**LAPORAN PRAKTIKUM PEMROGRAMAN BEORIENTASI OBJEK**

**“Tugas 07*”***

****

**Oleh:**

Nama : AIDA FITRIA

NPM : 4523210007

Kelas : A

**Dosen:**

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

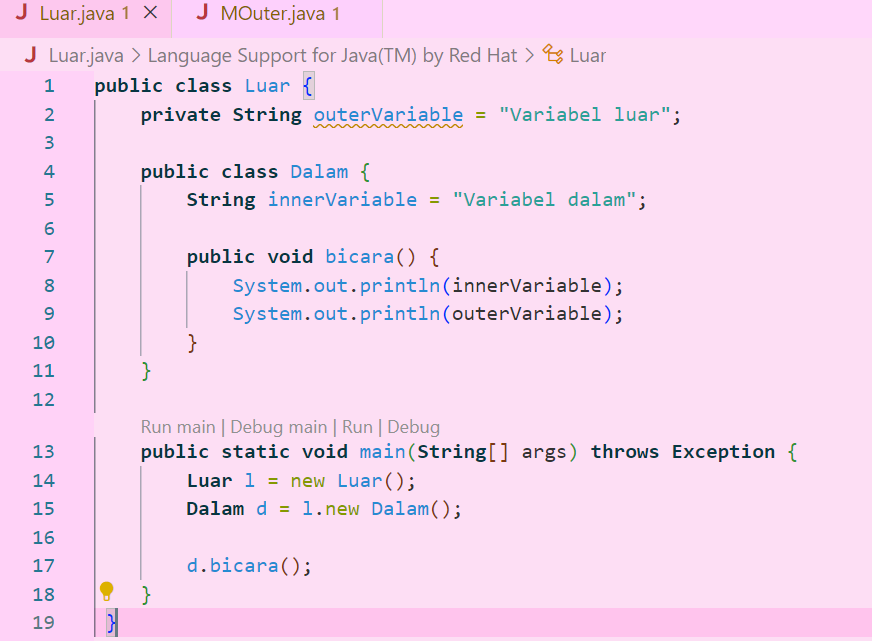
**S1-Teknik Informatika**

**Fakultas Teknik Universitas Pancasila**

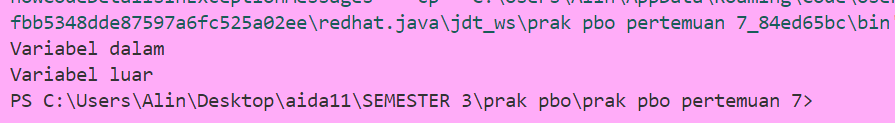
**2024/2025**

1. **Latihan 1**

* **Code Luar.java :**

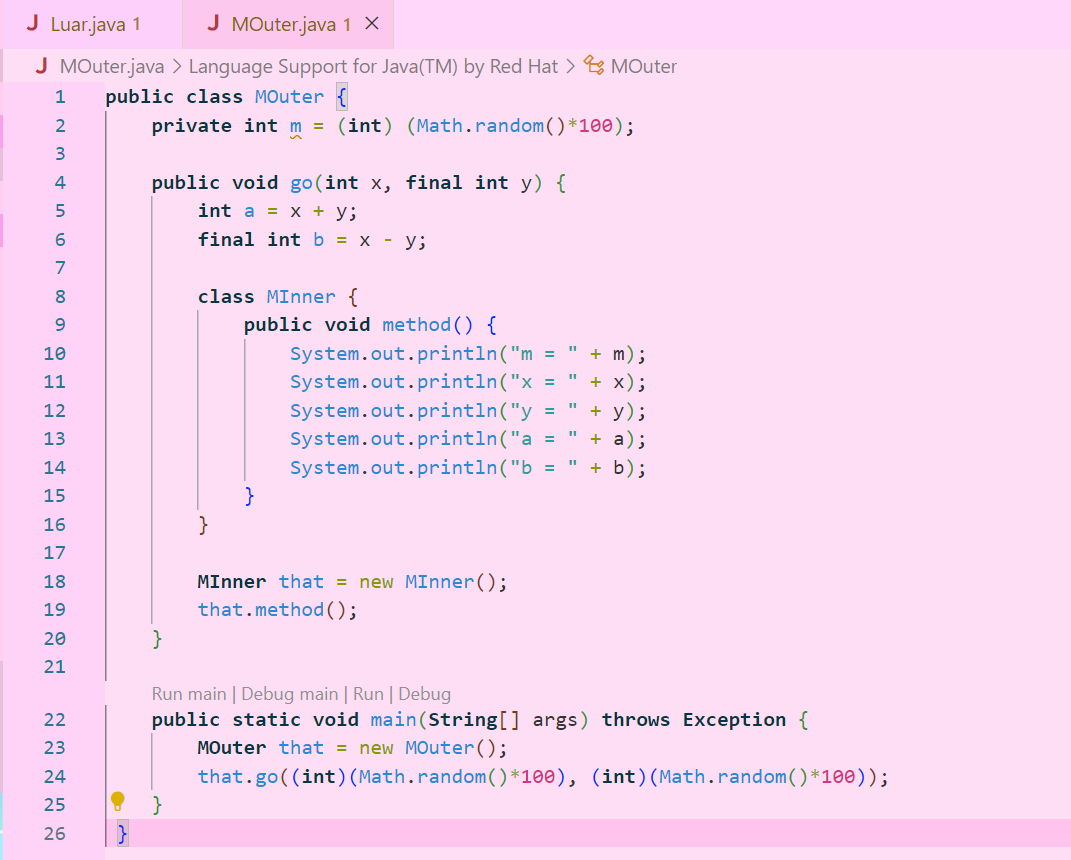


**Hasil Outputnya :**

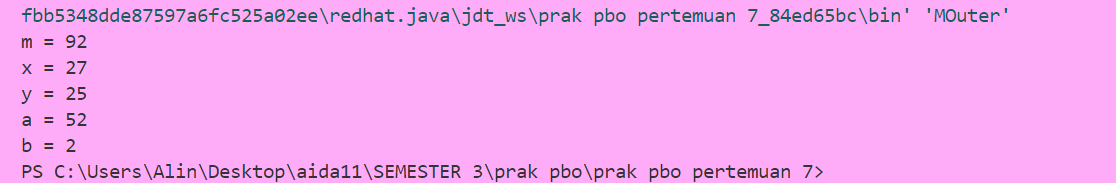


1. **Latihan 2**

* **Code MOuter.java :**



**Hasil outputnya :**



1. **Penjelasan mengapa hasil outputnya seperti diatas karena :**

pada code Luar.java terdapat inner class yang dapat mengakses variable dari outer class, sehingga dari kedua variabel dalam dan luar dicetak,

Sedangkan pada code MOuter.java terdapat inner class yang dapat mencetak variabel acal dan parameter yang diterima oleh metode go() mendapatkan hasil dari beberapa opersi aritmatika yang sederhana.

1. **Jelaskan kelebihan dan kelemahan Inner Class!**

* **Kelebihan**

1. Pemakaian Dengan Sruktur Event-Driven

Yaitu dalam event-driven programming seperti GUI atau Android development inner class sering digunakan sebagai mendefenisikan event handler contohnya seperti aksi tombol, karena inner class dapat merujuk langsung pada komponen outer class tanpa memerlukan banyak referensi.

1. Mendukung Pengguna Anonymous Class

Yaitu inner class mendukung konsep anonymous class yang dapat digunakan dalam situasi pembuatan implementasi interface atau class tertentu yang cepat tanpa harus membuat file class baru.

1. Peningkatan Keterbacaan Dan Organisasi Kode

Yaitu inner class dapat membantu dalam menjaga keterbacaan kode dengan mengelompokkan class yang saling terkait dalam satru tempat.Dan kelas hanya dapat digunakan oleh satu class lain yang dapat dideklarasikan sebagai inner class untuk memperjelas bahwa mereka memeiliki keterkaitan.

* **Kekurangan**

1. Menyebabkan Ukuran Kode Over

Yaitu jika inner class banyak digunakan di satu outer class program dapat lebih Panjang dan sulit diatur,terutama jika inner class terlalu besar hal ini dapat menyebabkan kode menjadi kurang efisien.

1. Meningkatkan Kompleksittas Debugging

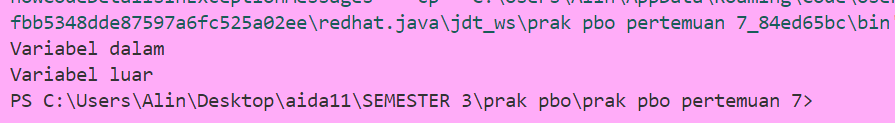
Karena inner class sering berinteraksi dengan anggota private outer class ini dapat menambah lapisan kompleksitas selama proses debugging terutama pada aplikasi yang banyak inner class.

1. Overhead Memori

Yaitu karena inner class bisa menambah overhead memori terutama jika kelas dalam tersebut menyimpan refrerensi yang tidak perlu ke kelas luar,dan penggunaan inner class dapat menyebabkan kebocoran memori (Memory Leaks).

1. **Latihan 4**

Luar.java



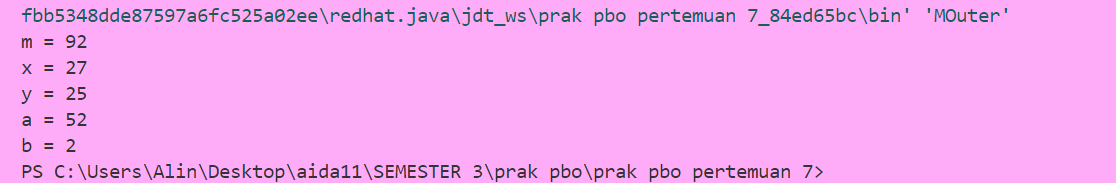
Dalam code contoh Latihan 1 terdapat :

1. Dua kelas yaitu kelas Luar dan kelas Dalam
2. Luar memiliki variabel instance yaitu outerVariabel yang diset dengan nilai (variabel dalam)
3. Didalem kelas luar ad akelas inner Dalam yang memiliki variabel instance yaitu innerVariable yang diset dengan nilai(variabel dalam)
4. Metode bicara() adalah mencetak nilai dari innerVariabel dan outerVariabel

Sedangkan di main()

1. Objek kelas luar dibuat dengan new Luar()
2. Objek dari kelas dalam dibuat menggunakan l.new Dakan(), Dimana l adalah instance dari kelas luar.
3. Ketika d.bicara() dipanggil maka outputnya akan muncul seperti
4. innerVariabel “Variabel dalam” dicetak dari kelas Dalam
5. outerVariabel “Variabel luar” dicetak dari kelas Luar karena kelas inner dapat mengaskses variabel instance dari keras luar.

**MOuter.java**



Dalam code contoh Latihan 2 terdapat :

1. ada kelas MOuter dengan variabel instance m yang diset secara acak menggunakan math.random () dikali 100, dan dicast ke int.
2. Metode go() menerima 2 parameter yaitu x dan y, yang nilai a dihitung sebagai x + y dan nilai b dihitung sebagai x – y .
3. Didalam metode go() terdapat inner class Minner yang mendefenisikan metode method(),Dimana variabel m,x,y,a, dan b dicetak.

Output dari kode tersebut akan bergantung pada nilai acak dari m, nilai x dan y yang diberikan