**LAPORAN PRAKTIKUM**

**Algoritma Pemrograman**

**MODUL 5 & 6**

**FOR-LOOP**



**Disusun oleh:**

**Akhmad Noval Annur**

**109082500100**

**S1IF-13-04**

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

**LATIHAN KELAS – GUIDED**

1. **Guided 1**

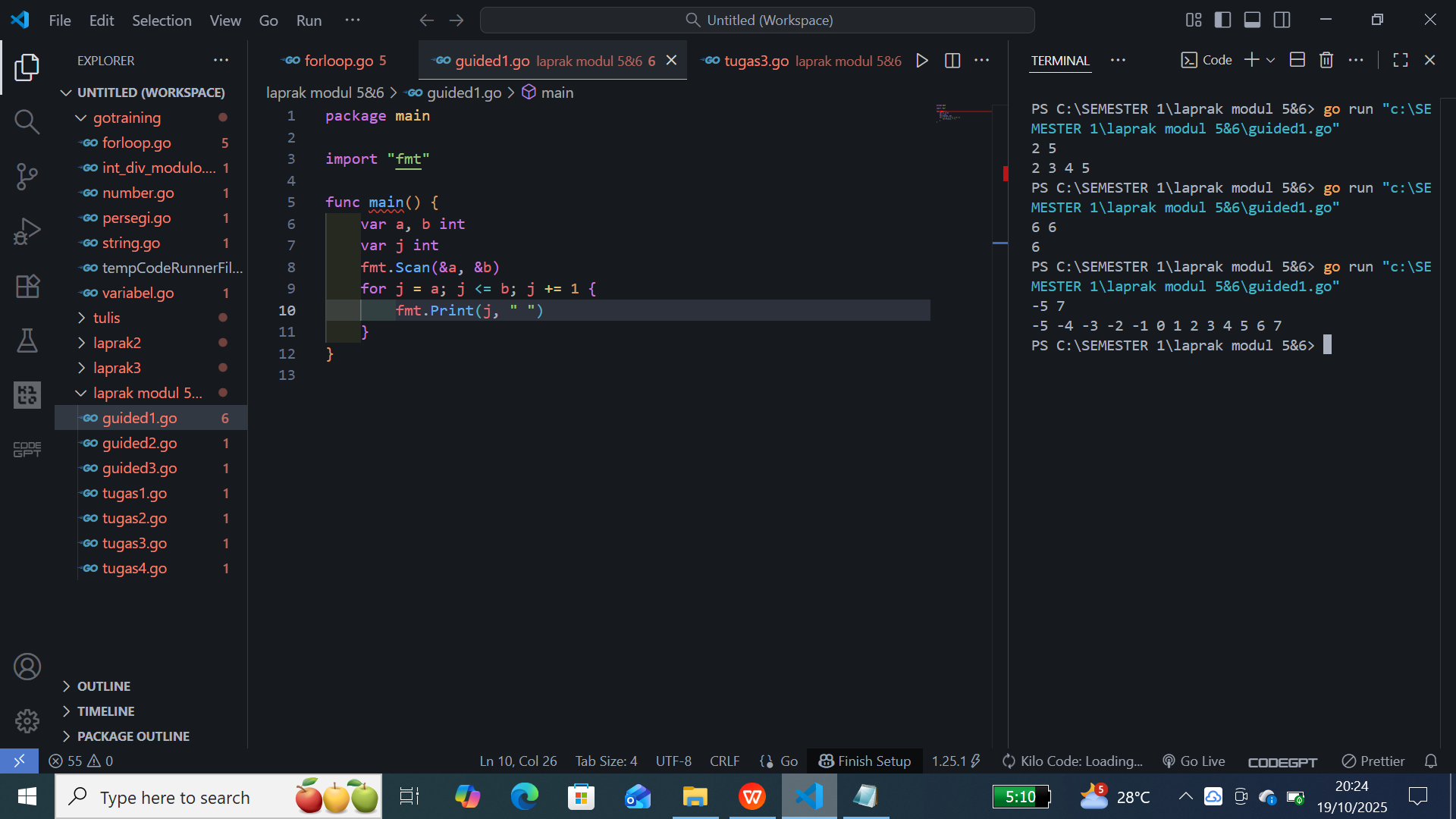
**Source Code**

|  |
| --- |
| 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**

**//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)**

**CONTOH TANGKAPAN LAYAR:**

****

**Deskripsi program**

Program ini bertujuan untuk menampilkan deret bilangan bulat dari nilai awal hingga nilai akhir yang dimasukkan oleh pengguna. Ketika dijalankan, program terlebih dahulu meminta dua buah input berupa bilangan bulat, yaitu nilai awal (a) dan nilai akhir (b). Setelah menerima input tersebut, program akan menjalankan struktur perulangan for yang dimulai dari nilai awal dan akan terus bertambah satu per satu hingga mencapai nilai akhir.

Setiap nilai yang dihasilkan pada proses perulangan ditampilkan ke layar dengan menggunakan fungsi fmt.Print, dan dipisahkan oleh spasi agar mudah dibaca oleh pengguna. Dengan demikian, program dapat menampilkan urutan bilangan dari nilai awal hingga nilai akhir secara berurutan dalam satu baris.

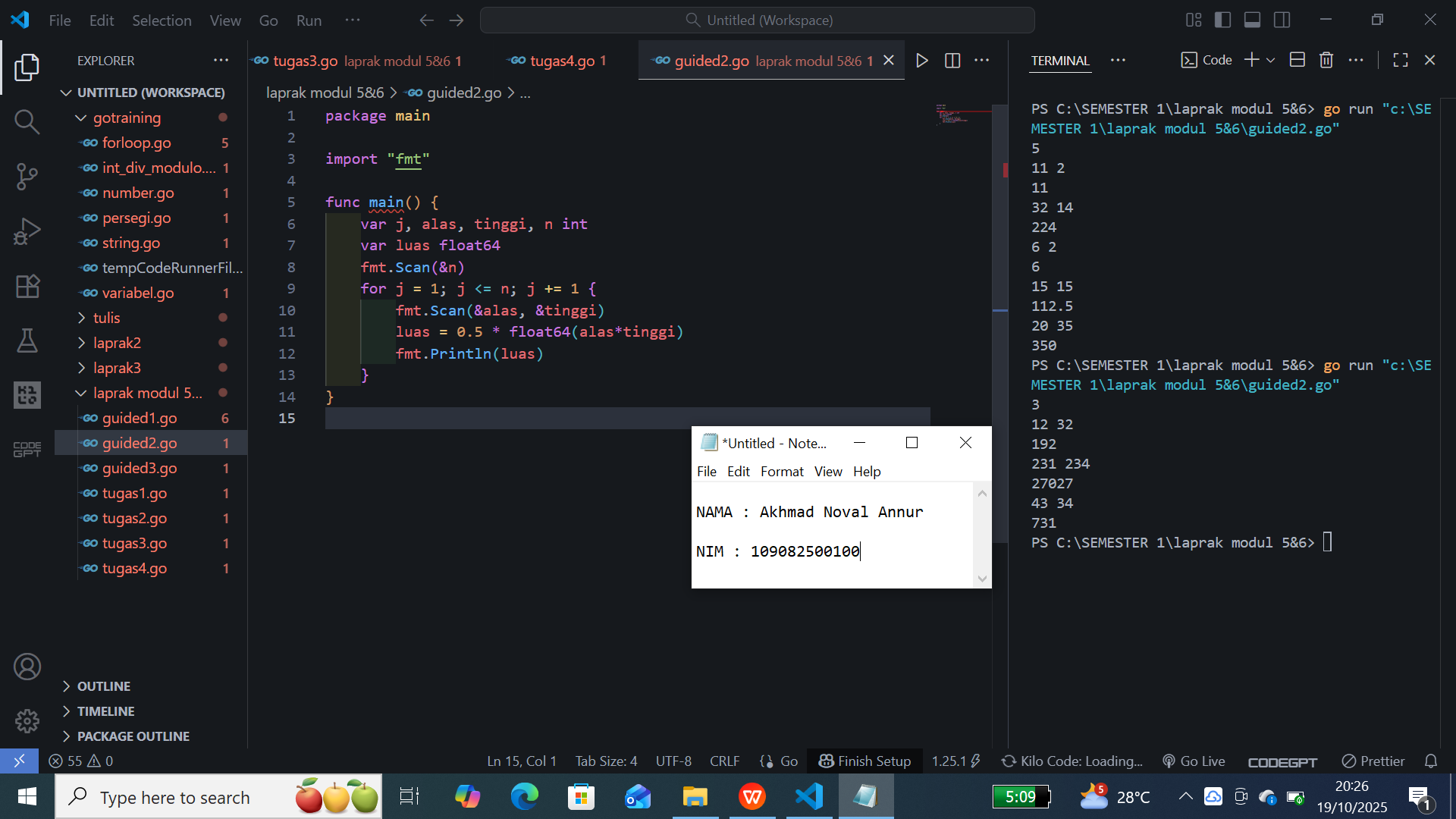
Program ini menerapkan konsep dasar counter-controlled loop atau perulangan yang dikendalikan oleh variabel penghitung. Pengguna dapat memahami bagaimana proses inisialisasi variabel, evaluasi kondisi batas, serta penggunaan inkremen bekerja secara bersamaan dalam struktur perulangan. Selain itu, program ini juga memperlihatkan penerapan konsep iterasi sederhana yang sering digunakan dalam pemrosesan data berurutan, seperti pencetakan bilangan, penghitungan deret, atau pembangkitan nilai secara bertahap.

1. **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 berfungsi untuk menghitung luas segitiga dengan menggunakan nilai alas dan tinggi yang dimasukkan oleh pengguna. Pada awal eksekusi, program meminta pengguna menentukan berapa banyak data segitiga yang akan dihitung, yaitu nilai n. Setelah itu, program menjalankan perulangan for sebanyak n kali untuk memproses setiap data segitiga.

Di setiap iterasi, pengguna diminta memasukkan dua nilai, yaitu alas dan tinggi. Nilai tersebut digunakan dalam perhitungan luas segitiga dengan rumus L = 1/2 × alas × tinggi. Hasil perhitungan disimpan dalam variabel bertipe float64 agar mendukung angka desimal dan ditampilkan langsung ke layar setelah proses input selesai.

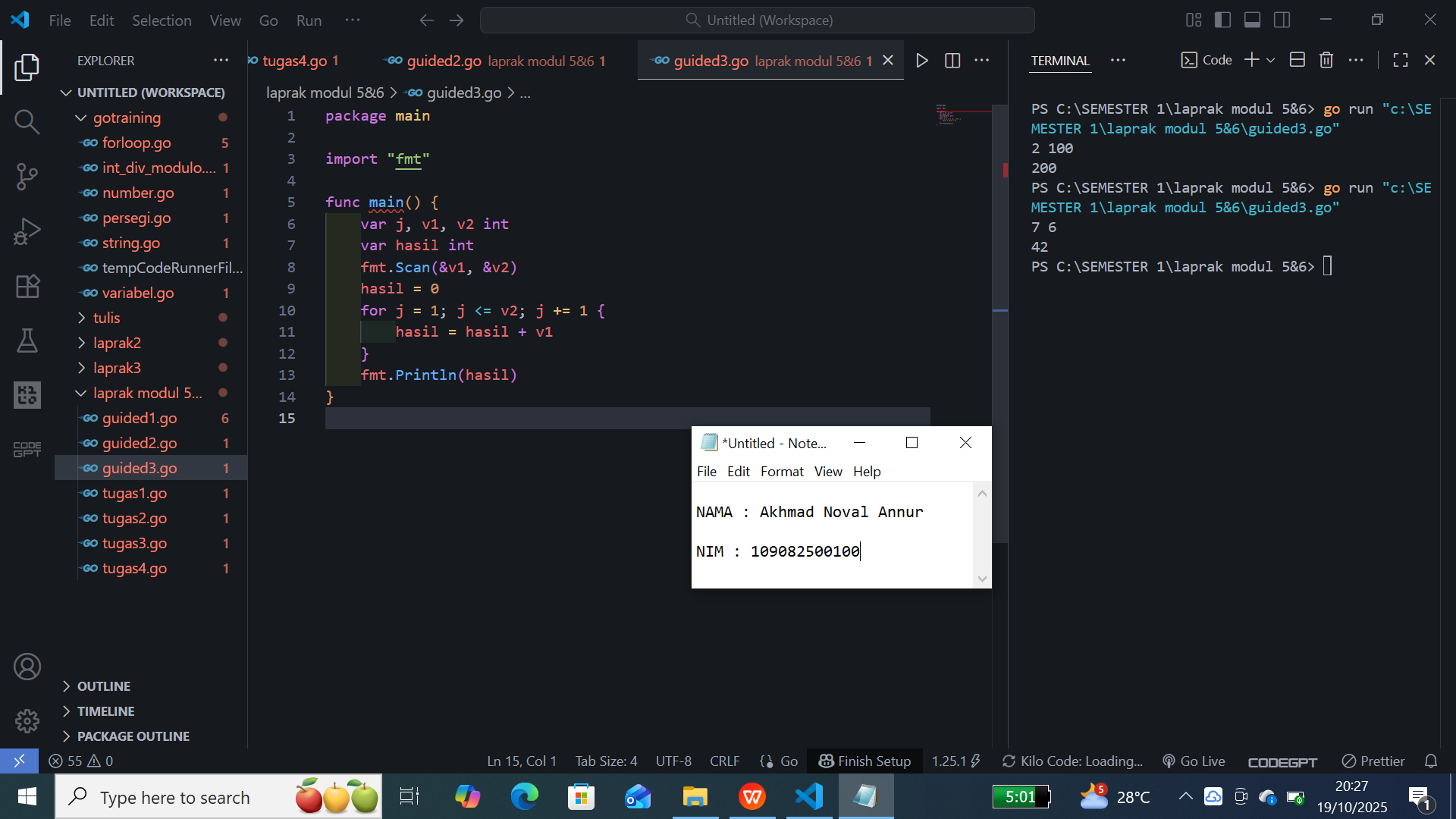
Program ini menggambarkan penerapan struktur perulangan untuk melakukan proses yang berulang secara sistematis dan efisien. Selain itu, mahasiswa dapat memahami cara penerapan tipe data float64 dalam operasi aritmetika serta bagaimana rumus matematis dapat diubah menjadi bentuk algoritmik yang terstruktur. Program ini juga memperlihatkan bagaimana setiap iterasi dapat bekerja secara independen untuk menghitung beberapa data dalam satu kali eksekusi program.

1. **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 digunakan untuk menghitung hasil perkalian dua bilangan bulat tanpa menggunakan operator perkalian (\*). Pengguna diminta memasukkan dua buah bilangan bulat, yaitu bilangan pertama (v1) dan bilangan kedua (v2). Program kemudian menjumlahkan nilai v1 sebanyak v2 kali menggunakan perulangan for.

Dalam setiap iterasi, nilai v1 akan ditambahkan ke dalam variabel hasil hingga jumlah perulangan mencapai nilai v2. Dengan cara ini, hasil akhir merupakan hasil perkalian antara dua bilangan tersebut. Konsep yang diterapkan adalah repetitive addition atau penjumlahan berulang, yang menjadi dasar dari operasi perkalian.

Program ini membantu mahasiswa memahami hubungan antara logika iteratif dan operasi aritmetika dasar. Selain itu, program ini memperlihatkan bagaimana penggunaan variabel penampung, inisialisasi nilai, dan proses akumulasi bekerja bersama dalam menyelesaikan masalah matematika sederhana. Program ini sangat efektif untuk memahami prinsip dasar algoritma perhitungan tanpa menggunakan operator langsung.

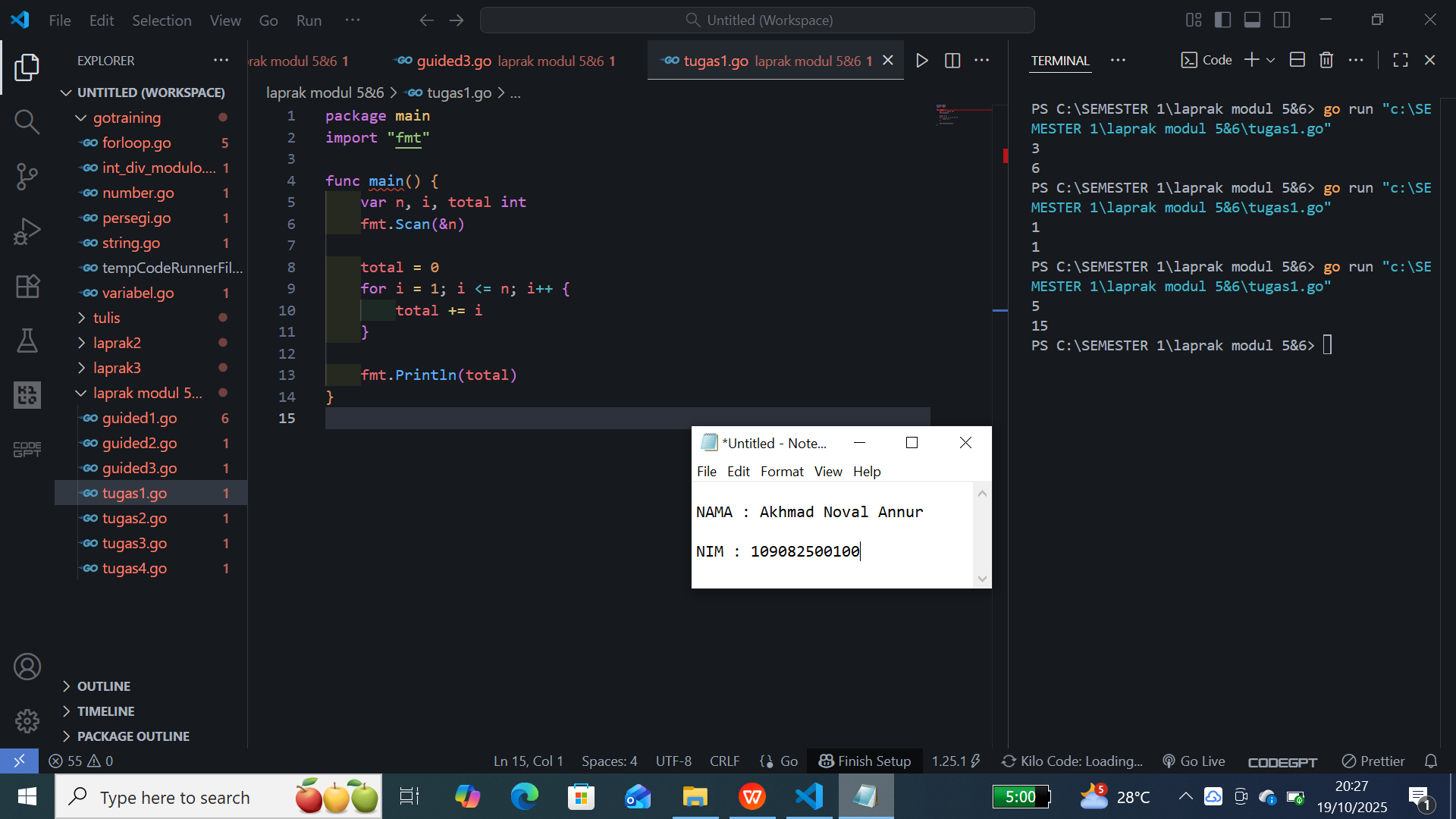
**TUGAS**

1. **Tugas 1**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {  var n, i, total int  fmt.Scan(&n)  total = 0  for i = 1; i <= n; i++ {  total += i  }  fmt.Println(total)  } |

**Screenshoot program**

****

**Deskripsi program**

Program ini dirancang untuk menghitung jumlah total dari deret bilangan bulat mulai dari angka 1 hingga angka n. Pada saat dijalankan, program meminta pengguna untuk memasukkan sebuah bilangan bulat positif yang menjadi batas akhir perhitungan. Setelah input diterima, program menginisialisasi variabel total dengan nilai awal 0.

Selanjutnya, program menjalankan perulangan for dengan variabel penghitung i yang dimulai dari 1 hingga mencapai nilai n. Dalam setiap iterasi, nilai i akan ditambahkan ke variabel total, sehingga ketika perulangan selesai, total berisi hasil penjumlahan dari semua bilangan 1 sampai n. Hasil akhir kemudian ditampilkan ke layar menggunakan fungsi fmt.Println.

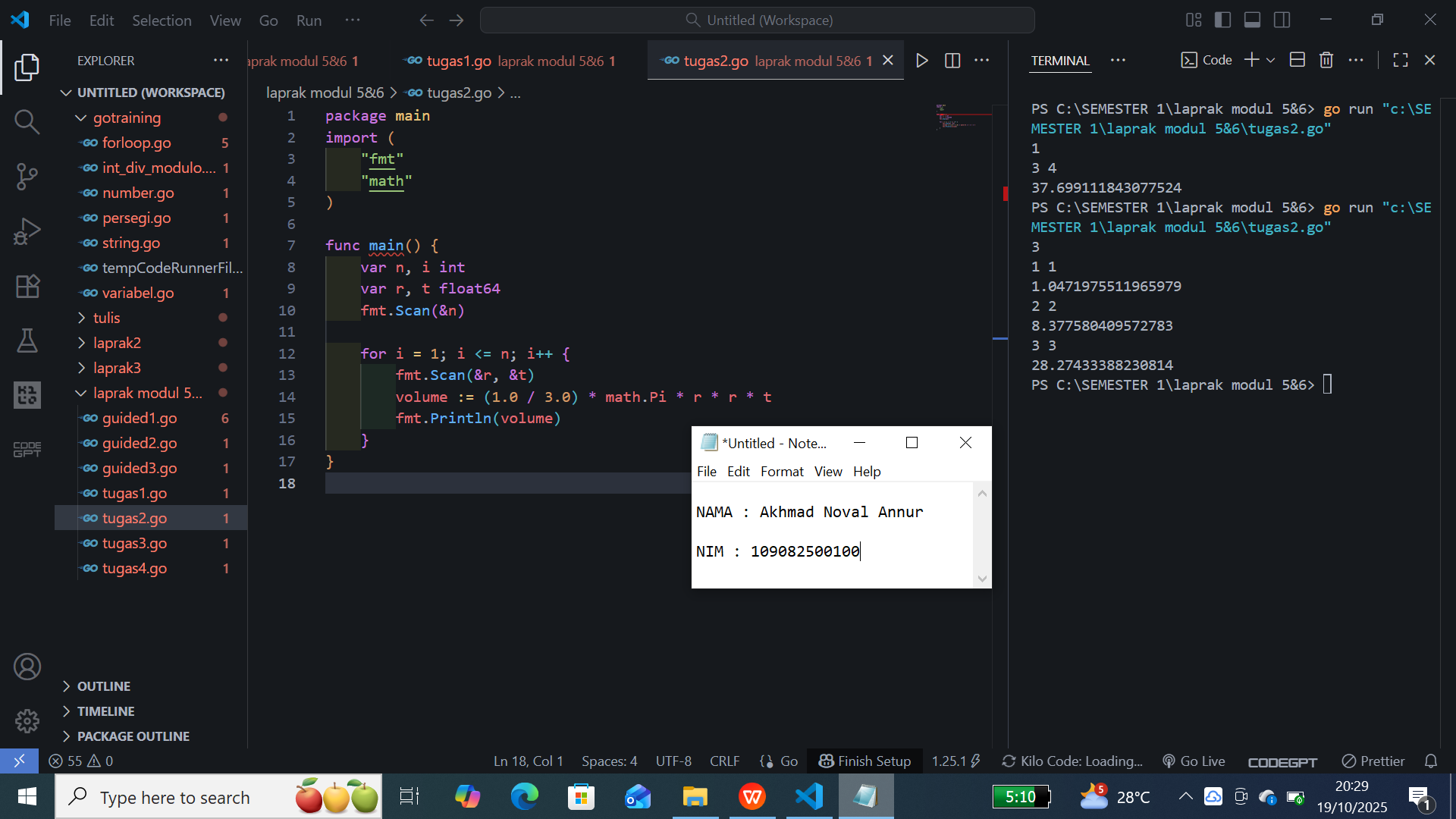
Program ini menunjukkan implementasi logika counting loop yang sederhana namun penting dalam pembelajaran algoritma dasar. Melalui program ini, pengguna dapat memahami bagaimana operasi akumulasi bekerja serta bagaimana alur input, proses, dan output diintegrasikan dalam pemrograman Go.

1. **Tugas 2**

**Source code**

|  |
| --- |
| package main  import (  "fmt"  "math"  )  func main() {  var n, i int  var r, t float64  fmt.Scan(&n)  for i = 1; i <= n; i++ {  fmt.Scan(&r, &t)  volume := (1.0 / 3.0) \* math.Pi \* r \* r \* t  fmt.Println(volume)  }  } |

**Screenshoot program**

****

**Deskripsi program**

Program ini digunakan untuk menghitung volume kerucut berdasarkan nilai jari-jari alas dan tinggi yang dimasukkan oleh pengguna. Pada awal program, pengguna diminta menentukan jumlah data yang akan dihitung (n). Untuk setiap data, program meminta dua buah nilai, yaitu jari-jari (r) dan tinggi (t).

Program kemudian menghitung volume kerucut menggunakan rumus V = (1/3) × π × r² × t, dengan π diambil dari paket math bawaan bahasa Go. Nilai hasil perhitungan disimpan dalam variabel bertipe float64 agar hasil yang ditampilkan memiliki tingkat presisi yang baik. Setelah setiap perhitungan selesai, hasil volume kerucut ditampilkan langsung ke layar.

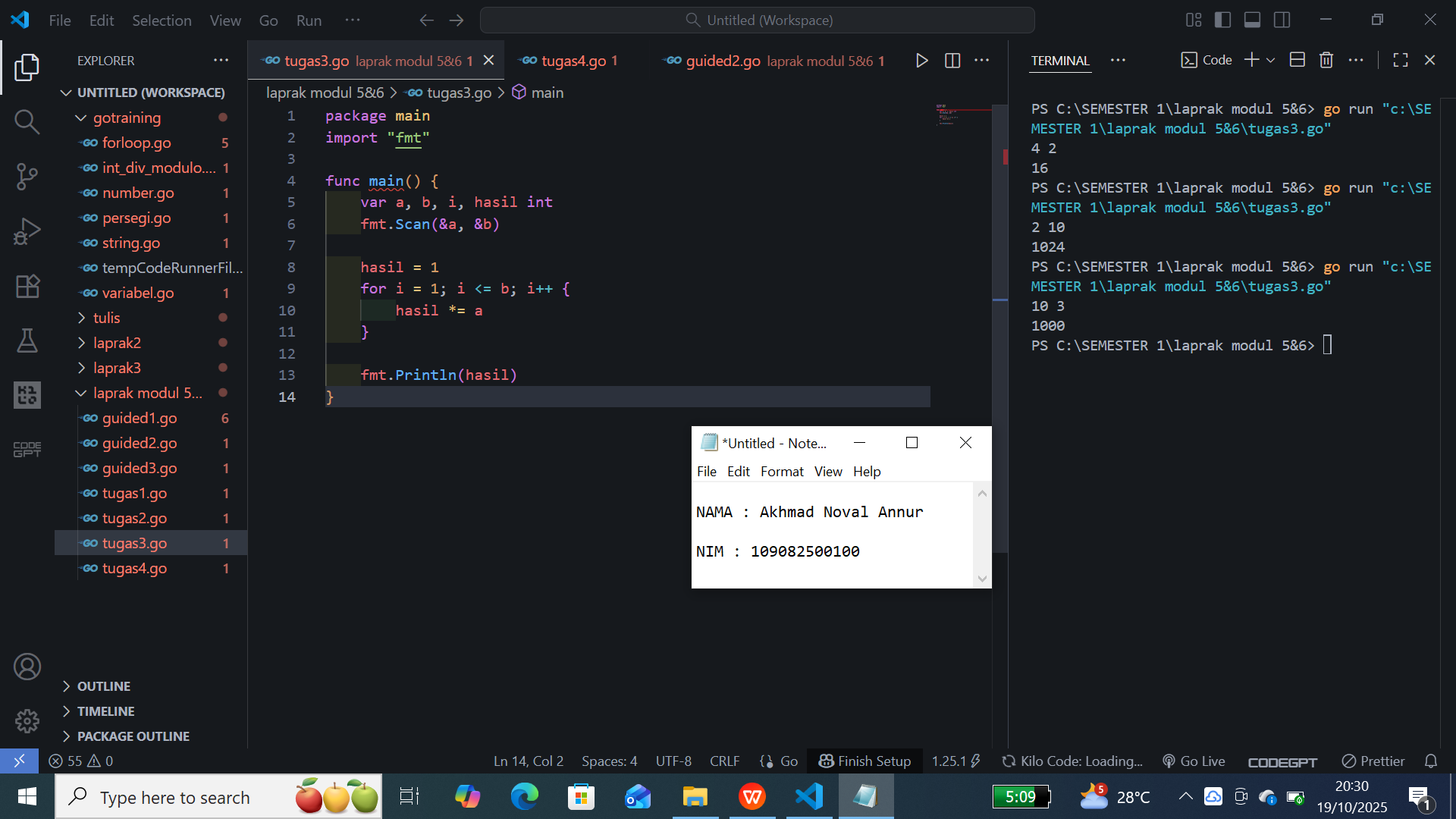
Program ini memperlihatkan penerapan rumus matematika dalam bentuk algoritmik yang dapat dijalankan oleh komputer. Selain itu, program juga membantu memahami penggunaan pustaka standar, tipe data desimal, serta logika perulangan untuk mengolah beberapa data sekaligus. Melalui implementasi ini, mahasiswa dapat menghubungkan konsep matematis dan algoritma pemrograman secara praktis.

1. **Tugas 3**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {  var a, b, i, hasil int  fmt.Scan(&a, &b)  hasil = 1  for i = 1; i <= b; i++ {  hasil \*= a  }  fmt.Println(hasil)  } |
|  |

**Screenshoot program**

****

**Deskripsi program**

Program ini berfungsi untuk menghitung hasil perpangkatan suatu bilangan bulat. Ketika dijalankan, pengguna diminta untuk memasukkan dua buah bilangan, yaitu bilangan dasar (a) dan bilangan pangkat (b). Program menginisialisasi variabel hasil dengan nilai awal 1.

Selanjutnya, dilakukan perulangan for sebanyak b kali, di mana setiap iterasi akan mengalikan nilai hasil dengan a. Setelah semua iterasi selesai, program menampilkan hasil akhir ke layar. Proses ini menggambarkan konsep perpangkatan sebagai hasil dari perkalian berulang.

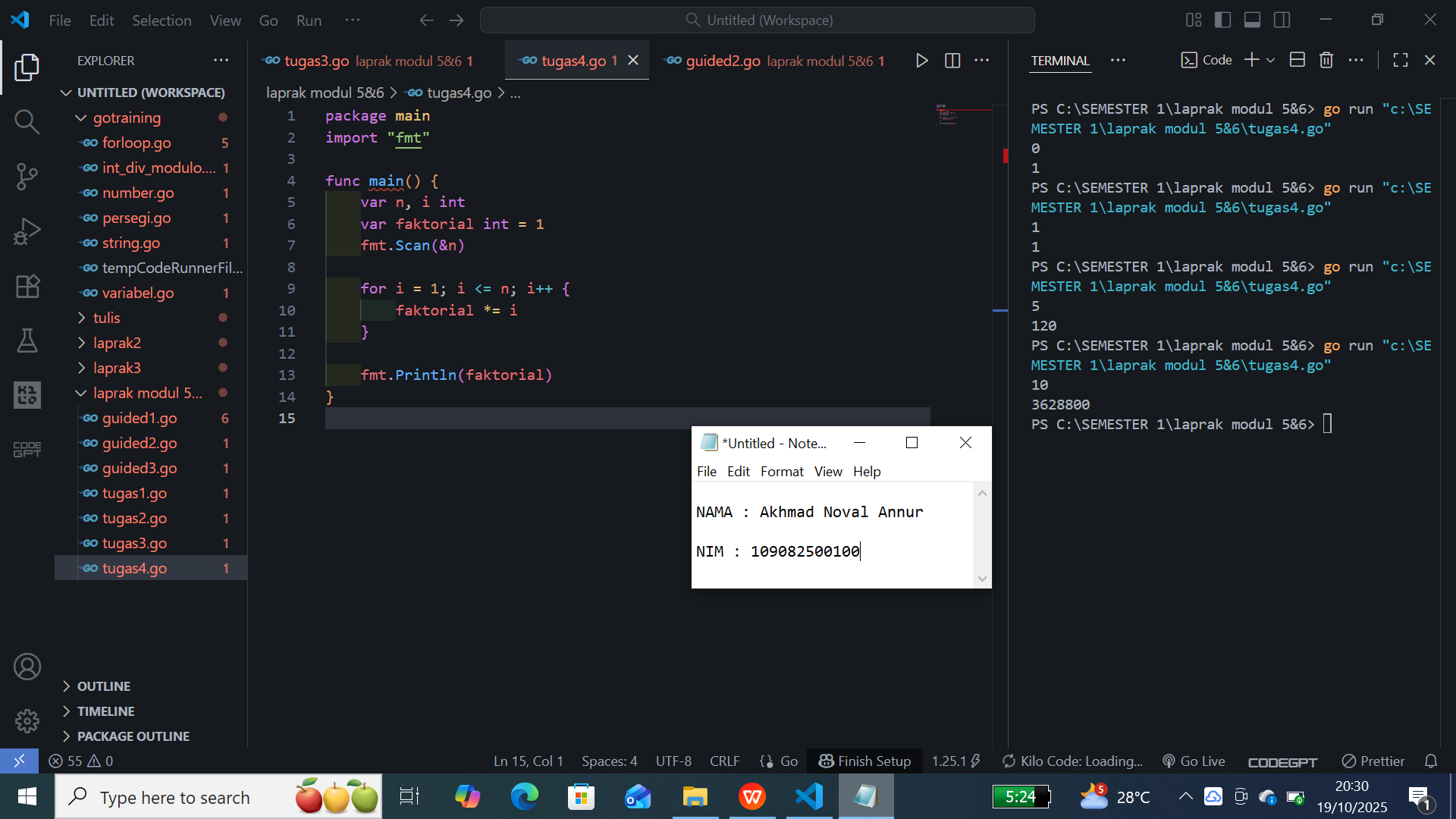
Program ini membantu pengguna memahami bagaimana operasi eksponensial dapat diterapkan melalui logika iteratif tanpa memanfaatkan fungsi matematika bawaan. Selain itu, program ini menunjukkan pentingnya inisialisasi variabel, urutan eksekusi, serta akurasi dalam melakukan proses komputasi yang berulang.

1. **Tugas 4**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {  var n, i int  var faktorial int = 1  fmt.Scan(&n)  for i = 1; i <= n; i++ {  faktorial \*= i  }  fmt.Println(faktorial)  } |
|  |

**Screenshoot program**

****

**Deskripsi program**

Program ini dibuat untuk menghitung nilai faktorial dari sebuah bilangan bulat positif. Saat dijalankan, pengguna diminta memasukkan nilai n. Program menginisialisasi variabel faktorial dengan nilai awal 1, yang kemudian digunakan sebagai penampung hasil perhitungan.

Perulangan for dijalankan mulai dari 1 hingga n, dan dalam setiap iterasi, nilai faktorial dikalikan dengan nilai i yang sedang diproses. Setelah seluruh proses selesai, program menampilkan hasil akhir berupa nilai faktorial dari bilangan yang dimasukkan.

Program ini memberikan pemahaman mendalam tentang konsep iterasi dan penggandaan berurutan yang digunakan dalam perhitungan faktorial. Selain itu, program juga memperkuat pemahaman mengenai pentingnya logika perulangan (looping), inisialisasi variabel, dan struktur kontrol dalam penyusunan algoritma yang efisien.