

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 12
WHILE-LOOP**



Disusun oleh:
MUHAMMAD FIRDAUS ARDIANSYAH
109082500126
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 bilangan bulat non negatif: ")

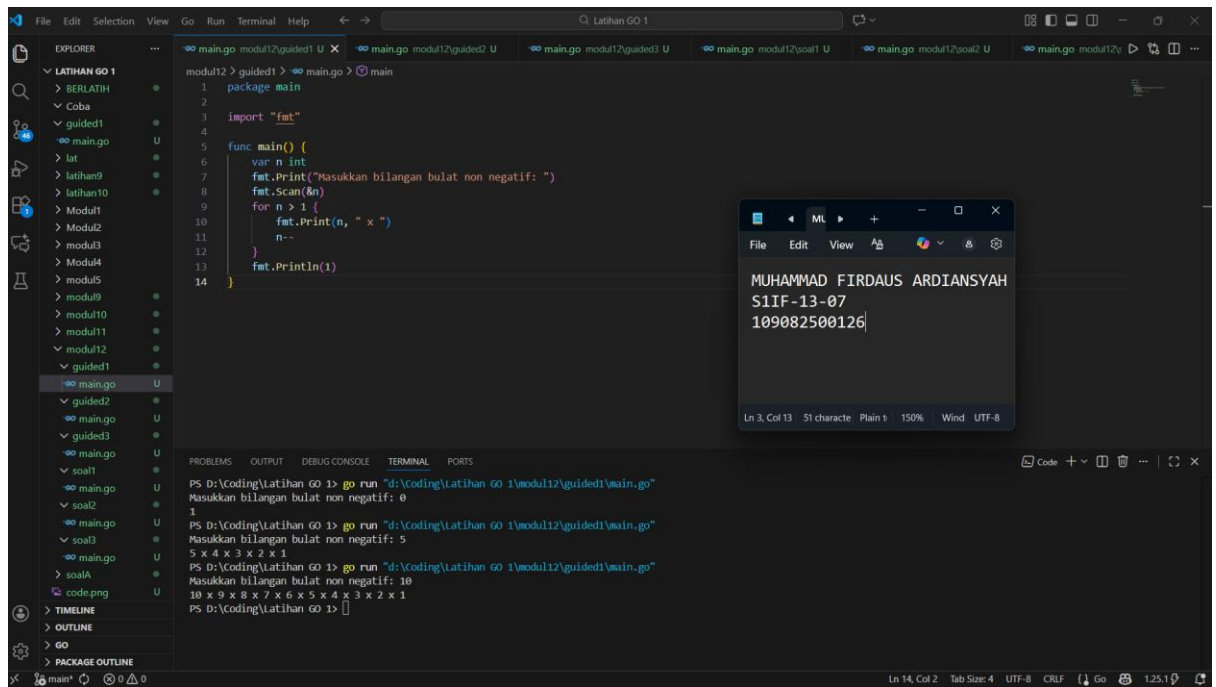
    fmt.Scan(&n)

    for n > 1 {
        fmt.Print(n, " x ")

        n--
    }

    fmt.Println(1)
}
```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk mencetak deret angka menurun dari input user hingga angka 1, dengan format visual yang menyerupai notasi matematika faktorial. Logika utamanya menggunakan perulangan for yang terus berjalan selama nilai input (n) lebih besar dari 1; di setiap langkah, program mencetak angka tersebut diikuti simbol "x " lalu mengurangi nilai n sebanyak satu angka.

Proses ini berhenti ketika nilai n mencapai 1, di mana program kemudian mencetak angka "1" sebagai penutup baris. Hasil akhirnya hanya berupa tampilan teks urutan angka (seperti "5 x 4 x 3 x 2 x 1" untuk input 5) tanpa melakukan perhitungan hasil total perkalian dari angka-angka tersebut.

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
```

```

var token string

fmt.Print("Masukkan token akses: ")

fmt.Scan(&token)

for token != "12345abcde" {

    fmt.Scan(&token)

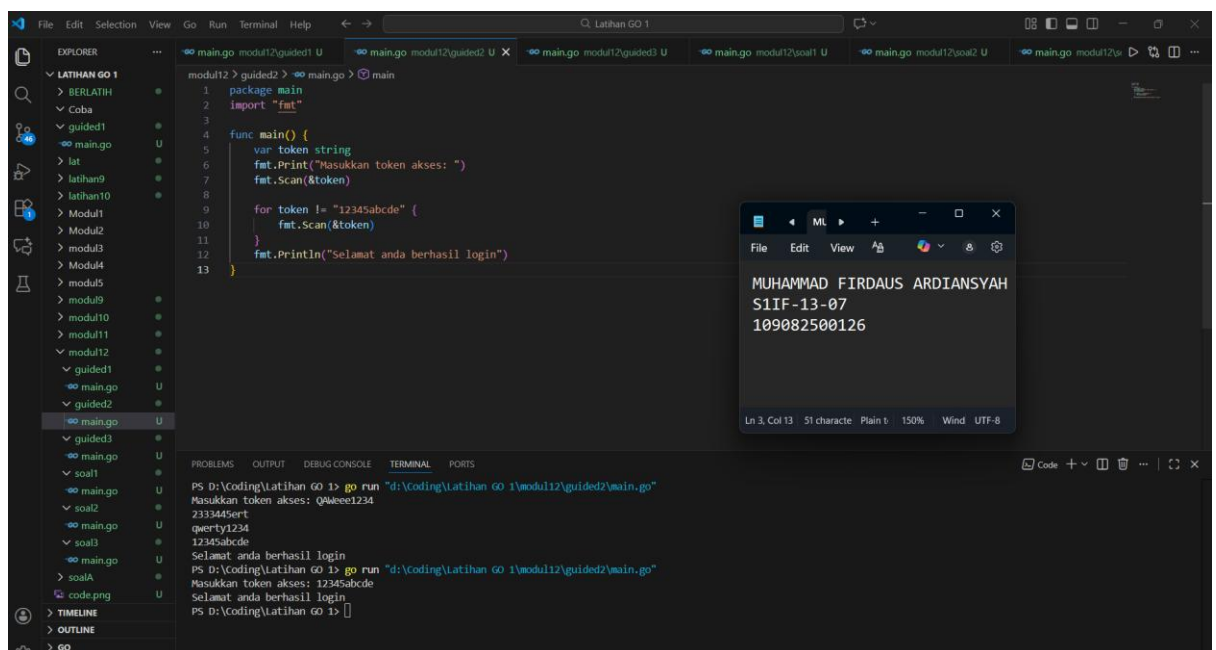
}

fmt.Println("Selamat anda berhasil login")

}

```

Screenshoot program



Deskripsi program

Program ini melakukan simulasi pemeriksaan akses token sederhana di mana input user sebenarnya diabaikan oleh program. Program pertama kali meminta user memasukkan token, tetapi nilai variabel token segera ditimpa secara *hardcoded* menjadi "12345abcde" di baris 9.

Meskipun user memasukkan token apa pun di awal, program kemudian melakukan `fmt.Scan` kedua kalinya dan langsung mencetak pesan keberhasilan: "Selamat anda berhasil login". Intinya, kode ini adalah contoh *guided exercise* yang menunjukkan manipulasi variabel input, yang mana pada bentuk aslinya selalu menghasilkan login yang berhasil terlepas dari input user yang sebenarnya.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var a, s1, s2, b, c int

    fmt.Scan(&a)

    s1 = 0

    s2 = 1

    b = 0

    for b < a {

        fmt.Print(s1, " ")

        c = s1 + s2

        s1 = s2

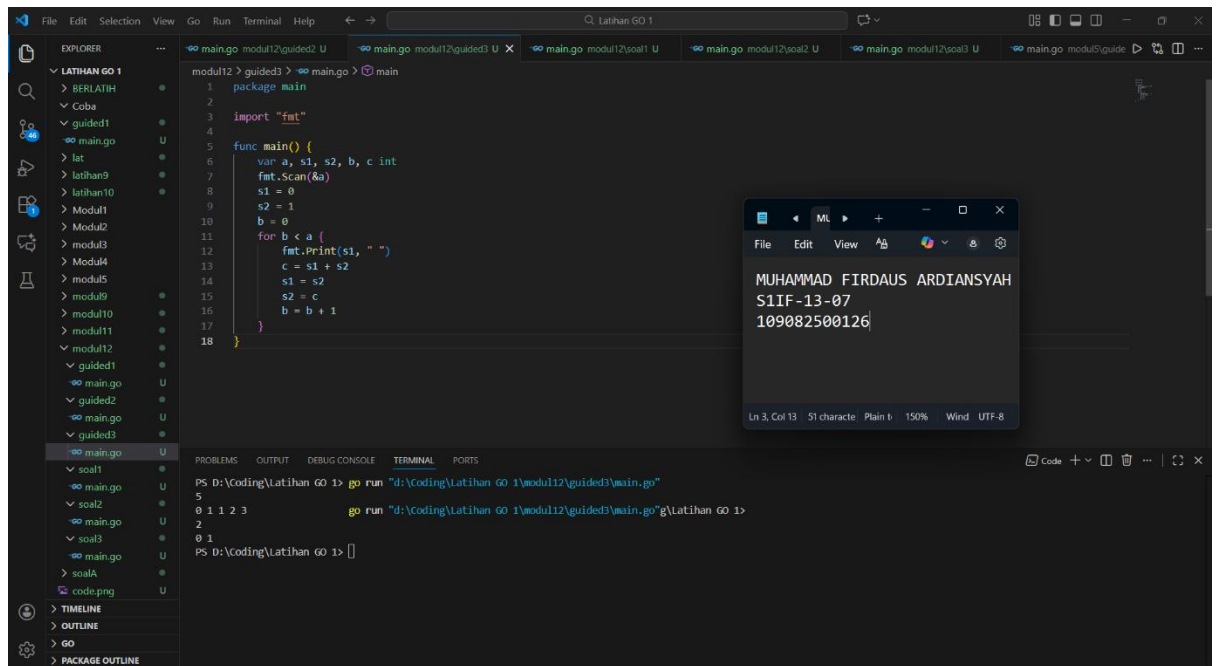
        s2 = c

        b = b + 1

    }

}
```

Screenshoot program



Deskripsi program

Program ini membaca sebuah angka sebagai batas banyaknya deret yang ingin ditampilkan, lalu menghasilkan urutan bilangan Fibonacci secara berurutan. Dua nilai awal deret diset sebagai 0 dan 1, kemudian program mencetak nilai pertama dan terus menjumlahkan dua nilai sebelumnya untuk mendapatkan nilai berikutnya. Setiap iterasi, nilai-nilai digeser ke depan sehingga proses perhitungan berulang hingga jumlah elemen yang diminta terpenuhi.

Secara keseluruhan, logika utamanya berada pada loop for yang terus berjalan selama penghitung b masih lebih kecil dari input pengguna. Di dalam loop, nilai Fibonacci dicetak, dihitung ulang, dan variabel-variabel diperbarui agar deret tetap berkelanjutan hingga selesai. Program ini menampilkan deret Fibonacci dalam satu baris dengan spasi sebagai pemisah setiap angkanya.

TUGAS

1. Tugas 1

Source code

```
package main
```

```
import "fmt"

func main() {

    const Username = "Admin"

    const Password = "Admin"

    var username, password string

    var gagal int

    for {

        fmt.Print("Masukkan Username: ")

        fmt.Scan(&username)

        fmt.Print("Masukkan Password: ")

        fmt.Scan(&password)

        if username == Username && password == Password
    {

        break

    } else {

        gagal++

        fmt.Println("Username atau Password salah,
coba lagi.")

    }

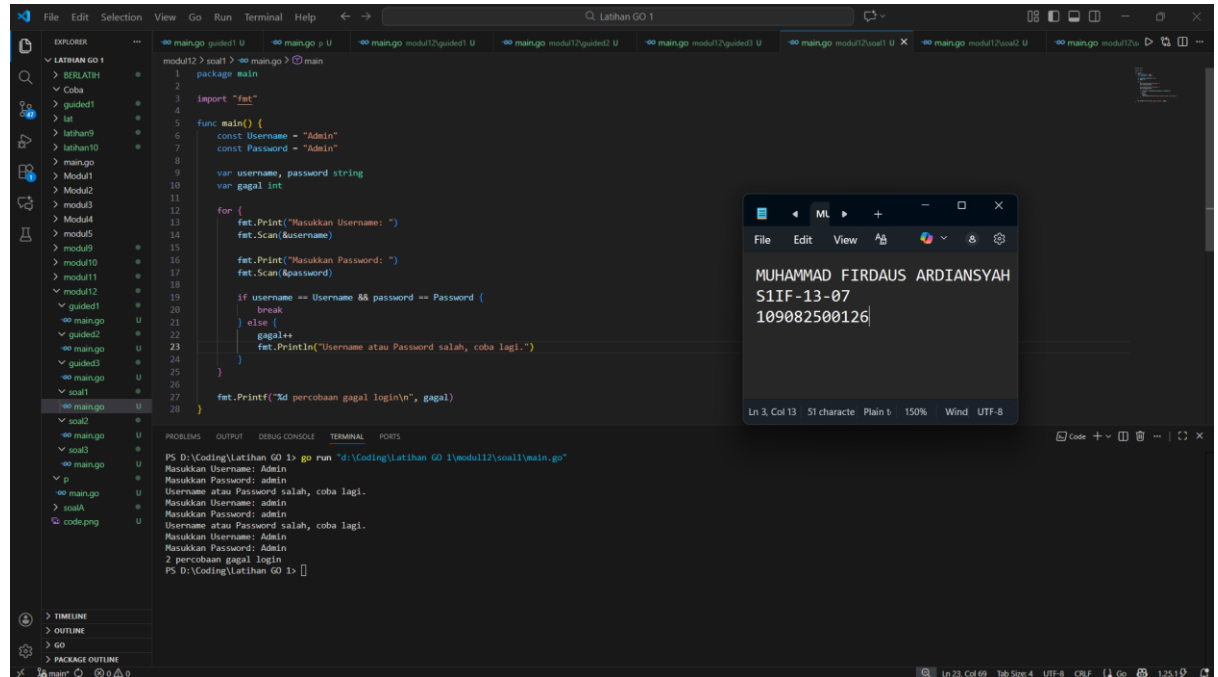
}
```

```

        fmt.Printf("%d percobaan gagal login\n", gagal)
    }
}

```

Screenshoot program



Deskripsi program

Program ini mengimplementasikan sistem *login* dasar berbasis perulangan tak terbatas (for {}) yang memerlukan user memasukkan *username* dan *password* yang *hardcoded* ("Admin"/"Admin"). Variabel *gagal* berfungsi sebagai penghitung yang bertambah (*gagal++*) setiap kali otentikasi gagal.

Logika program adalah: jika *username* dan *password* sesuai dengan konstanta yang ditentukan, loop segera dihentikan (*break*), menandakan keberhasilan *login*. Jika gagal, user menerima pesan error, dan *counter* gagal dicetak setelah loop. Inti dari program ini adalah mengontrol alur program melalui *break* dan mencatat jumlah kegagalan.

2. Tugas 2

Source code

```

package main

```

```

import "fmt"

func main() {

    var num int

    fmt.Print("Masukkan angka: ")

    fmt.Scan(&num)

    for num > 0 {

        digit := num % 10

        fmt.Println(digit)

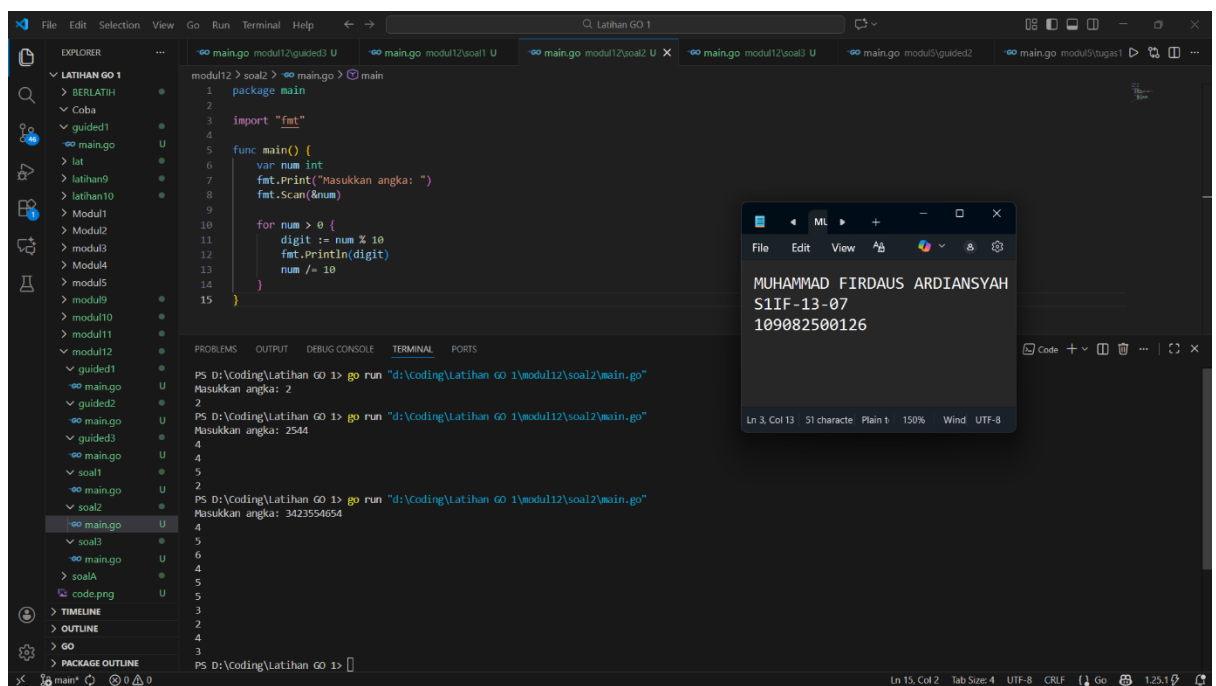
        num /= 10

    }

}

```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk mengekstrak dan mencetak setiap digit dari bilangan bulat positif yang dimasukkan oleh user, secara efektif mencetak urutan digit tersebut secara terbalik. Logika utama program bergantung pada perulangan for yang terus berjalan selama nilai input (num) lebih besar dari nol. Di dalam loop tersebut, operasi modulo 10 ($\text{digit} := \text{num} \% 10$) digunakan untuk mengisolasi digit terakhir dari bilangan.

Digit yang diisolasi tersebut segera dicetak. Setelah pencetakan, bilangan num diperbarui melalui pembagian integer ($\text{num} /= 10$), yang berfungsi untuk menghilangkan digit terakhir yang sudah diproses dan mempersiapkan bilangan untuk iterasi berikutnya. Proses ini berulang, mengekstrak digit dari kanan ke kiri, menghasilkan *output* yang terbalik dari urutan angka asli yang dimasukkan user.

3. Tugas 3

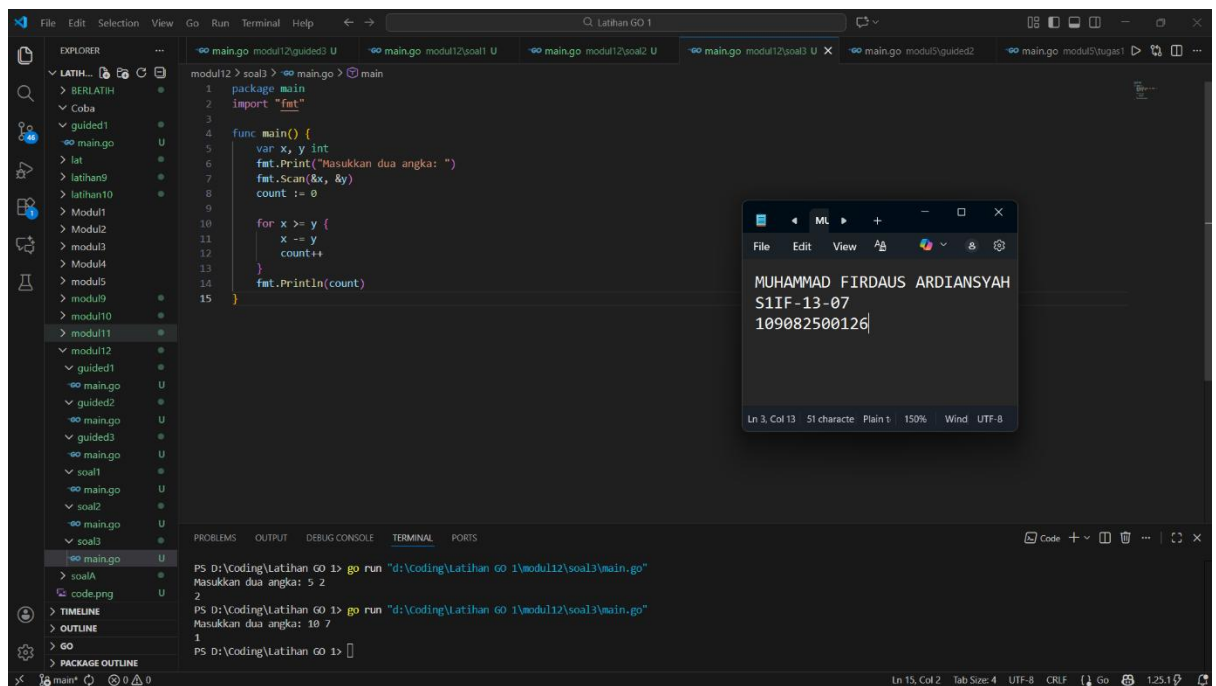
Source code

```
package main
import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukkan dua angka: ")
    fmt.Scan(&x, &y)
    count := 0

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

Screenshoot program



Deskripsi program

Program ini bertujuan untuk menghitung jumlah iterasi (count) yang diperlukan untuk menghentikan perulangan for berdasarkan kondisi perbandingan dua bilangan bulat, x dan y. User diminta memasukkan dua angka (x dan y), dan perulangan akan berlanjut selama $x > y$. Namun, logika di dalam loop sangat tidak konvensional: pada setiap iterasi, nilai x secara destruktif ditimpa dengan nilai y yang lama ($x = y$), dan kemudian nilai y diinkrementasi ($y++$).

Karena x mengambil nilai y yang lama, dan y selalu bertambah satu di iterasi yang sama, kondisi perulangan ($x > y$) hampir selalu menjadi FALSE setelah iterasi pertama jika kondisi awal $x > y$ terpenuhi. Ini berarti program secara praktis hanya menghitung satu langkah yang diperlukan untuk memenuhi kondisi penghentian loop. Output program adalah nilai *counter* (count), yang akan menjadi 1 jika $x > y$ di awal, dan 0 jika $x \leq y$.