**LAPORAN PRAKTIKUM PEMROGRAMAN BEORIENTASI OBJEK**

“Tugas 07 – Nested Class”

****

Oleh:

Nama : Daffa Yassar Ahmad

NPM : 4523210032

Kelas : A

Dosen:

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

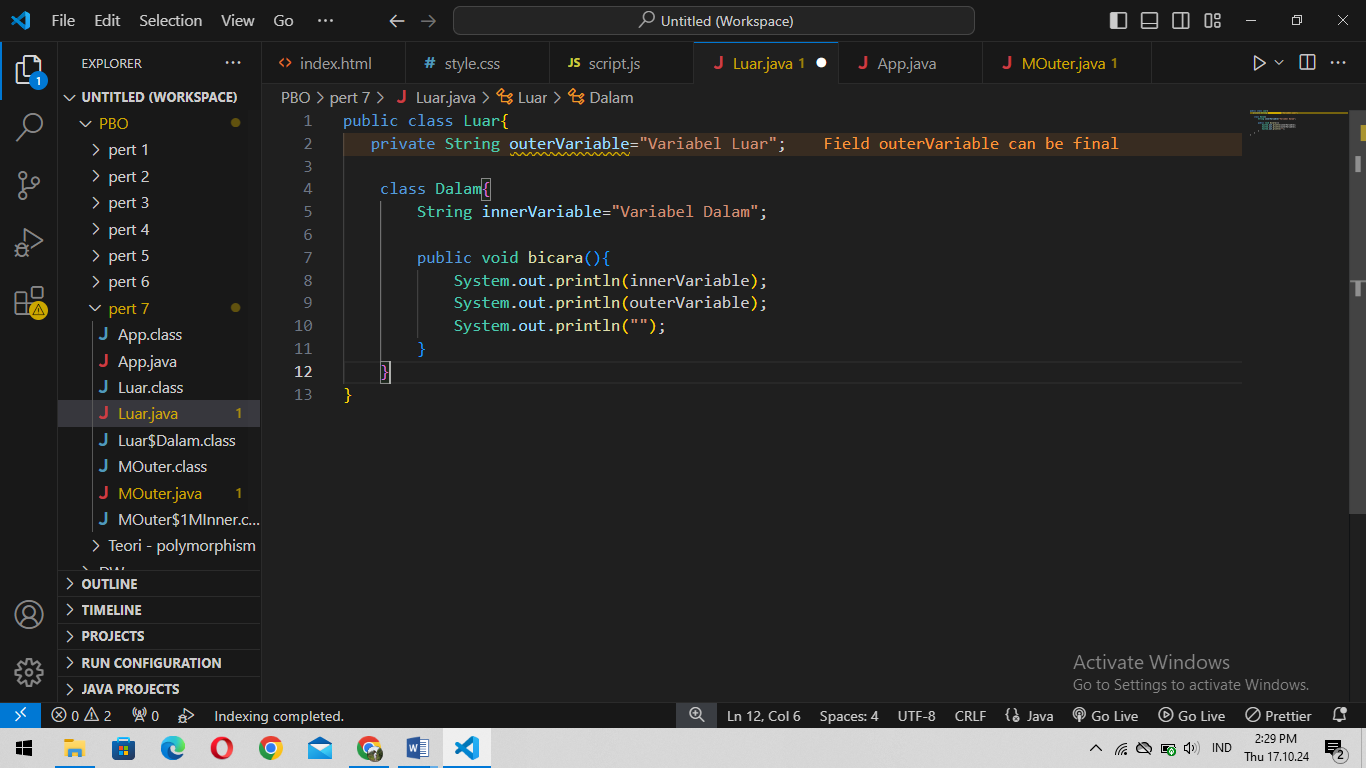
**S1-Teknik Informatika**

**Fakultas Teknik Universitas Pancasila**

**2023/2024**

**LATIHAN 1**

Luar,java



App.java

public class App{

    public static void main(String[] args)throws Exception {

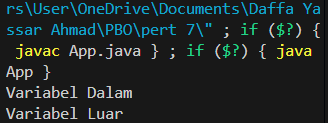
        Luar l = new Luar();

        Luar.Dalam d = l.new Dalam();

        d.bicara();

    }

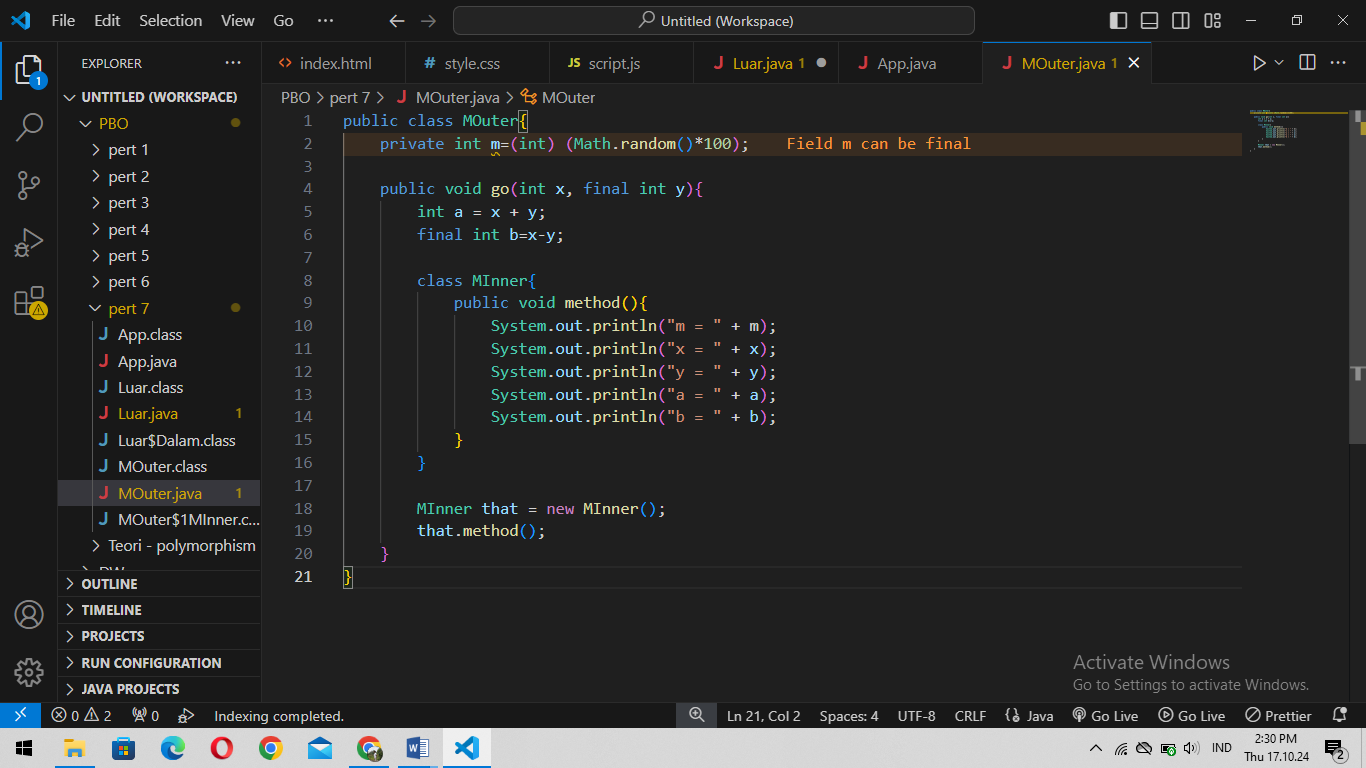
}



**PENJELASAN :** Pada codingan tersebut , Dalam adalah inner class yang bisa mengakses variabel instance outerVariable dari kelas luar Luar. Untuk membuat instance dari inner class, Kita perlu membuat instance dari kelas luar terlebih dahulu. Maka jadilah output tersebut

**LATIHAN 2**

MOuter.java



App.java

public class App{

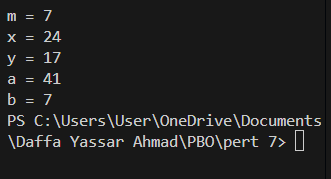
    public static void main(String[] args)throws Exception {

        MOuter that = new MOuter();

        that.go((int)(Math.random()\*100), (int)(Math.random()\*100));

    }

}



**PENJELASAN :** Moueter adalah bagian kelas luar yang memiliki sebuah variabel sebuah variabel instance bernama m, nilai dihasikan tersebut dihasilkan secara acak menggunakan fungsi Math.random() dan dikalikan 100. Nilai ini kemudian di casting ke tipe data int.

Sedangkan variabel m akan menyimpan nilai accak antara 0-99 setiap kali objek dari kelas Mouter dibuat, lalu go bertugas menerima dua parameter yaitu x dan y. Parameter y bersifat final, yang artinya tidak dapat diubah setelah diinisialisasi.

Minner adalah local inner class, yaitu kelas dalam yang dideklarasikan di dalam metode go. Local inner ini cuman dapat diakses dari dalam metode go dan tidak bisa diakses dari luar metode tersebut.Di dalam metode go,intance dari local inner class Minner dibuat dengan memanggil new Minner(), dan metode method() dipanggil pada instance tersebut. Ketika that.method() dipanggil akan mencetak nilai variabel-variabel tersebiut.

**LATIHAN 3**

Kelebihan Inner Class :

1. Enkapulasi menjadi lebih baik dikarenakan kita untuk mengelompokkan kelas yang hanya relavan dalam konteks kelas luar
2. Akses ke anggota kelas luar , misalnya jika inner class ingin memanipulasi atau menggunakan data dari kelas luar . ia dapat melakukannya secara langsung
3. Ketertarikan logis, misalnya dalam aplikasi GUI memungkinkan kita mengatikan event handler dengan elemen UI. Dalam hal ini, event handler bisa diimplementasikan sebagai inner class untuk menunjukkan ketertarikan logis dengan elemen UI
4. Meminimalisasi Namespace
5. Penggunaan di dalam Metode

Kelemahan Inner Class :

1. Meningkatnya kompleksitas kode
2. Ukuran file yang menjadi lebih besar
3. Keterikatan menjadi erat antara kelas luar dan kelas dalam
4. Tidak dapat digunakan diluar kelas luar
5. Aksesibilitas menjadi membingungkan

**LATIHAN 4**

RUN semua contoh latihan pada pertemuan hari ini

