

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 5 & 6  
FOR-LOOP**



**Disusun oleh:**

**RAFI OKTARINO RAMADHAN**

**109082500217**

**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)

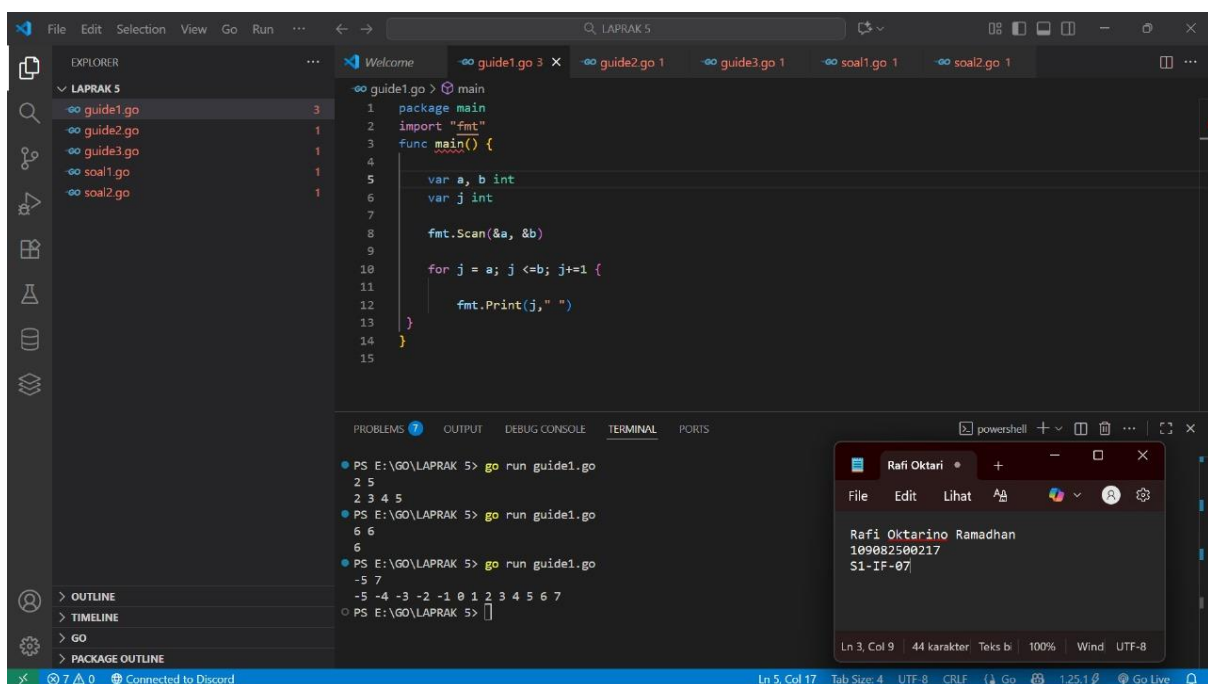
    for j = a; j <=b; j+=1 {

        fmt.Print(j, " ")

    }

}
```

### Screenshoot program



### Deskripsi program

Saya di perintahkan untuk menampilkan baris bilangan dari a sampai b. Gunakan *var* untuk menyimpan data a,b yang berfungsi untuk mengetahui awalan dan akhiran bilangan bulat. *fmt.Scan* untuk membaca apa yang sudah di simpan di *var* yang berisi a,b dan langkah selanjutnya memasukan rumus perulangan untuk menampilkan bilangan a sampai dengan b yang code nya berisi `for j = a; j <=b; j+=1`. Penjelasanya, untuk *j* yang berfungsi untuk menghitung angka yang akan di hitung, jadi jika *a* = 3 dan *b* = 7 maka perulangan tersebut dimulai dari 3 dan berakhir pada 7. Fungsi dari *j+=1* jika perulangan nya selesai satu kali maka *j* akan bertambah satu hingga seterusnya. Lalu yang terakhir *fmt.Print(j, " ")* yang berfungsi untuk mencetak nilai *j* di layar diikuti dengan spasi, supaya rapih.

## 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)

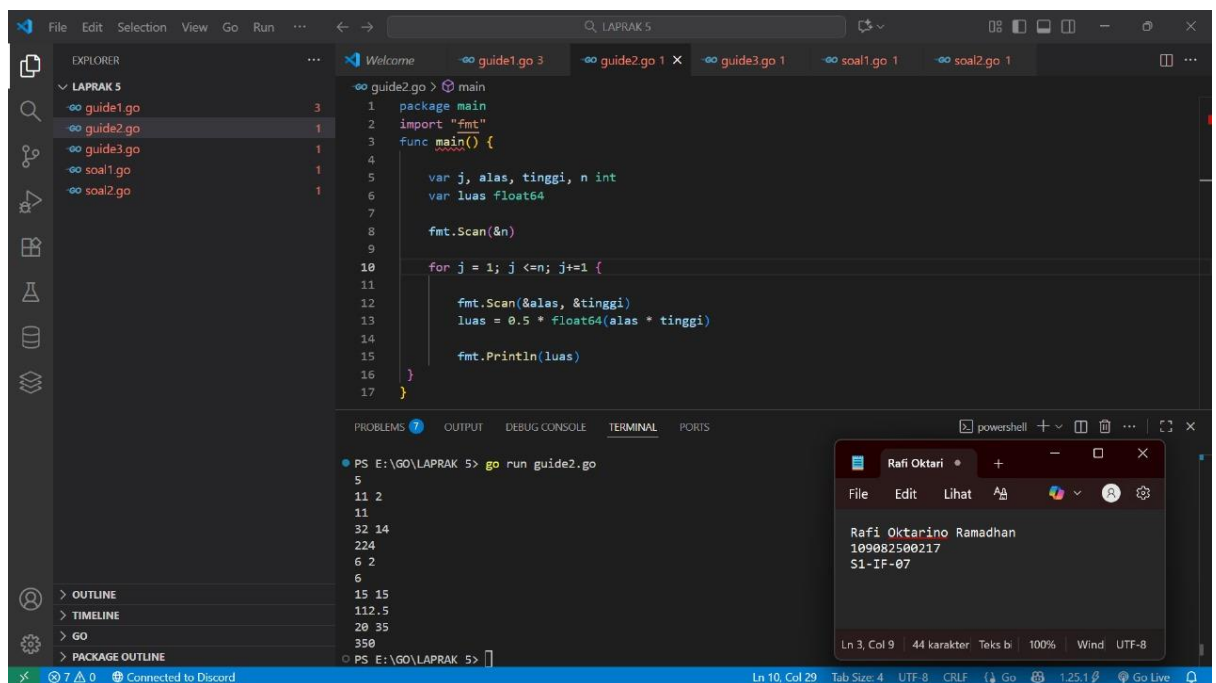
    }

    fmt.Println(luas)

}

```

## Screenshoot program



## Deskripsi program

Saya di perintahkan untuk menghitung luas segitiga secara berulang berdasarkan jumlah data yang dimasukkan pengguna. Di awal program, dideklarasikan beberapa variabel menggunakan var, yaitu alas dan tinggi untuk menyimpan nilai sisi segitiga, n untuk menentukan berapa kali perhitungan dilakukan, serta luas bertipe float64 untuk menyimpan hasil perhitungan agar dapat menampung angka desimal. Program membaca input nilai n menggunakan `fmt.Scan(&n)` yang berarti pengguna akan memasukkan berapa banyak data segitiga yang mau dihitung. Setelah itu, digunakan perulangan `for j = 1; j <= n; j += 1` yang berfungsi untuk mengulang proses perhitungan sebanyak n kali. Di dalam perulangan, program meminta pengguna untuk memasukkan nilai alas dan tinggi menggunakan `fmt.Scan(&alas, &tinggi)`, kemudian menghitung luas segitiga dengan rumus `luas = 0.5 * float64(alas * tinggi)`,

dan hasilnya ditampilkan menggunakan `fmt.Println(luas)`. Dengan demikian, setiap kali perulangan berjalan, program akan menampilkan hasil luas segitiga sesuai nilai alas dan tinggi yang dimasukkan hingga perulangan berakhir.

### 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 {

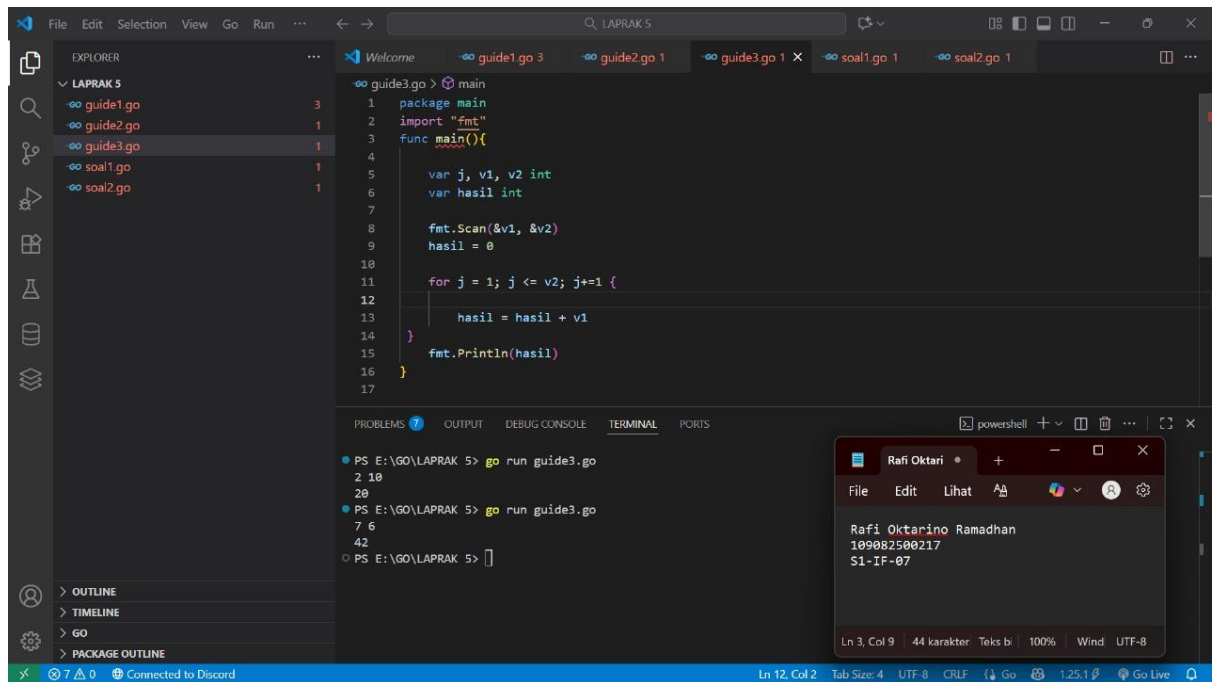
        hasil = hasil + v1

    }

    fmt.Println(hasil)

}
```

#### Screenshoot program



### Deskripsi program

Sesuai dengan perintah saya diminta untuk menghitung hasil perkalian dua buah bilangan tanpa menggunakan operator “\*”. Package main, import “fmt” dan func main adalah code agar program bisa berjalan dengan baik. Dilanjut dengan menyimpan nilai j , v1, dan v2 dengan var dan fungsi j tersebut digunakan untuk menghitung angka dalam perulangan, disini saya menggunakan int dikarenakan output nya bilangan bulat . Setelah itu masukan v1 dan v2 di fmt.Scan, jangan lupa tanda “&” untuk memberikan alamat pada v1 dan v2. Setelah itu masukan rumus perulangan. Langkah terakhir tampilkan outputnya menggunakan fmt.Println(hasil).

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

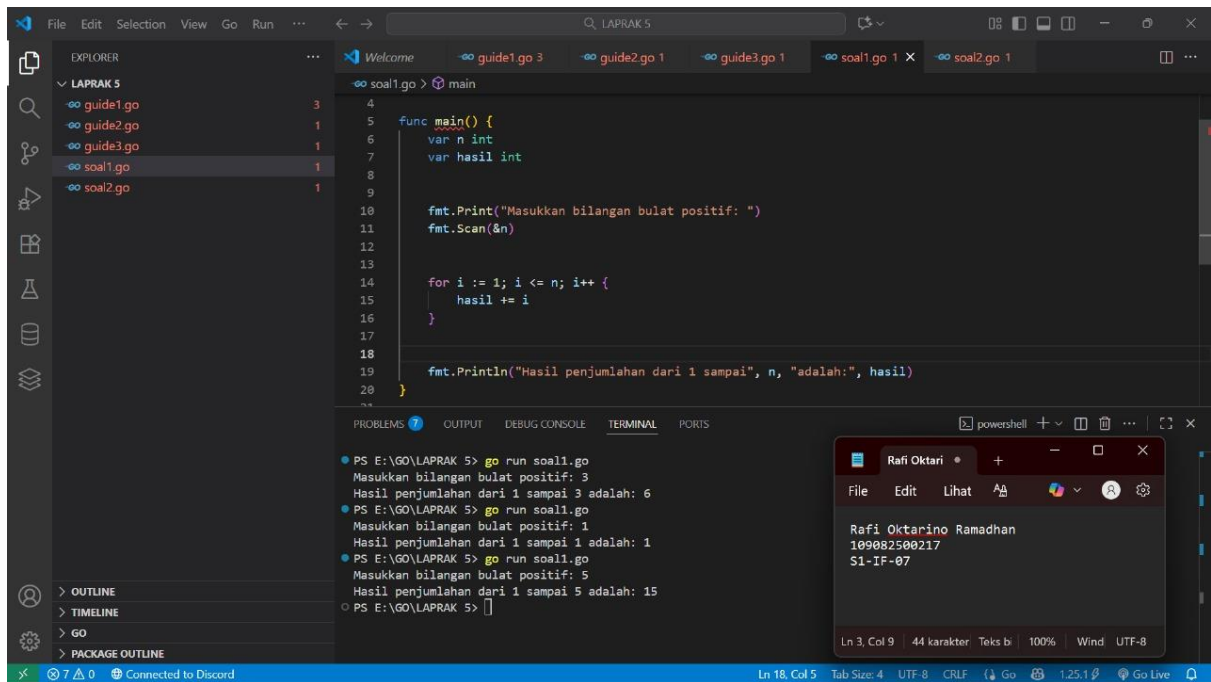
func main() {
    var n int
    var hasil int

    fmt.Print("Masukkan bilangan bulat positif: ")
    fmt.Scan(&n)

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

    fmt.Println("Hasil penjumlahan dari 1 sampai", n,
        "adalah:", hasil)
}
```

## Screenshoot program



## Deskripsi program

Sesuai dengan perintah saya diminta untuk menjumlahkan bilangan yang Dimana bilangan tersebut berurutan missal dari 1 sampai 3 itu hasilnya 6. Package main, import "fmt" dan func main adalah code agar program bisa berjalan dengan baik. Dilanjut dengan print untuk text nya menggunakan fmt.Print(). Setelah itu gunakan fmt.Scan(&n) agar inputan bisa terbaca dan tersimpan di var. Jangan lupa masukan rumus perulangan for i := 1; i <= n; i++ { hasil+= i } supaya bisa menghasilkan jawaban perulangan dan mengatur program melakukan perulangan dan tampilkan outputnya menggunakan fmt.Println.



## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var n int

    const pi = 3.141592653589793

    fmt.Print("Masukkan jumlah kerucut: ")

    fmt.Scan(&n)

    for i := 1; i <= n; i++ {
        var r, t float64

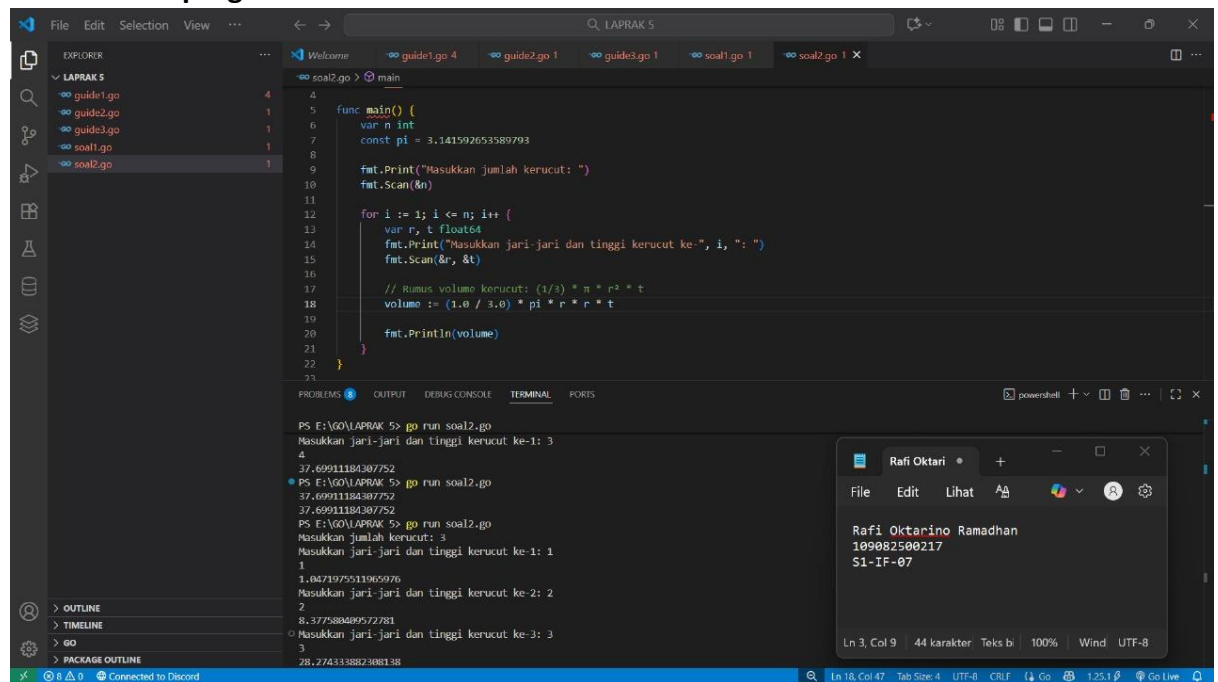
        fmt.Print("Masukkan jari-jari dan tinggi
kerucut ke-", i, ": ")

        fmt.Scan(&r, &t)

        volume := (1.0 / 3.0) * pi * r * r * t

        fmt.Println(volume)
    }
}
```

## Screenshoot program



```
4
5 func main() {
6     var n int
7     const pi = 3.141592653589793
8
9     fmt.Print("Masukkan jumlah kerucut: ")
10    fmt.Scan(&n)
11
12    for i := 1; i <= n; i++ {
13        var r, t float64
14        fmt.Print("Masukkan jari-jari dan tinggi kerucut ke-", i, ": ")
15        fmt.Scan(&r, &t)
16
17        // Rumus volume kerucut: (1/3) * pi * r^2 * t
18        volume := (1.0 / 3.0) * pi * r * r * t
19
20        fmt.Println(volume)
21    }
22 }
```

PS E:\GO\LAPRAK 5> go run soal2.go  
Masukkan jari-jari dan tinggi kerucut ke-1: 3  
4  
37.69911184387752  
PS E:\GO\LAPRAK 5> go run soal2.go  
37.69911184387752  
37.69911184387752  
PS E:\GO\LAPRAK 5> go run soal2.go  
Masukkan jumlah kerucut: 3  
Masukkan jari-jari dan tinggi kerucut ke-1: 1  
1  
1.0471975511965976  
Masukkan jari-jari dan tinggi kerucut ke-2: 2  
2  
8.377580499572781  
Masukkan jari-jari dan tinggi kerucut ke-3: 3  
3  
28.274333882386138

## Deskripsi program

Sesuai dengan perintah saya diminta untuk menghitung volume sejumlah n kerucut, apabila diketahui panjang jari-jari alas kerucut dan tinggi dari kerucut. Package main, import "fmt" dan func main adalah code agar program bisa berjalan dengan baik. Dilanjut dengan menyimpan n di var dan menggunakan int dikarenakan hasilnya bilangan bulat dan ada tambahan const pi = 3.141592653589793 digunakan untuk menghitung kerucut, setelah itu gunakan fmt.Scan(&n) agar inputan bisa terbaca dan tersimpan di var. dan disini saya membuat variabel baru untuk menyimpan input jari jari dan tinggi kerucut dan menambahkan code perulangan. Jangan lupa fmt.Scan untuk menerima inputannya. Dan masukan rumus menghitung volume kerucut, yang terakhir tampilkan outpunya menggunakan fmt.Println(volume).

### 3. Tugas 3

#### Source code

```
package main

import "fmt"

func main() {

    var a, b int

    var hasil int = 1


    fmt.Print("Masukkan dua bilangan bulat positif
basis dan pangkat: ")

    fmt.Scan(&a, &b)


    for i := 1; i <= b; i++ {

        hasil *= a

    }


    fmt.Println("Hasil:", hasil)

}
```

## Screenshoot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b int
7     var hasil int = 1
8
9
10    fmt.Print("Masukkan dua bilangan bulat positif basis dan pangkat: ")
11    fmt.Scan(&a, &b)
12
13
14    for i := 1; i <= b; i++ {
15        hasil *= a
16    }
17
18    fmt.Println("Hasil:", hasil)
19 }
20
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS E:\GO\LAPRAK 5> go run soal3.go  
Masukkan dua bilangan bulat positif basis dan pangkat: 4 2  
Hasil: 16
- PS E:\GO\LAPRAK 5> go run soal3.go  
Masukkan dua bilangan bulat positif basis dan pangkat: 2 10  
Hasil: 1804
- PS E:\GO\LAPRAK 5> 10 3  
At line:1 char:4  
+ 10 3  
+ ~~~~  
Unexpected token '3' in expression or statement.  
+ categoryInfo : ParserError: (1) [], ParentContainsErrorRecordException  
+ fullyQualifiedErrorId : UnexpectedToken
- PS E:\GO\LAPRAK 5> go run soal3.go  
Masukkan dua bilangan bulat positif basis dan pangkat: 10 3  
Hasil: 1000
- PS E:\GO\LAPRAK 5> ]

## Deskripsi program

Sesuai dengan perintah saya diminta untuk menghitung hasil pemangkatan dari dua buah bilangan. Program dibuat dengan menggunakan operator perkalian dan struktur kontrol perulangan. Package main, import "fmt" dan func main adalah code agar program bisa berjalan dengan baik. Dilanjut dengan menyimpan nilai a ,b, int dan ada tambahan hasil int = 1 untuk memulai nilai awalnya dari angka 1 dengan var, setelah itu gunakan fmt.Scan(&a, &b) agar inputan bisa terbaca dan tersimpan di var. dan masukan code perulangan agar bisa menghitung sebanyak apa yang di perintahkan dan hasil\*= a untuk mengalikan hasil yang diperintahkan, jangan lupa tambahkan fmt.Println("Hasil:", hasil) untuk menampilkan outputnya.

#### 4. Tugas 4

##### Source code

```
package main

import "fmt"

func main() {

    var n int

    var hasil int = 1


    fmt.Print("Masukkan bilangan bulat non-negatif: ")

    fmt.Scan(&n)


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

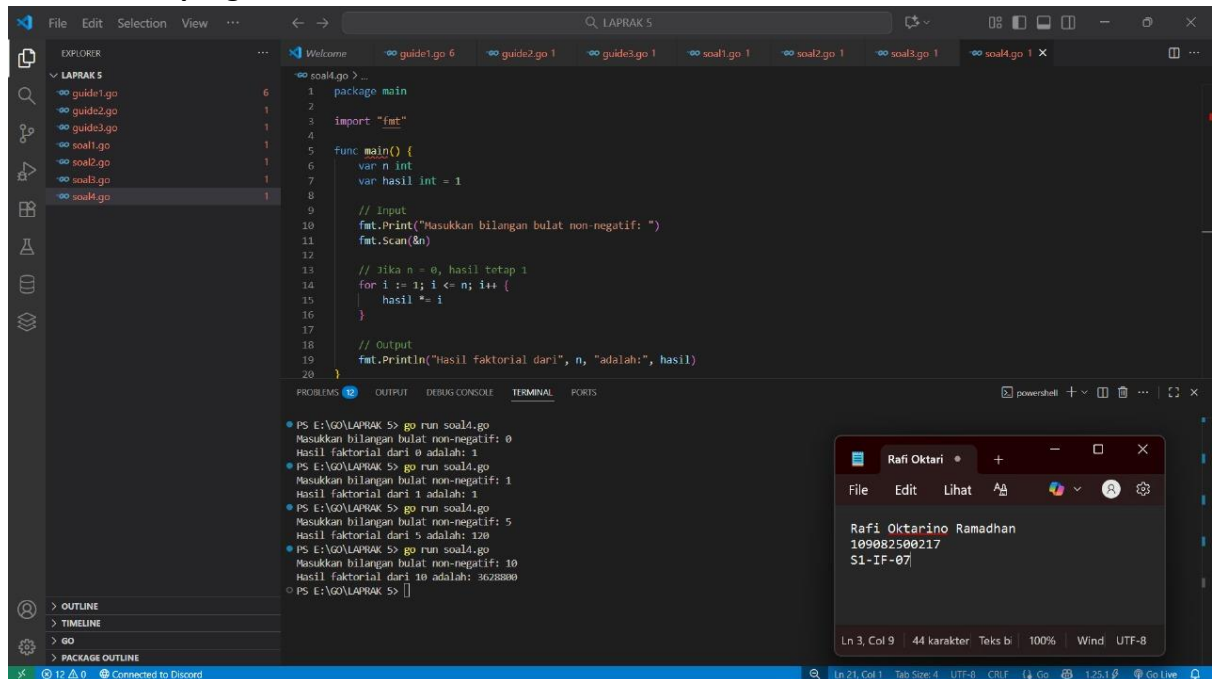
        hasil *= i

    }


    fmt.Println("Hasil faktorial dari", n, "adalah:",
hasil)

}
```

## Screenshoot program



The screenshot displays a Go IDE with a project named 'LAPRAK 5'. The Explorer panel on the left shows a directory structure with files: guide1.go, guide2.go, guide3.go, soal1.go, soal2.go, soal3.go, and soal4.go. The main editor window shows the code for 'soal4.go', which is a Go program to calculate the factorial of a non-negative integer. The code includes package declarations, imports, variable declarations, input handling, a loop for calculation, and output printing. The terminal panel at the bottom shows the execution of the program for inputs 0, 1, 5, and 10, with corresponding factorial results. A small window titled 'Rafi Oktari' is also visible in the background.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     var hasil int = 1
8
9     // Input
10    fmt.Print("Masukkan bilangan bulat non-negatif: ")
11    fmt.Scan(&n)
12
13    // Jika n = 0, hasil tetap 1
14    for i := 1; i <= n; i++ {
15        hasil *= i
16    }
17
18    // Output
19    fmt.Println("Hasil faktorial dari", n, "adalah:", hasil)
20 }
```

Terminal Output:

```
PS E:\GO\LAPRAK 5> go run soal4.go
Masukkan bilangan bulat non-negatif: 0
Hasil faktorial dari 0 adalah: 1
PS E:\GO\LAPRAK 5> go run soal4.go
Masukkan bilangan bulat non-negatif: 1
Hasil faktorial dari 1 adalah: 1
PS E:\GO\LAPRAK 5> go run soal4.go
Masukkan bilangan bulat non-negatif: 5
Hasil faktorial dari 5 adalah: 120
PS E:\GO\LAPRAK 5> go run soal4.go
Masukkan bilangan bulat non-negatif: 10
Hasil faktorial dari 10 adalah: 3628800
PS E:\GO\LAPRAK 5>
```

## Deskripsi program

Sesuai dengan perintah saya diminta menghitung hasil faktorial dari suatu bilangan. Package main, import "fmt" dan func main adalah code agar program bisa berjalan dengan baik. Dilanjut dengan menyimpan nilai n int dan ada tambahan hasil int = 1 untuk memulai nilai awalnya dari angka 1 dengan var, setelah itu gunakan fmt.Scan(&n) agar inputan bisa terbaca dan tersimpan di var. dan masukan code perulangan agar bisa menghitung sebanyak apa yang di perintahkan dan hasil\*= i untuk mengalikan hasil yang diperintahkan, jangan lupa tambahkan fmt.Println("Hasil:", hasil) untuk menampilkan outputnya.