

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL PRAKTIKUM 5 & 6
FOR-LOOP**



Disusun oleh:

Didi Hermawanto

109082500088

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

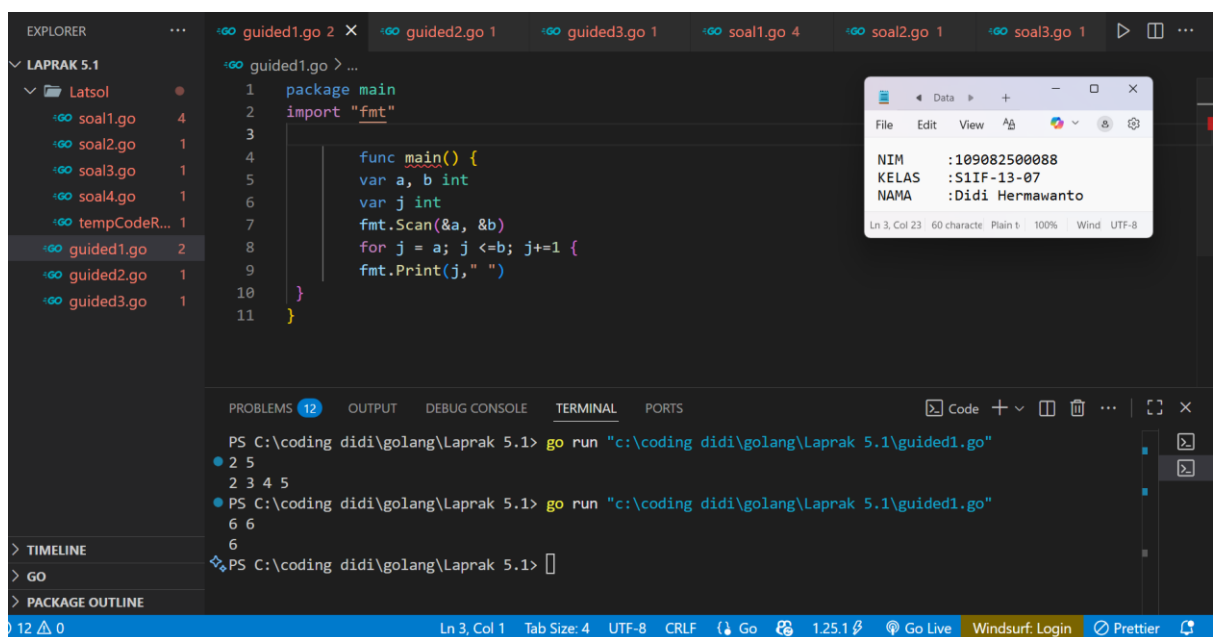
1. Guided 1 Source Code

```
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 digunakan untuk menampilkan deretan bilangan bulat dari nilai awal a hingga nilai akhir b berdasarkan input dari pengguna. Program ini bertujuan untuk melatih pemahaman tentang struktur perulangan for dalam bahasa Go dan cara menampilkan data secara berurutan.

Penjelasan Kode Program:

1. package main → mendefinisikan bahwa program ini merupakan program utama.
2. import "fmt" → mengimpor paket fmt yang digunakan untuk proses input dan output.
3. var a, b int → mendeklarasikan dua variabel bertipe integer sebagai batas awal dan batas akhir.
4. var j int → mendeklarasikan variabel j sebagai penghitung dalam perulangan.
5. fmt.Scan(&a, &b) → membaca dua nilai input dari pengguna, yaitu nilai a dan b.
6. for j = a; j <= b; j += 1 → perulangan yang dimulai dari nilai a dan terus bertambah 1 sampai j mencapai nilai b.
7. fmt.Print(j, " ") → menampilkan nilai j di layar dengan spasi di antara setiap angka.

Algoritma Program:

1. Mulai program.
2. Input dua bilangan bulat, yaitu a sebagai batas awal dan b sebagai batas akhir.
3. Lakukan perulangan mulai dari j = a hingga j <= b.
4. Cetak nilai j setiap kali perulangan berjalan.
5. Ulangi sampai j sama dengan b.
6. Program selesai.

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

```
1 package main
2 import "fmt"
3
4 func main() {
5     var j, alas, tinggi, n int
6     var luas float64
7     fmt.Scan(&n)
8     for j = 1; j <= n; j += 1 {
9         fmt.Scan(&alas, &tinggi)
10        luas = 0.5 * float64(alas * tinggi)
11        fmt.Println(luas)
12    }
13 }
```

Terminal Output:

```
PS C:\coding didi\golang\Laprak 5.1> go run "c:\coding didi\golang\Laprak 5.1\tempCodeRunnerFile.go"
3
12 32
192
231 234
27027
43 34
731
```

Deskripsi Program:

Program ini digunakan untuk menghitung **luas sejumlah segitiga** berdasarkan input dari pengguna. Pengguna diminta memasukkan jumlah segitiga yang ingin dihitung (n), kemudian untuk setiap segitiga dimasukkan nilai **alas** dan **tinggi**. Program akan menghitung luas masing-masing segitiga menggunakan rumus $\frac{1}{2} \times \text{alas} \times \text{tinggi}$, lalu menampilkan hasilnya satu per satu. Program ini bermanfaat untuk memahami penggunaan perulangan dan konversi tipe data dalam perhitungan matematika menggunakan bahasa Go.

Penjelasan Kode Program:

1. package main → menandakan bahwa program ini adalah program utama yang dapat dijalankan.
2. import "fmt" → mengimpor paket fmt untuk menangani input dan output.
3. var j, alas, tinggi, n int → mendeklarasikan variabel j sebagai penghitung perulangan, alas dan tinggi untuk menyimpan nilai setiap segitiga, serta n untuk jumlah segitiga yang akan dihitung.
4. var luas float64 → mendeklarasikan variabel luas bertipe float64 agar hasil perhitungan bisa berupa bilangan desimal.
5. fmt.Scan(&n) → membaca input jumlah segitiga dari pengguna.
6. for j = 1; j <= n; j += 1 → perulangan dijalankan sebanyak n kali untuk menghitung luas setiap segitiga.
7. fmt.Scan(&alas, &tinggi) → pada setiap perulangan, pengguna memasukkan nilai alas dan tinggi segitiga.
8. luas = 0.5 * float64(alas * tinggi) → menghitung luas segitiga dengan rumus $\frac{1}{2} \times \text{alas} \times \text{tinggi}$. Nilai alas dan tinggi dikonversi ke tipe float64 agar hasilnya tidak dibulatkan.
9. fmt.Println(luas) → menampilkan hasil perhitungan luas segitiga ke layar.

Algoritma Program:

1. Mulai program.
2. Input jumlah segitiga (n).
3. Ulangi proses sebanyak n kali.
4. Untuk setiap segitiga:
 - Input nilai alas dan tinggi.
 - Hitung luas dengan rumus $0.5 \times \text{alas} \times \text{tinggi}$.
 - Tampilkan hasil luas.
5. Setelah semua segitiga dihitung, program selesai.

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

```
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)
}
```

PS C:\coding didi\golang\Laprak 5.1> go run "c:\coding didi\golang\Laprak 5.1\tempCodeRunnerFile.go"

2 100
200

PS C:\coding didi\golang\Laprak 5.1> go run "c:\coding didi\golang\Laprak 5.1\tempCodeRunnerFile.go"

7 6
42

PS C:\coding didi\golang\Laprak 5.1>

Deskripsi Program:

Program ini digunakan untuk menghitung hasil perkalian dua bilangan bulat dengan cara menjumlahkan bilangan pertama secara berulang sebanyak nilai bilangan kedua. Program ini tidak menggunakan operator perkalian (*), melainkan memanfaatkan perulangan untuk memahami konsep dasar perkalian melalui penjumlahan berulang.

Penjelasan Kode Program:

1. package main → mendefinisikan bahwa program ini merupakan program utama.
2. import "fmt" → mengimpor paket fmt yang digunakan untuk membaca input dan menampilkan output.
3. var j, v1, v2 int → mendeklarasikan tiga variabel bertipe integer, yaitu v1 (bilangan pertama), v2 (bilangan kedua), dan j (variabel penghitung perulangan).
4. var hasil int → mendeklarasikan variabel hasil untuk menyimpan hasil akhir dari penjumlahan berulang.
5. fmt.Scan(&v1, &v2) → membaca dua input dari pengguna, yaitu nilai v1 dan v2.
6. hasil = 0 → menginisialisasi nilai awal hasil dengan 0 sebelum dilakukan penjumlahan.
7. for j = 1; j <= v2; j += 1 → melakukan perulangan sebanyak v2 kali, dimulai dari j = 1 sampai j = v2.
8. hasil = hasil + v1 → pada setiap perulangan, nilai v1 akan ditambahkan ke variabel hasil.
9. fmt.Println(hasil) → setelah perulangan selesai, hasil akhir dari perkalian ditampilkan ke layar.

Algoritma Program:

1. Mulai program.
2. Input dua bilangan bulat v1 dan v2.
3. Set nilai awal hasil = 0.
4. Lakukan perulangan dari j = 1 sampai j = v2.

5. Tambahkan nilai v1 ke hasil pada setiap perulangan.
6. Setelah selesai, tampilkan hasil akhir ke layar.
7. Program selesai.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

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

    sum := 0
    for i := 1; i <= n; i++ {
        sum += i
    }

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

Screenshoot program

```
package main

import "fmt"

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

    sum := 0
    for i := 1; i <= n; i++ {
        sum += i
    }
}
```

PS C:\coding\didid\golang\Laprak 5.1> go run "c:\coding\didid\golang\Laprak 5.1\Latsol\tempCodeRunnerFile.go"

Masukkan bilangan bulat positif n: 1
Hasil penjumlahan dari 1 sampai 1 adalah: 1

PS C:\coding\didid\golang\Laprak 5.1> go run "c:\coding\didid\golang\Laprak 5.1\Latsol\tempCodeRunnerFile.go"

Masukkan bilangan bulat positif n: 5
Hasil penjumlahan dari 1 sampai 5 adalah: 15

PS C:\coding\didid\golang\Laprak 5.1>

Deskripsi program

Program ini digunakan untuk menghitung hasil penjumlahan bilangan bulat dari 1 sampai dengan n menggunakan bahasa pemrograman Go. Program ini dimulai dengan mendeklarasikan variabel `n` sebagai bilangan bulat untuk menampung input dari pengguna. Setelah itu, pengguna diminta memasukkan nilai `n` melalui perintah `fmt.Print("Masukkan bilangan bulat positif n: ")` dan nilainya disimpan menggunakan `fmt.Scan(&n)`. Selanjutnya, dibuat variabel `sum` dengan nilai awal 0 yang berfungsi sebagai penampung hasil penjumlahan. Proses penjumlahan dilakukan menggunakan struktur perulangan `for i := 1; i <= n; i++ { sum += i }`, di mana variabel `i` dimulai dari 1 dan terus bertambah satu sampai mencapai nilai `n`. Setiap kali perulangan berjalan, nilai `i` akan ditambahkan ke dalam `sum`, sehingga pada akhir perulangan variabel `sum` menyimpan total jumlah seluruh bilangan dari 1 hingga `n`. Setelah proses selesai, hasil akhir ditampilkan menggunakan perintah `fmt.Println("Hasil penjumlahan dari 1 sampai", n, "adalah:", sum)`.

Alur Algoritma didalam sebuah program :

1. Mulai program.
2. Deklarasikan variabel `n` dan `sum`.
3. Tampilkan pesan untuk meminta input nilai `n` dari pengguna.
4. Baca input nilai `n`.
5. Set nilai awal `sum = 0`.
6. Lakukan perulangan dari `i = 1` sampai `i <= n`.
7. Pada setiap langkah, tambahkan nilai `i` ke dalam `sum` (`sum = sum + i`).
8. Setelah perulangan selesai, tampilkan hasil nilai `sum`.
9. Program selesai.

Contoh jika pengguna memasukkan `n = 5`, maka program akan menghitung $1 + 2 + 3 + 4 + 5$, dan hasil yang ditampilkan adalah 15. Program ini memanfaatkan logika dasar pemrograman berupa input/output, variabel, dan perulangan untuk menyelesaikan

persoalan matematis sederhana secara sistematis tanpa menggunakan rumus langsung. Dengan demikian, program ini tidak hanya berguna untuk menghitung jumlah deret, tetapi juga melatih pemahaman konsep kontrol alur perulangan dalam bahasa Go.

2. Tugas 2

Source code

```
package main

import (
    "fmt"
    "math"
)

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

    for i := 1; i <= n; i++ {
        var r, t float64
        fmt.Printf("Masukkan jari-jari dan tinggi\nkerucut ke-%d: ", i)
        fmt.Scan(&r, &t)

        volume := (1.0 / 3.0) * math.Pi * r * r * t

        fmt.Println(volume)
    }
}
```

Screenshoot program

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

    for i := 1; i <= n; i++ {
        var r, t float64
        fmt.Printf("Masukkan jari-jari dan tinggi kerucut ke-%d: ", i)
        fmt.Scan(&r, &t)

        volume := (1.0 / 3.0) * math.Pi * r * r * t
    }
}
```

PS C:\coding didi\golang\Laprak 5.1> go run "c:\coding didi\golang\Laprak 5.1\Latsol\soal2.go"

Masukkan jumlah kerucut: 1

Masukkan jari-jari dan tinggi kerucut ke-1: 3 4

37.699111843077524

PS C:\coding didi\golang\Laprak 5.1>

Deskripsi program

Program ini dibuat untuk menghitung volume sejumlah kerucut berdasarkan data yang dimasukkan oleh pengguna. Program diawali dengan mengimpor dua paket yaitu `fmt` dan `math`. Paket `fmt` digunakan untuk menampilkan teks ke layar serta membaca input dari pengguna, sedangkan paket `math` digunakan untuk mengakses konstanta `math.Pi` yang merepresentasikan nilai π (pi). Pada bagian utama program (`func main()`), pertama-tama dideklarasikan variabel `n` bertipe integer untuk menyimpan jumlah kerucut yang ingin dihitung volumenya. Program kemudian meminta pengguna untuk memasukkan nilai `n` melalui perintah `fmt.Print("Masukkan jumlah kerucut: ")`, dan nilai tersebut dibaca menggunakan `fmt.Scan(&n)`.

Setelah jumlah kerucut diketahui, program masuk ke dalam perulangan `for i := 1; i <= n; i++ { ... }` yang berfungsi untuk menghitung volume masing-masing kerucut satu per satu sesuai jumlah yang dimasukkan. Di dalam perulangan ini, dideklarasikan dua variabel bertipe `float64`, yaitu `r` untuk menyimpan jari-jari alas dan `t` untuk menyimpan tinggi kerucut. Pengguna diminta memasukkan kedua nilai tersebut melalui perintah `fmt.Printf("Masukkan jari-jari dan tinggi kerucut ke-%d: ", i)` dan nilainya dibaca menggunakan `fmt.Scan(&r, &t)`.

Setelah data dimasukkan, volume kerucut dihitung menggunakan rumus $V = \frac{1}{3} \times \pi \times r^2 \times t$, yang dituliskan dalam kode sebagai:

```
volume := (1.0 / 3.0) * math.Pi * r * r * t
```

Rumus tersebut berasal dari rumus geometri dasar untuk menghitung volume kerucut, di mana π (pi) bernilai sekitar 3.14159, `r` adalah jari-jari alas kerucut, dan `t`

adalah tinggi kerucut. Hasil perhitungan kemudian ditampilkan ke layar dengan perintah `fmt.Println(volume)` untuk setiap kerucut yang dihitung.

Alur kerja mengenai algoritma sebuah program berikut:

1. Mulai program.
2. Deklarasikan variabel `n` untuk jumlah kerucut.
3. Minta pengguna memasukkan jumlah kerucut yang ingin dihitung.
4. Lakukan perulangan sebanyak `n` kali.
5. Pada setiap perulangan, minta pengguna memasukkan nilai jari-jari (`r`) dan tinggi (`t`) kerucut.
6. Hitung volume kerucut dengan rumus $(1/3) * \pi * r * r * t$.
7. Tampilkan hasil perhitungan ke layar.
8. Ulangi langkah 5 sampai semua kerucut selesai dihitung.
9. Program selesai.

Sebagai contoh, jika pengguna memasukkan `n = 3` dan data masing-masing kerucut adalah (1, 1), (2, 2), dan (3, 3), maka hasil yang ditampilkan berturut-turut adalah 1.0471975511965976, 8.377580409572781, dan 28.274333882308138. Program ini menunjukkan bagaimana konsep perulangan, input/output, dan penggunaan rumus matematika dapat digabungkan dalam satu alur logika untuk menghasilkan perhitungan yang efisien dan akurat.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var a, b int
    fmt.Print("Masukkan dua bilangan (basis dan pangkat): ")
    fmt.Scan(&a, &b)

    hasil := 1
    for i := 1; i <= b; i++ {
        hasil *= a
    }

    fmt.Println("Hasil:", hasil)
}
```

Screenshoot program

```
package main

import "fmt"

func main() {
    var a, b int
    fmt.Print("Masukkan dua bilangan (basis dan pangkat): ")
    fmt.Scan(&a, &b)

    hasil := 1
    for i := 1; i <= b; i++ {
        hasil *= a
    }

    fmt.Println("Hasil:", hasil)
}
```

Terminal Output:

```
PS C:\coding didi\golang\Laprak 5.1> go run "c:\coding didi\golang\Laprak 5.1\Latsol\tempCodeRunnerFile.go"
Masukkan dua bilangan (basis dan pangkat): 4 2
Hasil: 16
PS C:\coding didi\golang\Laprak 5.1> go run "c:\coding didi\golang\Laprak 5.1\Latsol\tempCodeRunnerFile.go"
Masukkan dua bilangan (basis dan pangkat): 2 10
Hasil: 1024
```

Deskripsi program

Program ini dibuat untuk menghitung hasil pemangkatan dari dua bilangan bulat positif menggunakan operator perkalian dan perulangan, tanpa memanfaatkan fungsi bawaan seperti `math.Pow`. Program diawali dengan mendeklarasikan dua variabel bertipe integer, yaitu `a` sebagai **basis (bilangan pokok)** dan `b` sebagai **pangkat**. Pengguna diminta untuk memasukkan kedua bilangan tersebut melalui perintah `fmt.Print("Masukkan dua bilangan (basis dan pangkat): ")`, dan nilainya dibaca menggunakan `fmt.Scan(&a, &b)`.

Setelah nilai `a` dan `b` diperoleh, program mendeklarasikan variabel `hasil` dengan nilai awal 1. Variabel ini berfungsi sebagai penampung hasil perkalian yang akan digunakan untuk menghitung nilai pangkat. Proses perhitungan dilakukan menggunakan struktur perulangan `for i := 1; i <= b; i++ { hasil *= a }`. Pada setiap iterasi, nilai `hasil` dikalikan dengan `a`. Perulangan berlangsung sebanyak `b` kali, sehingga setelah selesai, nilai `hasil` akan berisi hasil dari `a` pangkat `b` (ditulis secara matematis sebagai a^b)).

Sebagai contoh, jika pengguna memasukkan `a = 2` dan `b = 5`, maka perulangan akan melakukan proses seperti berikut:

- Iterasi 1: $hasil = 1 \times 2 = 2$
- Iterasi 2: $hasil = 2 \times 2 = 4$
- Iterasi 3: $hasil = 4 \times 2 = 8$
- Iterasi 4: $hasil = 8 \times 2 = 16$
- Iterasi 5: $hasil = 16 \times 2 = 32$

Setelah perulangan selesai, nilai akhir `hasil` ditampilkan menggunakan perintah `fmt.Println("Hasil:", hasil)`.

Secara algoritmik, langkah kerja program ini dapat dijelaskan sebagai berikut:

1. Mulai program.
2. Deklarasikan variabel `a` dan `b`.
3. Minta pengguna memasukkan dua bilangan: bilangan basis (`a`) dan bilangan pangkat (`b`).

4. Set nilai awal hasil = 1.
5. Lakukan perulangan sebanyak b kali.
6. Pada setiap langkah perulangan, kalikan nilai hasil dengan a.
7. Setelah perulangan selesai, tampilkan hasil akhir pemangkatan.
8. Program selesai.

4. Tugas 4

Source code

```
package main

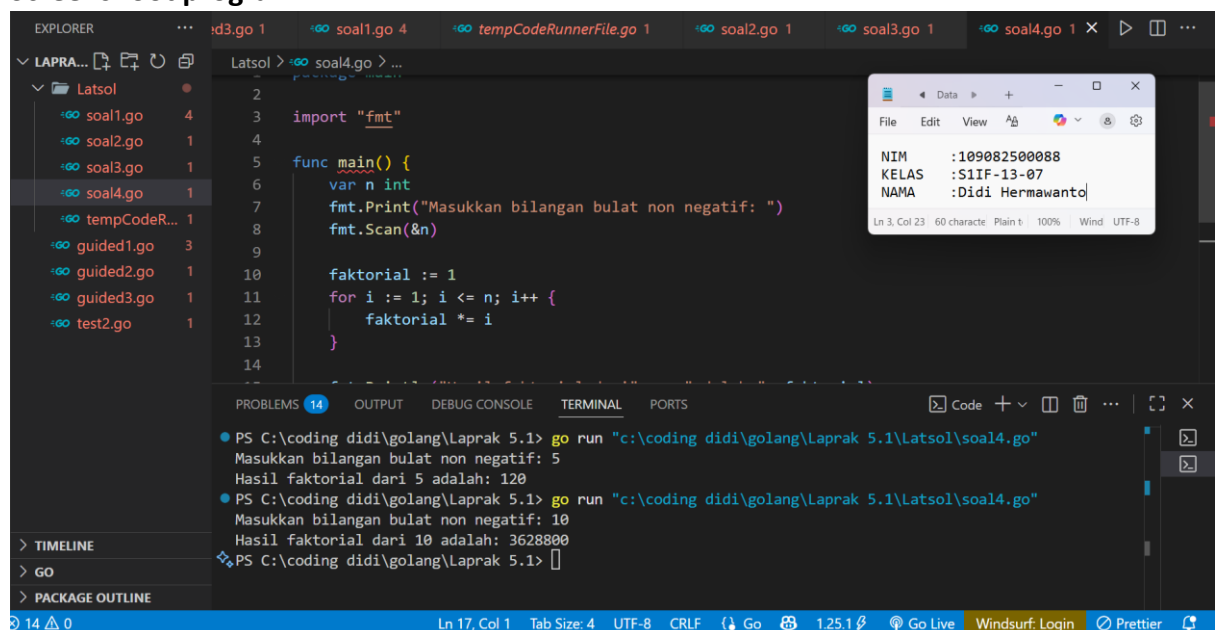
import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat non negatif: ")
    fmt.Scan(&n)

    faktorial := 1
    for i := 1; i <= n; i++ {
        faktorial *= i
    }

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

Screenshoot program



Deskripsi Program:

Program ini dibuat untuk menghitung nilai faktorial dari sebuah bilangan bulat non-negatif yang dimasukkan oleh pengguna. Faktorial sendiri adalah hasil perkalian semua bilangan bulat positif dari 1 hingga n (ditulis $n!$). Program ini menggunakan perulangan `for` untuk melakukan perkalian berulang dari 1 sampai n dan menampilkan hasil akhirnya ke layar.

Penjelasan Kode Program:

1. `package main` → Menandakan bahwa program ini merupakan program utama di bahasa Go.
 2. `import "fmt"` → Mengimpor paket `fmt` yang digunakan untuk input dan output seperti `Print` dan `Scan`.
 3. `var n int` → Mendeklarasikan variabel `n` dengan tipe data integer untuk menyimpan bilangan yang akan dihitung faktorialnya.
 4. `fmt.Print("Masukkan bilangan bulat non negatif: ")` → Menampilkan pesan agar pengguna memasukkan angka.
 5. `fmt.Scan(&n)` → Membaca input dari pengguna dan menyimpannya ke dalam variabel `n`.
 6. `faktorial := 1` → Inisialisasi variabel faktorial dengan nilai 1 sebagai nilai awal perkalian.
 7. `for i := 1; i <= n; i++` → Melakukan perulangan dari angka 1 hingga n .
 8. `faktorial *= i` → Setiap iterasi, nilai faktorial dikalikan dengan nilai `i` untuk menghitung hasil faktorial secara bertahap.
 9. `fmt.Println("Hasil faktorial dari", n, "adalah:", faktorial)` → Menampilkan hasil akhir perhitungan faktorial ke layar.
-

Algoritma Program:

1. Mulai program.
 2. Deklarasikan variabel `n` dan faktorial.
 3. Tampilkan pesan untuk meminta input bilangan dari pengguna.
 4. Baca input dan simpan ke dalam variabel `n`.
 5. Set nilai awal faktorial = 1.
 6. Lakukan perulangan dari `i = 1` sampai `i = n`.
 - a. Kalikan nilai faktorial dengan `i`.
 - b. Simpan hasilnya kembali ke faktorial.
 7. Setelah perulangan selesai, tampilkan hasil faktorial.
 8. Selesai.
-

Contoh Eksekusi Program:

Masukkan bilangan bulat non negatif: 5

Hasil faktorial dari 5 adalah: 120

Penjelasan Contoh:

Perhitungan dilakukan dengan cara:

$$1 \times 2 \times 3 \times 4 \times 5 = 120$$

Maka hasil faktorial dari 5 adalah **120**.