**LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**

“Nested Classes*”*

**

**Anavalis Ridho Abdee Nugroho**

**4523210012**

Dosen Pembimbing :

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

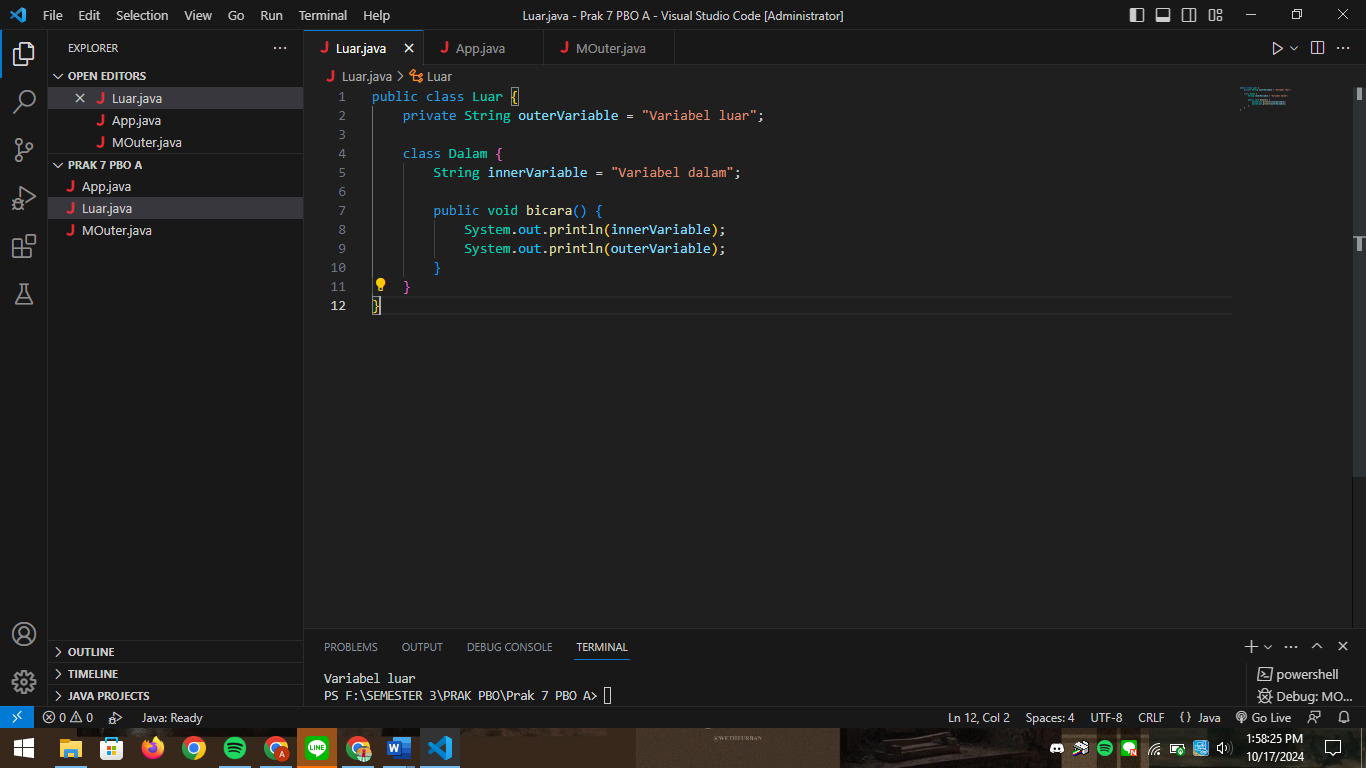
**FAKULTAS TEKNIK**

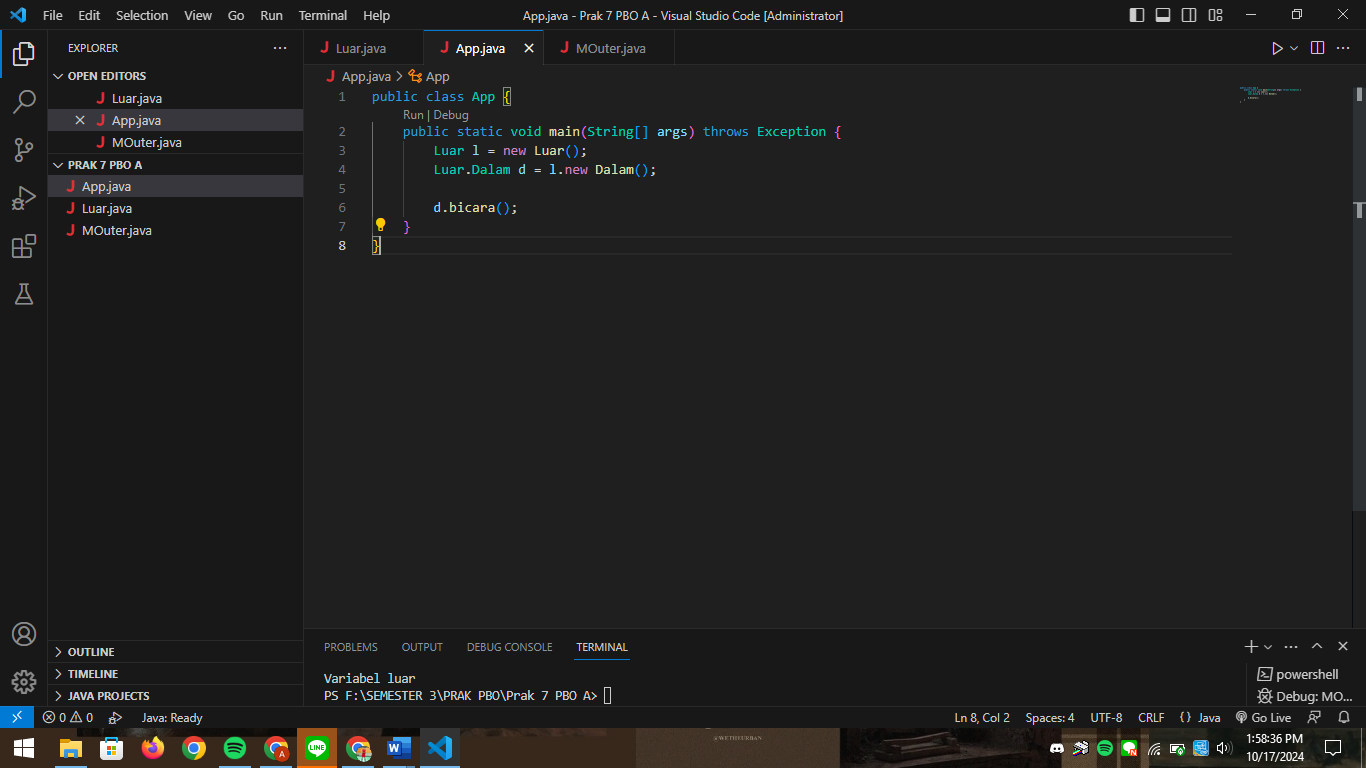
**UNIVERSITAS PANCASILA**

**JAKARTA**

**2024**

1. **Source Code Latihan 1**

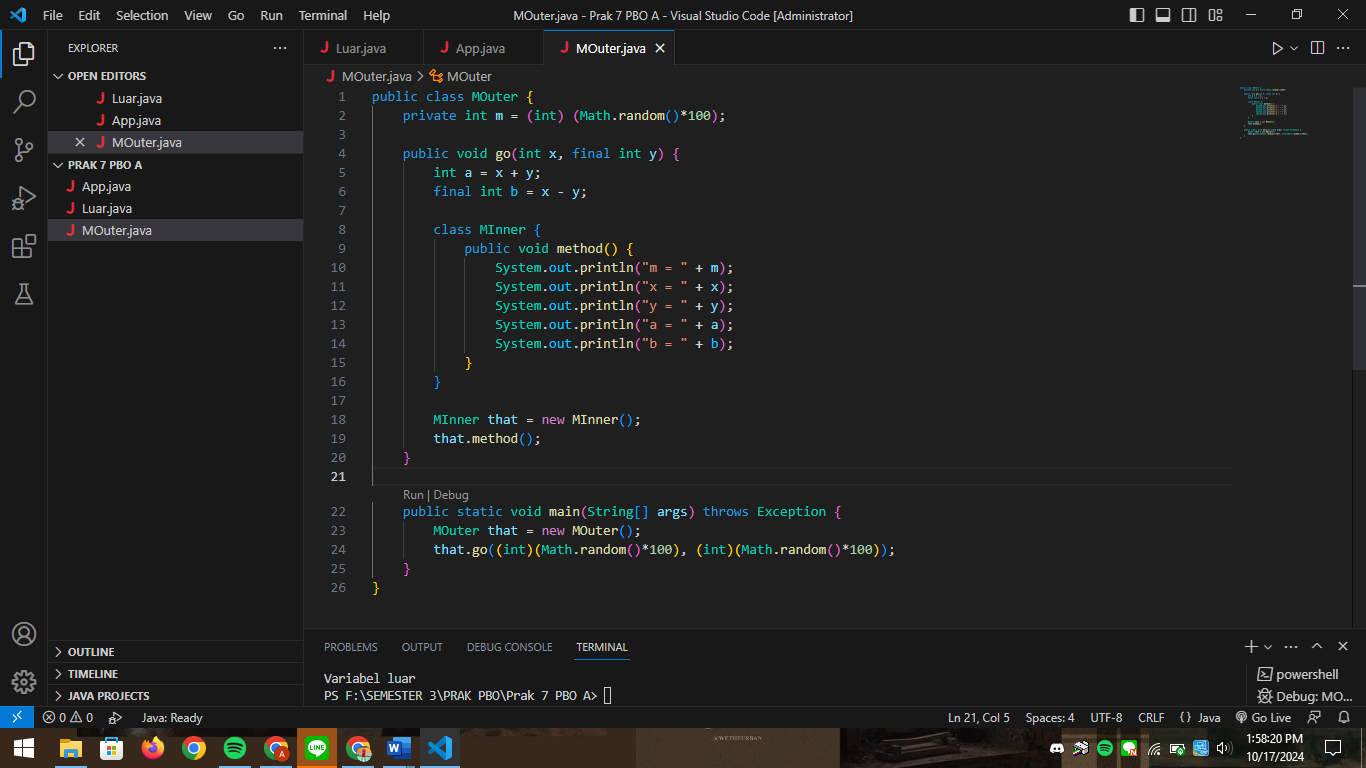
****

****

**Penjelasan**

* **Kelas Dalam** (Luar.Dalam) memiliki akses penuh ke variabel-variabel dari kelas luarnya (kelas Luar), termasuk variabel outerVariable meskipun variabel tersebut bersifat private. Ini adalah salah satu keistimewaan dari inner class di Java.
* Ketika metode bicara() dipanggil, ia mencetak terlebih dahulu nilai dari variabel innerVariable yang dimiliki oleh kelas Dalam, yaitu "Variabel dalam".
* Kemudian, ia mencetak nilai dari variabel outerVariable dari kelas Luar, yaitu "Variabel luar".

1. **Latihan 2**

****

Penjelasan

* Nilai m adalah variabel instance kelas MOuter, yang diinisialisasi dengan nilai acak antara 0 hingga 99.
* Nilai x dan y dikirimkan sebagai argumen ketika metode go() dipanggil.
* Nilai a dihitung sebagai hasil dari x + y sehingga merupakan penjumlahan dari dua angka acak. dan Variabel b dihitung sebagai hasil dari x - y, yang merupakan pengurangan antara dua angka acak tersebut.

Setiap kali program dijalankan, hasilnya akan berbeda karena nilai acak yang digunakan untuk variabel m, x, dan y. Namun, urutan nilai yang dicetak akan selalu sama:

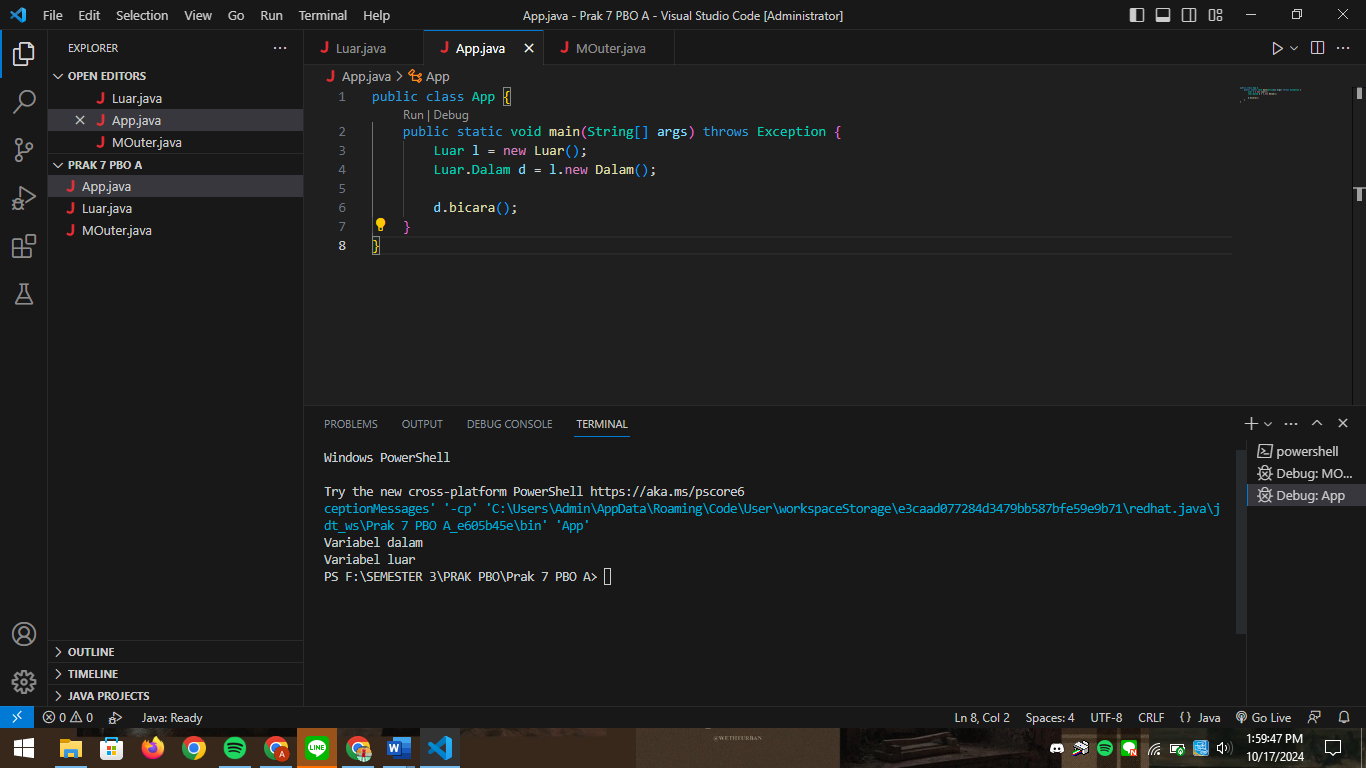
* Nilai m dari outer class.
* Nilai x yang diteruskan ke metode go().
* Nilai y yang juga diteruskan ke metode go().
* Nilai a, hasil penjumlahan x + y.
* Nilai b, hasil pengurangan x - y.

1. Latihan 3

* Kelebihan inner class :
* Inner class memiliki akses langsung ke variabeldan metode dari outer class
* Inner class sangat berguna dalam konteks di mana sebuah kelas hanya masuk akal digunakan bersama dengan outer class-nya.
* Dengan inner class, kita bisa mengimplementasikan beberapa interface atau abstract class secara lokal tanpa harus mendeklarasikan banyak kelas eksternal, yang dapat memperumit struktur program.
* Kelemahan inner class :
* Penggunaan inner class dapat membuat kode lebih sulit dimengerti, terutama bagi pengembang yang baru.
* Inner class terlalu bergantung pada outer class. Ini berarti jika outer class diubah secara drastis, inner class mungkin perlu dimodifikasi juga, yang dapat meningkatkan kerumitan dalam pemeliharaan kode.
* Inner class yang kompleks atau inner class yang memiliki banyak fungsionalitas terkadang lebih sulit untuk diuji secara terpisah. Pengujian menjadi lebih kompleks karena hubungan erat antara inner class dan outer class.

1. Latihan 4

Hasil run Luar.java dan App.java



Hasil run dari MOuter.java

