**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 05**

**FOOR LOOP**

**Sebuah gambar berisi logo, teks, simbol, Grafis

Konten yang dihasilkan AI mungkin salah.**

**Disusun oleh:**

**Muhammad Fu'ad Nurhakim**

**109082500072**

**S1IF-13-02**

**Asisten Praktikum**

Adithana Dharma Putra

Alfin Ilham Berlianto

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

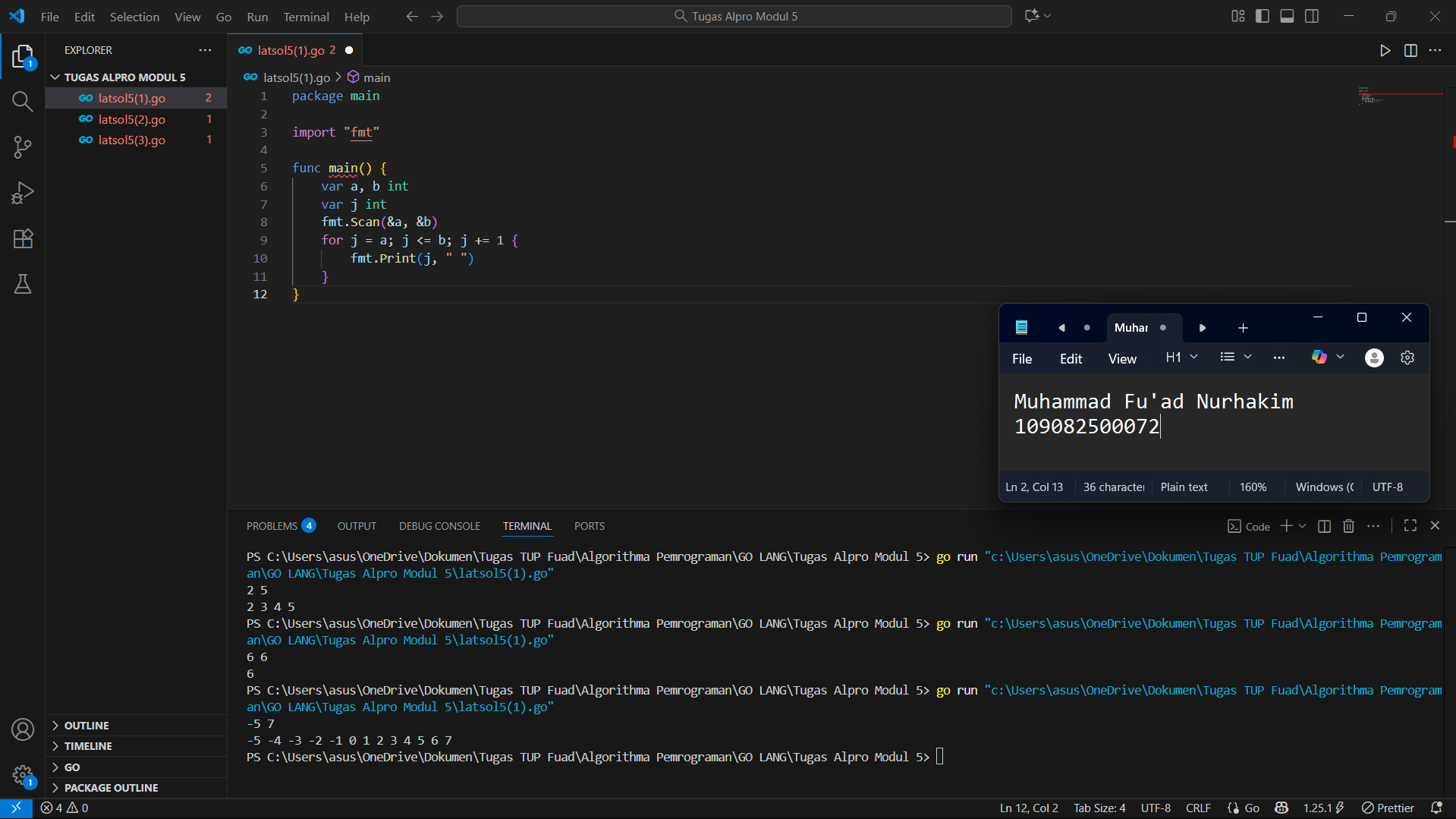
**LATIHAN KELAS – GUIDED**

1. **Guided 1**

**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**

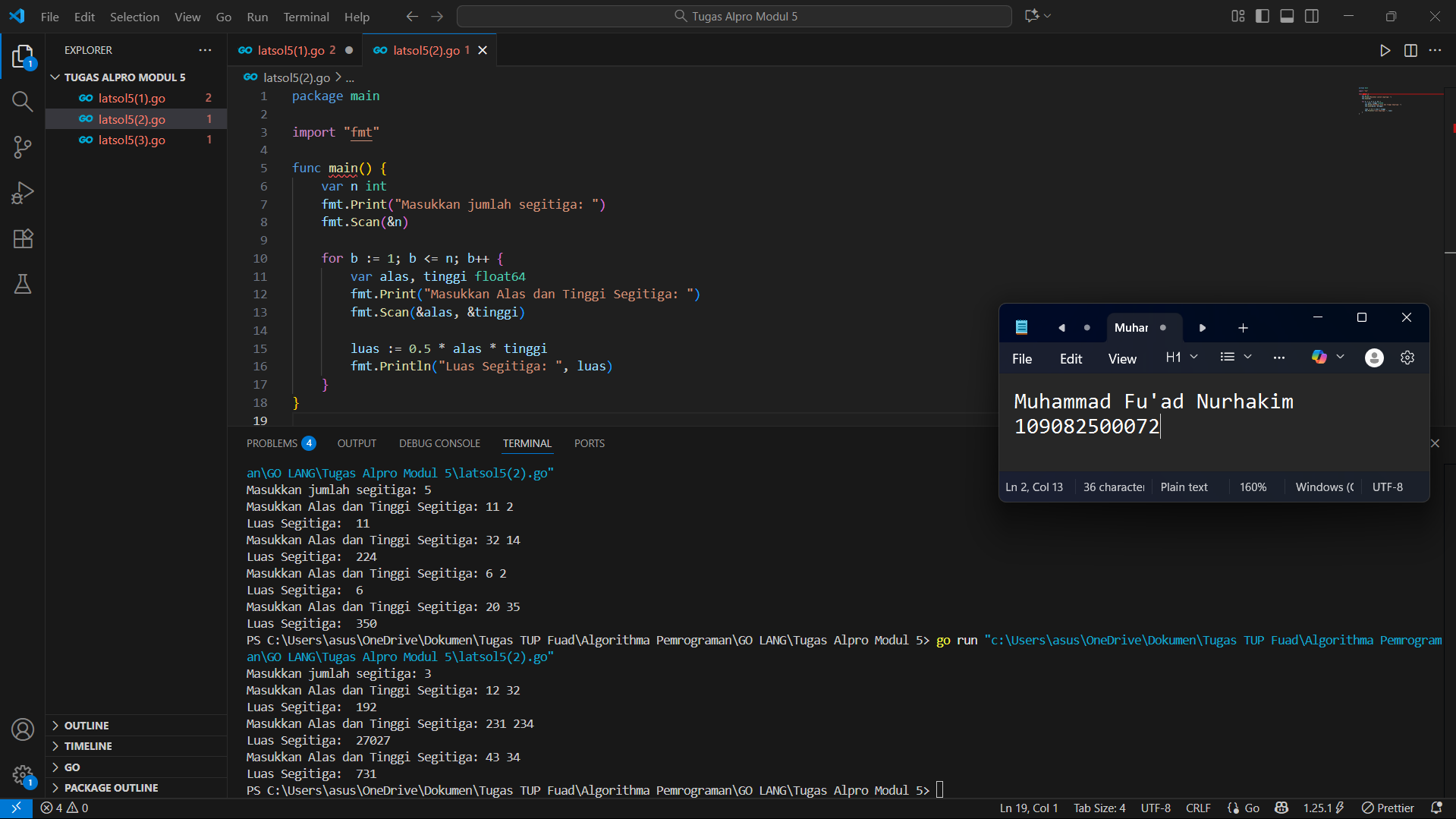
Program ini bertujuan untuk mencetak barisan bilangan bulat secara berurutan yang dimulai dari suatu nilai awal (a) hingga mencapai nilai akhir (b), di mana diketahui bahwa (a <= b). Program akan membaca dua bilangan bulat (a) dan (b) sebagai input, kemudian menggunakan struktur perulangan untuk menampilkan setiap bilangan bulat dalam rentang [a, b] tersebut ke layar, dengan setiap bilangan dipisahkan oleh spasi

1. **Guided 2**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {  var n int  fmt.Print("Masukkan jumlah segitiga: ")  fmt.Scan(&n)  for b := 1; b <= n; b++ {  var alas, tinggi float64  fmt.Print("Masukkan Alas dan Tinggi Segitiga: ")  fmt.Scan(&alas, &tinggi)  luas := 0.5 \* alas \* tinggi  fmt.Println("Luas Segitiga: ", luas)  }  } |

**Screenshoot program**

****

**Deskripsi program**

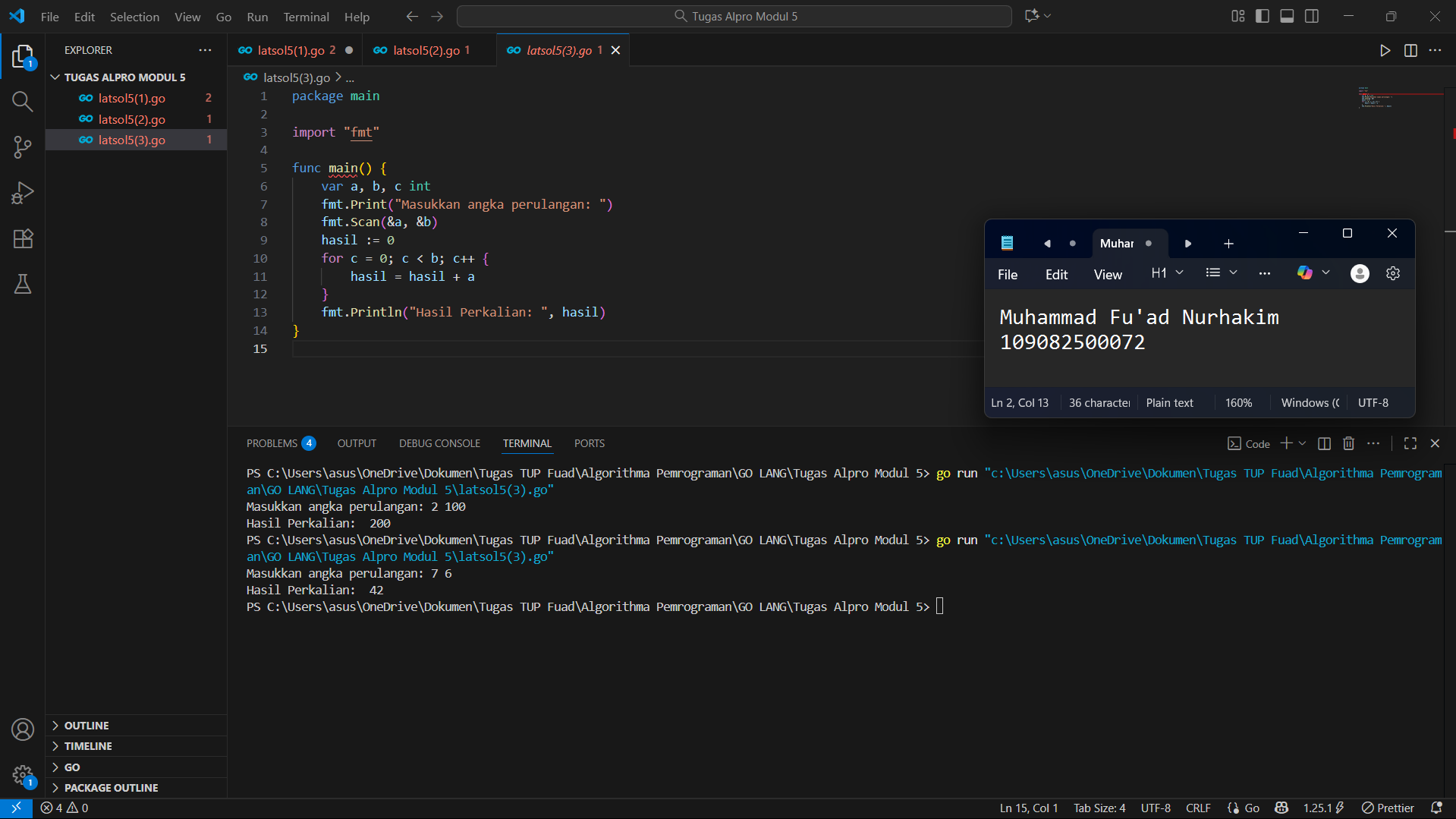
Program ini bertujuan untuk menghitung dan menampilkan luas dari sejumlah buah segitiga sesuai dengan input yang diberikan oleh pengguna. Program akan menerima angka yang diinputkan pengguna (bilangan bulat) pada baris pertama, yang menunjukkan jumlah segitiga yang akan diproses. Selanjutnya, program akan membaca angka tersebut baris input tambahan, di mana setiap baris berisi sisi alas dan tinggi dari masing-masing segitiga. Dengan menggunakan rumus luas segitiga (luas = 0.5 \* alas \* tinggi), program akan mencetak angka baris keluaran, yang masing-masing merupakan luas dari setiap segitiga tersebut.

1. **Guided 3**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {  var a, b, c int  fmt.Print("Masukkan angka perulangan: ")  fmt.Scan(&a, &b)  hasil := 0  for c = 0; c < b; c++ {  hasil = hasil + a  }  fmt.Println("Hasil Perkalian: ", hasil)  } |

**Screenshoot program**

****

**Deskripsi program**

ini berfungsi untuk menghitung hasil perkalian dua bilangan bulat positif tanpa menggunakan operator perkalian (\*), melainkan menggunakan metode penjumlahan berulang (repeated addition). Setelah membaca input dua bilangan dari pengguna (fmt.Scan(&a, &b)), program menginisialisasi variabel hasil menjadi nol. Selanjutnya, program menjalankan loop for yang akan mengulangi proses penjumlahan sebanyak angka yang pengguna input di (b) , di mana pada setiap iterasi, nilai (a) ditambahkan ke variabel hasil. Setelah loop selesai, nilai akhir yang merupakan hasil perkalian kedua bilangan tersebut akan dicetak menjadi hasil akhir.

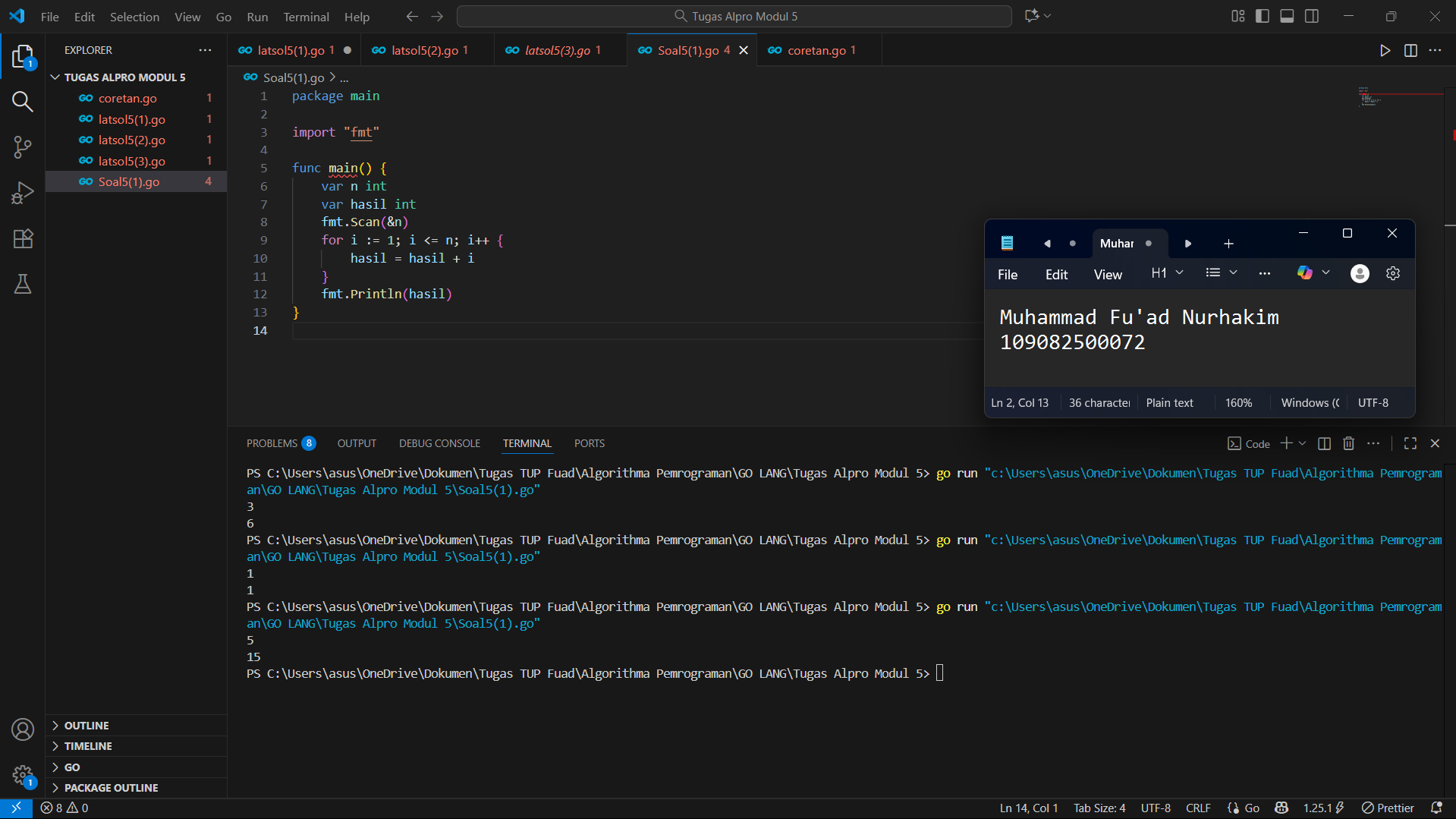
**TUGAS**

1. **Tugas 1**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {  var totalBelanja, diskon int  fmt.Print("Masukkan total Belanja: ")  fmt.Scanln(&totalBelanja)  fmt.Print("Masukkan diskon: ")  fmt.Scanln(&diskon)  Harga := totalBelanja - (totalBelanja \* diskon / 100)  fmt.Printf("Total belanja akhir : %d\n", Harga)  } |

**Screenshoot program**

****

**Deskripsi program**

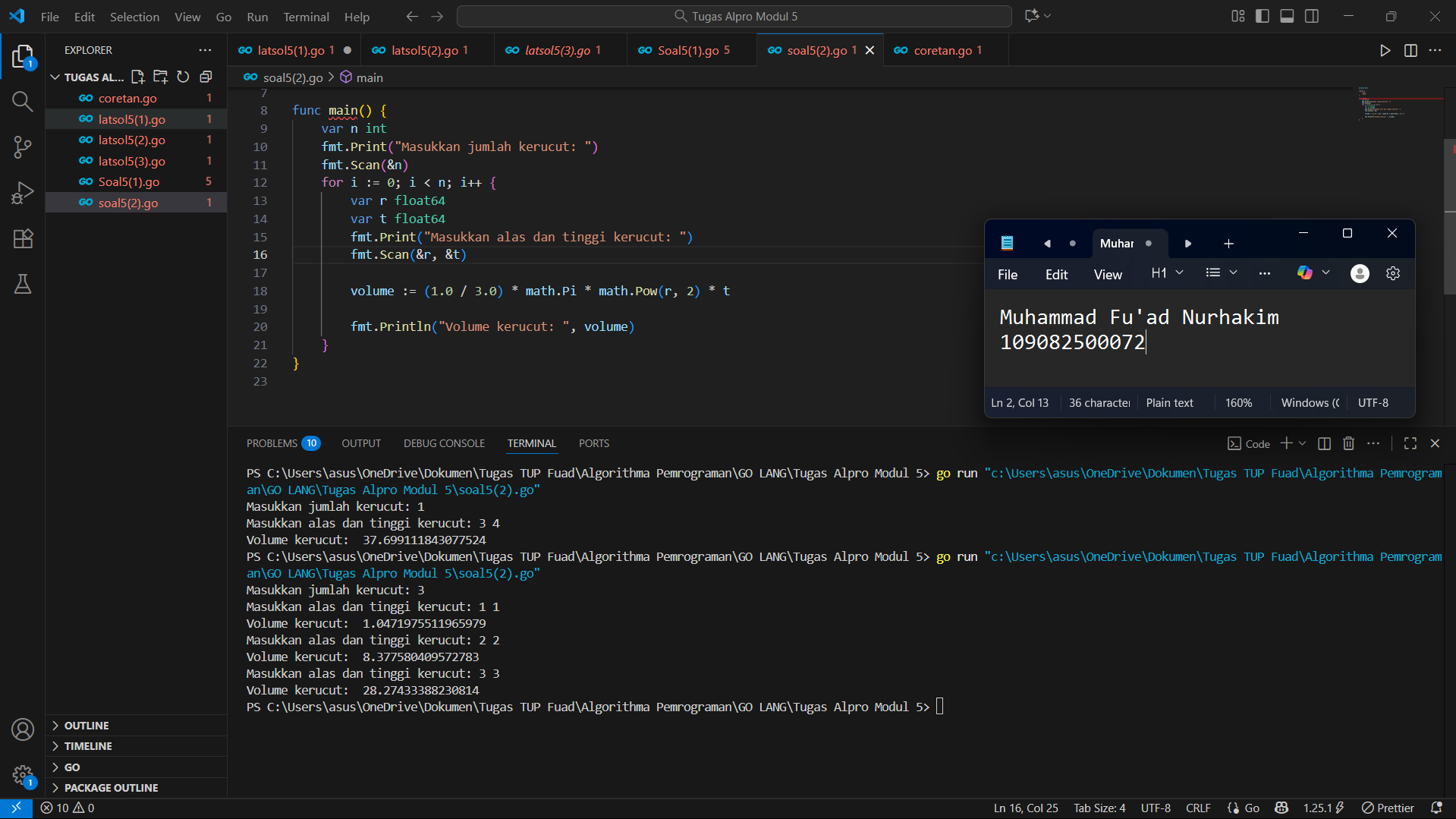
Tujuan program ini untuk menghitung jumlah total dari sekumpulan bilangan bulat positif berurutan, dimulai dari angka 1 hingga batas maksimum yang dimasukkan oleh pengguna (n). Program ini mencapai tujuan tersebut dengan membaca satu bilangan bulat positif (n) sebagai input, kemudian menggunakan struktur perulangan (for loop) untuk secara iteratif menambahkan setiap bilangan bulat (i) dari 1 hingga (n) ke dalam variabel hasil. Hasil akhirnya adalah nilai total penjumlahan deret aritmatika tersebut, seperti menghitung 1+2+3+...+n.

1. **Tugas 2**

**Source code**

|  |
| --- |
| package main  import (  "fmt"  "math"  )  func main() {  var n int  fmt.Print("Masukkan jumlah kerucut: ")  fmt.Scan(&n)  for i := 0; i < n; i++ {  var r float64  var t float64  fmt.Print("Masukkan alas dan tinggi kerucut: ")  fmt.Scan(&r, &t)  volume :=(1.0 / 3.0)\*math.Pi\*math.Pow(r, 2)\*t  fmt.Println("Volume kerucut: ", volume)  }  } |

**Screenshoot program**

****

**Deskripsi program**

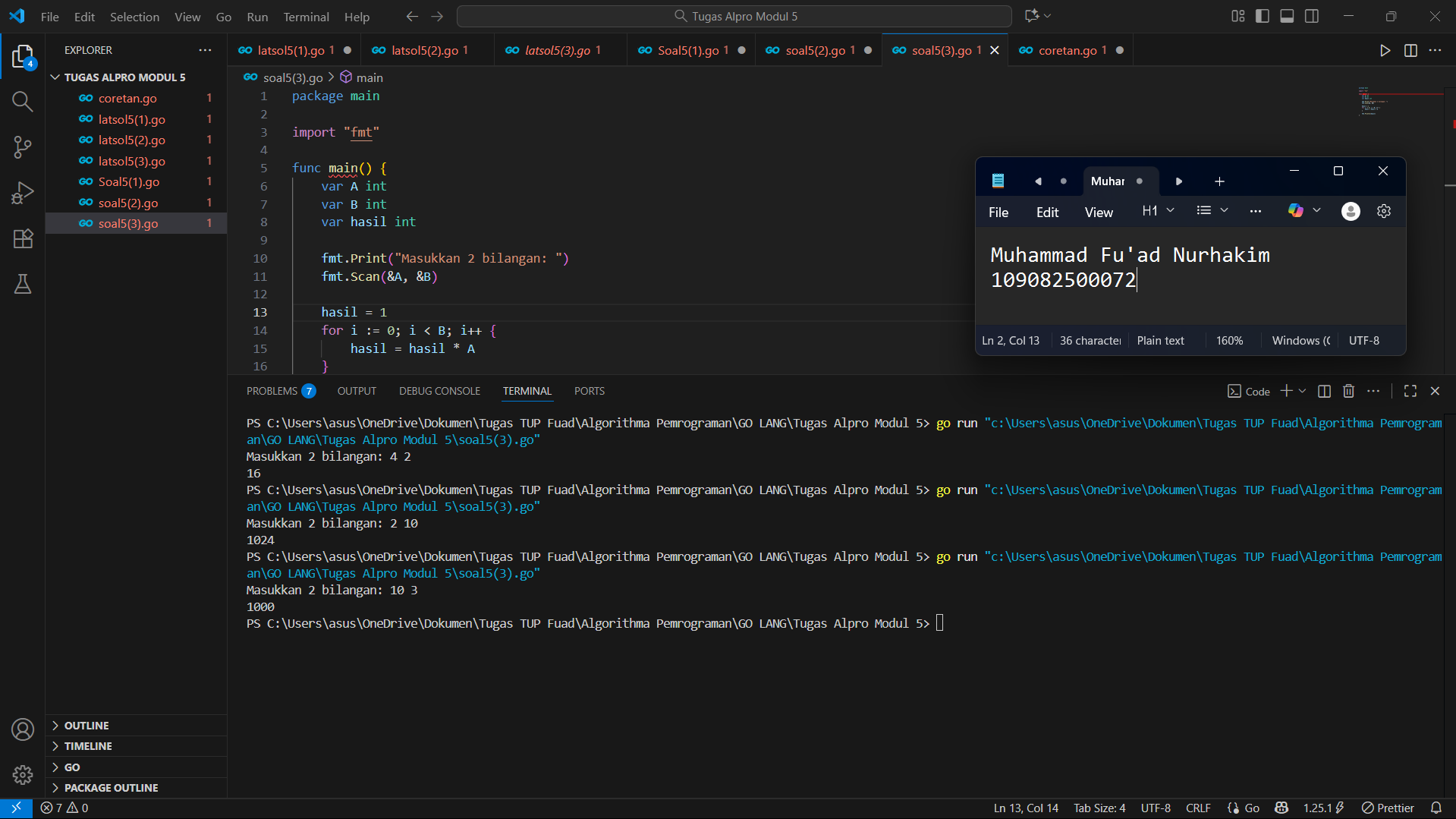
Program ini bertujuan untuk menghitung dan menampilkan volume dari sejumlah (n) buah kerucut yang ditentukan oleh pengguna. Program akan membaca bilangan bulat (n) pada baris pertama, yang menunjukkan jumlah kerucut yang akan diproses. Selanjutnya, program akan membaca (n) baris data tambahan, di mana setiap baris berisi panjang jari-jari alas (r) dan tinggi (t) dari kerucut. Dengan menggunakan rumus volume kerucut {Volume} = 1/3 x pi x r2 x t), program akan melakukan perhitungan untuk setiap kerucut dan mencetak (n) baris keluaran, yang masing-masing merupakan volume dari setiap kerucut tersebut dengan hasil yang akurat.

1. **Tugas 3**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {  var A int  var B int  var hasil int  fmt.Print("Masukkan 2 bilangan: ")  fmt.Scan(&A, &B)  hasil = 1  for i := 0; i < B; i++ {  hasil = hasil \* A  }  fmt.Println(hasil)  } |

**Screenshoot program**

****

**Deskripsi program**

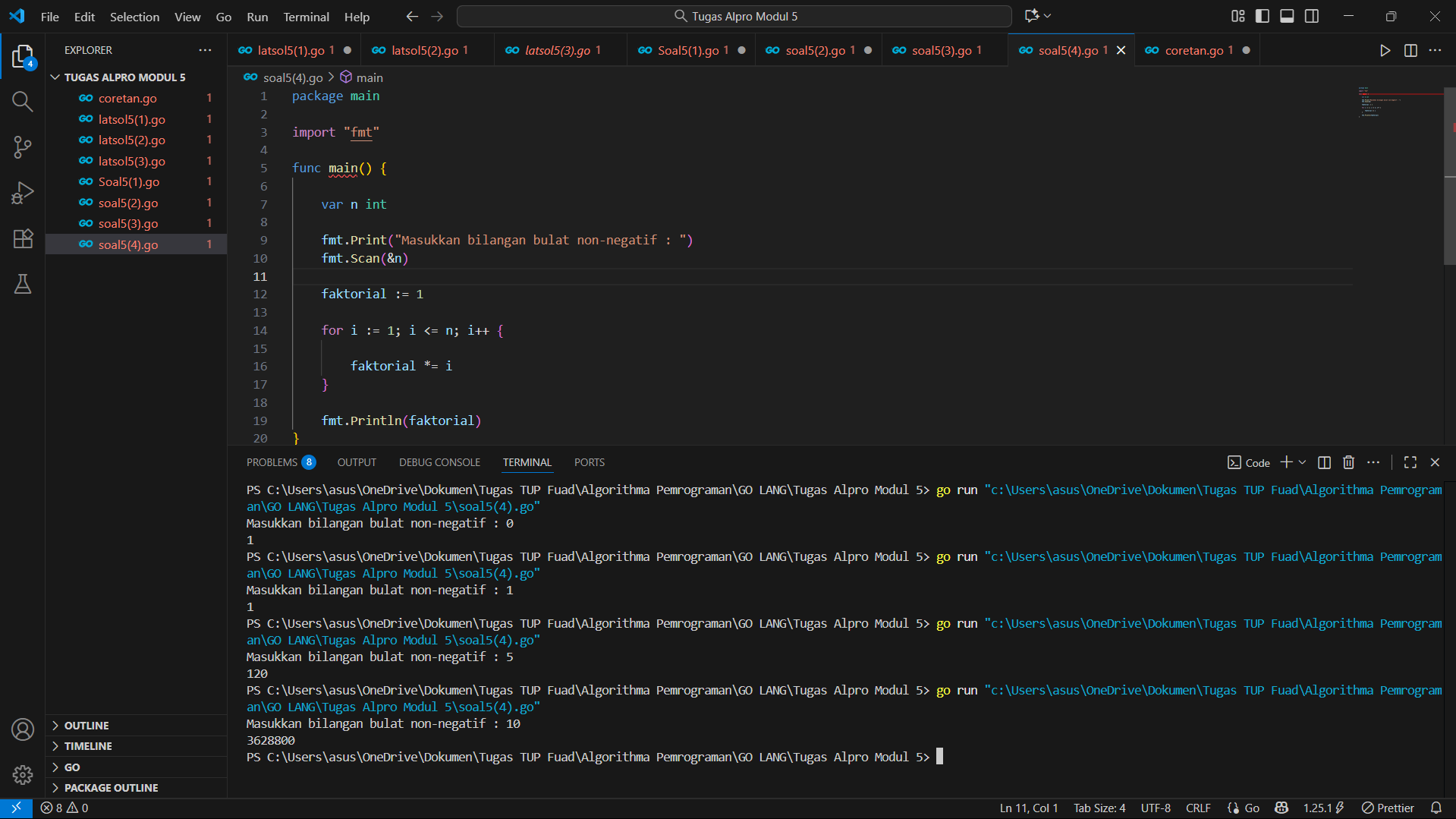
Program ini bertujuan untuk menghitung hasil pemangkatan (AB) dari dua bilangan bulat positif yang dimasukkan pengguna, di mana (A) adalah bilangan basis dan (B) adalah bilangan eksponen. Setelah membaca nilai (A) dan (B) sebagai input, program menginisialisasi variabel hasil menjadi 1. Kemudian, program menggunakan struktur kontrol perulangan (for loop) yang akan mengiterasi sebanyak (B) kali. Dalam setiap iterasi, variabel hasil dikalikan dengan bilangan basis (A), sehingga secara kumulatif mensimulasikan operasi A \* A \* … A sebanyak (B) kali. Hasil akhir pemangkatan tersebut kemudian ditampilkan sebagai keluaran.

1. **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(faktorial)  } |

**Screenshoot program**

****

**Deskripsi program**

Program ini bertujuan untuk menghitung dan menampilkan nilai faktorial (n!) dari suatu bilangan bulat non-negatif (n) yang dimasukkan pengguna. Faktorial (n!) didefinisikan sebagai hasil perkalian semua bilangan bulat positif dari (1) hingga (n), di mana secara khusus (0! = 1). Program diawali dengan membaca nilai (n) dari pengguna dan menginisialisasi variabel faktorial dengan nilai 1. Inisialisasi ini penting karena dua alasan: pertama, untuk memenuhi aturan(0! = 1) dan (! = 1); kedua, sebagai nilai awal perkalian berulang. Program kemudian menggunakan struktur kontrol perulangan (for) yang dimulai dari (i=1) hingga (i) mencapai nilai (n). Dalam setiap iterasi perulangan, nilai faktorial yang sementara akan dikalikan dengan nilai (i) saat itu (faktorial = faktorial \* i). Proses ini memastikan perkalian berantai (1 x 2 x 3 x …. x n) dilakukan secara bertahap. Setelah perulangan selesai, variabel faktorial akan menyimpan hasil akhir (n!), yang kemudian dicetak sebagai keluaran.