

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 5 & 6  
FOR-LOOP**



**Disusun oleh:**  
**Dharma Chandra Viriya**  
**109082500052**  
**S1IF-13-02**

**Asisten Praktikum**  
Adithana dharma putra  
Alfin Ilham Berlianto

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## LATIHAN KELAS - GUIDED

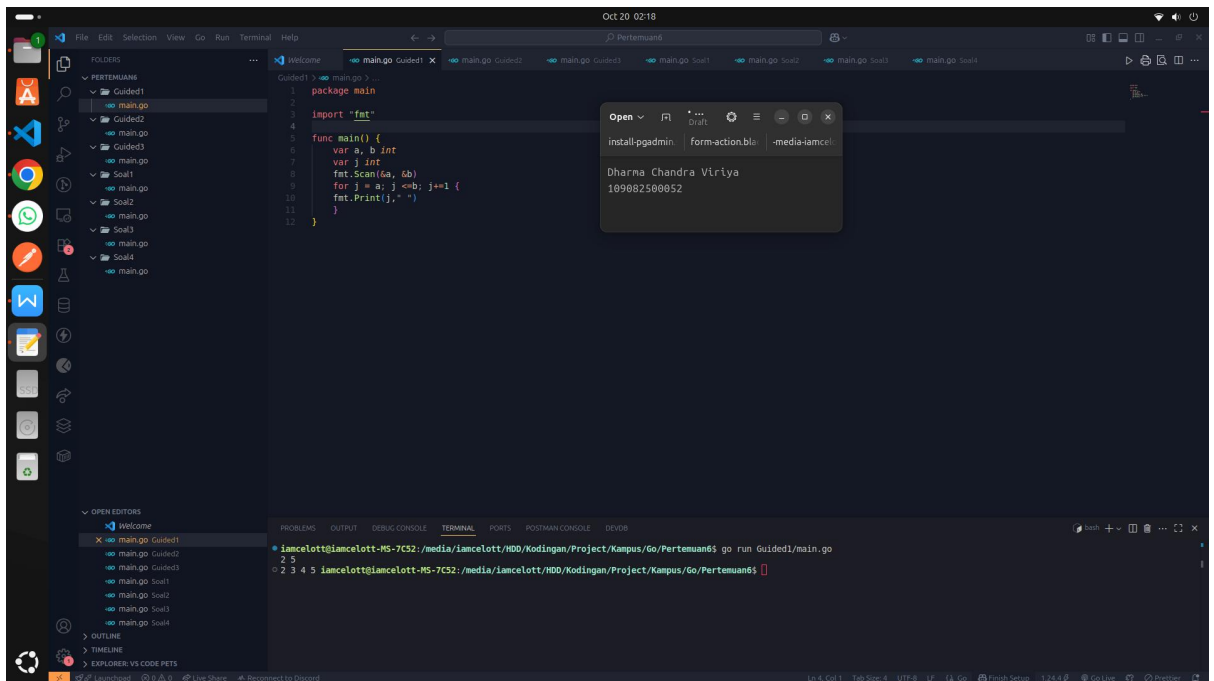
### 1. Guided 1 - Menampilkan Baris Bilangan Source Code

```
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

Kode program di atas merupakan program sederhana dalam bahasa Go yang digunakan untuk menampilkan deretan angka dari satu nilai awal hingga nilai akhir sesuai dengan input pengguna. Pada bagian awal kode, program mendeklarasikan package dengan nama main, kemudian mengimpor library "fmt" yang berfungsi untuk menangani proses input dan output. Selanjutnya, di dalam function main, terdapat tiga variabel yang dideklarasikan, yaitu a dan b bertipe integer untuk menampung nilai awal dan nilai akhir, serta j sebagai variabel penghitung dalam perulangan. Program kemudian meminta input dari pengguna menggunakan fmt.Scan(&a, &b) agar nilai yang dimasukkan dapat disimpan ke dalam variabel a dan b. Setelah itu, program menjalankan perulangan for yang dimulai dari nilai a dan akan terus berjalan hingga mencapai nilai b, dengan setiap iterasi menambah nilai j sebanyak 1. Di dalam perulangan tersebut, program mencetak nilai j ke layar menggunakan fmt.Print, sehingga menghasilkan urutan angka dari nilai awal hingga nilai akhir sesuai input pengguna.

## 2. Guided 2 - Menampilkan Sejumlah n Luas Segitiga

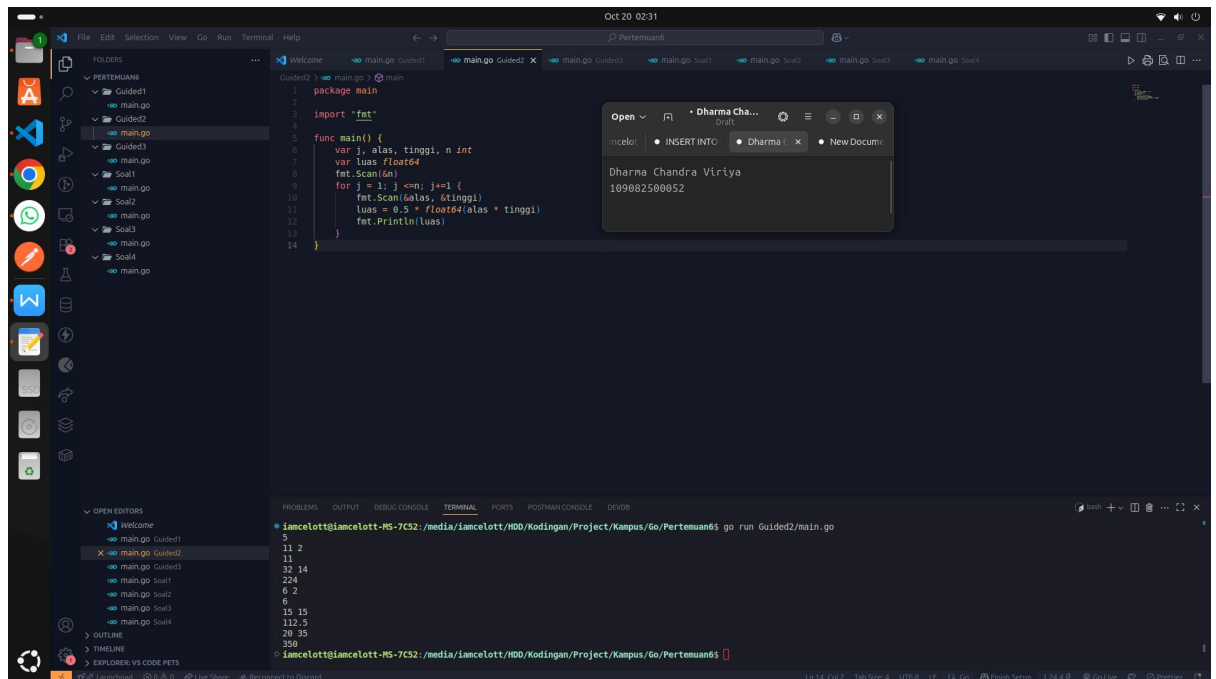
### Source Code

```
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

Kode program di atas merupakan program sederhana dalam bahasa Go yang digunakan untuk menghitung luas segitiga berdasarkan nilai alas dan tinggi yang dimasukkan oleh pengguna secara berulang. Pada bagian awal, program mendeklarasikan package dengan nama main dan mengimpor library "fmt" yang digunakan untuk menangani proses input dan output data. Selanjutnya, di dalam function main, terdapat beberapa variabel yang dideklarasikan, yaitu j, alas, tinggi, dan n dengan tipe data integer, serta luas dengan tipe data float64 untuk menyimpan hasil perhitungan. Program kemudian meminta pengguna memasukkan nilai n, yaitu jumlah segitiga yang ingin dihitung. Setelah itu, program menjalankan perulangan for yang dimulai dari 1 hingga n, di mana pada setiap perulangan pengguna diminta memasukkan nilai alas dan tinggi segitiga. Nilai-nilai tersebut kemudian digunakan untuk menghitung luas dengan rumus  $0.5 * \text{float64}(\text{alas} * \text{tinggi})$ . Hasil perhitungan luas segitiga tersebut selanjutnya ditampilkan ke layar menggunakan `fmt.Println`. Dengan cara ini, pengguna dapat menghitung luas beberapa segitiga secara berurutan hanya dalam satu kali eksekusi program.

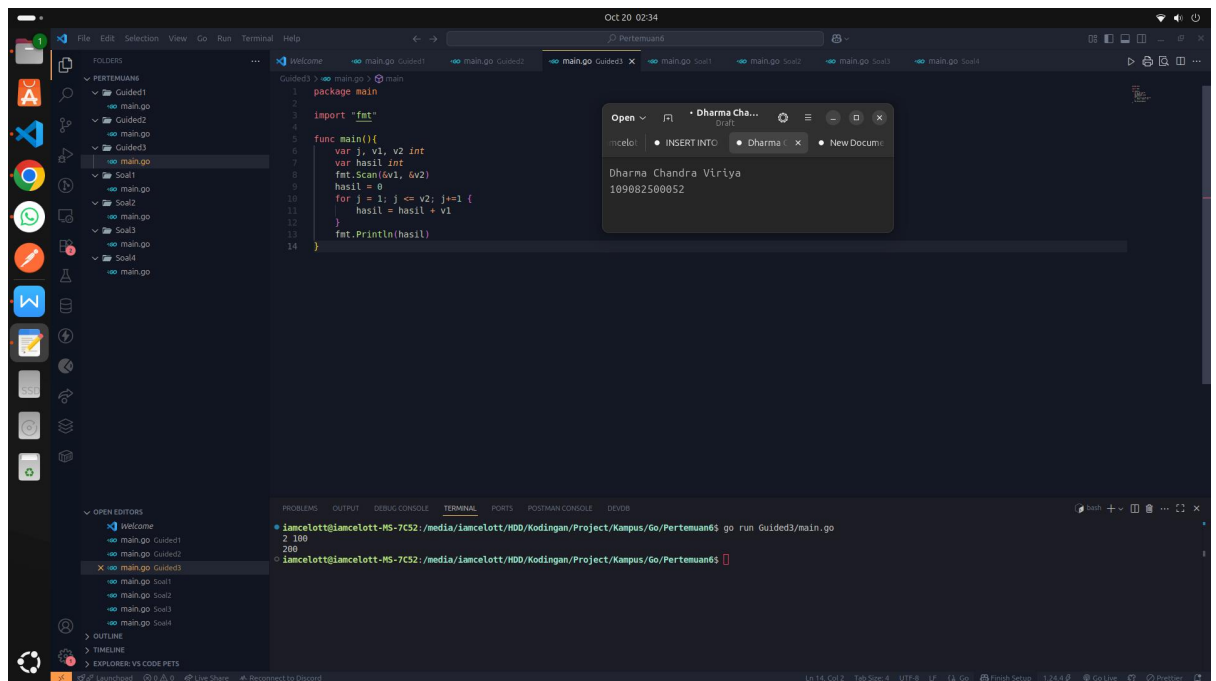
### 3. Guided 3 - Menghitung Hasil Perkalian Tanpa “\*” Source Code

```
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

Kode program di atas merupakan program sederhana dalam bahasa Go yang digunakan untuk melakukan operasi perkalian dua bilangan tanpa menggunakan operator perkalian langsung. Pada bagian awal, program mendeklarasikan package dengan nama main dan mengimpor library "fmt" yang digunakan untuk menangani proses input dan output. Selanjutnya, di dalam function main, terdapat beberapa variabel yang dideklarasikan, yaitu j, v1, dan v2 dengan tipe data integer, serta variabel hasil yang juga bertipe integer untuk menyimpan hasil perhitungan. Program kemudian meminta input dari pengguna berupa dua bilangan, yaitu v1 dan v2, melalui perintah `fmt.Scan(&v1, &v2)`. Setelah itu, nilai awal hasil diinisialisasi dengan 0. Program kemudian menjalankan perulangan `for` yang dimulai dari 1 hingga v2, di mana pada setiap iterasi nilai v1 akan ditambahkan ke variabel hasil. Dengan demikian, proses penjumlahan berulang ini menghasilkan nilai perkalian antara v1 dan v2. Setelah perulangan selesai, hasil akhir perhitungan akan ditampilkan ke layar menggunakan `fmt.Println`.

## TUGAS

### 1. Tugas 1 - Menjumlahkan Sekumpulan Bilangan

**Source code**

```
package main

import "fmt"

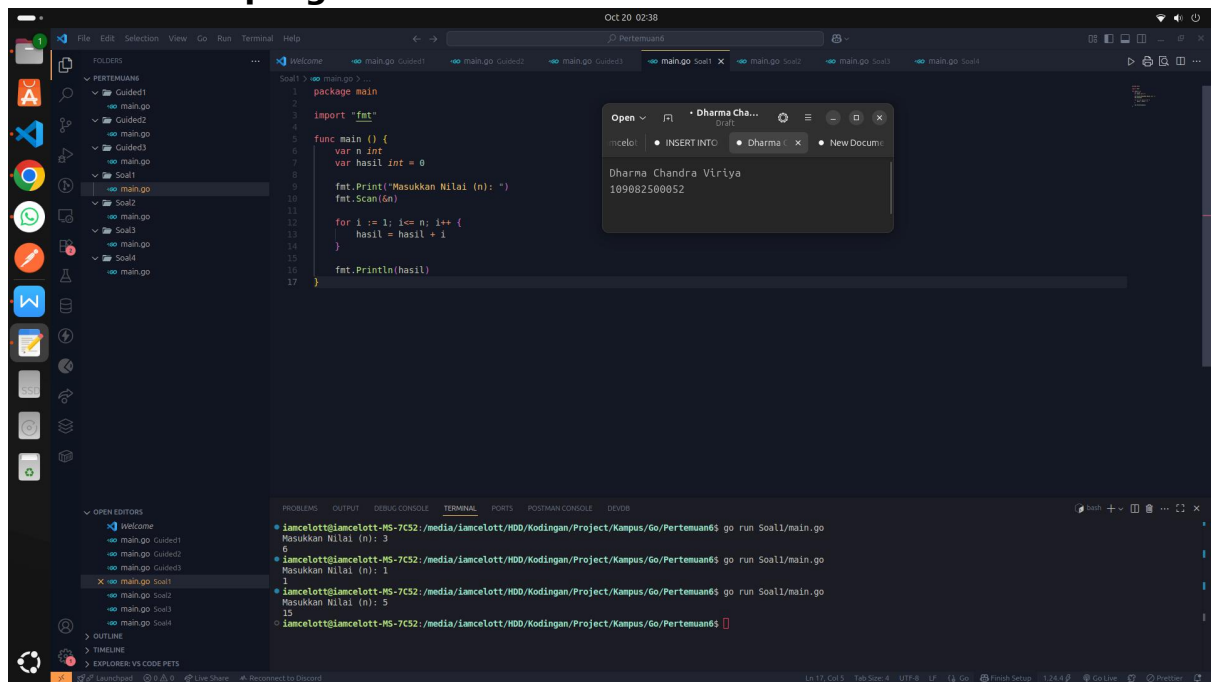
func main () {
    var n int
    var hasil int = 0

    fmt.Print("Masukkan Nilai (n): ")
    fmt.Scan(&n)

    for i := 1; i <= n; i++ {
        hasil = hasil + i
    }

    fmt.Println(hasil)
```

## Screenshoot program



## Deskripsi program

Kode program di atas merupakan program sederhana dalam bahasa Go yang berfungsi untuk menghitung jumlah dari deret bilangan mulai dari 1 hingga nilai yang dimasukkan oleh pengguna. Pada bagian awal, program mendeklarasikan package dengan nama main dan mengimpor library "fmt" yang digunakan untuk menangani proses input dan output. Di dalam function main, terdapat dua variabel yang dideklarasikan, yaitu n dengan tipe data integer untuk menampung nilai batas akhir dari deret bilangan, serta hasil yang diinisialisasi dengan nilai awal 0 untuk menyimpan hasil penjumlahan. Program kemudian menampilkan pesan ke layar menggunakan 'fmt.Print("Masukkan Nilai (n): ")' agar pengguna dapat memasukkan nilai yang diinginkan, dan nilai tersebut disimpan ke dalam variabel n menggunakan 'fmt.Scan(&n)'. Setelah itu, program menjalankan perulangan for yang dimulai dari 1 hingga n, di mana pada setiap iterasi nilai i akan ditambahkan ke dalam variabel hasil. Setelah perulangan selesai, hasil akhir penjumlahan ditampilkan ke layar menggunakan 'fmt.Println(hasil)'. Dengan cara ini, pengguna dapat mengetahui total dari penjumlahan deret bilangan secara otomatis berdasarkan nilai yang mereka inputkan.

## 2. Tugas 2 - Menghitung Volume Sejumlah n Kerucut

### Source code

```
package main

import (
    "fmt"
    "math"
)

func HitungVolumeKerucut(r, t int) float64 {
    return 1.0 / 3.0 * math.Pi * math.Pow(float64(r), 2) *
float64(t)
}

func main() {
    var n, r, t int

    fmt.Print("Masukkan Nilai (n): ")
    fmt.Scan(&n)

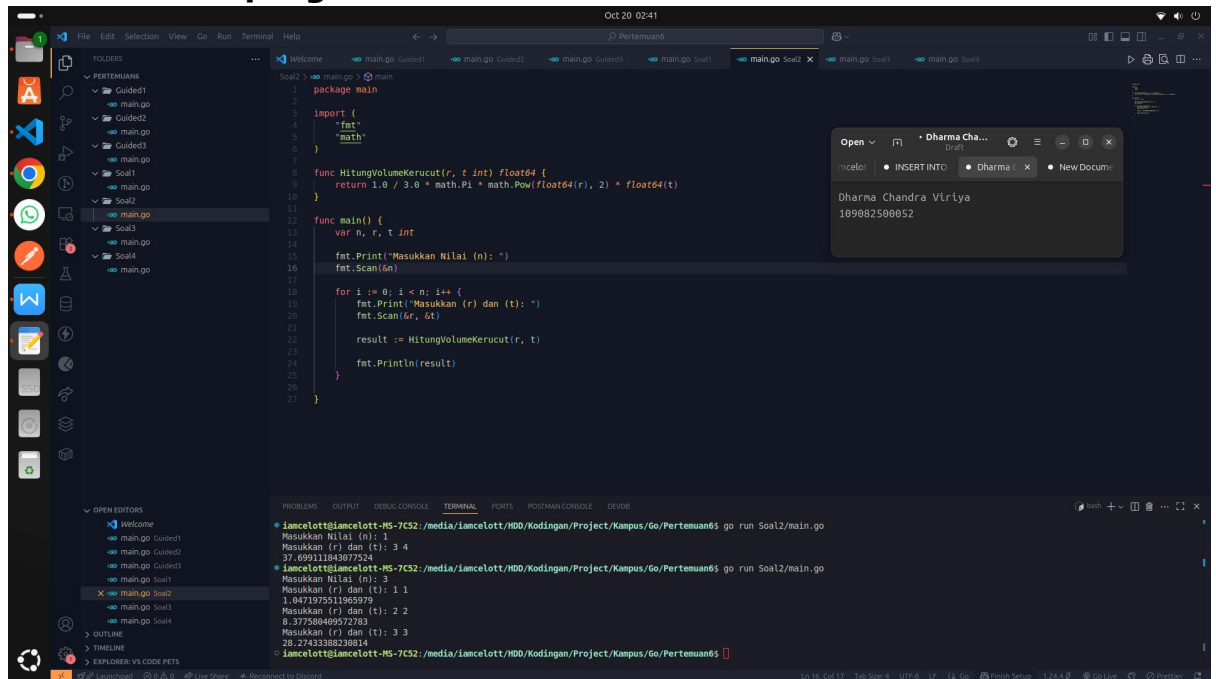
    for i := 0; i < n; i++ {
        fmt.Print("Masukkan (r) dan (t): ")
        fmt.Scan(&r, &t)

        result := HitungVolumeKerucut(r, t)

        fmt.Println(result)
    }
}
```



## Screenshoot program



```
package main

import (
    "fmt"
    "math"
)

func HitungVolumeKerucut(r, t int) float64 {
    return 1.0 / 3.0 * math.PI * math.Pow(float64(r), 2) * float64(t)
}

func main() {
    var n, r, t int
    fmt.Println("Masukkan Nilai (n): ")
    fmt.Scan(&n)

    for i := 0; i < n; i++ {
        fmt.Println("Masukkan (r) dan (t): ")
        fmt.Scan(&r, &t)

        result := HitungVolumeKerucut(r, t)
        fmt.Println(result)
    }
}
```

Terminal output:

```
iamcelott@iamcelott-MS-7C52: /media/iamcelott/HDD/Kodingan/Project/Kampus/Go/Pertemuan6$ go run Soal2/main.go
Masukkan Nilai (n): 1
Masukkan (r) dan (t): 3 4
31.499114877954
iamcelott@iamcelott-MS-7C52: /media/iamcelott/HDD/Kodingan/Project/Kampus/Go/Pertemuan6$ go run Soal2/main.go
Masukkan Nilai (n): 3
Masukkan (r) dan (t): 1 1
1.0471975511965979
Masukkan (r) dan (t): 2 2
8.377580409572783
Masukkan (r) dan (t): 3 3
28.27433388230814
iamcelott@iamcelott-MS-7C52: /media/iamcelott/HDD/Kodingan/Project/Kampus/Go/Pertemuan6$
```

## Deskripsi program

Kode program di atas merupakan program dalam bahasa Go yang digunakan untuk menghitung volume kerucut berdasarkan nilai jari-jari dan tinggi yang dimasukkan oleh pengguna. Pada bagian awal, program mendeklarasikan package dengan nama main serta mengimpor dua library, yaitu "fmt" untuk menangani input dan output, dan "math" untuk melakukan operasi matematika seperti pemangkatan dan penggunaan nilai  $\pi$  (pi). Selanjutnya, terdapat sebuah function bernama HitungVolumeKerucut yang menerima dua parameter bertipe integer, yaitu r untuk jari-jari dan t untuk tinggi, kemudian mengembalikan nilai bertipe float64 dengan rumus  $\frac{1}{3} \times \pi \times r^2 \times t$  untuk menghitung volume kerucut. Di dalam function main, dideklarasikan beberapa variabel yaitu n, r, dan t dengan tipe data integer. Program pertama-tama meminta input nilai n yang menentukan berapa kali perhitungan akan dilakukan, lalu menjalankan perulangan for sebanyak n kali. Pada setiap iterasi, pengguna diminta memasukkan nilai jari-jari (r) dan tinggi (t), kemudian nilai tersebut dikirim ke function HitungVolumeKerucut untuk dihitung. Hasil dari perhitungan volume kerucut kemudian ditampilkan ke layar menggunakan fmt.Println(result). Dengan cara ini, pengguna dapat menghitung beberapa volume kerucut sekaligus hanya dengan menjalankan satu kali program.

### 3. Tugas 3 - Hitung Hasil Pemangkatan Dari Dua Buah Bilangan

#### Source code

```
package main

import "fmt"

func main() {

    var angka, pangkat int

    var result int = 1

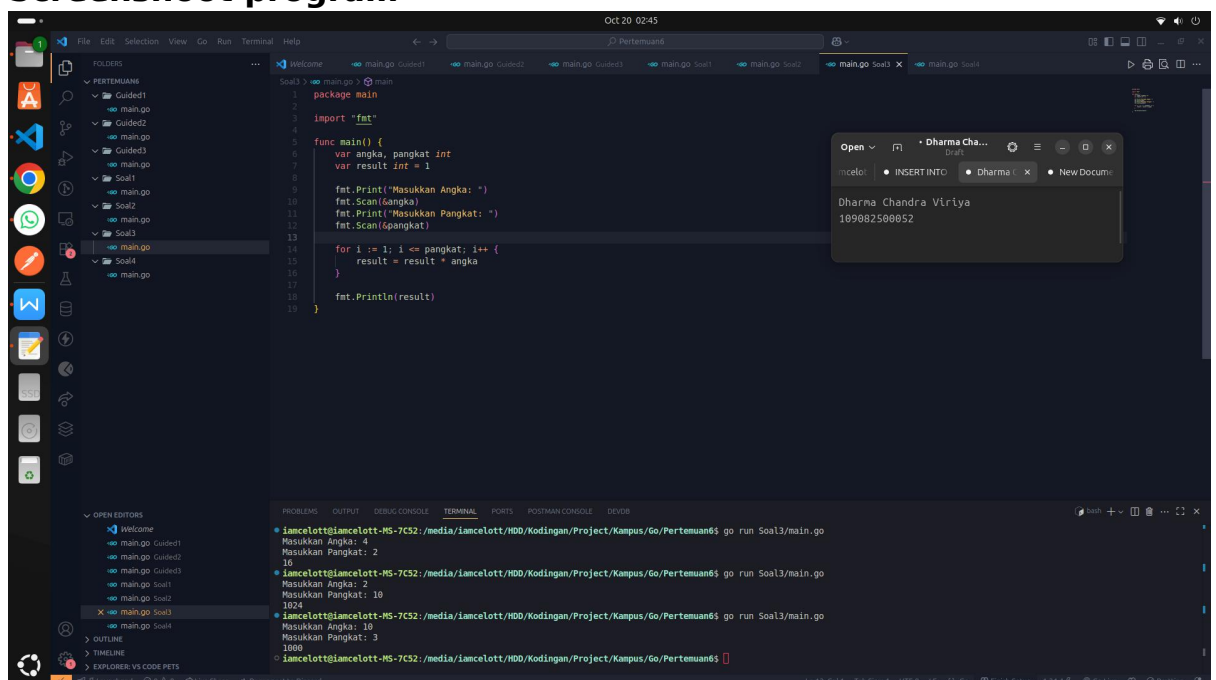

    fmt.Print("Masukkan Angka: ")
    fmt.Scan(&angka)
    fmt.Print("Masukkan Pangkat: ")
    fmt.Scan(&pangkat)


    for i := 1; i <= pangkat; i++ {
        result = result * angka
    }

    fmt.Println(result)

}
```

#### Screenshoot program



### Deskripsi program

Kode program di atas merupakan program sederhana dalam bahasa Go yang digunakan untuk menghitung hasil perpangkatan dari sebuah bilangan berdasarkan nilai yang dimasukkan oleh pengguna. Pada bagian awal, program mendeklarasikan package dengan nama main dan mengimpor library "fmt" yang berfungsi untuk menangani proses input dan output. Di dalam function main, terdapat tiga variabel yang dideklarasikan, yaitu angka dan pangkat dengan tipe data integer untuk menyimpan nilai yang akan diproses, serta result yang diinisialisasi dengan nilai awal 1 untuk menampung hasil perhitungan. Program kemudian meminta pengguna untuk memasukkan nilai angka melalui perintah `fmt.Scan(&angka)` dan nilai pangkat melalui `fmt.Scan(&pangkat)`. Setelah kedua nilai diperoleh, program menjalankan perulangan for yang berjalan dari 1 hingga nilai pangkat, di mana pada setiap iterasi variabel result dikalikan dengan angka. Proses ini akan mengalikan nilai angka sebanyak jumlah pangkat yang dimasukkan oleh pengguna. Setelah perulangan selesai, hasil akhir perhitungan ditampilkan ke layar menggunakan `fmt.Println(result)`. Dengan cara ini, program dapat menghitung nilai perpangkatan tanpa menggunakan fungsi matematika bawaan, melainkan melalui proses perkalian berulang.

### 4. Tugas 4 - Hitung Hasil Faktorial Dari Suatu Bilangan

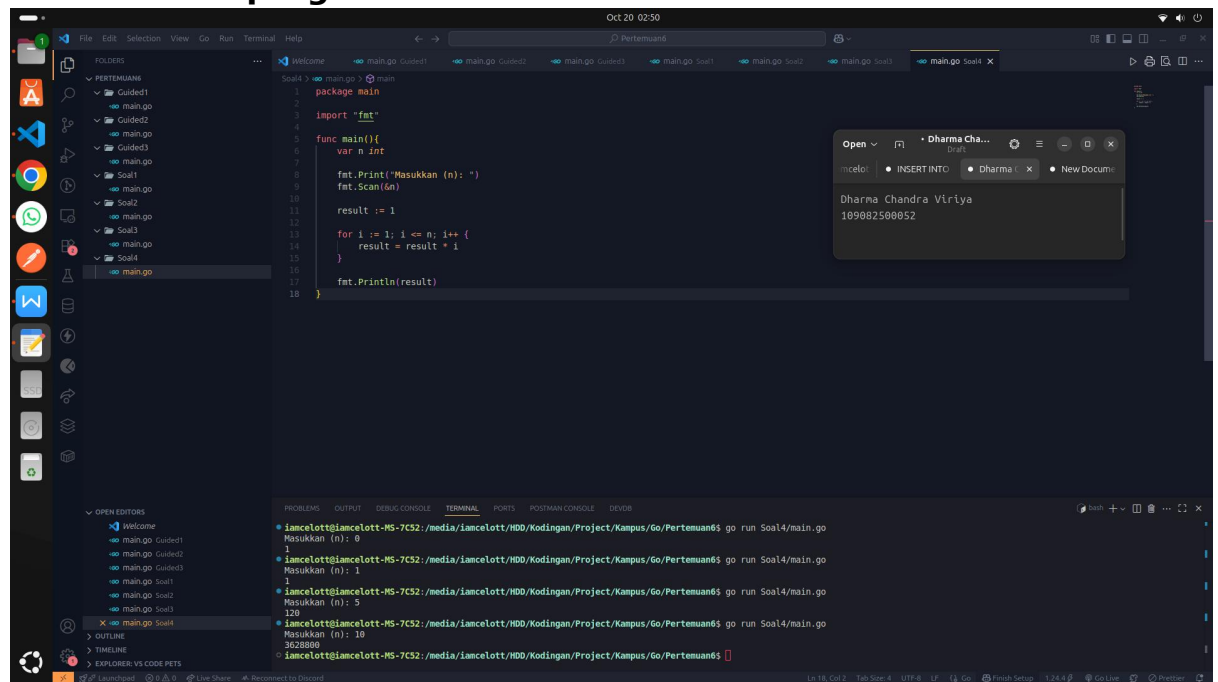
#### Source code

```
package main

import "fmt"

func main(){
    var n int
    fmt.Print("Masukkan (n): ")
    fmt.Scan(&n)
    result := 1
    for i := 1; i <= n; i++ {
        result = result * i
    }
    fmt.Println(result)
}
```

## Screenshoot program



## Deskripsi program

Kode program di atas merupakan program sederhana dalam bahasa Go yang digunakan untuk menghitung nilai faktorial dari suatu bilangan yang dimasukkan oleh pengguna. Pada bagian awal, program mendeklarasikan package dengan nama main dan mengimpor library "fmt" yang berfungsi untuk menangani proses input dan output. Di dalam function main, terdapat sebuah variabel n bertipe integer yang digunakan untuk menyimpan nilai yang akan dihitung, serta variabel result yang diinisialisasi dengan nilai awal 1 untuk menampung hasil perhitungan faktorial. Program kemudian menampilkan pesan ke layar menggunakan `fmt.Print("Masukkan (n): ")` untuk meminta pengguna memasukkan sebuah bilangan, dan nilai tersebut disimpan ke dalam variabel n menggunakan `fmt.Scan(&n)`. Setelah itu, program menjalankan perulangan for yang dimulai dari 1 hingga n, di mana pada setiap iterasi nilai result dikalikan dengan nilai i. Proses ini terus berulang sampai perulangan selesai, sehingga menghasilkan nilai faktorial dari bilangan yang dimasukkan. Setelah semua proses selesai, hasil perhitungan ditampilkan ke layar menggunakan `fmt.Println(result)`. Dengan cara ini, pengguna dapat mengetahui hasil faktorial dari bilangan yang mereka inputkan secara langsung.