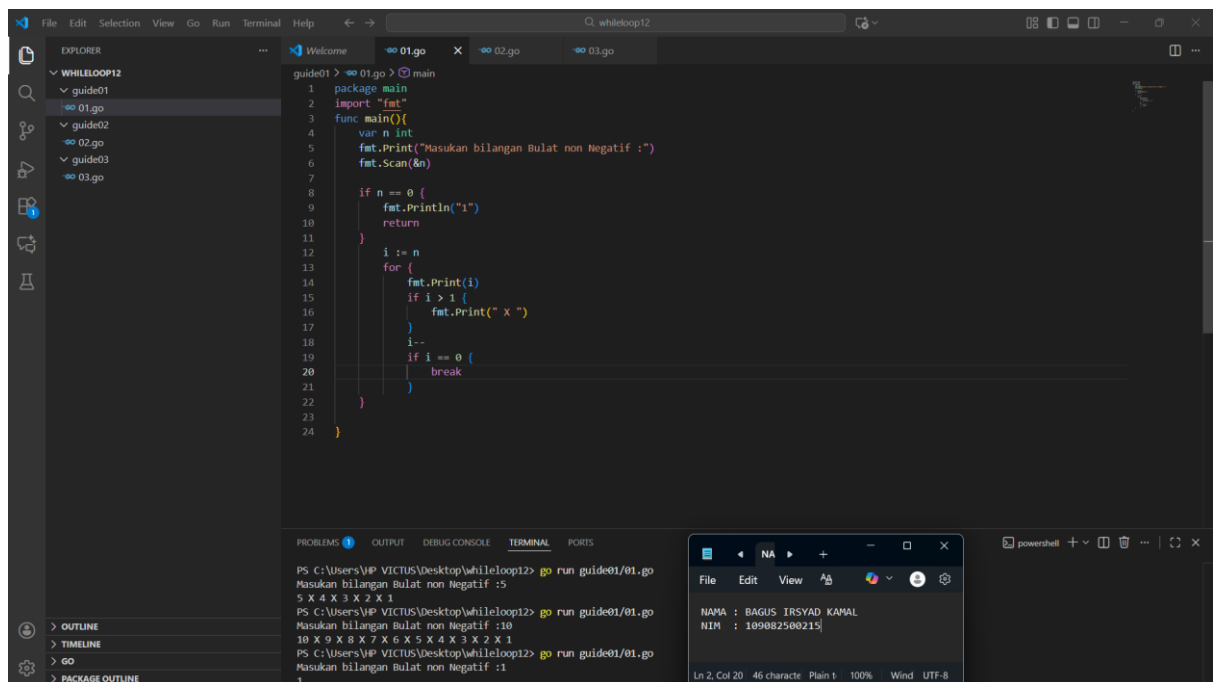
            break

        }

    }

}
```

Screenshoot program



```
1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Print("Masukan bilangan Bulat non Negatif :")
6     fmt.Scan(&n)
7
8     if n == 0 {
9         fmt.Println("1")
10        return
11    }
12    i := n
13    for {
14        fmt.Print(i)
15        if i > 1 {
16            fmt.Print(" X ")
17        }
18        i--
19        if i == 0 {
20            break
21        }
22    }
23 }
24 }
```

PS C:\Users\VP VICTUS\Desktop\whileloop12> go run guide01/01.go
Masukan bilangan Bulat non Negatif :5
5 X 4 X 3 X 2 X 1

PS C:\Users\VP VICTUS\Desktop\whileloop12> go run guide01/01.go
Masukan bilangan Bulat non Negatif :10
10 X 9 X 8 X 7 X 6 X 5 X 4 X 3 X 2 X 1

PS C:\Users\VP VICTUS\Desktop\whileloop12> go run guide01/01.go
Masukan bilangan Bulat non Negatif :1
1

Deskripsi program

Program ini digunakan untuk mencetak deret mundur dari sebuah bilangan bulat positif hingga 1, dengan setiap angka dipisahkan oleh tulisan " X ", kecuali angka terakhir. Program menerima satu input berupa bilangan bulat non-negatif. Ketika program dijalankan, pertama-tama ia membaca nilai input. Jika nilai input adalah 0, program langsung mencetak "1" sebagai hasil dan berhenti, karena tidak ada angka untuk dihitung mundur. Jika nilai input lebih besar dari 0, program mulai mencetak angka dari nilai tersebut turun satu per satu. Setiap kali mencetak angka, jika angka tersebut masih lebih besar dari 1, program menambahkan " X " sebagai pemisah. Proses ini berulang sampai angka mencapai 1, lalu program berhenti. Tujuan dari program ini adalah membuat deret angka menurun dengan format yang rapi dan konsisten.

Contoh 1

Input: 5

Output: 5 X 4 X 3 X 2 X 1

Contoh 2

Input: 10

Output: 10 X 9 X 8 X 7 X 6 X 5 X 4 X 3 X 2 X 1

Contoh 3

Input: 1

Output: 1

Contoh 4

Input: 0

Output: 1

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var token string

    fmt.Print("Masukkan Kode Password : ")

    fmt.Scan(&token)

    for token != "12345abcde" {

        fmt.Scan(&token)

    }

    fmt.Println("Selamat Anda berhasil login")

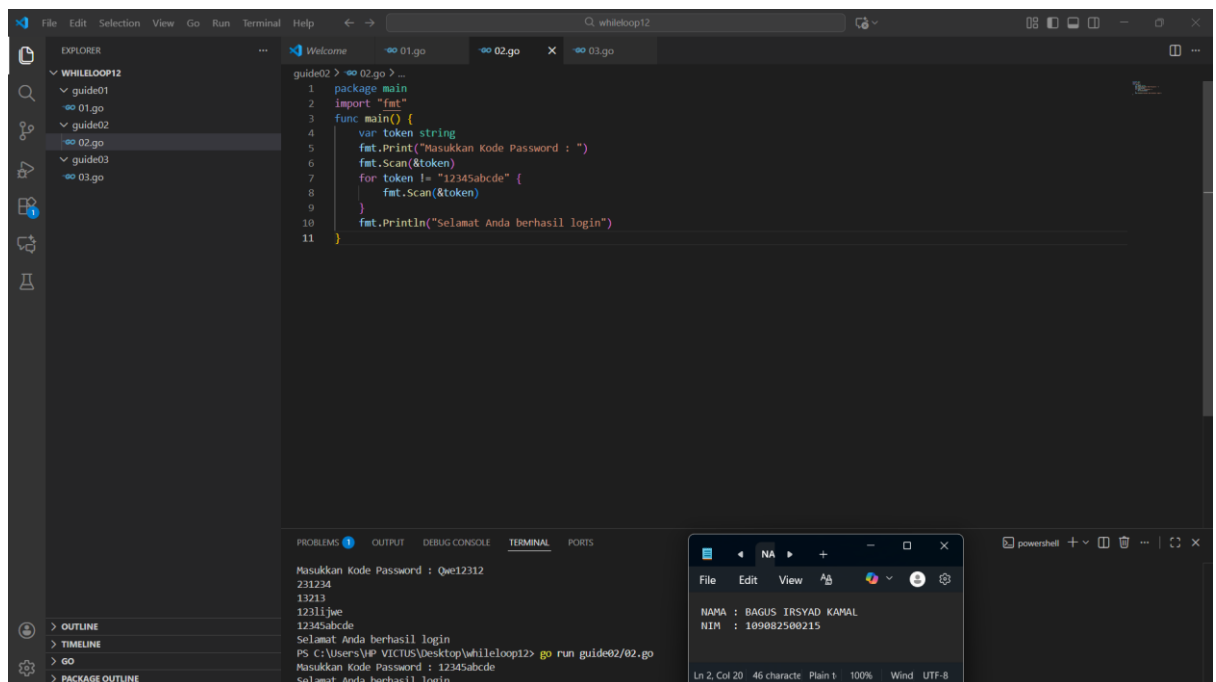
}

    fmt.Println("bukan huruf")

}

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk melakukan proses verifikasi password sederhana. Program meminta pengguna memasukkan sebuah kode password, kemudian memeriksa apakah password tersebut sudah benar. Program menerima satu input berupa teks string.

Ketika program dijalankan, ia pertama kali meminta pengguna memasukkan password. Jika password yang dimasukkan tidak sama dengan nilai yang benar, yaitu "12345abcde", program akan meminta pengguna untuk memasukkannya kembali. Proses ini berlangsung terus menerus selama password masih salah.

Setelah pengguna akhirnya memasukkan password yang benar, program menghentikan proses pengulangan dan menampilkan pesan "Selamat Anda berhasil login" sebagai tanda bahwa verifikasi telah berhasil.

Tujuan dari program ini adalah melakukan pengecekan password berulang hingga user memasukkan input yang benar.

Contoh 1

Input dari pengguna: Qwe12312

Program menolak.

User memasukkan lagi: 231234

Program menolak.

User mencoba lagi: 13213

Program menolak.

User mencoba lagi: 123ljiwe

Program menolak.

User akhirnya memasukkan: 12345abcde

Output: Selamat Anda berhasil login

Contoh 2

Input: 12345abcde

Output: Selamat Anda berhasil login

3. Guided 3

Source Code

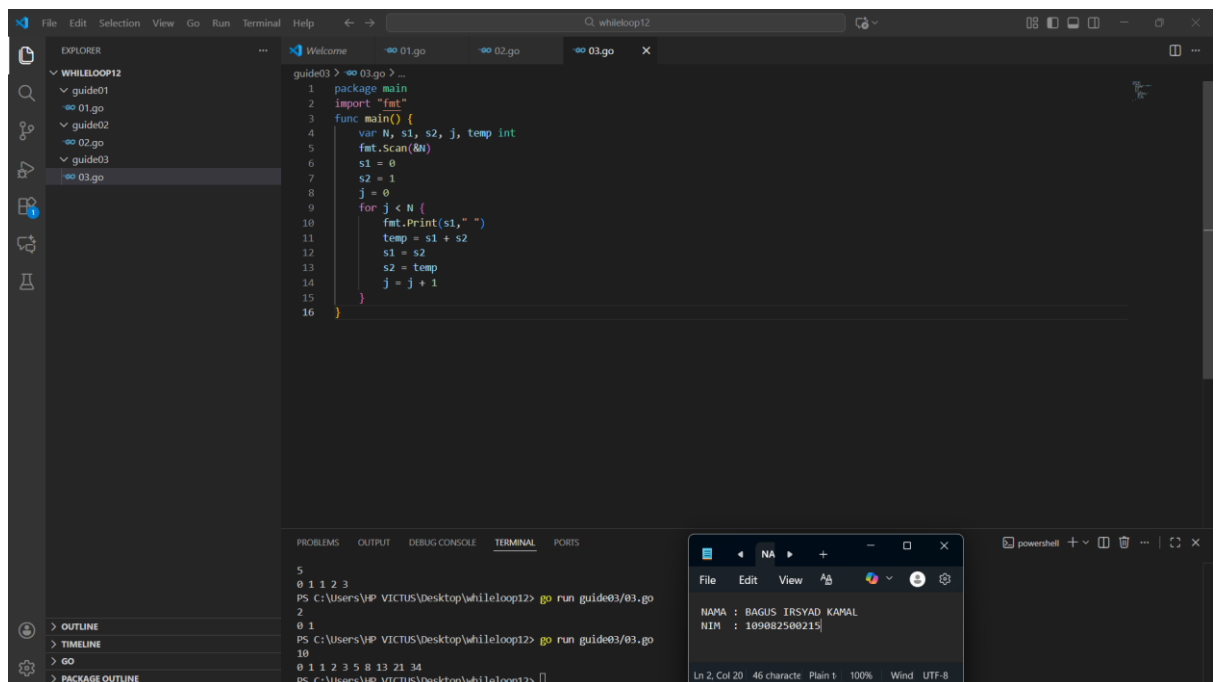
```
package main

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 project named 'WHILELOOP12'. The Explorer panel on the left shows a file structure with 'guide01', 'guide02', and 'guide03'. The main editor displays the code for 'guide03.go', which is a Go program that prints the first N numbers of the Fibonacci sequence. The code is as follows:

```
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 }
```

The bottom panel shows the output of the program, which is the first 5 numbers of the Fibonacci sequence: 0 1 1 2 3. The terminal also shows the command to run the program: 'go run guide03/03.go'.

Deskripsi program

Program ini digunakan untuk menampilkan deret Fibonacci sebanyak jumlah data yang diminta oleh pengguna. Program menerima satu input berupa bilangan bulat N, yaitu jumlah angka Fibonacci yang ingin ditampilkan.

Ketika program dijalankan, dua nilai awal Fibonacci disiapkan, yaitu 1 dan 0. Nilai pertama yang dicetak adalah s1 (nilai 1). Setelah mencetak angka pertama, program menghitung angka Fibonacci berikutnya dengan cara menjumlahkan s1 dan s2. Hasil penjumlahan tersebut disimpan dalam variabel sementara sebelum memperbarui nilai s1 dan s2. Proses ini terus berulang sampai jumlah angka yang dicetak mencapai nilai N.

Tujuan dari program ini adalah menampilkan deret Fibonacci secara berurutan sebanyak N angka, dimulai dari angka pertama yaitu 1.

Contoh 1

Input: 3

Program mencetak tiga angka pertama dari deret Fibonacci.

Output: 1 1 2

Contoh 2

Input: 5

Program mencetak lima angka pertama dari deret Fibonacci.

Output: 1 1 2 3 5

Contoh 3

Input: 10

Program mencetak sepuluh angka Fibonacci berurutan.

Output: 1 1 2 3 5 8 13 21 34 55

TUGAS

1. Tugas 1

Source code

```
package main

import (
    "fmt"
)

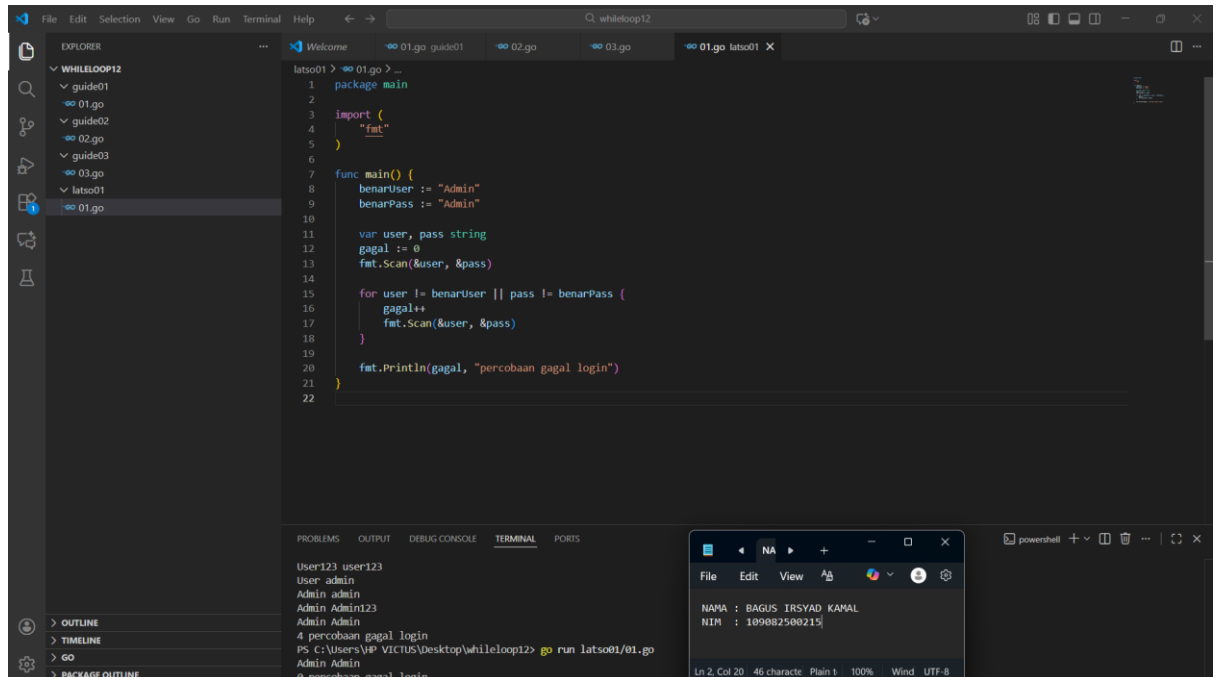
func main() {
    benarUser := "Admin"
    benarPass := "Admin"

    var user, pass string
    gagal := 0
    fmt.Scan(&user, &pass)

    for user != benarUser || pass != benarPass {
        gagal++
        fmt.Scan(&user, &pass)
    }

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

Screenshoot program



```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     benarUser := "Admin"
9     benarPass := "Admin"
10
11     var user, pass string
12     gagal := 0
13     fmt.Scan(&user, &pass)
14
15     for user != benarUser || pass != benarPass {
16         gagal++
17         fmt.Scan(&user, &pass)
18     }
19
20     fmt.Println(gagal, "percobaan gagal login")
21 }
22
```

Terminal output:

```
User123 user123
User admin
Admin admin
Admin Admin123
Admin Admin
4 percobaan gagal login
PS C:\Users\VP VICTUS\Desktop\whileloop12> go run latso01/01.go
Admin Admin
0 percobaan gagal login
```

Deskripsi program

Program ini digunakan untuk menghitung berapa kali seorang pengguna gagal melakukan login karena salah memasukkan username atau password. Program menerima pasangan input berupa username dan password pada setiap percobaan login.

Ketika program dijalankan, nilai username dan password yang benar diset sebagai "Admin" dan "Admin". Program menggunakan while loop (dalam Go menggunakan for dengan kondisi) untuk meminta pengguna memasukkan username dan password berulang-ulang hingga keduanya benar.

Setiap kali pengguna memasukkan username atau password yang salah, program menambahkan satu ke variabel penghitung percobaan gagal. Jika username dan password sudah benar, program berhenti dan menampilkan jumlah percobaan gagal login yang telah dilakukan sebelum berhasil.

Tujuan program ini adalah menampilkan jumlah percobaan gagal login yang dilakukan pengguna sampai login berhasil.

Contoh input dan output

Contoh 1

Input:

User123 user123

User admin

Admin admin1

Admin Admin123

Admin Admin

Output:

4 percobaan gagal login

Contoh 2

Input:

Admin Admin

Output:

0 percobaan gagal login

2. Tugas 2

Source code

```
package main

import (

    "fmt"

)

func main() {

    var n int

    fmt.Scan(&n)

    for n > 0 {

        digit := n % 10

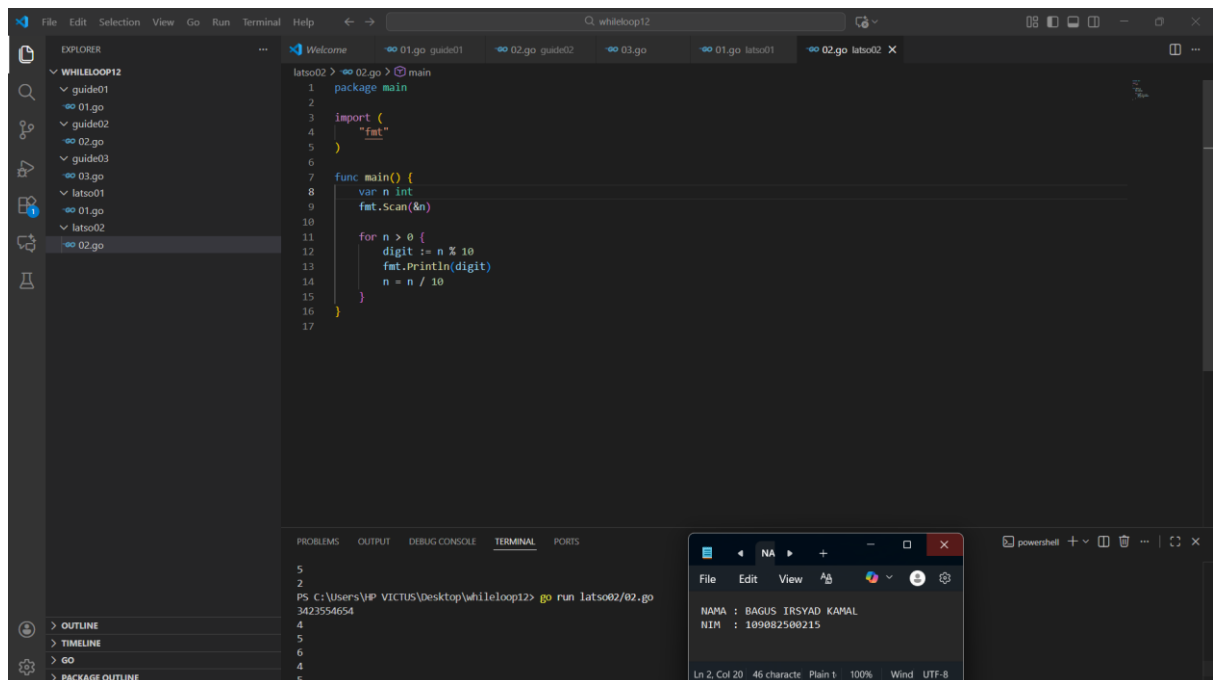
        fmt.Println(digit)

        n = n / 10

    }

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk mencacah atau menampilkan setiap digit yang terdapat di dalam suatu bilangan bulat positif. Program menerima satu input berupa sebuah bilangan bulat positif.

Ketika program dijalankan, program mengambil digit paling kanan dari bilangan tersebut dengan operasi sisa pembagian 10. Digit tersebut langsung ditampilkan sebagai keluaran. Setelah itu, bilangan dibagi 10 (tanpa koma) untuk menghapus digit terakhir. Proses ini diulang menggunakan while loop hingga nilai bilangan menjadi nol.

Dengan cara ini, program menampilkan digit-digit bilangan mulai dari digit paling kanan (terakhir) hingga digit paling kiri (pertama).

Tujuan program ini adalah menampilkan seluruh digit dari bilangan bulat positif dalam urutan terbalik, yaitu dari digit terakhir ke digit pertama.

Contoh input dan output

Contoh 1

Input:

2

Output:

2

Contoh 2

Input:

2544

Output:

4

4

5

2

Contoh 3

Input:

3423554654

Output:

4

5

6

4

5

5

3

2

4

3

Tugas 3

Source code

```
package main

import (
    "fmt"
)

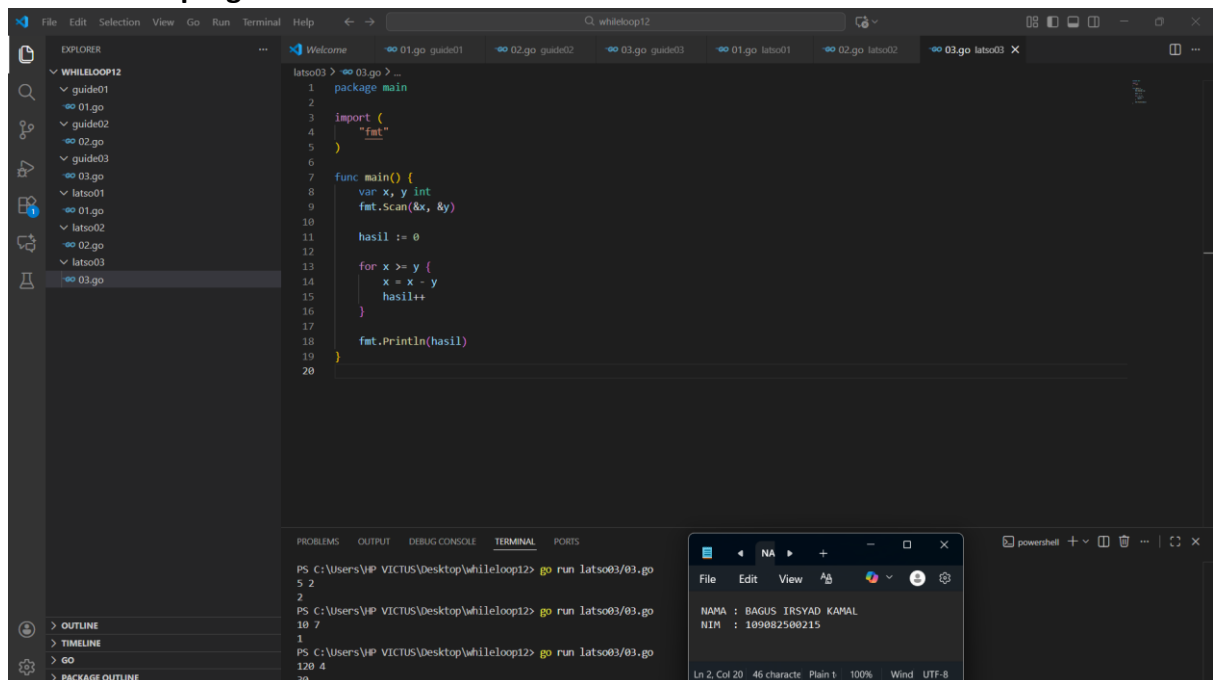
func main() {
    var x, y int
    fmt.Scan(&x, &y)

    hasil := 0

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

    fmt.Println(hasil)
}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk mencari hasil integer division (pembagian bulat) dari dua bilangan bulat positif x dan y tanpa menggunakan operator pembagian. Program menerima dua input, yaitu x dan y , dengan syarat $x \geq y$.

Ketika program dijalankan, proses pembagian dilakukan menggunakan perulangan. Program terus mengurangi nilai x dengan y secara berulang, dan setiap pengurangan dihitung sebagai satu kali pembagian. Perulangan berhenti ketika nilai x sudah lebih kecil daripada y . Jumlah pengurangan itulah yang menjadi hasil dari $x \text{ div } y$.

Dengan kata lain, program menentukan berapa kali y dapat dikurangkan dari x sebelum x menjadi kurang dari y . Itulah hasil pembagian bulat tanpa menggunakan operator pembagian.

Tujuan program ini adalah menampilkan hasil dari operasi $x \text{ div } y$, dilakukan melalui proses pengurangan berulang.

Contoh input dan output

Contoh 1

Input:

5 2

Output:

2

Contoh 2

Input:

10 7

Output:

1

Contoh 3

Input:

120 4

Output:

30