

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 12

WHILE-LOOP



**Telkom
University
PURWOKERTO**

Disusun oleh:

ITRHOH ANGGUN PAMUNGKAS

109082500117

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

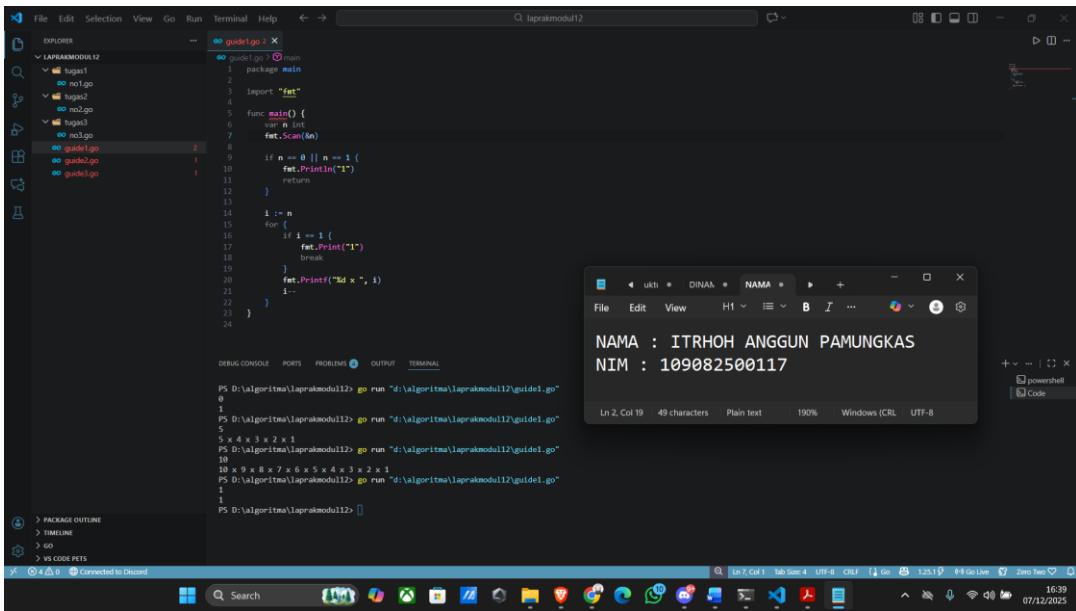
import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan angka: ")
    fmt.Scan(&n)

    if n == 0 || n == 1 {
        fmt.Println("1")
        return
    }

    i := n
    for {
        if i == 1 {
            fmt.Print("1")
            break
        }
        fmt.Printf("%d x ", i)
        i--
    }
}
```

Screenshot program



Deskripsi program

Program ini meminta pengguna memasukkan sebuah angka. Jika angkanya 0 atau 1, program langsung menampilkan 1 karena faktorial dari 0 dan 1 adalah 1. Jika angkanya lebih besar, program membuat perulangan dengan menurunkan nilai i dari angka tersebut hingga 1, dan setiap putaran mencetak angka tersebut dalam format “n x ...” sampai akhirnya mencapai angka 1 dan berhenti.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var input string

    tokenBenar := "12345abcde"

    for input != tokenBenar {
        fmt.Print("token: ")
    }
}
```

```

        fmt.Scan(&input)

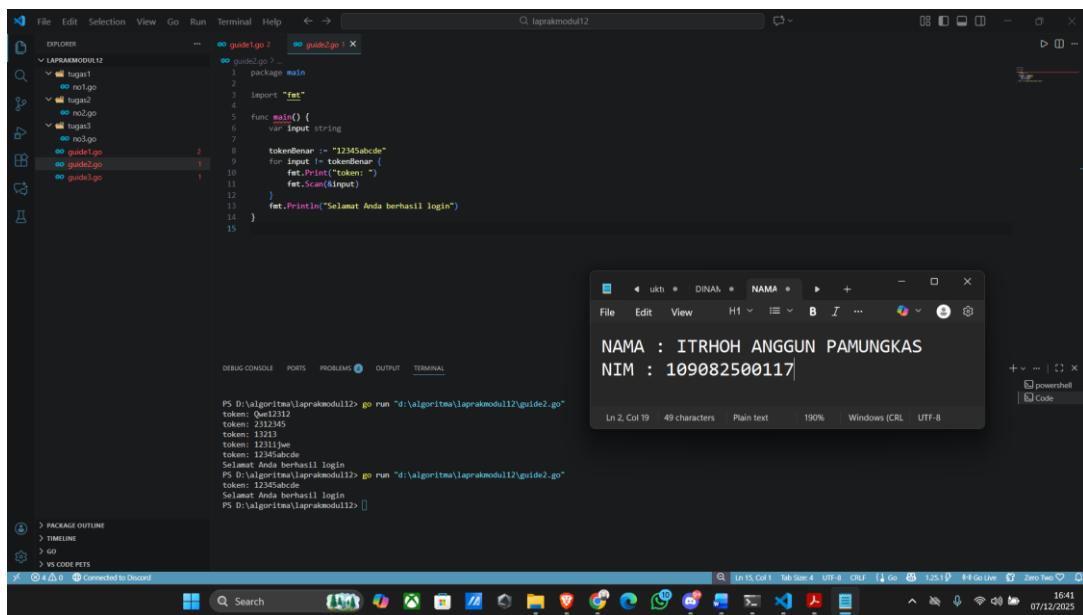
    }

    fmt.Println("Selamat Anda berhasil login")

}

```

Screenshot program



Deskripsi program

Program ini menyiapkan token yang benar yaitu "12345abcde". Kemudian program menjalankan perulangan yang terus meminta input token dari pengguna, dan perulangan hanya akan berhenti jika token yang dimasukkan sama dengan token yang benar. Jika sudah benar, program 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.Println(s1, " ")

    temp = s1 + s2

    s1 = s2

    s2 = temp

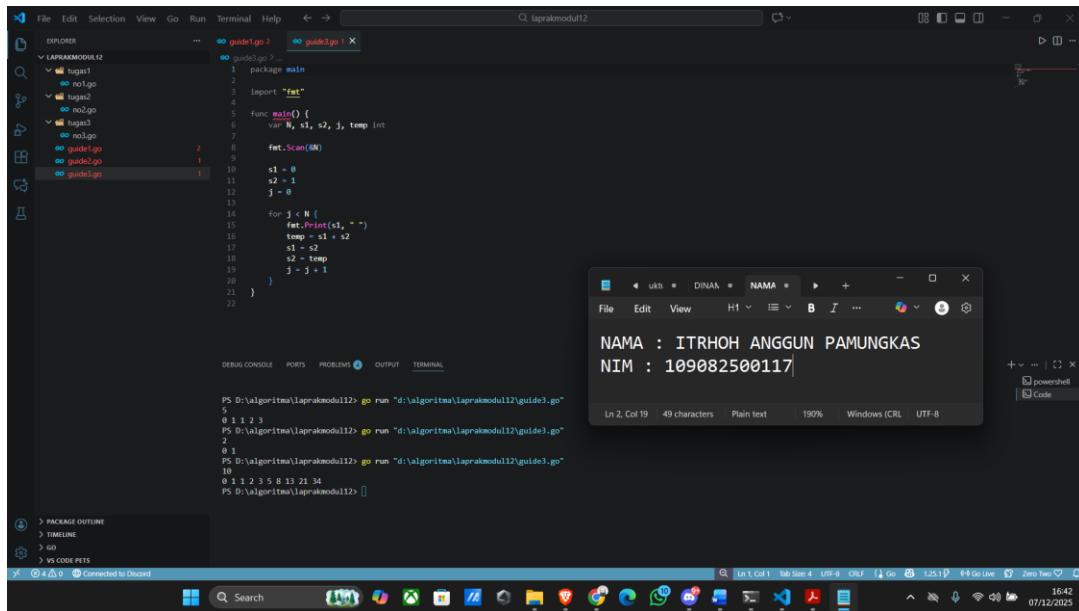
    j = j + 1

}

}

```

Screenshot program



Deskripsi program

Program ini membaca nilai N dari pengguna, lalu menampilkan N bilangan pertama dari deret Fibonacci. Dua variabel awal (s1 dan s2) diset menjadi 0 dan 1, kemudian program mencetak nilai s1, menghitung bilangan Fibonacci berikutnya, dan menggeser nilainya maju pada setiap iterasi hingga jumlah bilangan yang dicetak mencapai N.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

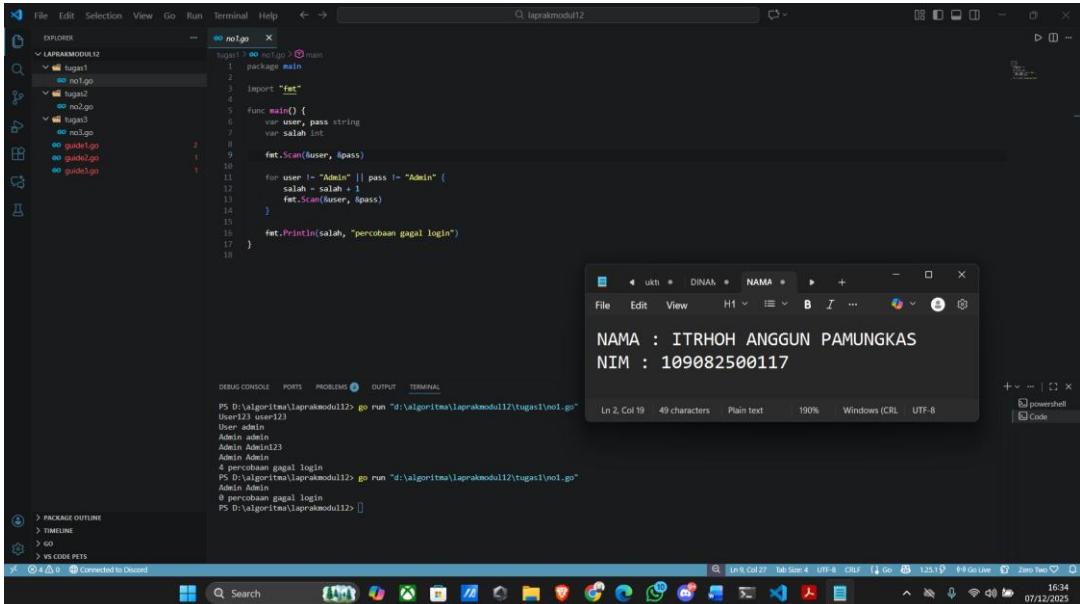
func main() {
    var user, pass string
    var salah int

    fmt.Scan(&user, &pass)

    for user != "Admin" || pass != "Admin" {
        salah = salah + 1
        fmt.Scan(&user, &pass)
    }

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

Screenshot program



Deskripsi program

Program ini meminta username dan password dari pengguna. Pertama, siapkan dua variabel bertipe string bernama user dan pass, lalu satu variabel bernama salah untuk menghitung berapa kali pengguna salah login. Setelah itu, program membaca input pertama dengan `fmt.Scan(&user, &pass)`. Pada saat itu, program belum tahu apakah inputnya benar atau salah. Karena itu, kita menggunakan sebuah perulangan while-loop yang ditulis dengan `for user != "Admin" || pass != "Admin" { ... }`. Artinya, selama username tidak sama dengan "Admin" atau password tidak sama dengan "Admin", maka program akan terus masuk ke dalam perulangan. Di dalam perulangan itu, setiap kali input yang dimasukkan salah, variabel salah ditambah satu sebagai tanda bahwa pengguna gagal login. Setelah itu, program kembali meminta input baru lagi dengan `fmt.Scan(&user, &pass)` supaya pengguna bisa mencoba login ulang. Perulangan akan berhenti ketika username dan password yang dimasukkan sudah benar, yaitu keduanya harus sama-sama "Admin". Setelah keluar dari perulangan, barulah program menampilkan jumlah percobaan gagal login menggunakan `fmt.Println(salah, "percobaan gagal 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)

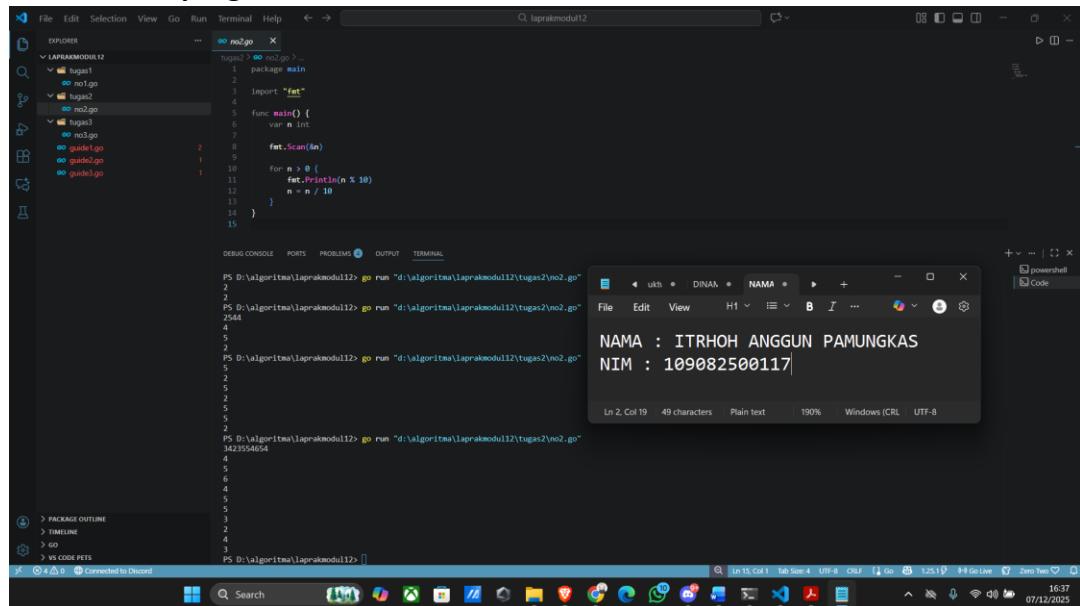
    n = n / 10

}

}

```

Screenshoot program



Deskripsi program

Program ini membaca satu bilangan bulat positif dan kemudian menampilkan setiap digitnya dari belakang ke depan. Pertama, program menyiapkan sebuah variabel bernama `n` untuk menyimpan bilangan tersebut, lalu input dibaca dengan `fmt.Scan(&n)`. Setelah angka masuk, program menggunakan while-loop `for n > 0 { ... }`, artinya selama nilai `n` masih lebih besar dari nol, maka program akan terus mengambil digit terakhir. Di dalam perulangannya, `fmt.Println(n % 10)` digunakan untuk mencetak digit paling kanan, karena operasi `n % 10` akan mengambil sisa bagi dari pembagian 10, yaitu digit terakhir. Setelah digit terakhir dicetak, angka diubah dengan `n = n / 10`, yang membuang digit terakhir dari angka tersebut. Proses ini terus berulang sampai semua digit habis dan nilai `n` menjadi nol, dan ketika itu perulangan otomatis berhenti. Hasilnya, setiap digit akan ditampilkan dari yang paling kanan sampai yang paling kiri.

3. Tugas 3

Source code

```
package main

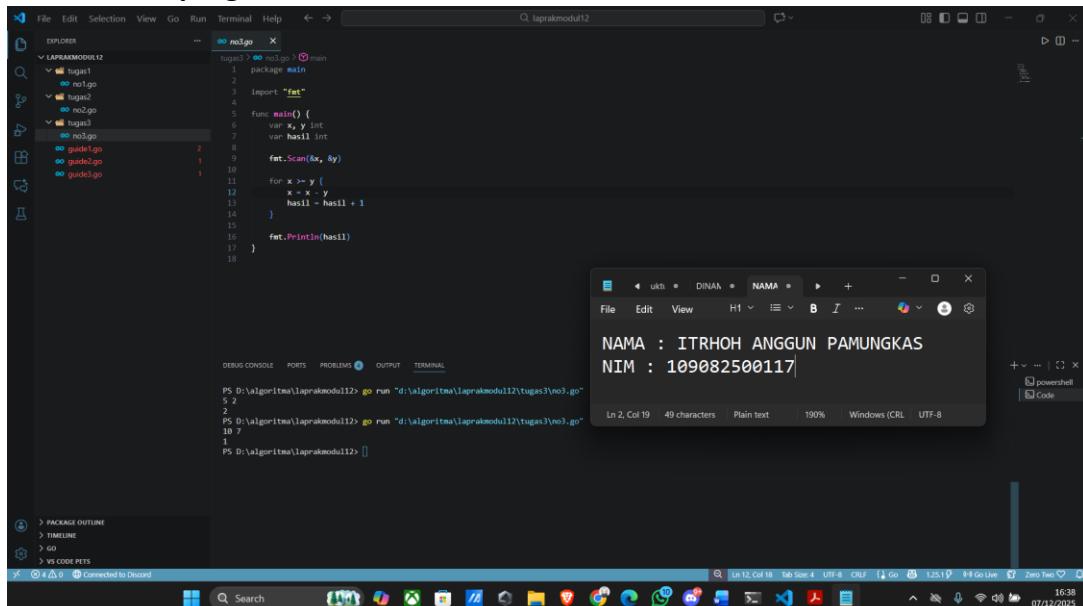
import "fmt"

func main() {
    var x, y int
    var hasil int

    fmt.Scan(&x, &y)

    for x >= y {
        x = x - y
        hasil = hasil + 1
    }
    fmt.Println(hasil)
}
```

Screenshot program



Deskripsi program

Program tersebut menghitung pembagian bilangan bulat secara manual tanpa menggunakan operator pembagian. Pertama, dua bilangan dimasukkan oleh pengguna dan disimpan pada variabel x dan y, lalu kita juga menyiapkan variabel hasil untuk menyimpan berapa kali y bisa dikurangkan dari x. Setelah input diterima,

program masuk ke bentuk while-loop for $x \geq y \{ \dots \}$. Kondisi ini berarti selama nilai x masih lebih besar atau sama dengan y , maka perulangan akan tetap berjalan. Di dalam perulangan, program melakukan pengurangan $x = x - y$, yaitu mengurangi x dengan y sekali setiap putaran. Setiap kali pengurangan berhasil dilakukan, variabel hasil ditambah satu sebagai tanda bahwa y sudah dikurangkan satu kali dari x . Perulangan akan berhenti ketika x sudah tidak cukup besar untuk dikurangkan lagi oleh y . Ketika kondisi berhenti tercapai, nilai hasil itulah yang nanti menjadi hasil integer division, dan program menampilkannya dengan `fmt.Println(hasil)`.