**Laporan Resmi**

1. Buatlah UML class diagram dari percobaan 3!

Jawaban :

|  |
| --- |
| acaraTV |
| -StasiunTV:String  -Acara:String  +getStasiunTV():String  +setStasiunTV(StasiunTV:String)  +getAcara():String  +setAcara(Acara:String)  +cetak() |

1. Apa perbedaan penentu akses privat dan publik! Berilah contoh program untuk memperjelas penjelasan.

Jawaban : Perbedaan penentu akses privat dan publik adalah pada cara aksesnya. Untuk akses publik dapat diakses dimana aja termasuk di luar kelas tersebut. Untuk akses privat cara mengaksesnya sangat terbatas, yaitu hanya dapat diakses di kelasnya. Untuk mengakses privat di kelas lain hanya dapat diakses melalui metode getter dan setter, sehingga data aman tidak dapat diubah melalui program utama.

Contoh program :

class Data {

private String dataRahasia = "Ini adalah data rahasia"; // hanya bisa diakses dalam kelas Data

public String dataUmum = "Ini adalah data umum"; // bisa diakses dari mana saja

// Metode untuk mengakses dataRahasia

public String getDataRahasia() {

return dataRahasia;

}

// Metode untuk mengubah dataRahasia

public void setDataRahasia(String dataBaru) {

this.dataRahasia = dataBaru;

}

}

public class AksesModifier {

public static void main(String[] args) {

Data obj = new Data();

// Mengakses data public langsung

System.out.println("Akses langsung ke public: " + obj.dataUmum);

// Tidak bisa mengakses langsung variabel private

// System.out.println(obj.dataRahasia); // ERROR

// Menggunakan metode public untuk mengakses private

System.out.println("Akses private melalui metode: " + obj.getDataRahasia());

// Mengubah nilai private melalui metode setter

obj.setDataRahasia("Data rahasia telah diubah");

System.out.println("Setelah perubahan: " + obj.getDataRahasia());

}

}

**Penjelasan Program**

1. *dataRahasia* bersifat private, sehingga hanya bisa diakses melalui metode *getDataRahasia()* dan *setDataRahasia()*.
2. *dataUmum* bersifat public, sehingga bisa langsung diakses dari luar kelas.
3. Program menggunakan metode *getter* dan *setter* untuk mengakses dan mengubah variabel private, menjaga prinsip enkapsulasi.
4. Kapan kata kunci this dipergunakan?

Jawaban : kata kunci this digunakan pada saat nama variabel sama seperti nama pada parameter sehingga memerlukan this agar tidak terjadi kesalahan saat menjalankan program’

Contoh program :

class Contoh {

int x;

Contoh(int x) {

this.x = x; // 'this.x' adalah variabel instance, sedangkan 'x' adalah parameter

}

}