

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 12
WHILE - LOOP**



Disusun oleh:

RAFIF ARFA DHIAPRAJA

109082500041

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 int

    fmt.Print("Masukkan sebuah Bilangan Non Negatif: ")

    fmt.Scan(&n)

    for n > 1 {

        fmt.Print(n, " x ")

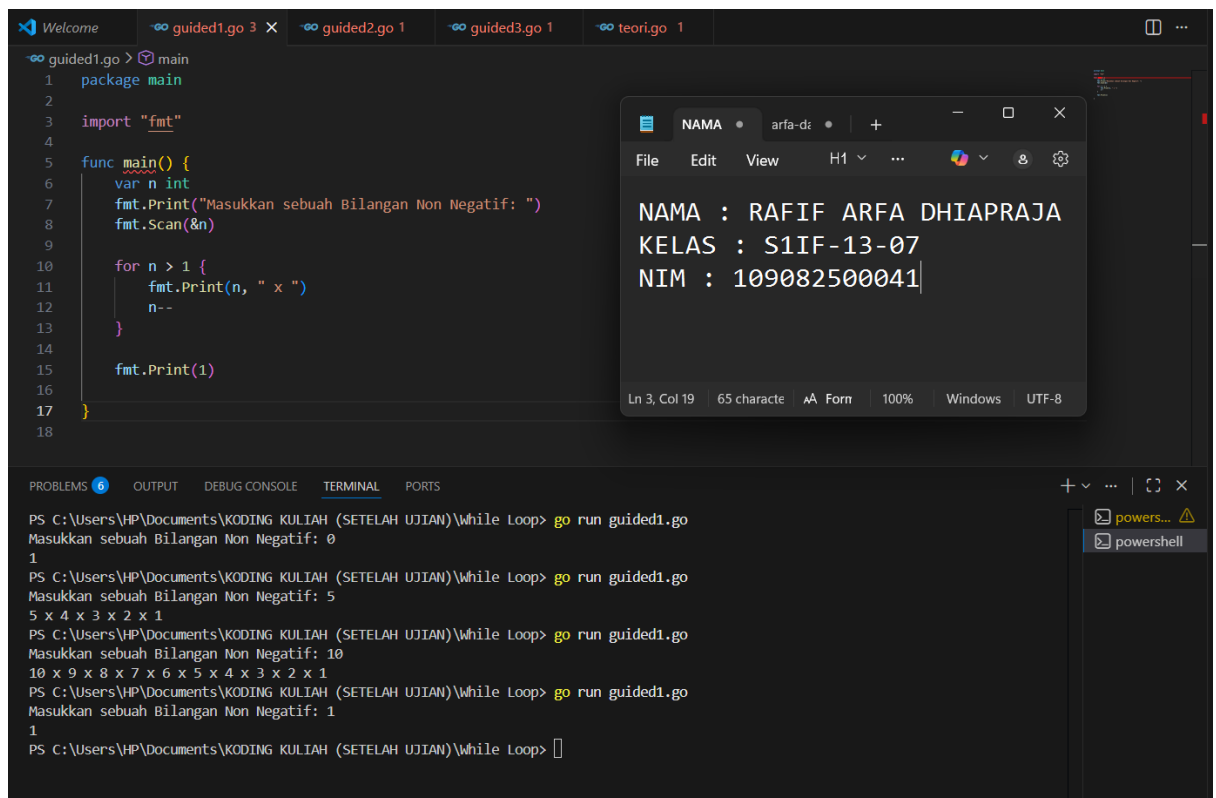
        n--

    }

    fmt.Print(1)

}
```

Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("Masukkan sebuah Bilangan Non Negatif: ")
8     fmt.Scan(&n)
9
10    for n > 1 {
11        fmt.Print(n, " x ")
12        n--
13    }
14
15    fmt.Print(1)
16
17 }
18
```

```
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop> go run guided1.go
Masukkan sebuah Bilangan Non Negatif: 0
1
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop> go run guided1.go
Masukkan sebuah Bilangan Non Negatif: 5
5 x 4 x 3 x 2 x 1
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop> go run guided1.go
Masukkan sebuah Bilangan 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\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop> go run guided1.go
Masukkan sebuah Bilangan Non Negatif: 1
1
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop>
```

Deskripsi program

Pada program ini, saya membuat sebuah program sederhana yang berfungsi untuk menampilkan urutan faktorial dari sebuah bilangan non-negatif yang dimasukkan oleh pengguna. Program ini tidak menghitung nilai faktorial secara numerik, tetapi hanya menampilkan bentuk perkaliannya saja, misalnya "5 x 4 x 3 x 2 x 1". Program diawali dengan mendeklarasikan sebuah variabel bertipe int, yaitu n, yang digunakan untuk menyimpan nilai bilangan yang dimasukkan oleh pengguna. Setelah deklarasi variabel, program menampilkan pesan "Masukkan sebuah Bilangan Non Negatif:" menggunakan fmt.Print. Pesan ini berfungsi sebagai pemberitahuan bahwa pengguna diminta memasukkan sebuah angka yang akan menjadi dasar perhitungan deret faktorial. Nilai yang dimasukkan oleh pengguna kemudian dibaca dan disimpan ke dalam variabel n menggunakan fmt.Scan(&n). Setelah nilai diperoleh, program masuk ke bagian utama yaitu proses mencetak urutan faktorial. Untuk menjalankan proses tersebut, program menggunakan sebuah perulangan for dengan kondisi $n > 1$. Kondisi ini memastikan bahwa selama nilai n masih lebih besar dari 1, program akan terus mencetak angka saat ini diikuti oleh simbol " x ". Pada setiap iterasi, nilai n dikurangi satu menggunakan n--, sehingga deret akan terus menurun secara bertahap. Perulangan ini berhenti ketika nilai n mencapai angka 1. Setelah keluar dari loop, program mencetak angka 1 sebagai elemen terakhir dari deret faktorial. Dengan demikian, program dapat menghasilkan output seperti: 5 x 4 x 3 x 2 x 1 ketika pengguna memasukkan angka 5.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var token string

    fmt.Scan(&token)

    for token != "12345abcde" {

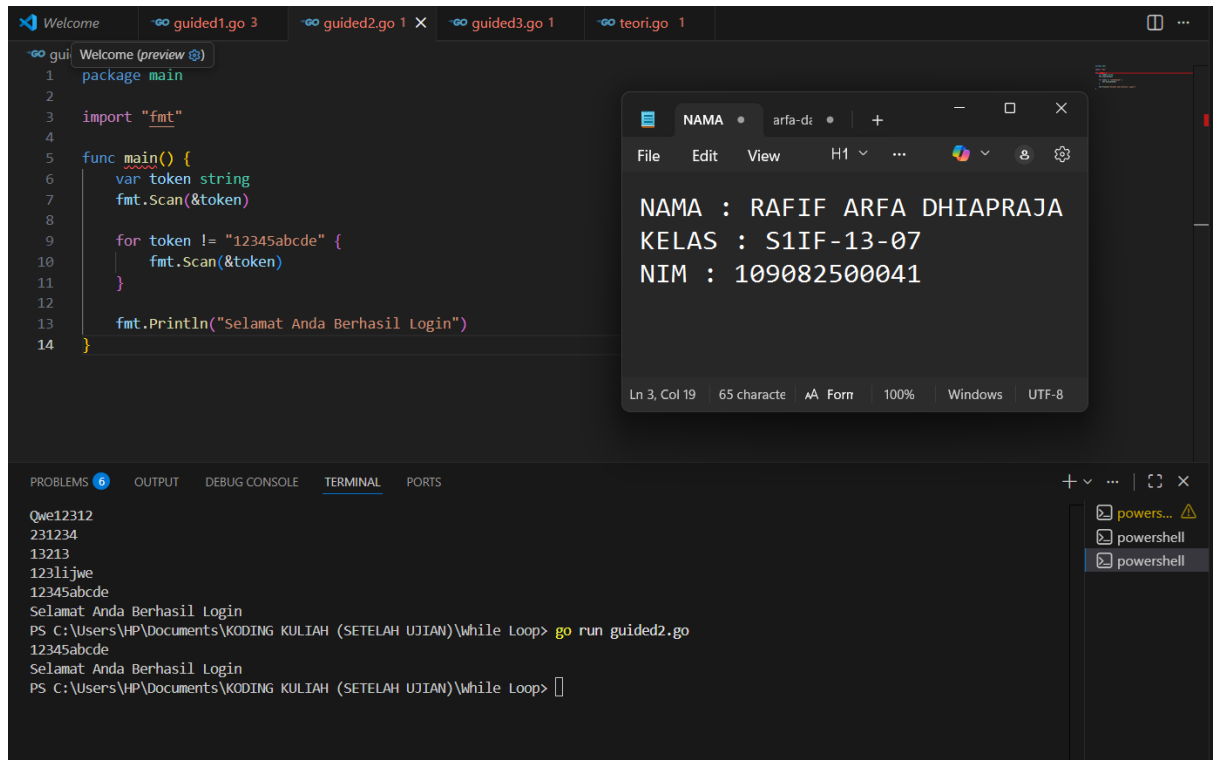
        fmt.Scan(&token)

    }

    fmt.Println("Selamat Anda Berhasil Login")

}
```

Screenshoot program



The screenshot shows a Go program in VS Code. The editor displays the following code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var token string
7     fmt.Scan(&token)
8
9     for token != "12345abcde" {
10        fmt.Scan(&token)
11    }
12
13    fmt.Println("Selamat Anda Berhasil Login")
14 }
```

The terminal output shows the program's execution:

```
Qwe12312
231234
13213
1231ijwe
12345abcde
Selamat Anda Berhasil Login
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop> go run guided2.go
12345abcde
Selamat Anda Berhasil Login
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop>
```

A separate window titled 'NAMA' displays the user's input details:

```
NAMA : RAFIF ARFA DHIAPRAJA
KELAS : S1IF-13-07
NIM : 109082500041
```

Deskripsi program

Pada program ini dibuat sebuah mekanisme login sederhana menggunakan token yang dimasukkan oleh pengguna. Program diawali dengan mendeklarasikan variabel token yang bertipe string. Variabel ini berfungsi untuk menampung input dari pengguna yang akan digunakan sebagai bahan pengecekan. Setelah variabel disiapkan, program langsung membaca token pertama yang dimasukkan menggunakan `fmt.Scan(&token)`. Setelah menerima input awal, program masuk ke proses validasi dengan menggunakan sebuah perulangan `for` yang memiliki kondisi `token != "12345abcde"`. Kondisi tersebut berarti bahwa selama token yang dimasukkan tidak sama dengan token yang benar, pengguna harus terus menginput ulang token. Pada setiap iterasi, program menjalankan `fmt.Scan(&token)` kembali, sehingga pengguna diberikan kesempatan untuk memasukkan token yang benar. Perulangan ini akan berhenti secara otomatis ketika pengguna berhasil memasukkan token yang tepat, yaitu "12345abcde". Ketika kondisi terpenuhi dan loop selesai, program kemudian menampilkan pesan "Selamat Anda Berhasil Login" sebagai konfirmasi bahwa proses login telah berhasil dan token yang diberikan sudah valid.

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

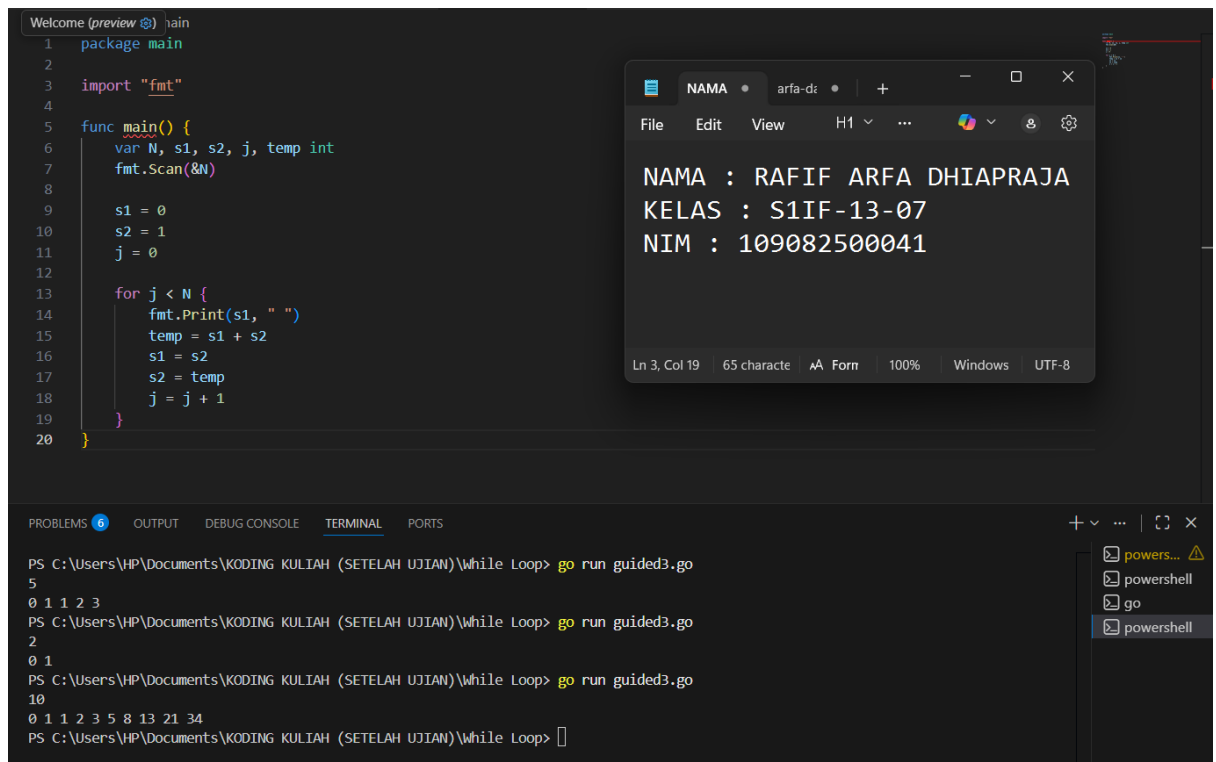
        s2 = temp

        j = j + 1

    }

}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor. The program is a simple implementation of the Fibonacci sequence. It uses a loop to calculate and print the sequence up to a user-defined number N. The output window shows the sequence: 0 1 1 2 3. The terminal window shows the command to run the program and the output.

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

Output:

```
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop> go run guided3.go
0 1 1 2 3
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop> go run guided3.go
0 1
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop> go run guided3.go
0 1 1 2 3 5 8 13 21 34
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\While Loop>
```

Deskripsi program

Program ini merupakan sebuah implementasi sederhana untuk menampilkan deret Fibonacci berdasarkan jumlah elemen yang diminta oleh pengguna. Program dimulai dengan deklarasi beberapa variabel bertipe integer, yaitu N, s1, s2, j, dan temp. Variabel N digunakan untuk menyimpan banyaknya bilangan Fibonacci yang ingin ditampilkan oleh pengguna, sedangkan s1 dan s2 digunakan sebagai dua bilangan awal deret Fibonacci. Setelah variabel disiapkan, program membaca input N menggunakan `fmt.Scan(&N)`. Nilai ini menentukan berapa banyak angka dalam deret Fibonacci yang akan dicetak. Program kemudian menginisialisasi s1 dengan nilai 0 dan s2 dengan nilai 1 sebagai dua angka pertama dalam deret Fibonacci. Variabel j diatur menjadi 0 sebagai penghitung iterasi. Proses utama program berada dalam perulangan for yang berjalan selama $j < N$. Pada setiap iterasi, program menampilkan nilai s1 sebagai bilangan Fibonacci berikutnya. Selanjutnya, program menghitung nilai Fibonacci berikutnya dengan menjumlahkan s1 dan s2, lalu menyimpannya ke variabel temp. Setelah itu, nilai s1 diperbarui menjadi nilai sebelumnya dari s2, dan s2 diperbarui menjadi nilai temp. Perubahan ini mengikuti prinsip dasar deret Fibonacci, yaitu setiap bilangan merupakan hasil penjumlahan dua bilangan sebelumnya. Variabel j kemudian ditambah satu untuk melanjutkan ke iterasi berikutnya.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var username, password string

    fmt.Print("Masukkan Username & Password: ")

    fmt.Scan(&username, &password)

    jumlah := 0

    for username != "Admin" || password != "Admin" {

        jumlah = jumlah + 1

        fmt.Print("Username & Password Salah! Masukkan  
Kembali: ")

        fmt.Scan(&username, &password)

    }

    fmt.Printf("%d Percobaan gagal", jumlah)

}
```


Screenshoot program

The screenshot shows a Go program in VS Code and its execution in the terminal. The program is a simple login simulation. It prompts the user for a username and password. If the input is incorrect (not 'Admin'), it increments a counter and asks for the credentials again. After four failed attempts, it displays the number of failed attempts.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var username, password string
7     fmt.Print("Masukkan Username & Password: ")
8     fmt.Scan(&username, &password)
9
10    jumlah := 0
11
12    for username != "Admin" || password != "Admin" {
13        jumlah = jumlah + 1
14        fmt.Print("Username & Password Salah! Masukkan Kembali: ")
15        fmt.Scan(&username, &password)
16    }
17
18    fmt.Printf("%d Percobaan gagal", jumlah)
19 }
```

The terminal output shows the program being run and the user's input. The first three attempts fail, and the fourth attempt is successful. The final output is "4 Percobaan gagal".

```
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Latsol 12> go run latsol1.go
Masukkan Username & Password: user123 user123
Username & Password Salah! Masukkan Kembali: User Admin
Username & Password Salah! Masukkan Kembali: Admin admin
Username & Password Salah! Masukkan Kembali: Admin Admin123
Username & Password Salah! Masukkan Kembali: Admin Admin
4 Percobaan gagal
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Latsol 12> go run latsol1.go
Masukkan Username & Password: Admin Admin
0 Percobaan gagal
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Latsol 12>
```

Deskripsi program

Program ini dibuat untuk mensimulasikan proses login sederhana menggunakan username dan password. Di awal program, pengguna diminta memasukkan kedua data tersebut melalui perintah "Masukkan Username & Password:". Nilai yang dimasukkan kemudian disimpan ke dalam variabel username dan password. Setelah input pertama diterima, program menyiapkan sebuah variabel bernama jumlah yang berfungsi sebagai penghitung berapa kali pengguna salah memasukkan data login. Nilai awalnya adalah 0. Selanjutnya, program melakukan pengecekan menggunakan sebuah perulangan for. Perulangan ini akan terus berjalan selama username atau password yang dimasukkan tidak sesuai dengan nilai yang benar, yaitu "Admin". Selama pengguna masih salah memasukkan salah satu dari keduanya, program akan menambah nilai jumlah sebanyak satu, lalu menampilkan pesan bahwa input sebelumnya salah, dan meminta pengguna mencoba lagi. Perulangan baru akan berhenti jika pengguna berhasil memasukkan username dan password yang sesuai. Setelah keluar dari loop, program menampilkan jumlah percobaan yang gagal dengan format "X Percobaan gagal", di mana X adalah nilai pada variabel jumlah.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var bilangan int

    fmt.Print("Masukkan Bilangan: ")

    fmt.Scan(&bilangan)

    fmt.Println("Digit bilangan dari Kanan ke Kiri: ")

    for bilangan != 0 {
        digit := bilangan % 10

        fmt.Println(digit)

        bilangan = bilangan / 10
    }
}
```

Screenshoot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7     fmt.Print("Masukkan Bilangan: ")
8     fmt.Scan(&bilangan)
9
10    fmt.Println("Digit bilangan dari Kanan ke Kiri: ")
11
12    for bilangan != 0 {
13        digit := bilangan % 10
14        fmt.Println(digit)
15        bilangan = bilangan / 10
16    }
17 }
```

```
Digit bilangan dari Kanan ke Kiri:
4
4
5
2
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Latsol 12> go run latsol2.go
Masukkan Bilangan: 342354654
Digit bilangan dari Kanan ke Kiri:
4
5
6
4
5
3
2
4
3
PS C:\Users\HP\Documents\KODING KULIAH (SETELAH UJIAN)\Latsol 12>
```

Deskripsi program

Program ini digunakan untuk menampilkan setiap digit dari sebuah bilangan, dimulai dari digit paling kanan. Di awal program, pengguna diminta memasukkan sebuah bilangan melalui perintah "Masukkan Bilangan:". Nilai yang dimasukkan kemudian disimpan dalam variabel bilangan. Setelah itu, program menampilkan judul "Digit bilangan dari Kanan ke Kiri:" sebagai penanda bahwa digit-digit akan ditampilkan satu per satu. Proses pengambilan digit dilakukan menggunakan perulangan for dengan kondisi bilangan != 0. Selama bilangan belum bernilai nol, program akan terus mengambil digit paling kanan. Pada tiap iterasi, program menghitung digit terakhir dengan operasi bilangan % 10, lalu menampilkannya. Setelah digit tersebut ditampilkan, bilangan dibagi 10 menggunakan pembagian bulat (bilangan = bilangan / 10) untuk menghilangkan digit yang baru saja diambil. Dengan cara ini, digit berikutnya menjadi digit terakhir yang akan diproses pada iterasi selanjutnya. Perulangan berhenti ketika seluruh digit sudah habis. Hasil akhirnya adalah daftar digit yang ditampilkan dari yang paling kanan hingga yang paling kiri.

3. Tugas 3

Source code

```
package main

import "fmt"

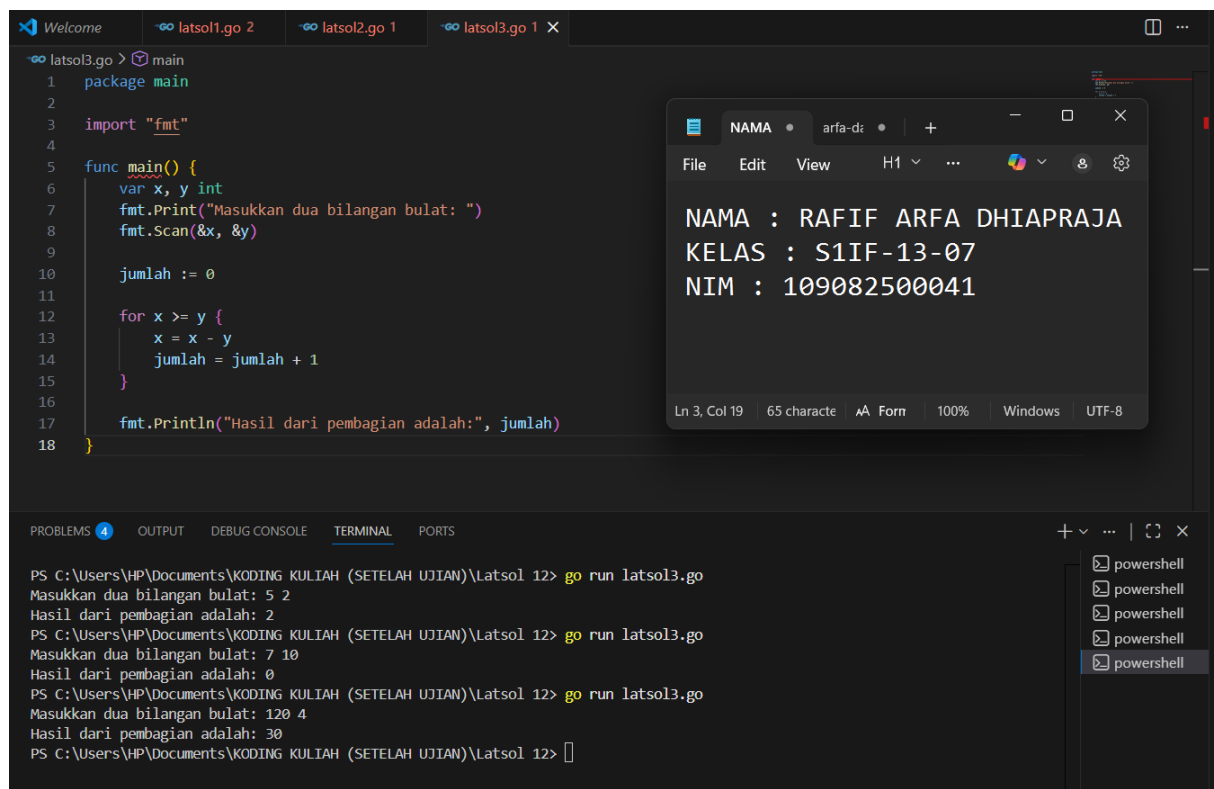
func main() {
    var x, y int
    fmt.Print("Masukkan dua bilangan bulat: ")
    fmt.Scan(&x, &y)

    jumlah := 0

    for x >= y {
        x = x - y
        jumlah = jumlah + 1
    }

    fmt.Println("Hasil dari pembagian adalah:", jumlah)
}
```

Screenshoot program



Deskripsi program

Program ini dibuat untuk menghitung hasil pembagian bilangan bulat menggunakan cara pengurangan berulang. Di awal program, pengguna diminta memasukkan dua bilangan bulat melalui perintah "Masukkan dua bilangan bulat:", lalu kedua nilai tersebut disimpan ke variabel x dan y. Setelah input diterima, program menyiapkan variabel jumlah dengan nilai awal 0. Variabel inilah yang nantinya berfungsi sebagai penghitung berapa kali nilai y dapat dikurangkan dari x. Pada dasarnya, proses ini meniru cara pembagian manual, di mana kita terus mengurangi bilangan pembagi dari bilangan yang dibagi hingga nilainya tidak cukup untuk dikurangi lagi. Proses perhitungan dilakukan melalui perulangan for dengan kondisi $x \geq y$. Selama nilai x masih lebih besar atau sama dengan y, program akan terus mengurangi x dengan y. Setiap kali pengurangan dilakukan, nilai jumlah akan bertambah satu. Dengan cara ini, nilai jumlah akan berakhir sebagai hasil pembagian bulat antara x dan y (tanpa menghitung sisa bagi). Setelah nilai x sudah tidak cukup lagi untuk dikurangi, perulangan berhenti. Program kemudian menampilkan hasil pembagian dalam bentuk pesan "Hasil dari pembagian adalah:" diikuti nilai jumlah.