**PRAKTIKUM 07**

**Nested Classes**

(Disusun Untuk Tugas Kuliah Praktikum Pemograman Berorientasi Objek (A) Semester 3)



**Karin Lina Safitri**

**4523210056**

Dosen Pembimbing :

**Adi Wahyu Pribadi , S.Si., M.Kom.**

**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

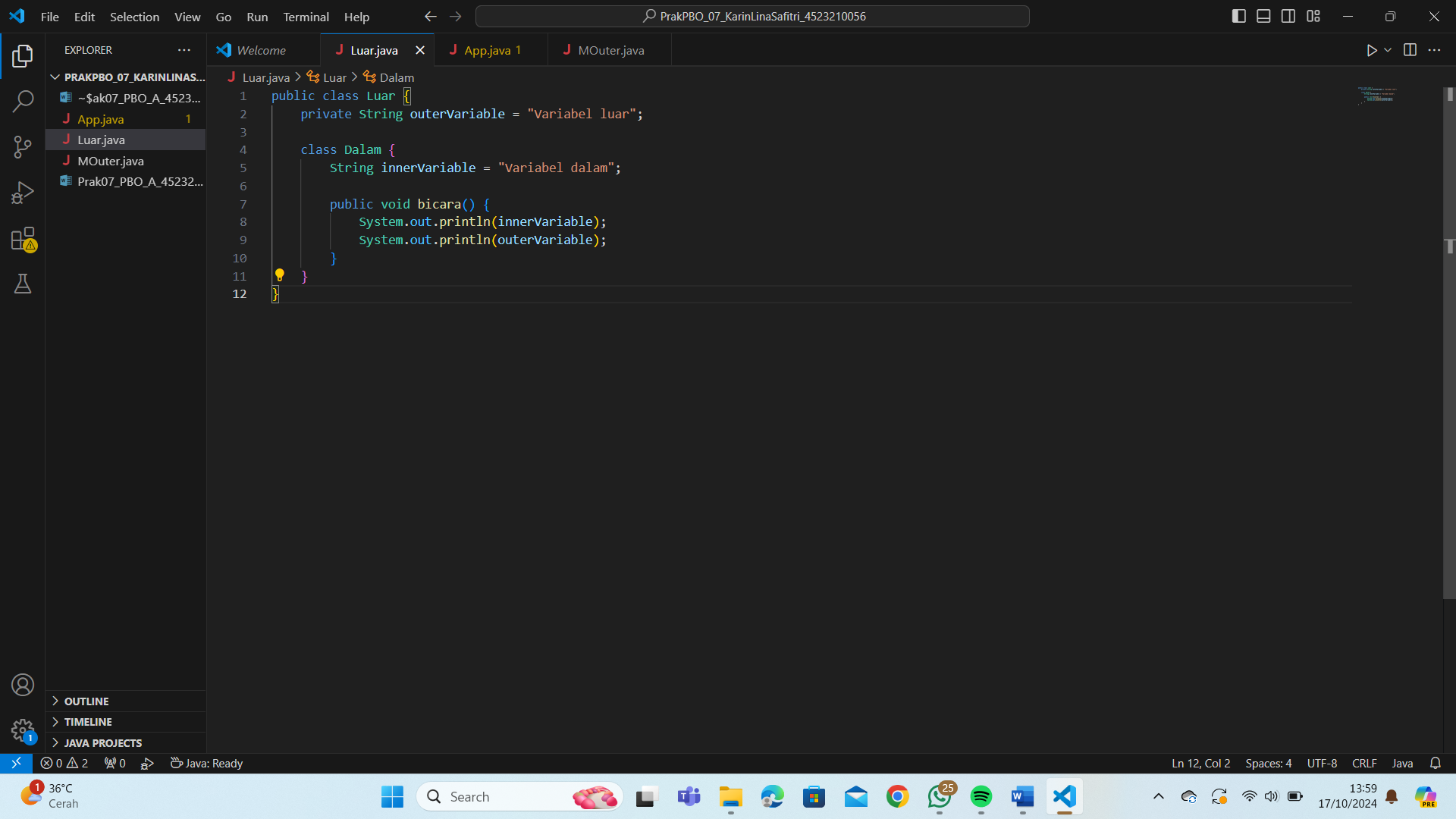
**JAKARTA**

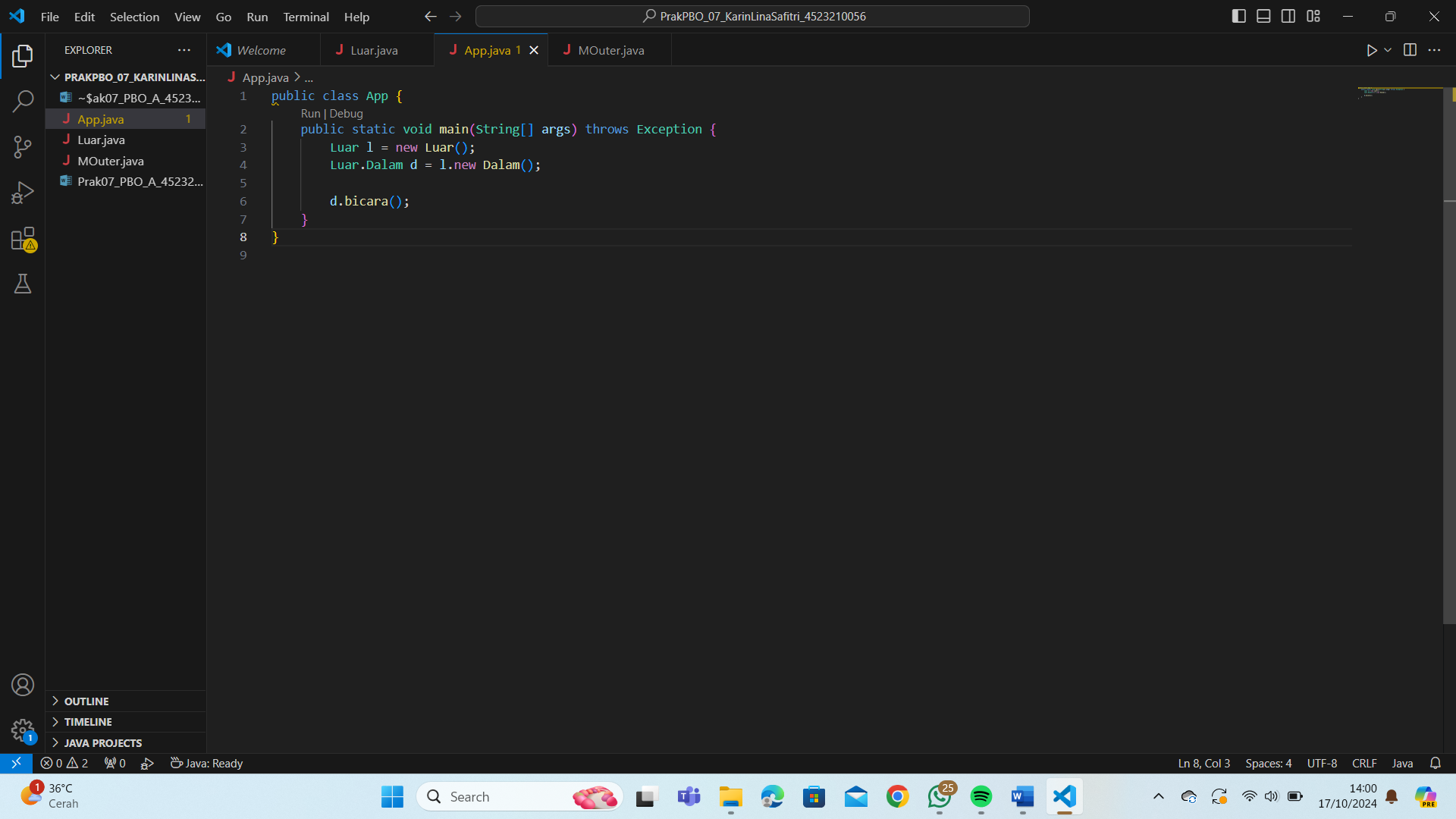
**2024**

1. **Tuliskan program ini di Visual Code kalian, Run dan tangkap Outputnya.**
2. **Jelaskan mengapa outputnya bisa seperti itu!**

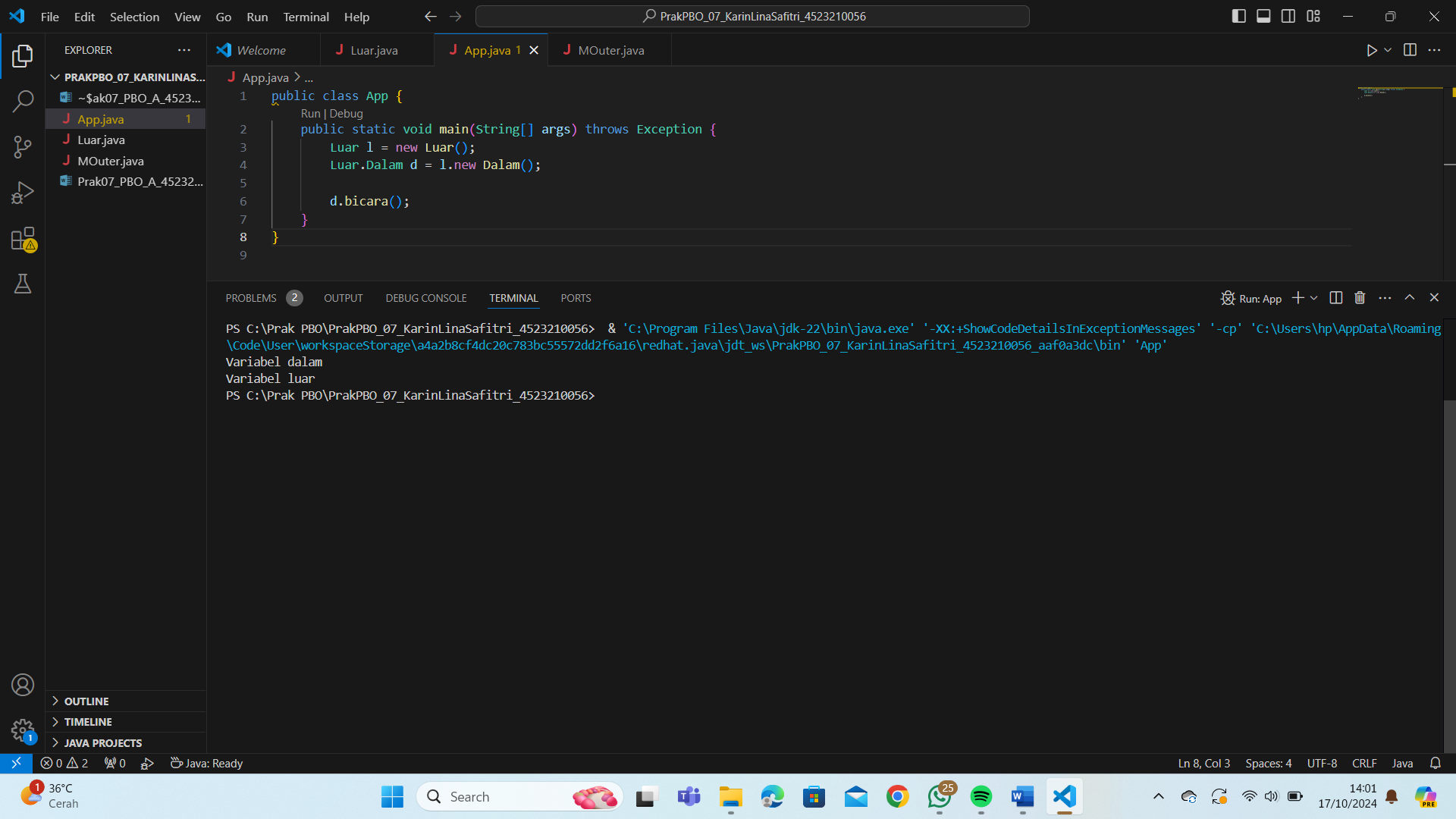
* **Latihan 1**

**Program :**





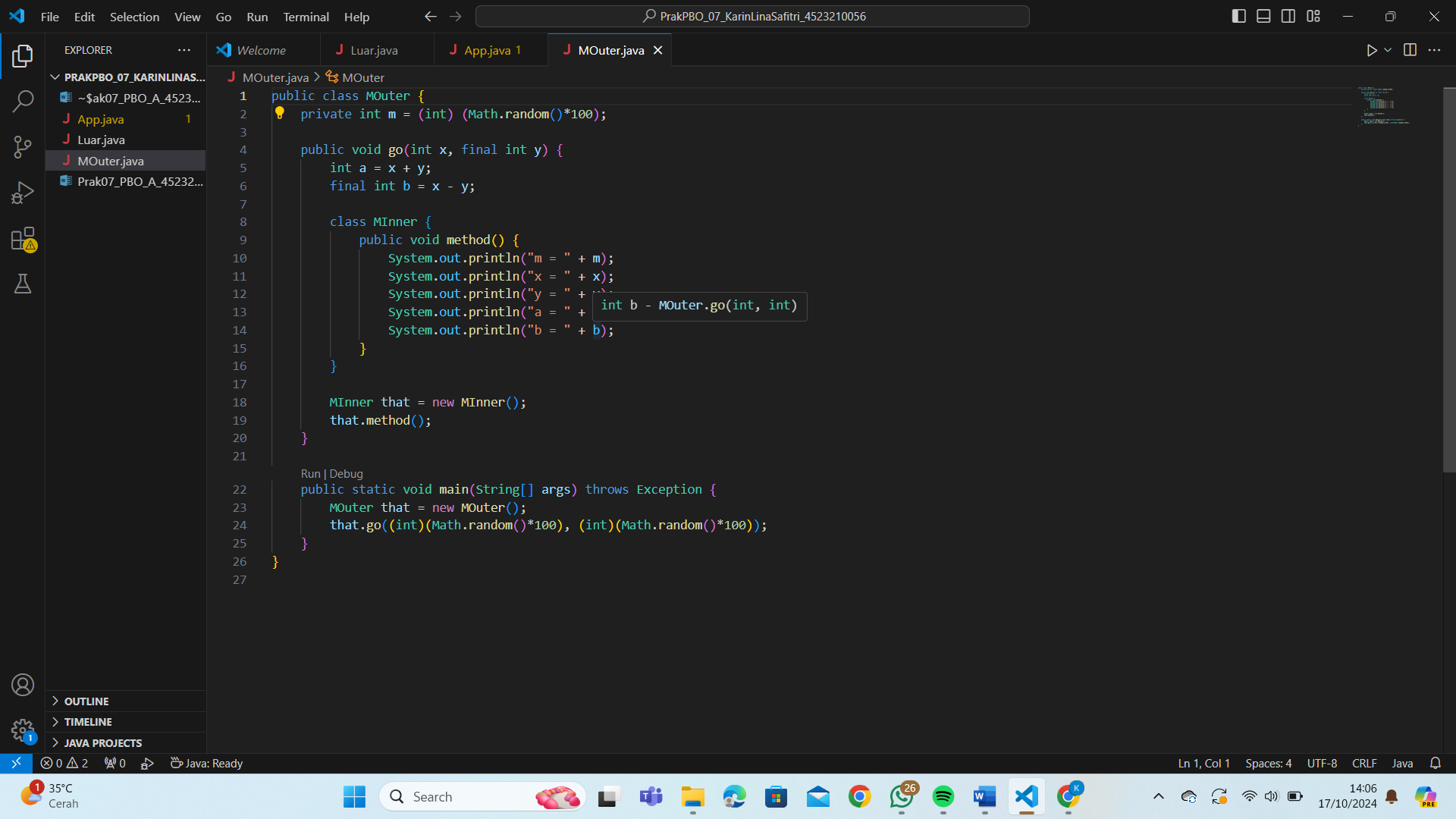
**Hasil Run:**



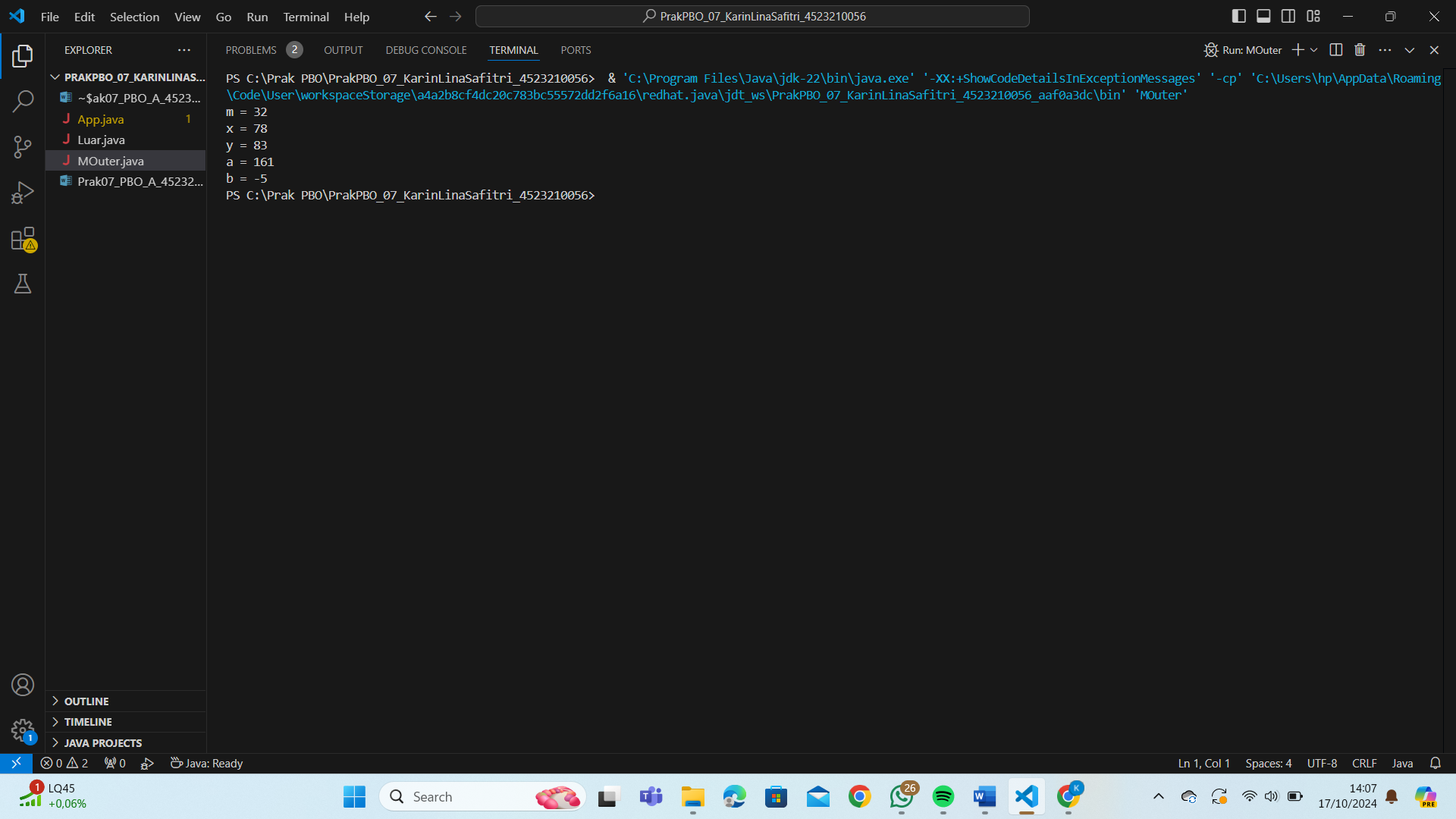
Outputnya bisa seperti ini karena menunjukkan konsep aksesibilitas variabel di dalam dan di luar kelas dalam (inner class). "Variabel dalam" menunjukkan bahwa program dapat mengakses atau mencetak variabel di dalam kelas, dan "Variabel luar" menunjukkan bahwa program dapat mengakses variabel di luar kelas.Output muncul dalam urutan yang sesuai dengan variabel yang dipanggil.

* **Latihan 2**

**Program :**



**Hasil Run :**



Outputnya bisa seperti ini kemungkinan besar terjadi karena melibatkan operasi pada beberapa variabel atau metode, setiap variabel memiliki nilai yang beda, dan setiap kali variabel tersebut dipanggil atau diakses, nilai tersebut dicetak.

* **Latihan 3**

Jelaskan kelebihan dan kelemahan Inner Class!

Kelebihan dan Kelemahan Kelas Dalam dalam Pemrograman Java Kelas dalam, atau "kelas dalam", adalah kelas yang didefinisikan di dalam kelas lainnya, yang memberikan fleksibilitas dan kemampuan untuk mengorganisasi kode dengan lebih baik. Namun, seperti konsep pemrograman lainnya, ada kelebihan dan kelemahan dari konsep ini.  
  
Kelebihan Inner Class:

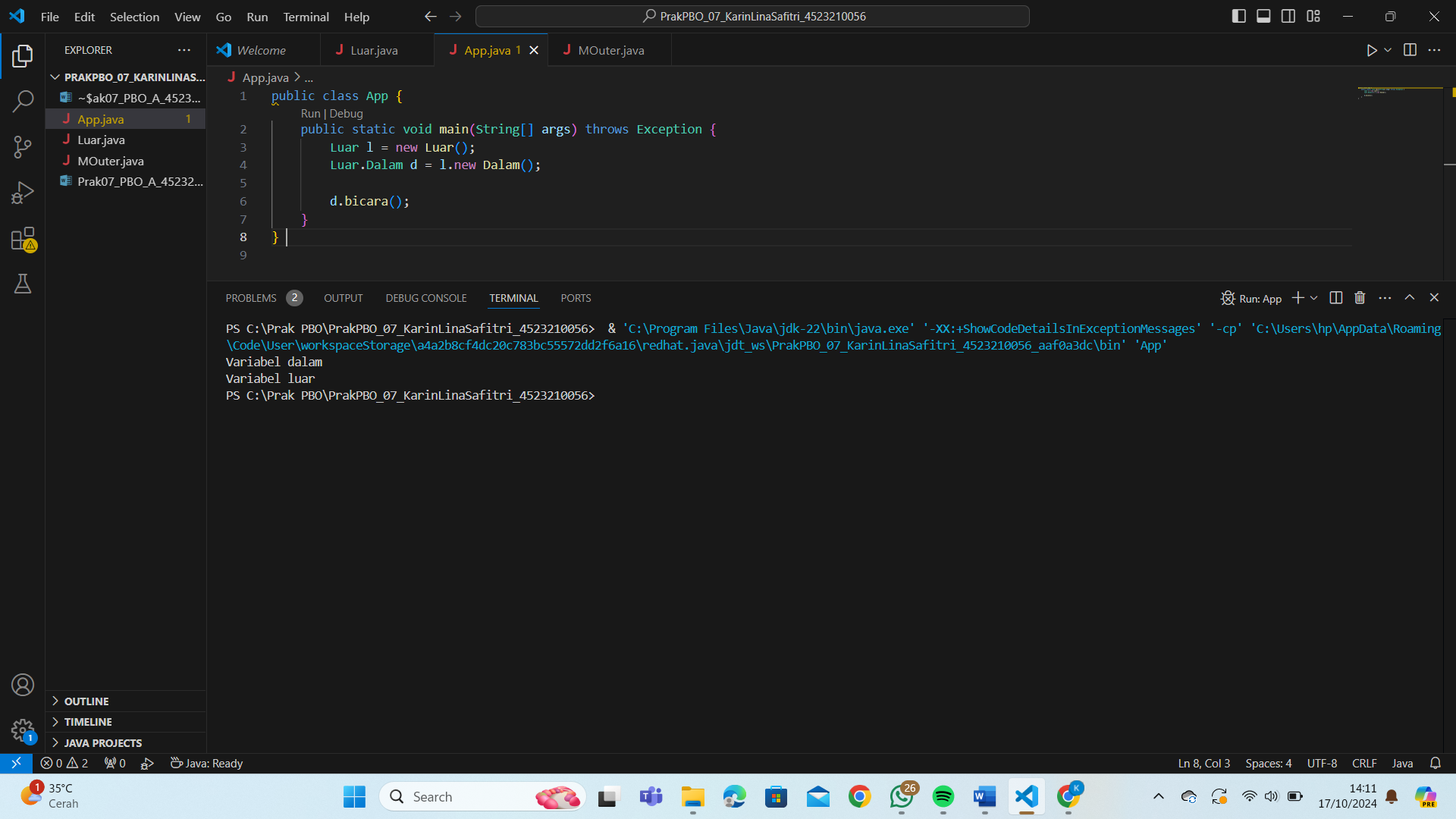
* Enkapsulasi yang Lebih Baik: Kelas internal dapat secara langsung mengakses anggota pribadi kelas luar, yang memungkinkan Anda membuat hubungan yang lebih erat antara kelas-kelas yang terkait, yang memungkinkan Anda menyembunyikan implementasi detail dan meningkatkan modularitas kode.
* Peningkatan Readability: Kelas internal dapat digunakan untuk mengelompokkan kelas-kelas yang sangat terkait, membuat kode lebih mudah dibaca dan dipahami.

Kelemahan Inner Class:

* Kompleksitas: Kode dapat menjadi lebih kompleks dan sulit untuk dipahami, terutama bagi pemula.  
  Melacak aliran eksekusi program dapat menjadi sulit karena hierarki kelas yang dalam.
* Debugging: Mengedit kelas dalam bisa lebih sulit daripada mengedit kelas biasa, terutama dalam kasus di mana ada banyak tingkat nesting.
* Ketergantungan: Kelas dalam sangat bergantung pada kelas luarnya, dan jika kelas luar diubah, kelas dalam juga harus diubah.
* **Latihan 4**

Run semua contoh yang ada di Pertemuan hari ini (Delapan UTS)

* **Luar.java dan App.java**



* **MOuter.java**

