

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL 05 & 06
FOR-LOOP**



Disusun oleh:

RAFFI YULIANSYAH

109082500101

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 a, b int

    var j int

    fmt.Scan(&a, &b)

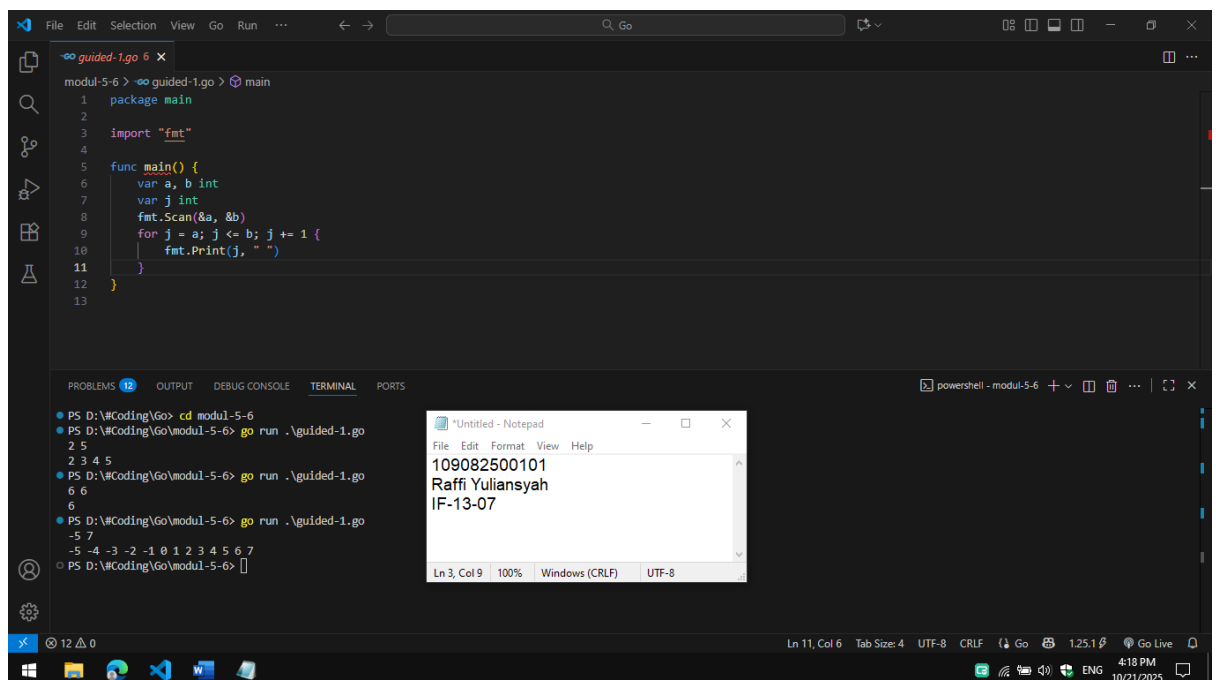
    for j = a; j <= b; j += 1 {

        fmt.Print(j, " ")

    }

}
```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk mencetak deret angka berurutan dalam satu baris, sesuai dengan input a sampai dengan input b.

Program dimulai dengan mendeklarasikan dua variabel bilangan bulat (int): a (untuk batas awal) dan b (untuk batas akhir). Program kemudian akan diberi input dua angka, di mana angka pertama disimpan di a dan angka kedua di b. Setelah itu, program melakukan perulangan for . (for j = a; j <= b; j += 1) Perulangan ini diinisialisasi dengan mengatur nilai variabel j sama dengan a. Kemudian memberikan kondisi, selama nilai j masih lebih kecil atau sama dengan b. Pada setiap loop-nya, akan dicetak angka berurutan, `fmt.Print(j, " ")` dengan memanfaatkan variabel j. Pada setiap akhir iterasi nilai j akan ditambah satu (`j += 1`), dan perulangan akan dilakukan hingga kondisi bernilai false (berulang dari a sampai dengan b).

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var j, alas, tinggi, n int

    var luas float64

    fmt.Scan(&n)

    for j = 1; j <= n; j += 1 {

        fmt.Scan(&alas, &tinggi)

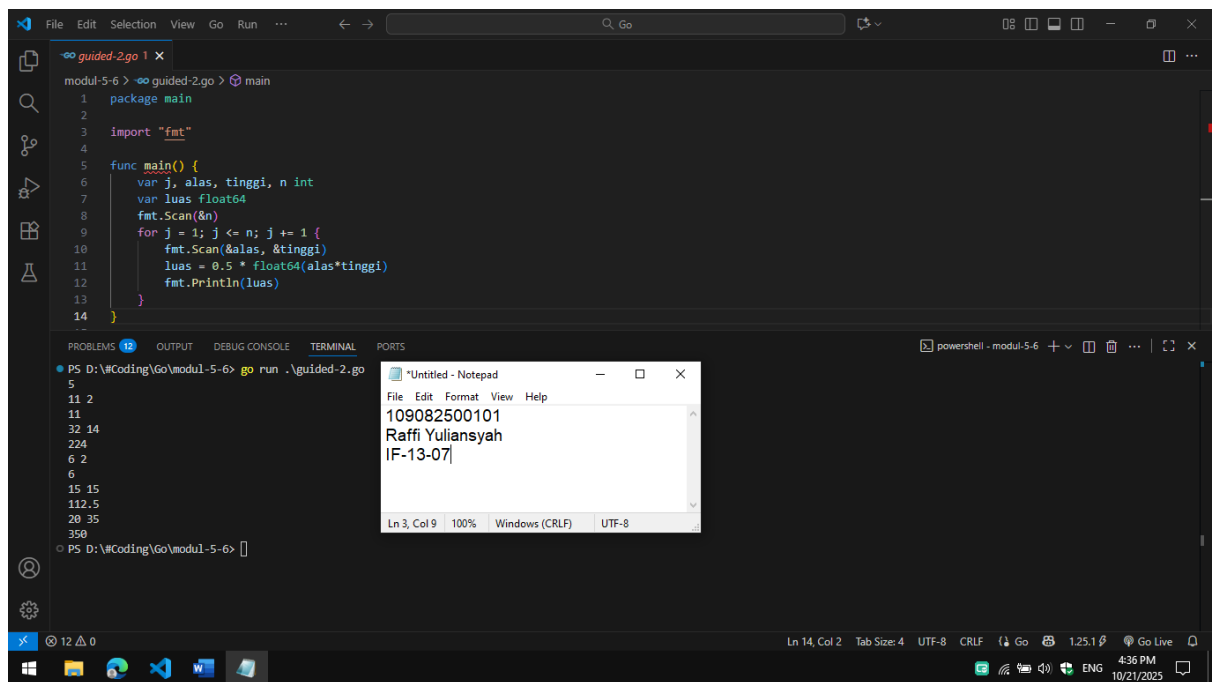
        luas = 0.5 * float64(alas*tinggi)

        fmt.Println(luas)

    }

}
```

Screenshoot program



Deskripsi program

Program ini akan menghitung luas segitiga dari banyak n segitiga secara berurutan. Program dimulai dengan mendeklarasikan variabel n bertipe data integer untuk menentukan banyak segitiga yang akan dihitung, kemudian variabel (int) alas, (int) tinggi serta (float64) luas. Pertama, program akan meminta input satu angka (n) untuk menentukan berapa kali perhitungan segitiga akan dilakukan.

Setelah itu, program akan melakukan perulangan for yang akan berjalan sebanyak n kali (for $j = 1; j \leq n; j += 1$). Di dalam setiap iterasi, pertama program meminta untuk memasukkan nilai alas dan tinggi untuk satu segitiga. Program kemudian menghitung luas segitiga tersebut dengan rumus $0.5 * \text{alas} * \text{tinggi}$ (karena variabel alas dan tinggi bertipe integer maka hasil dari perkalian keduanya perlu di-casting dengan float64) dan menyimpan hasilnya di variabel luas. Terakhir, program akan langsung mencetak hasil luas segitiga berdasarkan isi variabel luas. Setelahnya blok program akan berulang hingga semua n segitiga selesai dihitung.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var j, v1, v2 int
```

```

    var hasil int

    fmt.Scan(&v1, &v2)

    hasil = 0

    for j = 1; j <= v2; j += 1 {

        hasil = hasil + v1

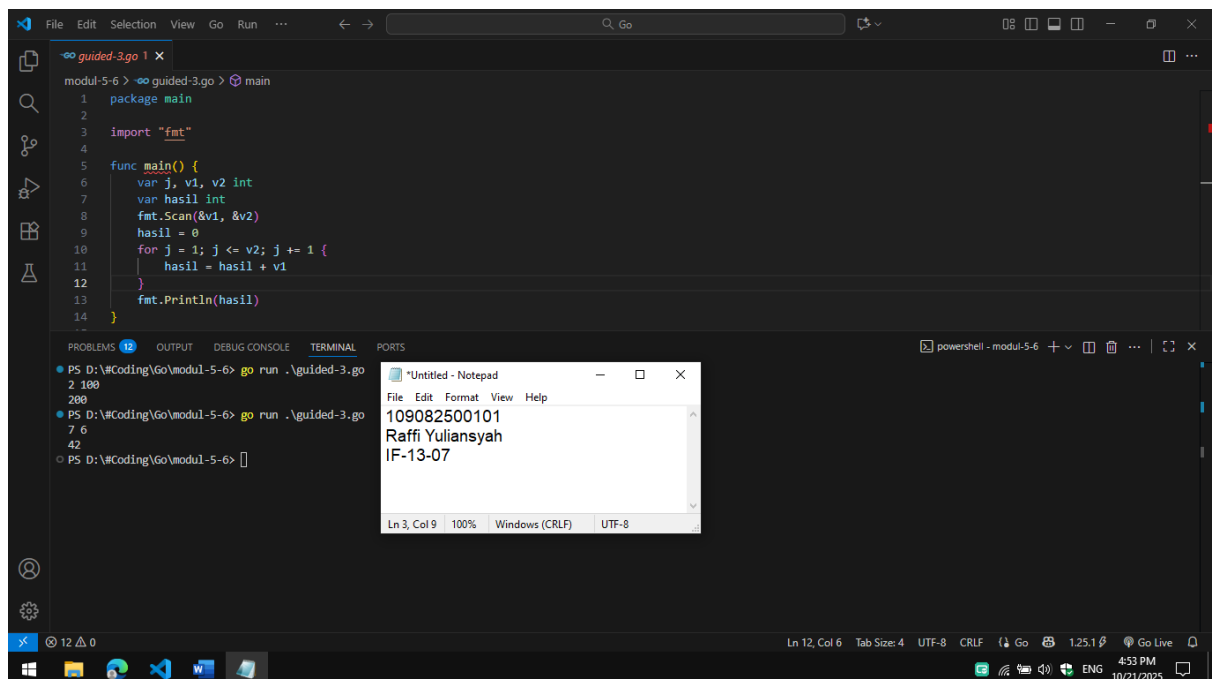
    }

    fmt.Println(hasil)

}

```

Screenshoot program



Deskripsi program

Program ini akan melakukan operasi perkalian ($v1$ dikali $v2$) dengan menggunakan metode penjumlahan berulang memanfaatkan for-loop.

Programnya dimulai dengan mendeklarasikan tiga variabel bilangan bulat (int): $v1$ dan $v2$ untuk menampung input dan $hasil$ untuk menyimpan hasil perkalian $v1$ dan $v2$. Program kemudian akan diberi input untuk variabel $v1$ kemudian $v2$, yang akan disimpan ke dalam variabel $v1$ dan $v2$. Sebelum perulangan dimulai, variabel $hasil$ diberi nilai awal 0.

Setelah itu, program akan melakukan perulangan for, dengan nilai awal $j = 1$, yang akan berjalan sebanyak $v2$ kali (artinya akan berjalan selama kondisi $j \leq v2$ masih bernilai true) dan nilai j akan bertambah 1 setelah setiap iterasi selesai. Di dalam setiap perulangan, nilai $v1$ akan ditambahkan ke nilai hasil. Proses ini berulang terus-menerus, menjumlahkan $v1$ dengan dirinya sendiri sebanyak $v2$ kali. Setelah perulangan selesai, program akan mencetak nilai akhir yang tersimpan di variabel hasil.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main(){
    var bilangan, hasil int

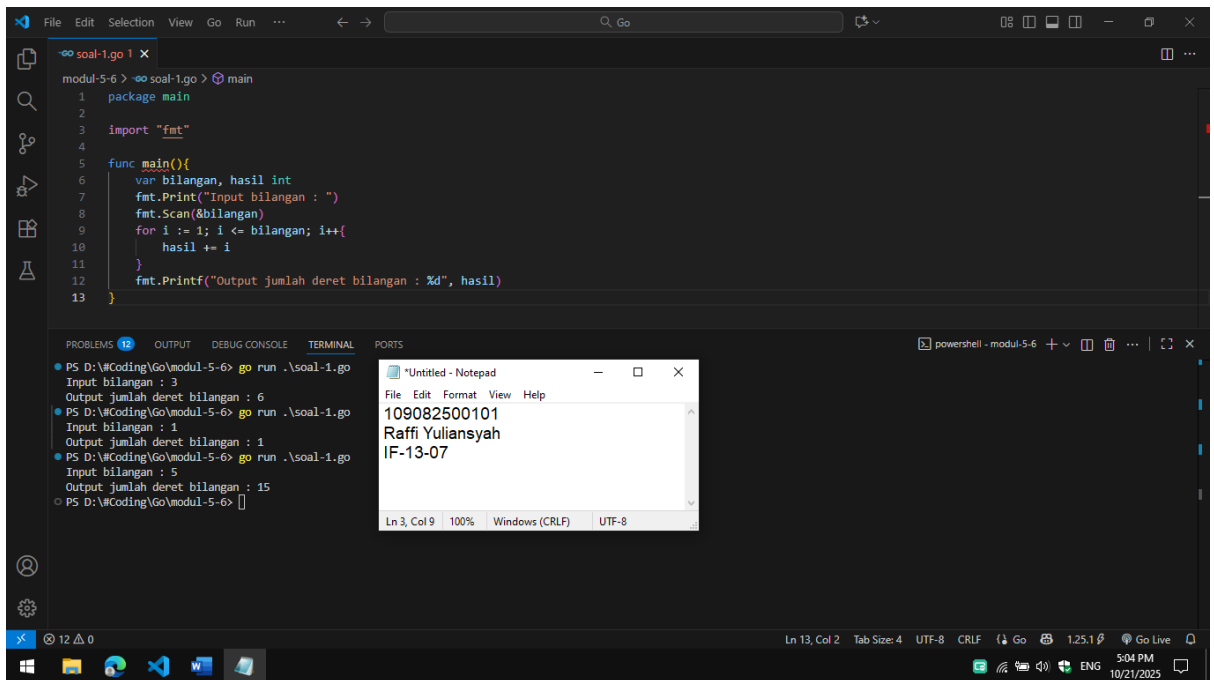
    fmt.Print("Input bilangan : ")

    fmt.Scan(&bilangan)

    for i := 1; i <= bilangan; i++){
        hasil += i
    }

    fmt.Printf("Output jumlah deret bilangan : %d", hasil)
}
```

Screenshoot program



Deskripsi program

Sebuah program yang berfungsi untuk menghitung jumlah sekumpulan bilangan, mulai dari 1 hingga angka yang diinputkan.

Alur program dimulai dengan mendeklarasikan dua variabel bilangan bulat (int), yaitu bilangan (menyimpan input berapa banyak deret yang akan ditambahkan) dan hasil (untuk menyimpan total penjumlahan, yang bernilai awal 0). Program kemudian meminta untuk memasukkan sebuah angka yang menunjukkan berapa banyak deret, yang selanjutnya disimpan ke dalam variabel bilangan.

Selanjutnya program akan melakukan sebuah perulangan for untuk menghitung hasil penjumlahan deret angka yang dimulai dari 1. Perulangan ini diinisialisasi dengan i yang bernilai 1. Selanjutnya kondisi dimana selama nilai i masih lebih kecil atau sama dengan nilai variabel bilangan yang diinput, program akan menambahkan nilai i saat ini ke dalam variabel hasil (hasil += i). Nilai i kemudian ditambah satu, dan proses ini berulang. Setelah perulangan selesai (ketika i telah melebihi variabel bilangan, dan kondisi false), program akan mencetak nilai akhir dari variabel hasil yang berisi penjumlahan deret bilangan.

2. Tugas 2

Source code

```
package main

import "fmt"

func main(){
```

```

var n int

var jariJari, tinggi, hasil float64

const pi = 3.14159

fmt.Print("Input banyak kerucut : ")

fmt.Scan(&n)

for i := 0; i < n ; i++){

    fmt.Scan(&jariJari)

    fmt.Scan(&tinggi)

    hasil = pi * jariJari * jariJari * tinggi /
3

    fmt.Printf("Output Kerucut %d : %f\n",i+1,
hasil)

}

}

```

Screenshoot program

The screenshot shows the Visual Studio Code interface with a Go file named 'soal-2.go' open. The code defines a program that calculates the volume of a cone for a given number of iterations. The terminal output shows the program running successfully, displaying the input values and the calculated output for each iteration.

```

modul-5-6 > go run .\soal-2.go
1 package main
2
3 import "fmt"
4
5 func main(){
6     var n int
7     var jariJari, tinggi, hasil float64
8     const pi = 3.14159
9     fmt.Print("Input banyak kerucut : ")
10    fmt.Scan(&n)
11    for i := 0; i < n ; i++){
12        fmt.Scan(&jariJari)
13        fmt.Scan(&tinggi)
14        hasil = pi * jariJari * jariJari * tinggi / 3
15        fmt.Printf("Output Kerucut %d : %f\n",i+1, hasil)
16    }
17 }

```

Terminal Output:

```

PS D:\Coding\Go\modul-5-6> go run .\soal-2.go
Input banyak kerucut : 1
3 4
Output Kerucut 1 : 37.699888
PS D:\Coding\Go\modul-5-6> go run .\soal-2.go
Input banyak kerucut : 3
1 1
Output Kerucut 1 : 1.047197
2 2
Output Kerucut 2 : 8.377573
3 3
Output Kerucut 3 : 28.274310
PS D:\Coding\Go\modul-5-6>

```

Notepad Content:

```

109082500101
Raffi Yuliansyah
IF-13-07

```

Deskripsi program

Program ini adalah program untuk menghitung volume beberapa kerucut.

Alur programnya dimulai dengan mendeklarasikan variabel: n (int) untuk jumlah kerucut, kemudian variabel bertipe float64: jariJari, tinggi, dan hasil untuk data perhitungan kerucut, serta menginisialisasi konstanta pi = 3.14159. Program kemudian meminta memasukkan jumlah kerucut (n) yang ingin dihitung. Setelah itu, program melakukan perulangan for yang akan berjalan sebanyak n kali. Di dalam setiap perulangan, program akan:

1. Meminta input nilai jariJari dan tinggi untuk satu kerucut.
2. Menghitung volume kerucut tersebut dengan rumus $\pi * \text{jariJari} * \text{jariJari} * \text{tinggi} / 3$ dan menyimpannya di variabel hasil.
3. Langsung mencetak hasil volume kerucut yang baru saja dihitung.

Proses ini (meminta input, menghitung, mencetak) akan diulangi sebanyak n kali, satu kerucut pada satu waktu.

3. Tugas 3

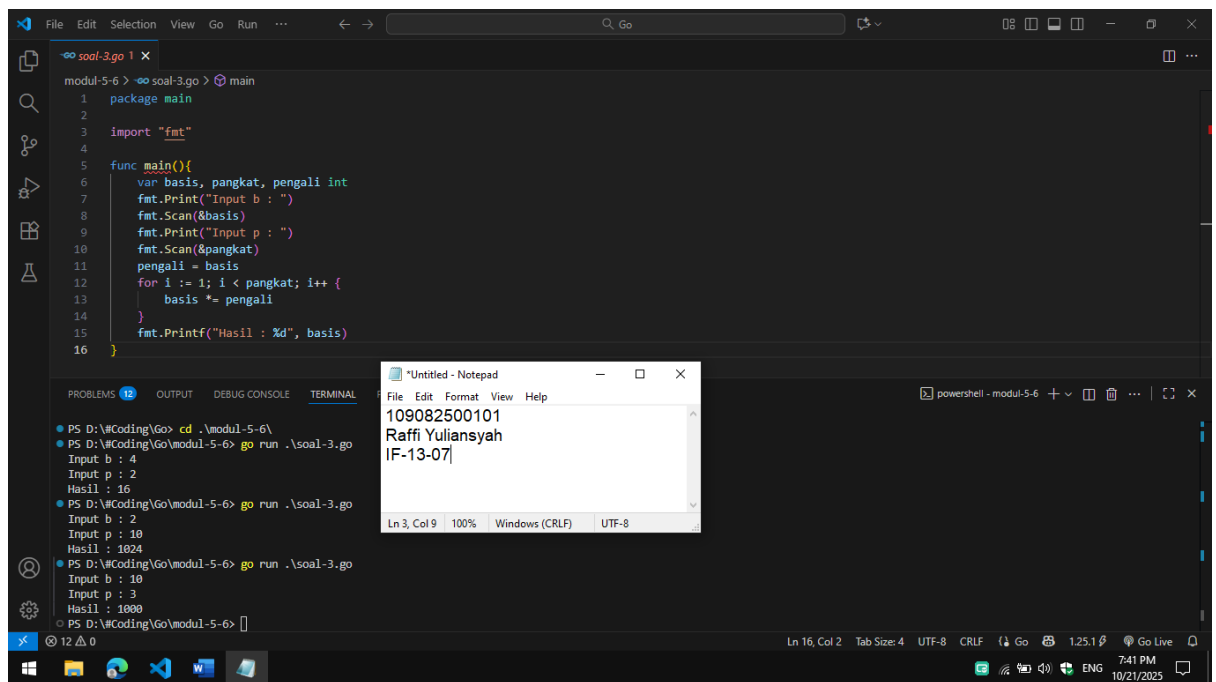
Source code

```
package main

import "fmt"

func main(){
    var basis, pangkat, pengali int
    fmt.Print("Input b : ")
    fmt.Scan(&basis)
    fmt.Print("Input p : ")
    fmt.Scan(&pangkat)
    pengali = basis
    for i := 1; i < pangkat; i++ {
        basis *= pengali
    }
    fmt.Printf("Hasil : %d", basis)
}
```

Screenshoot program



Deskripsi program

Program ini menghitung hasil dari perpangkatan (basis yang dipangkatkan) menggunakan cara perkalian berulang.

Alur program dimulai dengan mendeklarasikan tiga variabel bilangan bulat (int): basis, pangkat, dan pengali. Program kemudian meminta memasukkan dua nilai, yaitu basis (b) dan pangkat (p). Sebelum melakukan perhitungan, nilai basis yang asli disimpan ke dalam variabel pengali sebagai acuan dalam perhitungan.

Program ini adalah sebuah perulangan for yang akan berjalan sebanyak (pangkat - 1) kali. Di dalam setiap putaran perulangan, nilai basis yang sedang berjalan akan dikalikan dengan nilai pengali (yaitu nilai basis asli). Setelah perulangan selesai, variabel basis akan menyimpan nilai akhir hasil perpangkatan, yang kemudian dicetak.

4. Tugas 4

Source code

```
package main

import "fmt"

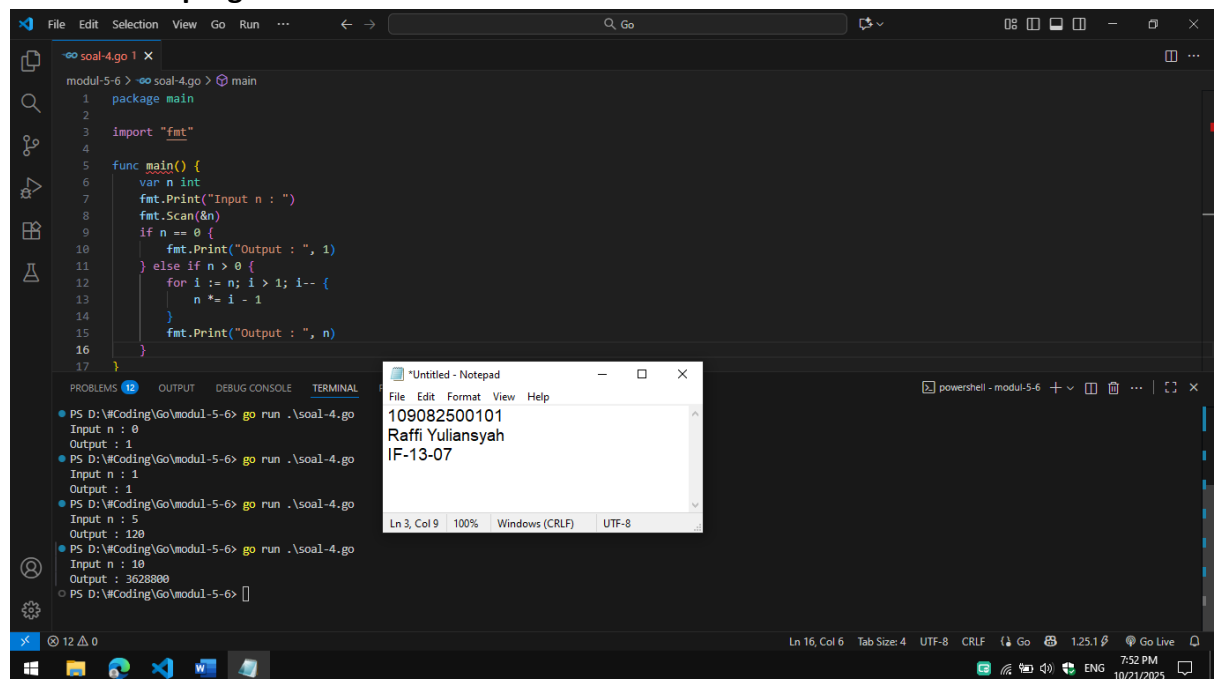
func main() {
    var n int
    fmt.Print("Input n : ")
    fmt.Scan(&n)
    if n == 0 {
        fmt.Print("Output : ", 1)
```

```

    } else if n > 0 {
        for i := n; i > 1; i-- {
            n *= i - 1
        }
        fmt.Print("Output : ", n)
    }
}

```

Screenshoot program



Deskripsi program

Program ini akan menghitung suatu faktorial (n!).

Alur program dimulai dengan mendeklarasikan variabel `n` (int), kemudian meminta memasukkan sebuah angka yang akan disimpan pada variabel `n`. Program kemudian memeriksa nilai `n` yang diinput: jika `n` adalah 0, program akan langsung mencetak "Output : 1" dan selesai. Jika `n` lebih besar dari 0, program akan menghitung nilai faktorial `n` dengan perulangan `for`. Perulangan ini akan mengalikan nilai `n` secara berulang (dengan memanfaatkan `i`) dengan angka yang terus menurun (`i--`), lalu hasilnya dikali, dan seterusnya hingga 1. Setelah perulangan selesai, program akan mencetak hasil faktorial akhir yang tersimpan di dalam variabel `n`.