**LAPORAN PRAKTIKUM PEMROGRAMAN BEORIENTASI OBJEK**

“Tugas 07*”*

****

Oleh:

Nama : Dendy Anugrahi Rabbi

NPM : 4522210034

Kelas : A

Dosen:

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

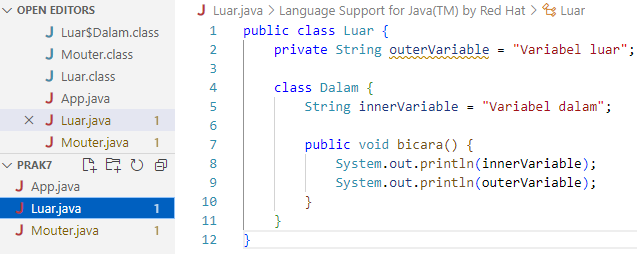
**S1-Teknik Informatika**

**Fakultas Teknik Universitas Pancasila**

**2023/2024**

**LATIHAN 1**

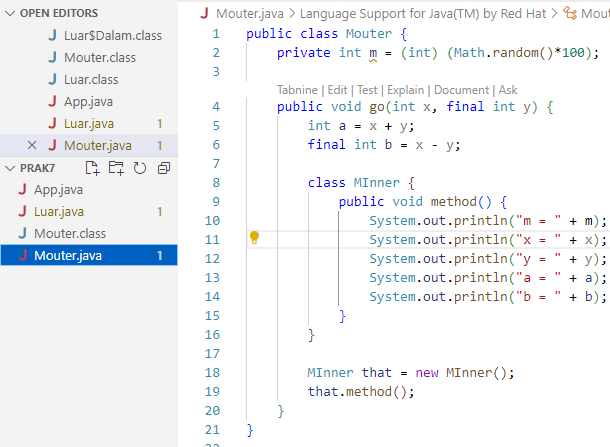
Pada contoh kode dibawah, terdapat dua kelas, yaitu kelas **Luar** (outer class) dan kelas **Dalam** (inner class). Inner class **Dalam** dideklarasikan di dalam kelas **Luar**, dan hal ini memungkinkan **Dalam** untuk mengakses variabel serta metode dari outer class **Luar**, bahkan jika variabel tersebut bersifat private

****

**LATIHAN 2**

Local inner class bisa mengakses:

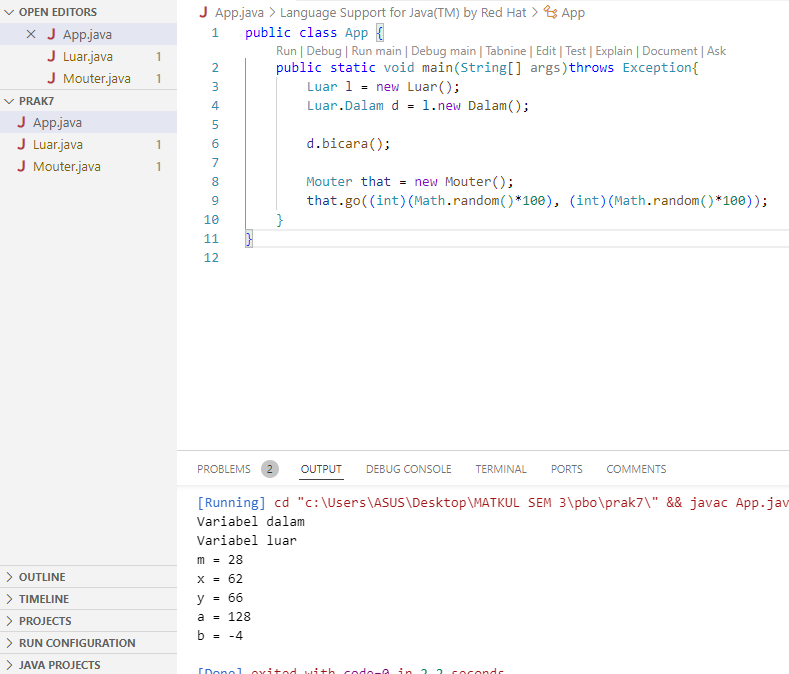
1. m: Karena variabel instance dari kelas luar.
2. x, y, a, b: Karena bersifat final atau effectively final.

Outputnya muncul karena inner class dapat mengakses variabel-variabel tersebut.

**LATIHAN 3**

Inner class memiliki kelebihan seperti dapat mengakses variabel dan metode dari kelas luar, termasuk yang bersifat private, sehingga memudahkan integrasi dan membuat kode lebih terstruktur. Selain itu, inner class mendukung enkapsulasi dengan menyembunyikan detail implementasi, serta cocok untuk pengelompokan kode yang hanya digunakan oleh kelas luar tertentu. Namun, kelemahannya adalah jika digunakan terlalu banyak, kode bisa menjadi kompleks dan sulit dibaca. Inner class juga membawa overhead performa karena menyimpan referensi ke kelas luar, yang dapat memperlambat aplikasi. Selain itu, pengujian inner class sering kali lebih sulit dilakukan secara terpisah karena keterkaitan dengan kelas luar.

**APP & RUN APP**

Kode ini menghasilkan output karena inner class **Dalam** di dalam kelas **Luar** dapat mengakses variabel instance dari kelas luar, meskipun bersifat private, sehingga mencetak "Variabel dalam" dan "Variabel luar". Pada bagian kedua, metode go() dari kelas **Mouter** menggunakan local inner class yang dapat mengakses variabel instance m dari kelas luar serta variabel lokal x, y, a, dan b yang bersifat final atau **effectively final**. Ini menghasilkan output berupa nilai acak yang dicetak untuk variabel-variabel tersebut sesuai hasil perhitungan saat runtime.