**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

}

}

**Analisa**

Pada praktikum kali ini membahas program berorientasi objek yaitu akses terhadap suatu variabel instan diluar kelas biasanya tidak diperkenankan. Sebagai penggantinya, disediakan metode yang diperlukan untuk mengakses variable instant. Berkaitan dengan boleh/tidaknya suatu variable instant diakses dari luar kelas ,Java menyediakan penentu akses. Dua diantara penentu akses yang tersedia adalah private dan public.

* public berarti bahwa pengaksesan statu variable instan atau metode dapat dilakukan dari luar kelas
* private berarti bahwa pengaksesan statu variable instan atau metode hanya dapat dilakukan didalam kelas;tidak bisa diakses dari luar kelas

Kata kunci this sangat berguna untuk menunjukkan suatu member dalam classnya sendiri. This dapat digunakan baik untuk data member maupun untuk function member, serta dapat juga digunakan untuk konstruktor.

Pada tugas pendahuluan yang pertama diminta untuk mengamati dan tampilkan dari program, yang mana keluaran dari program tersebut adalah 5. Untuk tugas pendahuluan kedua diminta untuk mengamati dan menampilkan hasil dari program, dan untuk keluaran dari program tersebut adalah 5 Budi.

Pada tugas percobaan yang pertama diminta untuk mengimplementasikan UML menjadi dalam bentuk listing program yaitu :

class Buku {

String judul;

String pengarang;

}

Untuk percobaan kedua mengubah UML class diagram ke dalam bentuk listing program dengan hasil :

class Mahasiswa {

private int nrp;

private String nama;

public void setNrp(int i) {

nrp = i;

}

public void setNama(String i) {

nama = i;

}

public int getNrp() {

return nrp;

}

public String getNama() {

return nama;

}

}

Pada percobaan ketiga terjadi kesalahan saat kompilasi dikarenakan pada class utama memanggil variabel privat class lain, sehingga terjadi kesalahan kompilasi. Sehingga untuk perbaikan programnya dapat dilakukan dengan mengganti perintahnya menjadi setter agar dapat mengubah variabel di class lain. Sehingga perbaikan di class utama :

public class percobaan3 {

public static void main(String[] args) {

acaraTV a = new acaraTV();

a.setStasiunTV ("Indosiar");

a.setAcara("Sinetron");

a.cetak();

}

}

**Kesimpulan**

Pada praktikum Pemrograman Berorientasi Objek ini Mahasiswa diharapkan dapat memahami konsep pemrograman berorientasi objek, menciptakan kelas, membuat objek dari suatu kelas, mengakses variabel dan method dari suatu kelas, kata kunci this, penentu akses public dan privat, kata kunci static.