**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 5 & 6**

**FOR-LOOP**

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

Konten yang dihasilkan AI mungkin salah.**

**Disusun oleh:**

**ANDIKA FATHUR RIZKI**

**109082530012**

**S1IF-13-07**

**Asisten Praktikum**

Adithana dharma putra

Apri pandu wicaksono

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

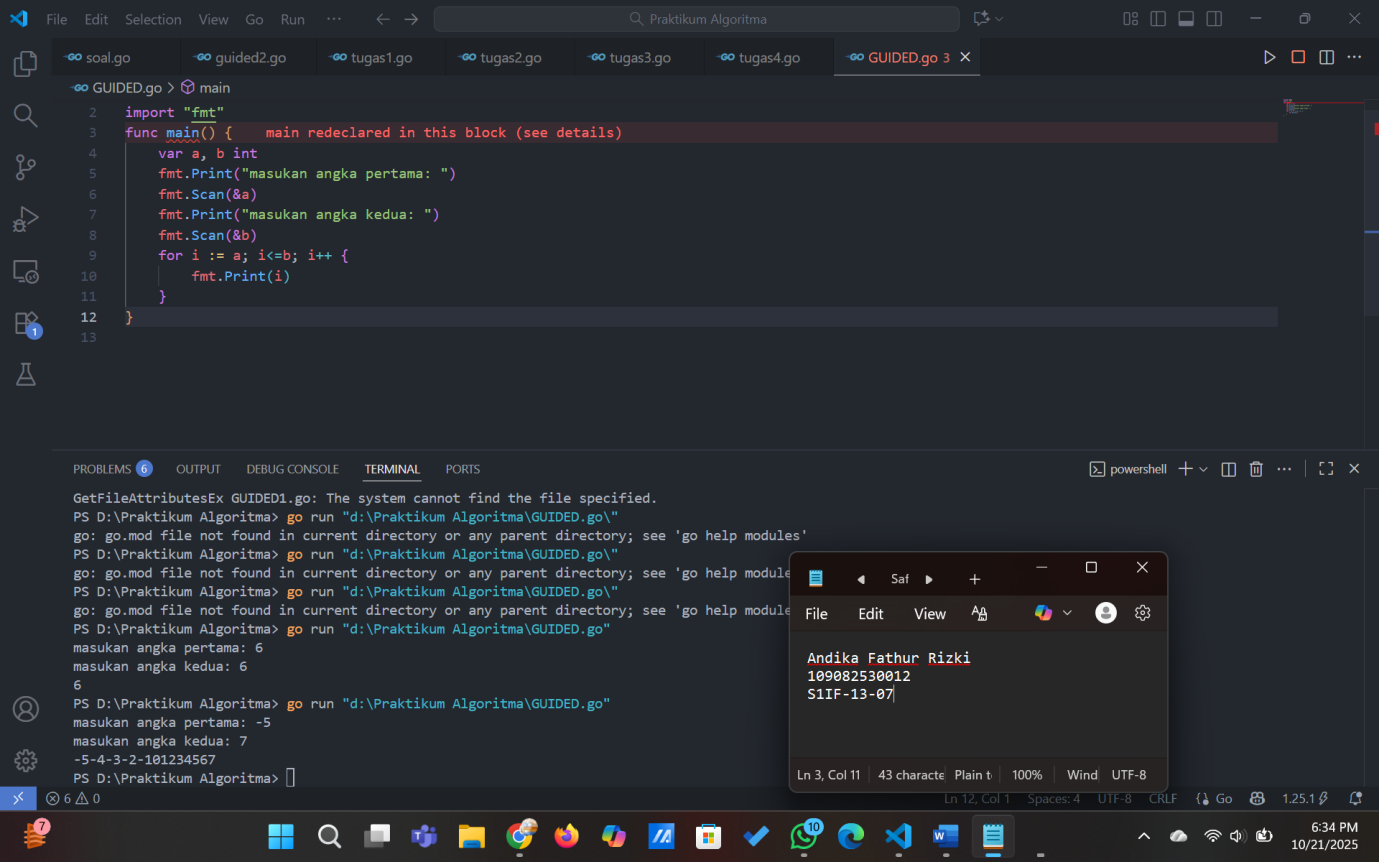
**LATIHAN KELAS – GUIDED**

1. **Guided 1**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {      var a, b int      fmt.Print("masukan angka pertama: ")      fmt.Scan(&a)      fmt.Print("masukan angka kedua: ")      fmt.Scan(&b)      for i := a; i<=b; i++ {          fmt.Print(i)      }  } |

**Screenshoot program**

****

**Deskripsi program**

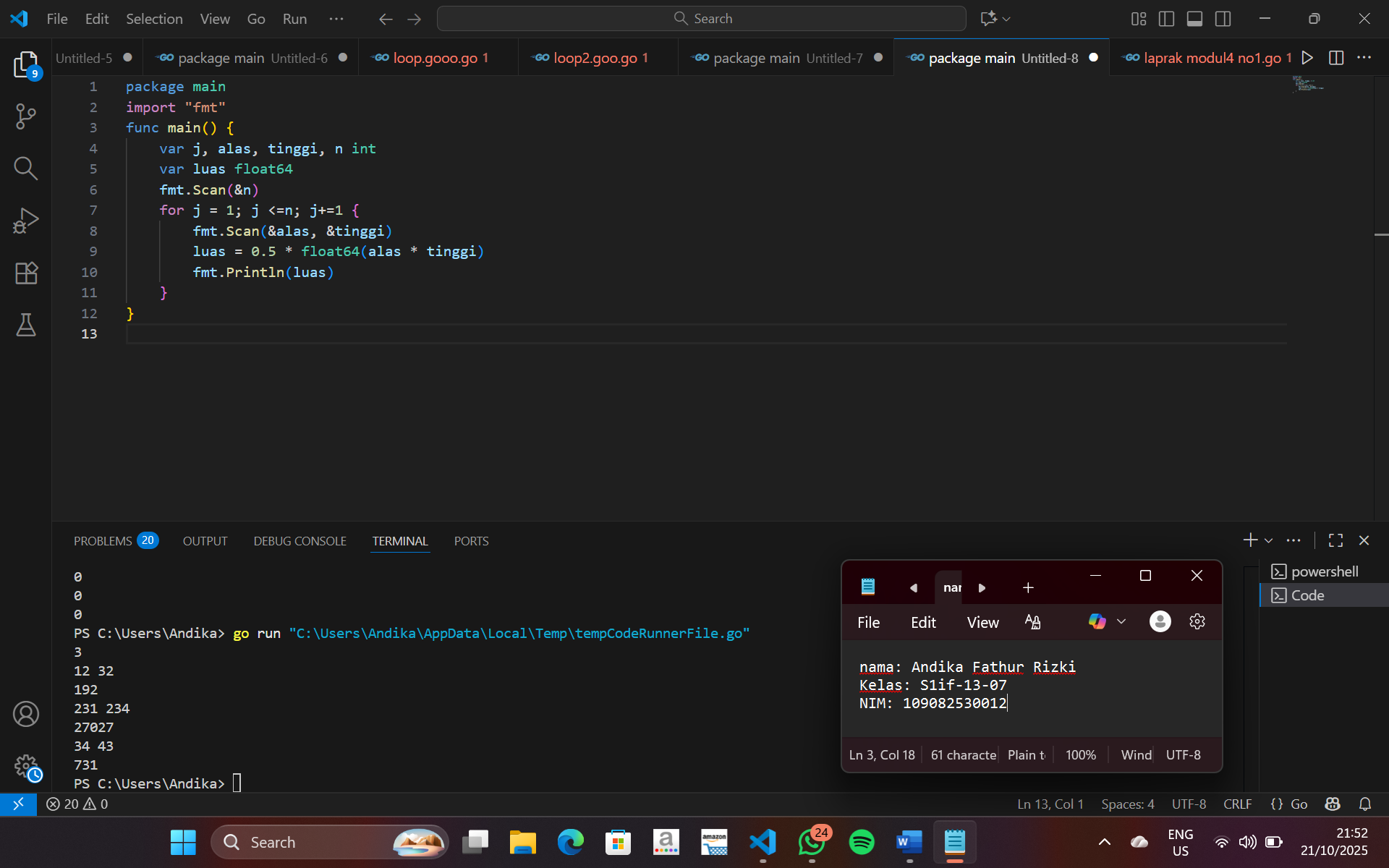
Program di atas berfungsi untuk menampilkan deretan angka dari nilai awal hingga nilai akhir yang dimasukkan oleh pengguna. Pertama, program mendeklarasikan dua variabel a dan b bertipe integer, lalu meminta pengguna memasukkan dua angka menggunakan fmt.Scan. Setelah itu, perulangan for i := a; i <= b; i++ digunakan untuk mencetak semua angka mulai dari a hingga b. Misalnya, jika pengguna memasukkan a = 2 dan b = 5, maka prosesnya dimulai dari i = 2 yang kemudian bertambah satu setiap perulangan dan mencetak angka 2, 3, 4 hingga 5. Setelah i mencapai nilai b, perulangan berhenti dan hasil akhir yang muncul di layar adalah 2345.

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**

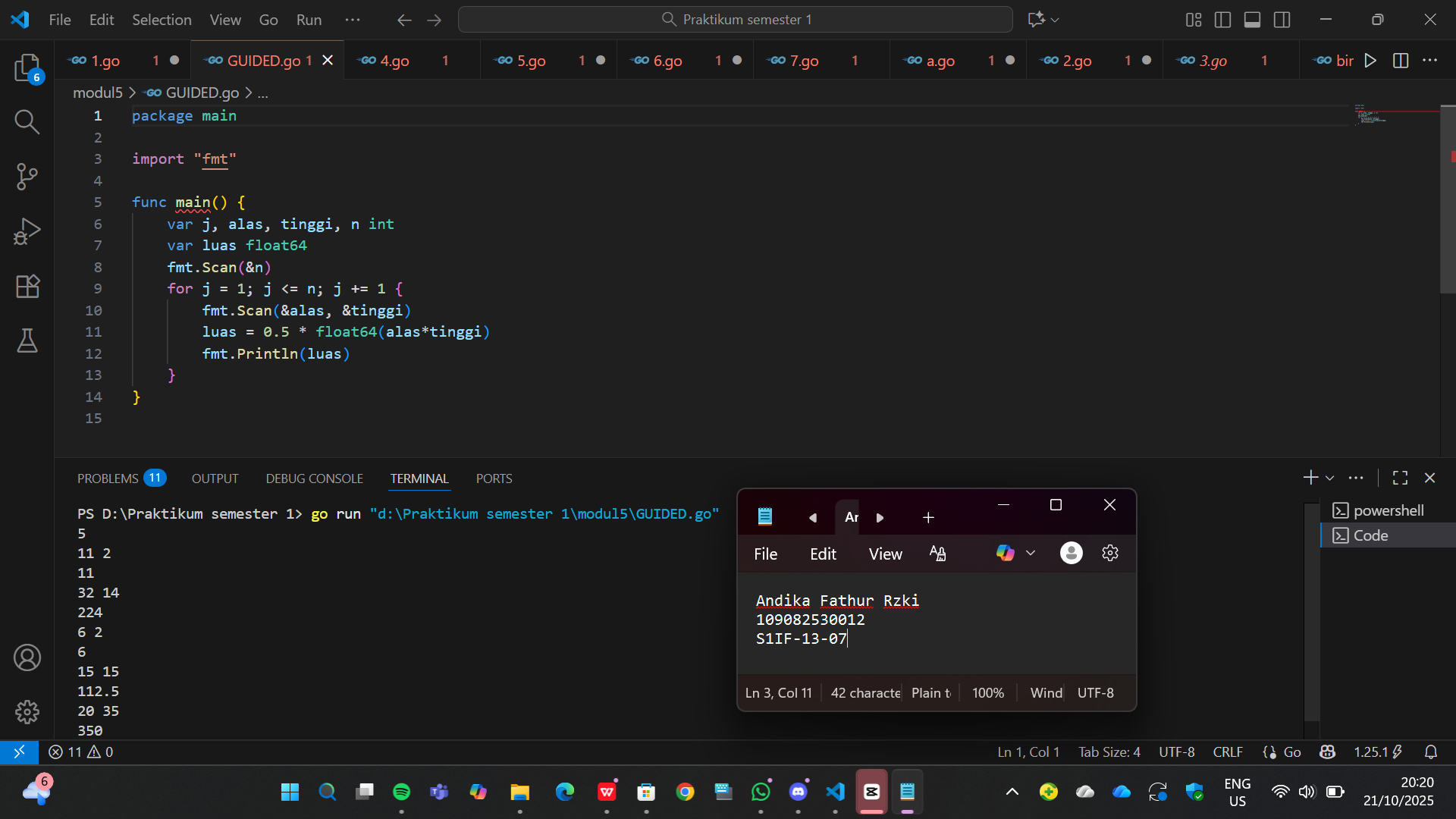
Program di atas digunakan untuk menghitung luas beberapa segitiga berdasarkan nilai alas dan tinggi yang dimasukkan oleh pengguna. Pertama, program mendeklarasikan variabel j, alas, tinggi, dan n bertipe integer serta luas bertipe float64 untuk menyimpan hasil perhitungan. Nilai n dibaca menggunakan fmt.Scan sebagai jumlah segitiga yang akan dihitung. Selanjutnya, perulangan for j = 1; j <= n; j++ digunakan agar program membaca nilai alas dan tinggi setiap segitiga satu per satu. Di dalam perulangan, rumus luas segitiga diterapkan yaitu luas = 0.5 \* float64(alas \* tinggi) dan hasilnya langsung ditampilkan dengan fmt.Println(luas). Misalnya, jika pengguna memasukkan input 5, lalu diikuti data 11 2, 32 14, 6 2, 15 15, dan 20 35, maka program akan menghitung luas setiap segitiga secara berurutan sehingga menghasilkan output 11, 224, 6, 112.5, dan 350.

1. **Guided 3**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {      var a, b, c int      var hasil int      fmt.Print("masukan angka pertama: ")      fmt.Scan(&b)      fmt.Print("masukan angka kedua: ")      fmt.Scan(&c)      hasil = 0          for a = 1; a <= c; a+=1 {          hasil = hasil + b      }      fmt.Println(hasil)  } |

**Screenshoot program**



**Deskripsi program**

Program di atas digunakan untuk menghitung hasil perkalian dua bilangan menggunakan operasi penjumlahan berulang. Pertama, program mendeklarasikan variabel a, b, dan c bertipe integer untuk menyimpan nilai yang akan diproses, serta variabel hasil untuk menyimpan hasil akhir perhitungan. Program kemudian meminta pengguna memasukkan dua angka melalui fmt.Scan, di mana nilai pertama disimpan pada variabel b dan nilai kedua pada variabel c. Nilai awal hasil diatur menjadi 0, kemudian perulangan for a = 1; a <= c; a++ dijalankan sebanyak c kali. Di dalam perulangan, setiap iterasi menambahkan nilai b ke variabel hasil, sehingga secara logika proses ini sama dengan operasi b × c. Setelah perulangan selesai, program mencetak nilai hasil sebagai output. Misalnya, jika pengguna memasukkan angka pertama 7 dan angka kedua 6, maka program akan melakukan penjumlahan 7 + 7 + 7 + 7 + 7 + 7 atau menjumlahkan sebanyak enam kali dan menghasilkan output 42.

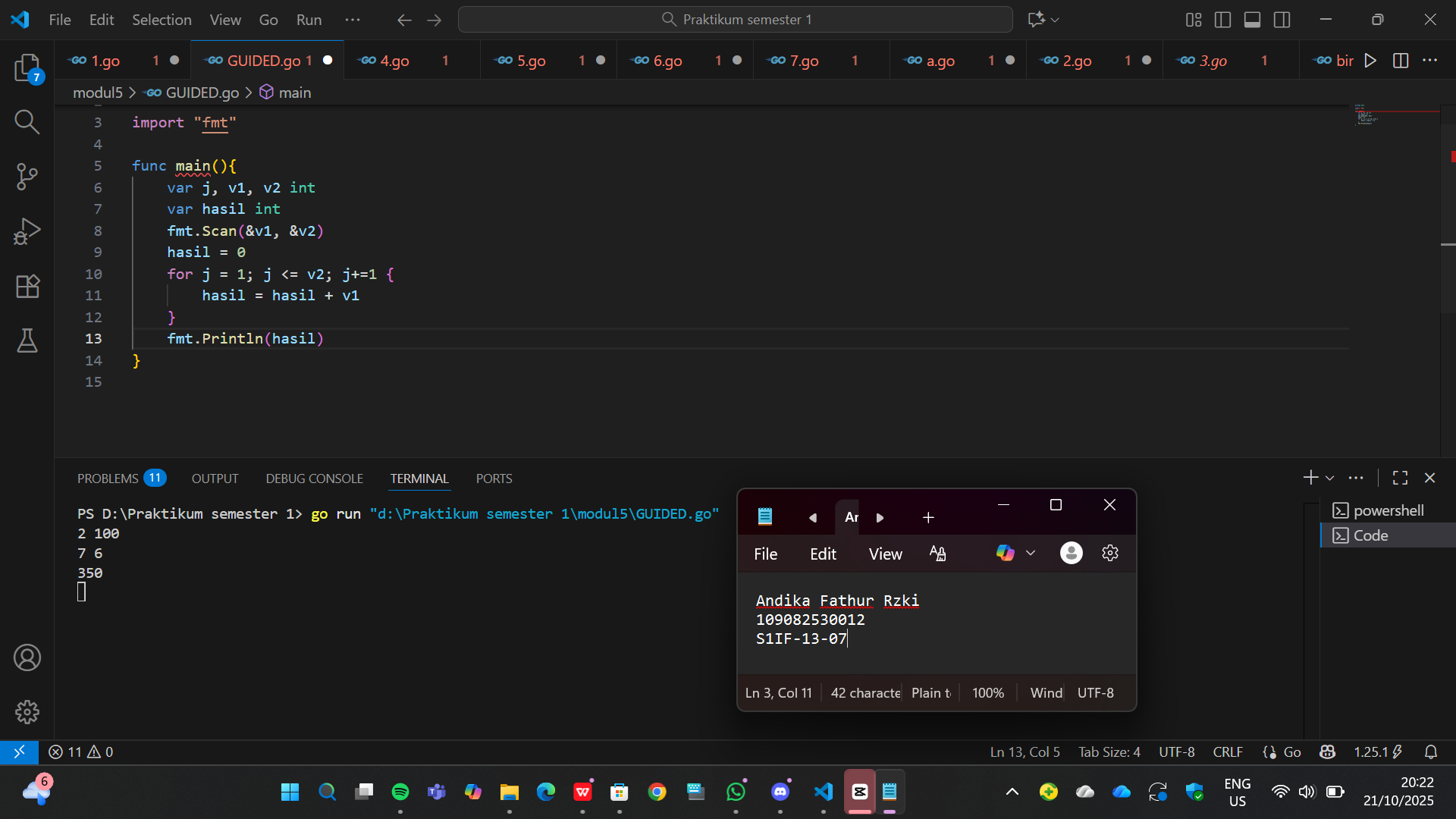
**TUGAS**

1. **Tugas 1**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      var n int      var hasil int      fmt.Print("Masukkan bilangan n: ")      fmt.Scan(&n)      hasil = 0      for i := 1; i <= n; i++ {          hasil += i //sama dengan hasil = hasil + i      }      fmt.Println(hasil)  } |

**Screenshoot program**



**Deskripsi program**

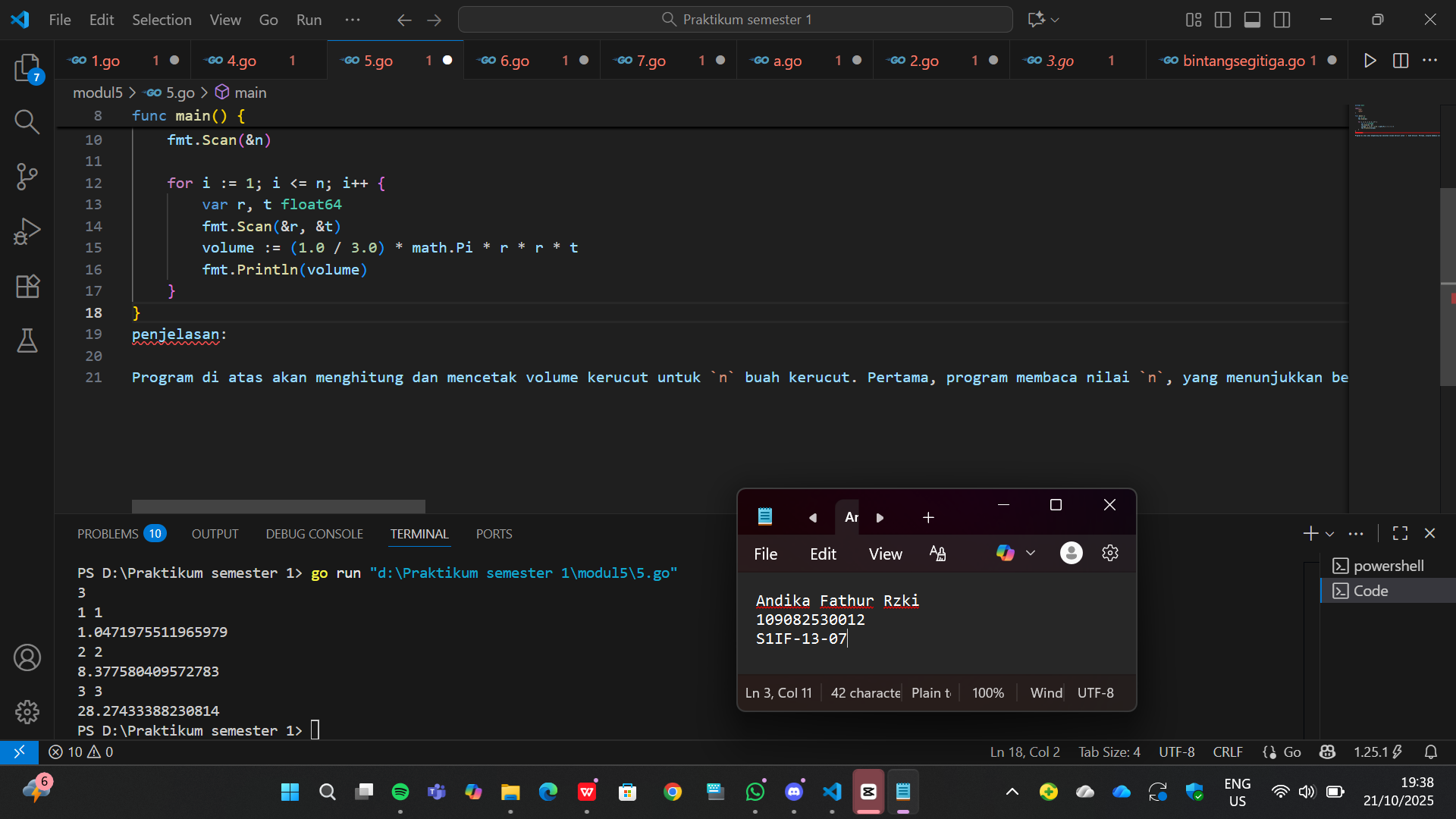
Program di atas digunakan untuk menghitung jumlah dari deretan bilangan mulai dari 1 hingga n yang dimasukkan oleh pengguna. Pertama, program mendeklarasikan variabel n dan hasil bertipe integer untuk menyimpan input dan hasil penjumlahan. Program kemudian meminta pengguna memasukkan nilai n menggunakan fmt.Scan, lalu menginisialisasi hasil dengan nilai 0. Setelah itu, perulangan for i := 1; i <= n; i++ digunakan untuk menambahkan setiap bilangan dari 1 sampai n ke dalam variabel hasil dengan perintah hasil += i. Misalnya, jika pengguna memasukkan input 1, maka perulangan hanya dijalankan satu kali dengan nilai i = 1, sehingga hasil berubah dari 0 menjadi 1. Setelah perulangan selesai, program menampilkan hasil akhirnya yaitu 1.

1. **Tugas 2**

**Source code**

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

**Screenshoot program**



**Deskripsi program**

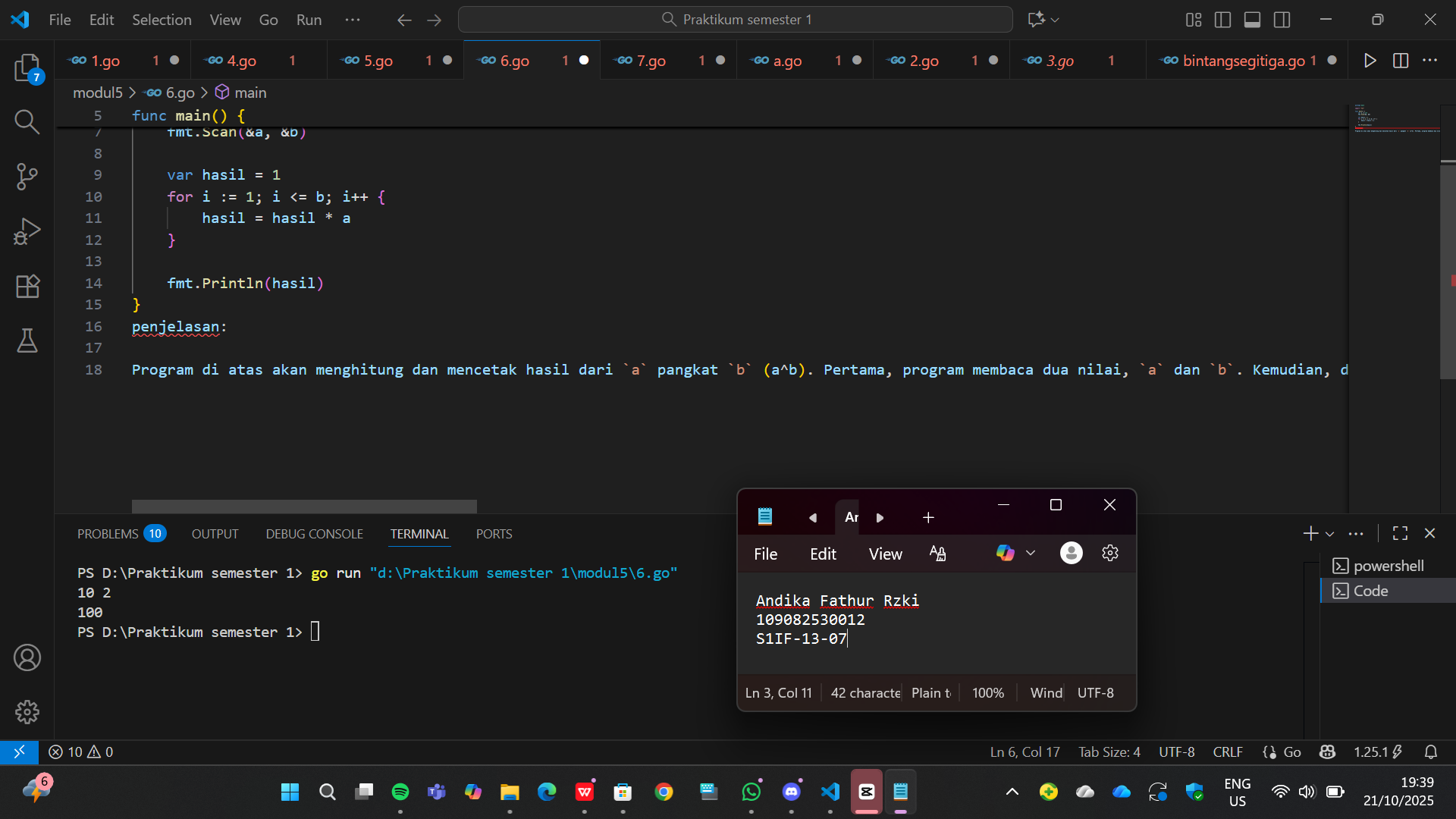
Program di atas digunakan untuk menghitung volume beberapa kerucut berdasarkan nilai jari-jari dan tinggi yang dimasukkan oleh pengguna. Pertama, program mengimpor paket fmt untuk input-output dan math untuk operasi matematika seperti π (pi) dan pemangkatan. Variabel n bertipe integer digunakan untuk menampung jumlah kerucut yang akan dihitung, kemudian program meminta pengguna memasukkan nilai n melalui fmt.Scan. Setelah itu, perulangan for i := 0; i < n; i++ digunakan untuk membaca data setiap kerucut satu per satu. Di dalam perulangan, program meminta input r (jari-jari) dan t (tinggi) bertipe float64, lalu menghitung volumenya menggunakan rumus volume kerucut yaitu volume = (1/3) × π × r² × t, di mana math.Pow(r, 2) digunakan untuk menghitung r² dan menggunakan math pow ini lebih menjadikan codingan kita simple daripada make r \* r ataupun r\*2 itu tergantung perspektif masing masing. Setelah perhitungan selesai, hasil volume langsung ditampilkan menggunakan fmt.Println(volume).

Sebagai contoh, jika pengguna memasukkan input n = 3 untuk jumlah kerucut, kemudian untuk setiap kerucut memasukkan nilai r = 2 dan t = 2, maka proses perhitungan untuk masing-masing kerucut adalah volume = (1/3) × 3.141592653589793 × 2² × 2 = 8.377580409572781. Itu salah satu outputnya jika seperti yang ingin diminta di modul Latihan ada di screenshoot an code inputnya sama dengan modul dan dipenjelasan ini hanya untuk contoh agar simple untuk dimengerti.

1. **Tugas 3**

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

**Screenshoot program**



**Deskripsi program**

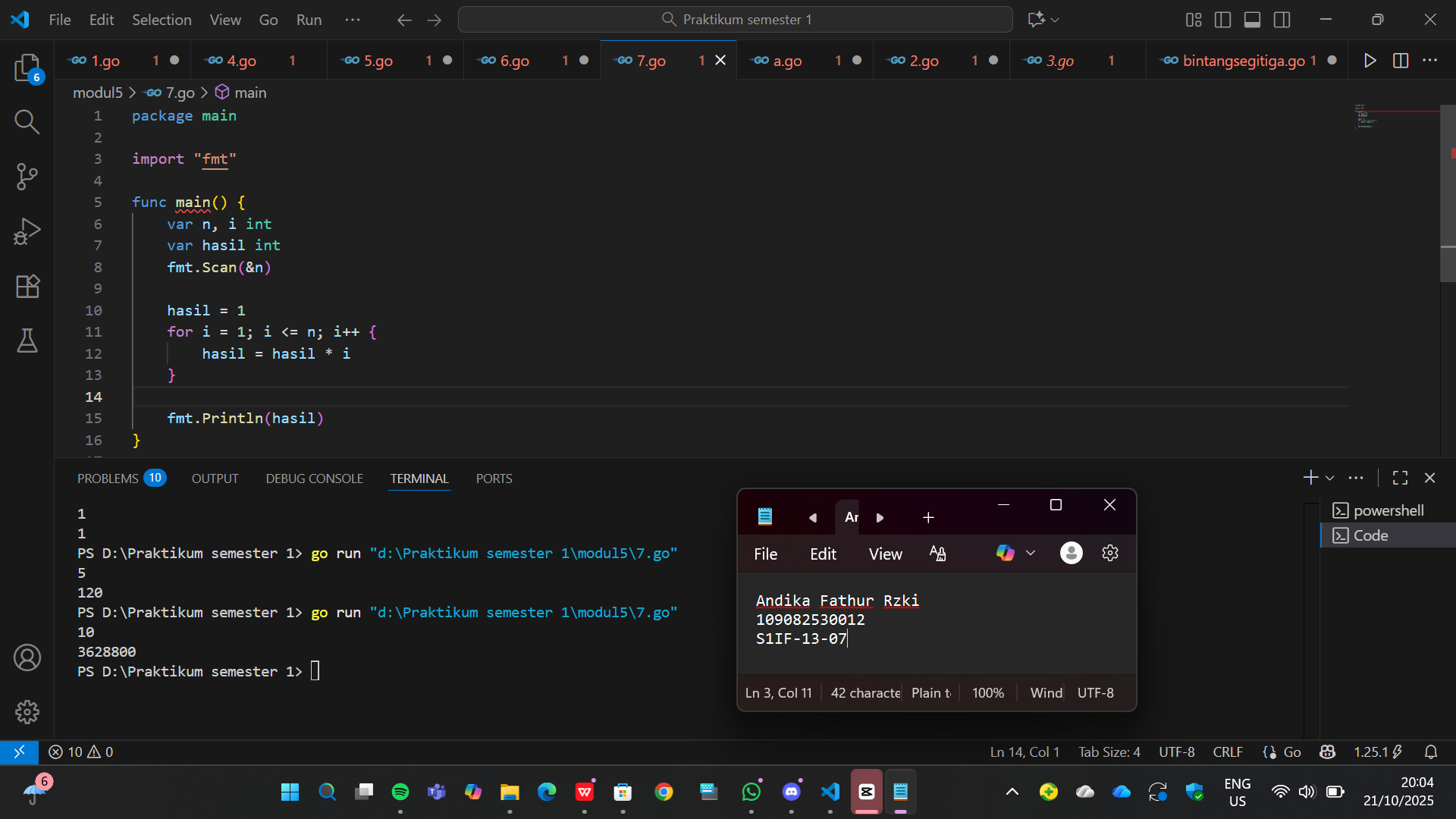
Program di atas digunakan untuk menghitung hasil pemangkatan dua bilangan dengan menggunakan operasi perkalian berulang. Pertama, program mendeklarasikan variabel a dan b bertipe integer, kemudian meminta pengguna untuk memasukkan kedua bilangan tersebut melalui fmt.Scan. Setelah itu, variabel hasil diinisialisasi dengan nilai 1 sebagai nilai awal sebelum perulangan dimulai. Perulangan for i := 0; i < b; i++ dijalankan sebanyak b kali, dan di setiap iterasi nilai hasil dikalikan dengan a menggunakan perintah hasil \*= a, yang berarti hasil = hasil \* a. Setelah perulangan selesai, program menampilkan nilai akhir dari hasil yang merupakan hasil dari pemangkatan a^b.

Sebagai contoh, jika pengguna memasukkan nilai a = 2 dan b = 10, maka program akan melakukan proses perkalian berulang sebanyak 10 kali, yaitu 1×2×2×2×2×2×2×2×2×2×2, sehingga hasil akhirnya adalah 1024. Dengan demikian, output yang ditampilkan program adalah 1024, yang merupakan hasil dari perhitungan (2^{10}).

1. **Tugas 4**

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

**Screenshoot program**



**Deskripsi program**

Program di atas digunakan untuk menghitung nilai faktorial dari sebuah bilangan bulat non-negatif. Pertama, program mendeklarasikan variabel n bertipe integer untuk menampung input dari pengguna, kemudian meminta pengguna memasukkan nilai n melalui fmt.Scan. Variabel hasil diinisialisasi dengan 1 sebagai nilai awal untuk perhitungan faktorial. Selanjutnya, perulangan for i := 1; i <= n; i++ dijalankan dari 1 hingga n, dan pada setiap iterasi nilai hasil dikalikan dengan i menggunakan hasil \*= i, sehingga menghitung perkalian beruntun dari 1 sampai n. Setelah perulangan selesai, program mencetak nilai hasil sebagai hasil faktorial.

Sebagai contoh, jika pengguna memasukkan input n = 5, maka perhitungannya adalah 1 × 2 × 3 × 4 × 5, sehingga output akhir yang ditampilkan program adalah 120.