

**LAPORAN TUGAS 2 KELAS DAN OBJEK MATA KULIAH
PEMROGRAMAN BERBASIS OBJEK**

Dosen Pengampu: Siroj Nur Ulum, S.T., M.T.



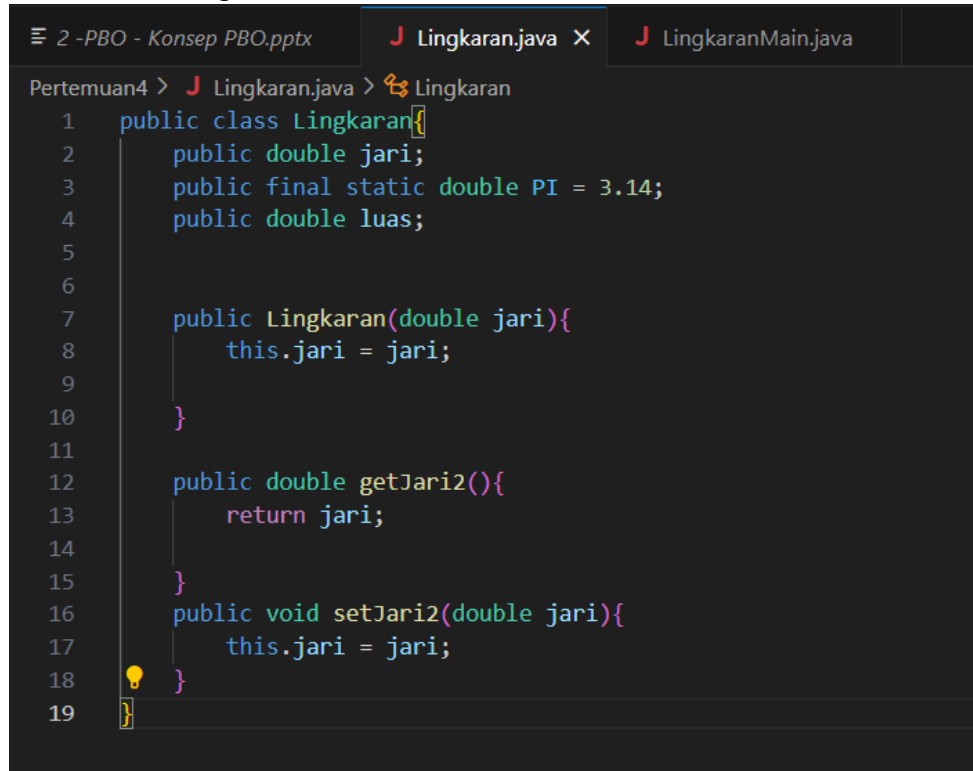
Dipersiapkan Oleh :
Chandra Harkat Raharja (233040089)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PASUNDAN
BANDUNG
2025**

LATIHAN SOAL:

- Konversi contoh-contoh kode program yang ada di slide ke dalam kode program yang bisa di-running
- Screenshotkan hasilnya dan berikan penjelasan mengenai hal tersebut!

Code Untuk Lingkaran:



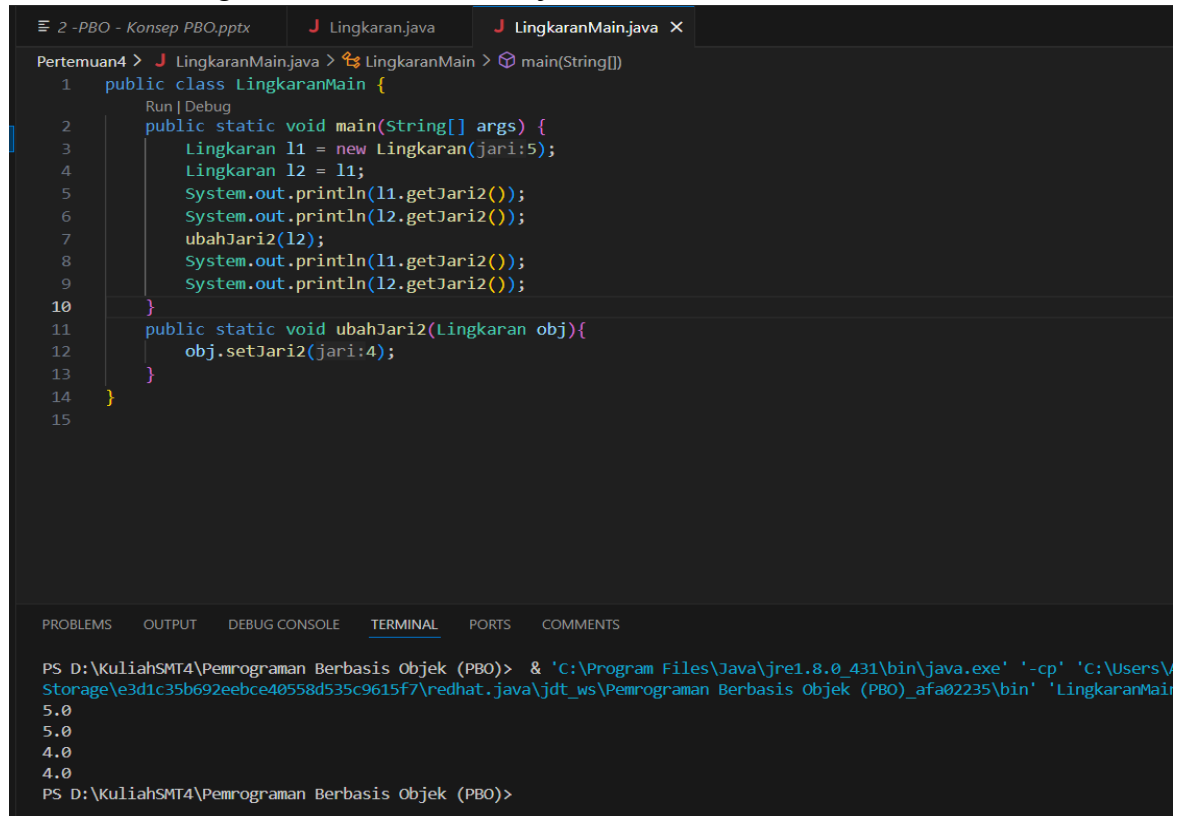
```
2 -PBO - Konsep PBO.pptx  J Lingkaran.java X  J LingkaranMain.java
Pertemuan4 > J Lingkaran.java > J Lingkaran
1 public class Lingkaran{
2     public double jari;
3     public final static double PI = 3.14;
4     public double luas;
5
6
7     public Lingkaran(double jari){
8         this.jari = jari;
9     }
10
11
12     public double getJari2(){
13         return jari;
14     }
15
16     public void setJari2(double jari){
17         this.jari = jari;
18     }
19 }
```

Penjelasan:

Kode dimulai dengan deklarasi kelas `Lingkaran` yang memiliki tiga atribut bertipe `double`: `jari`, `PI`, dan `luas`. Atribut `jari` digunakan untuk menyimpan nilai jari-jari lingkaran, sedangkan `PI` adalah konstanta dengan nilai 3.14 yang bersifat `final static`, artinya nilainya tetap dan tidak dapat diubah. Atribut `luas` disiapkan untuk menyimpan hasil perhitungan luas lingkaran.

Selanjutnya, terdapat konstruktor `public Lingkaran(double jari)` yang berfungsi untuk menginisialisasi nilai `jari` saat objek `Lingkaran` pertama kali dibuat. Kelas ini juga memiliki dua metode utama, yaitu `getJari2()` dan `setJari2(double jari)`. Metode `getJari2()` berfungsi untuk mengembalikan nilai jari-jari yang tersimpan dalam atribut `jari`, sedangkan metode `setJari2(double jari)` digunakan untuk mengubah nilai atribut `jari` dengan nilai baru yang diberikan sebagai parameter.

Code Untuk LingkaranMain, Beserta Penjelasan dan bukti run:



```
Pertemuan4 > J LingkaranMain.java > Run | Debug
1 public class LingkaranMain {
2     public static void main(String[] args) {
3         Lingkaran l1 = new Lingkaran(jari:5);
4         Lingkaran l2 = l1;
5         System.out.println(l1.getJari2());
6         System.out.println(l2.getJari2());
7         ubahJari2(l2);
8         System.out.println(l1.getJari2());
9         System.out.println(l2.getJari2());
10    }
11    public static void ubahJari2(Lingkaran obj){
12        obj.setJari2(jari:4);
13    }
14 }
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\KuliahSMT4\Pemrograman Berbasis Objek (PBO)> & 'C:\Program Files\Java\jre1.8.0_431\bin\java.exe' '-cp' 'C:\Users\
Storage\3d1c35b692eebce40558d535c9615f7\redhat.java\jdt_ws\Pemrograman Berbasis Objek (PBO)_afa02235\bin' 'LingkaranMai
5.0
5.0
4.0
4.0
PS D:\KuliahSMT4\Pemrograman Berbasis Objek (PBO)>
```

Penjelasan:

Kode dimulai dengan membuat objek Lingkaran bernama l1 dengan jari-jari 5. Kemudian, referensi l2 diberikan nilai yang sama dengan l1, sehingga keduanya menunjuk ke objek yang sama di memori. Saat nilai jari-jari dicetak menggunakan getJari2(), hasilnya adalah 5 untuk l1 dan l2.

Selanjutnya, metode ubahJari2(l2) dipanggil, di mana di dalamnya nilai jari-jari diubah menjadi 4 melalui setJari2(4). Karena l1 dan l2 menunjuk ke objek yang sama, perubahan ini berdampak pada keduanya.

Setelah pemanggilan metode tersebut, pencetakan nilai jari-jari menunjukkan bahwa baik l1 maupun l2 kini memiliki nilai 4. Hal ini menunjukkan bahwa perubahan pada satu referensi juga mempengaruhi referensi lain yang menunjuk ke objek yang sama. Ini mengilustrasikan bagaimana referensi objek bekerja dalam Java; ketika beberapa variabel referensi menunjuk ke objek yang sama, perubahan melalui salah satu referensi akan terlihat oleh semua referensi tersebut. Bukti Run sudah tercantum di dalam gambar.