

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 12
WHILE-LOOP



Disusun oleh:

Sukma Aditya Rafindra

109082500189

S1IF-13-04

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: ")

    fmt.Scan(&n)

    for n > 1 {

        fmt.Print(n, " x ")

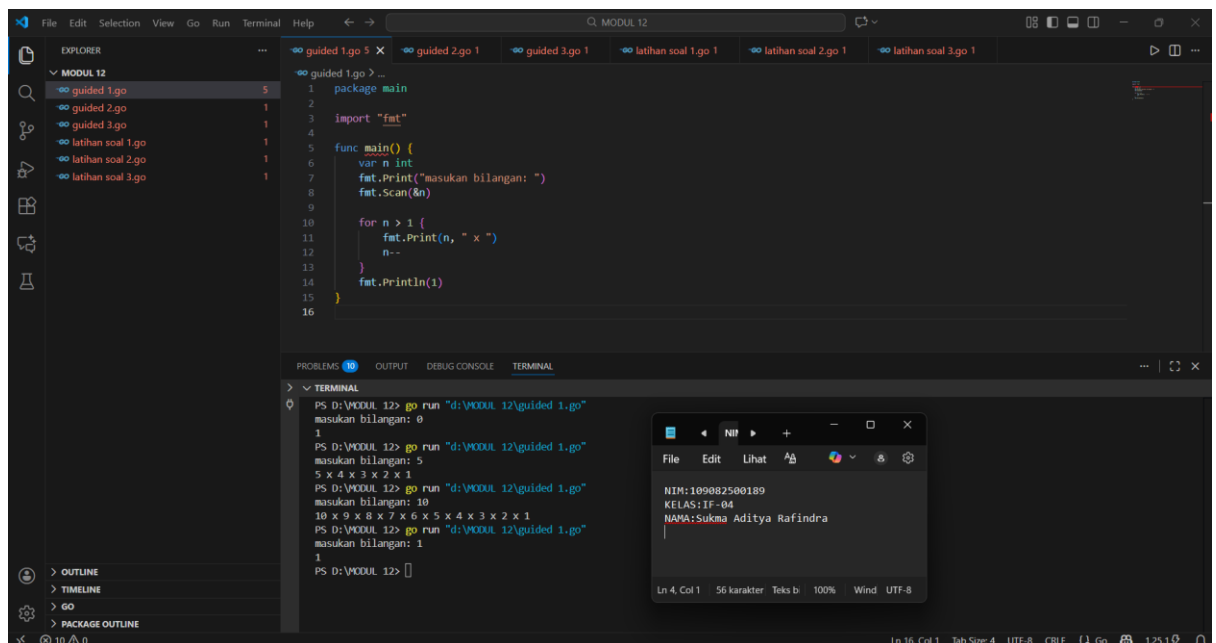
        n--

    }

    fmt.Println(1)

}
```

Screenshoot program



Deskripsi program

Program tersebut membaca sebuah angka dari input, lalu menampilkan proses perkalian mundur dari angka tersebut sampai 1, sambil menghitung hasil akhirnya. Misalnya jika pengguna memasukkan angka 5, program akan menampilkan $5 \times 4 \times 3 \times 2 \times 1 = 120$, karena program memulai dari nilai 5, mencetak angka tersebut, mengalikan hasil dengan angka itu, kemudian menurunkannya satu per satu hingga mencapai 1, dan akhirnya menampilkan total perkaliannya yaitu 120.

2. Guided 2

Source Code

```
package main

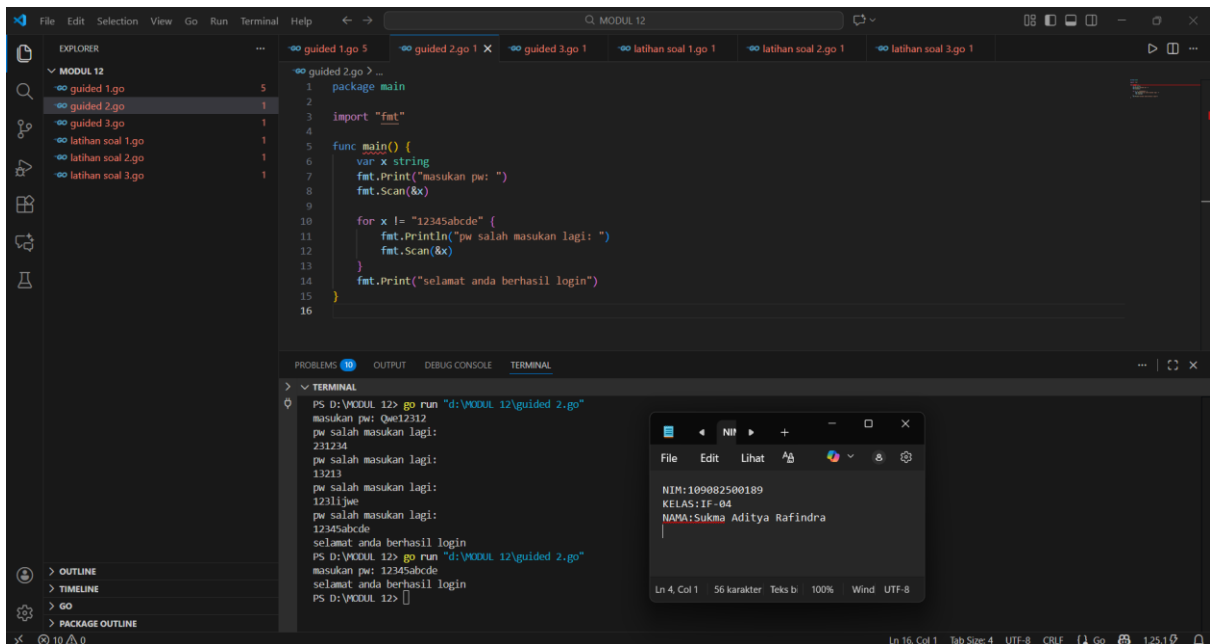
import "fmt"

func main() {
    var x string
    fmt.Print("masukan pw: ")
    fmt.Scan(&x)

    for x != "12345abcde" {
        fmt.Println("pw salah masukan lagi: ")
        fmt.Scan(&x)
    }

    fmt.Print("selamat anda berhasil login")
}
```

Screenshoot program



Deskripsi program

Program ini adalah sistem login yang meminta pengguna memasukkan token. Token yang dimasukkan dibandingkan dengan token yang valid, yaitu 12345abcde. Jika token salah, program akan terus meminta input hingga token yang benar dimasukkan. Setelah token benar, program menampilkan pesan bahwa login berhasil. Contohnya, jika pengguna memasukkan token yang salah beberapa kali, program akan terus meminta input sampai akhirnya mengetik 12345abcde, lalu menampilkan pesan “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)

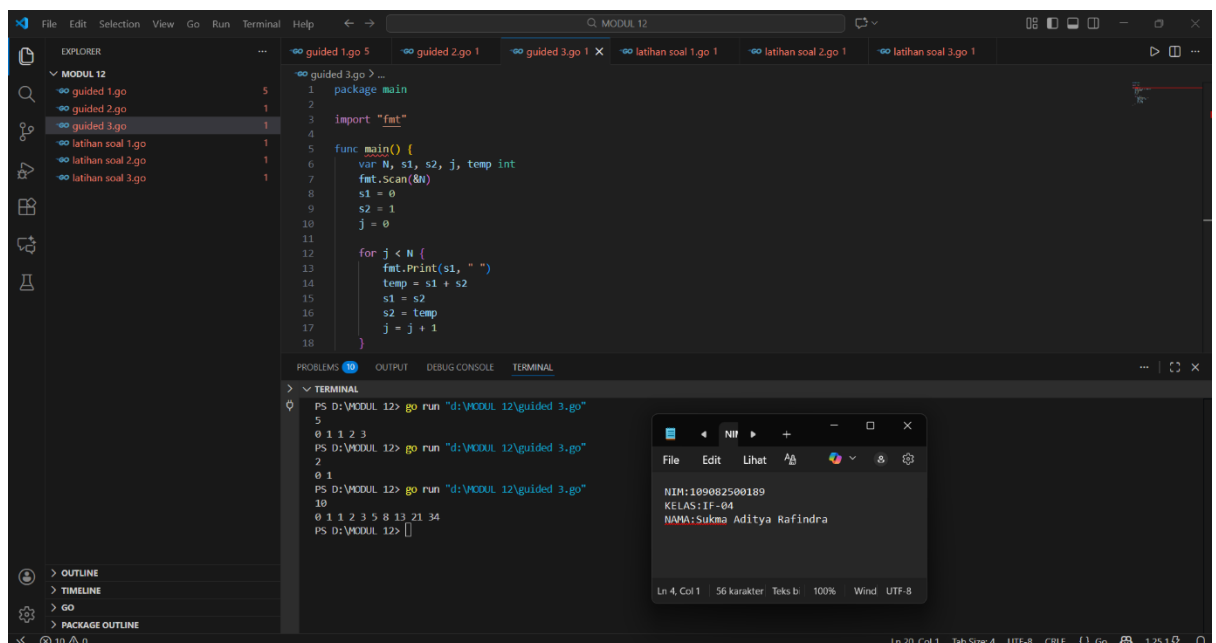
    s1 = 0
    s2 = 1
    j = 0
```

```

        for j < N {
            fmt.Print(s1, " ")
            temp = s1 + s2
            s1 = s2
            s2 = temp
            j = j + 1
        }
    }
}

```

Screenshoot program



Deskripsi program

Program ini menampilkan deret Fibonacci sebanyak n bilangan. Pertama diminta memasukkan jumlah bilangan yang diinginkan. Program memulai deret dengan 0 dan 1, kemudian mencetak setiap bilangan satu per satu sambil menghitung bilangan berikutnya sebagai penjumlahan dari dua bilangan sebelumnya. Proses ini diulang sampai tercetak sebanyak n bilangan. Misalnya, jika pengguna memasukkan 5, program akan menampilkan 0 1 1 2 3.

TUGAS

1. Tugas 1

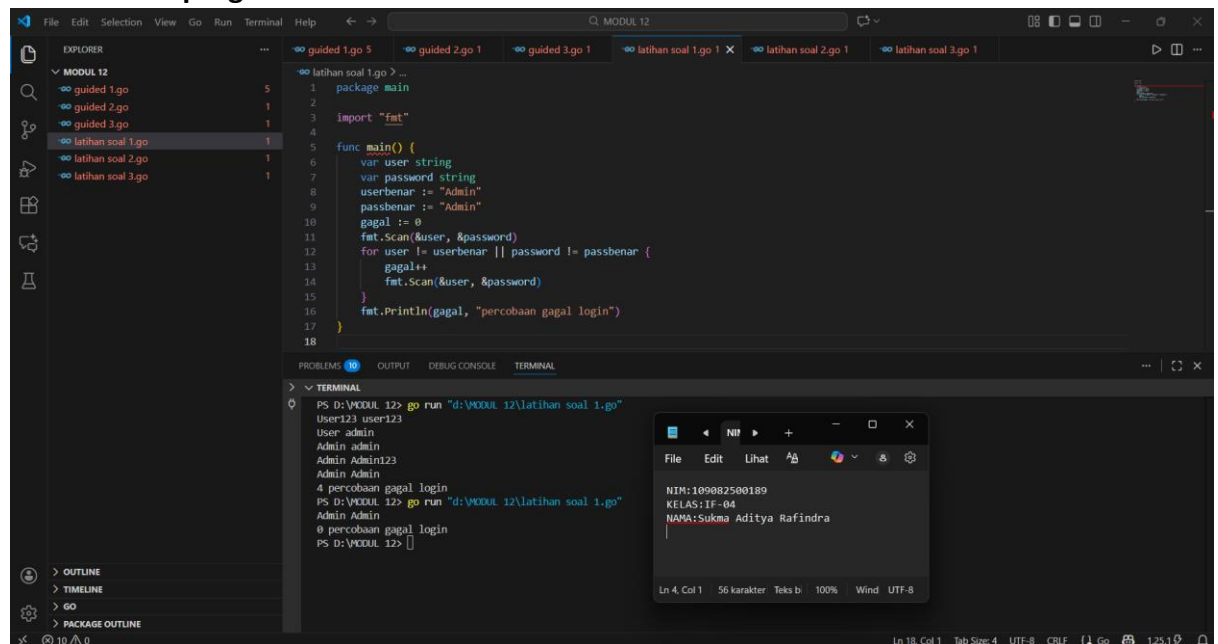
Source code

```
package main

import "fmt"

func main() {
    var user string
    var password string
    userbenar := "Admin"
    passbenar := "Admin"
    gagal := 0
    fmt.Scan(&user, &password)
    for user != userbenar || password != passbenar {
        gagal++
        fmt.Scan(&user, &password)
    }
    fmt.Println(gagal, "percobaan gagal login")
}
```

Screenshoot program



Deskripsi program

Program ini membuat simulasi proses login sederhana dengan username dan password yang benar bernilai "Admin". Saat dijalankan, program meminta pengguna memasukkan username dan password, lalu memeriksa kecocokannya. Jika salah, program tidak berhenti, tetapi meminta input ulang dan menambahkan hitungan pada variabel *gagal* untuk mencatat jumlah percobaan login yang keliru. Proses ini berlangsung dalam sebuah loop yang terus berulang sampai pengguna memasukkan username dan password yang benar. Setelah berhasil login, program menampilkan jumlah total percobaan gagal yang terjadi sebelum akhirnya pengguna berhasil masuk, sehingga menggambarkan cara kerja validasi login beserta pencatatan kesalahan input.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var digit int
```

```

    fmt.Println("masukan digit: ")

    fmt.Scan(&digit)

    for digit > 0 {

        sisa := digit % 10

        fmt.Println(sisa)

        digit = digit / 10

    }

}

```

Screenshoot program

The screenshot shows a Go IDE with the following components:

- EXPLORER:** Shows a project structure with files like `guided 1.go`, `guided 2.go`, `guided 3.go`, and `latihan soal 1.go` through `latihan soal 3.go`.
- EDITOR:** Displays the code for `latihan soal 2.go`, which is identical to the code provided in the first block.
- TERMINAL:** Shows the execution of the program. It runs `go run "d:\MODUL 12\latihan soal 2.go"` three times with different inputs:
 - Input: `2`, Output: `2`
 - Input: `2544`, Output: `4`, `4`, `5`, `2`
 - Input: `3423554654`, Output: `4`, `6`, `4`, `5`, `5`, `3`, `2`, `4`, `3`
- OUTPUT WINDOW:** A small window titled "NIP" showing personal information:
 - NIM: 109082500189
 - KELAS: IF-04
 - NAMA: Sukma Aditya Rafindra

Deskripsi program

Program ini meminta pengguna memasukkan sebuah bilangan bulat, kemudian program akan memecah bilangan tersebut menjadi digit-digit penyusunnya mulai dari digit paling belakang. Setiap iterasi, program mengambil angka terakhir dengan operasi `digit % 10`, menampilkannya, lalu membuang digit tersebut dengan membagi digit menggunakan 10. Proses ini diulang terus sampai semua digit habis. Program ini pada dasarnya digunakan untuk menampilkan setiap digit dari suatu bilangan secara terbalik (dari belakang ke depan).

3. Tugas 3

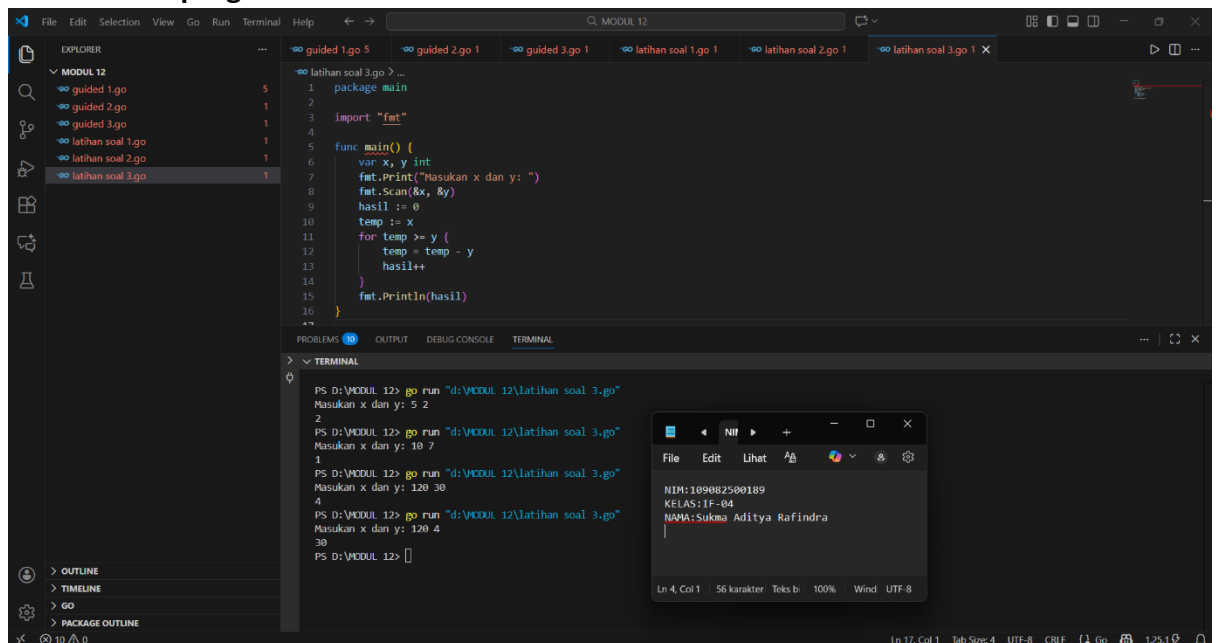
Source code

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukan x dan y: ")
    fmt.Scan(&x, &y)
    hasil := 0
    temp := x
    for temp >= y {
        temp = temp - y
        hasil++
    }
    fmt.Println(hasil)
}
```

Screenshoot program



Deskripsi program

Program ini meminta pengguna memasukkan dua angka, yaitu **x** dan **y**. Setelah itu, program menghitung berapa kali nilai **y** bisa dikurangkan dari **x** secara berulang sampai nilai sementara (**temp**) menjadi lebih kecil dari **y**. Setiap kali pengurangan dilakukan, sebuah counter bernama **hasil** ditambahkan satu. Setelah loop selesai, nilai **hasil** ditampilkan. Intinya, program ini menghitung berapa kali **x** dapat dibagi oleh **y** menggunakan metode pengurangan berulang, atau dengan kata lain menghitung hasil pembagian bulat tanpa memakai operator pembagian.