

Nama : Hanifa Salsabila

NIM : 2100018466

Kelas : I

URL Git : <https://github.com/hanifa000/Tugas-PBO-2>

Petunjuk: Realisasi Class Diagram dalam kode program anda dan tunjukkan realisasi tersebut berdasarkan beberapa hal berikut:

Catatan: jika diperlukan boleh gunakan perintah komentar di kode program untuk menjelaskan kode program dan tunjukkan kelas main untuk mensimulasikan realisasi.

Komponen Dasar PBO

- Daftar Objek, daftar property/state, daftar behavior

NO	Object	Variabel	Method
1.	Komputer	<ul style="list-style-type: none">• merk: String,• kecProsesor : int,• sizeMemory: int,• jnsProsesor : String	<ul style="list-style-type: none">• <u>Komputer()</u>• Komputer(merk: String, kecProsesor : int, sizeMemory: int, jnsProsesor : String)• <u>tampilData</u>: void
2.	Pc	<ul style="list-style-type: none">• ukuranMonitor : int	<ul style="list-style-type: none">• <u>Pc()</u>• Pc(merk: String, kecProsesor : int, sizeMemory: int, jnsProsesor : String, ukuranMonitor: int)• tampilPc() : void
3.	Laptop	<ul style="list-style-type: none">• jnsBatrai: String	<ul style="list-style-type: none">• <u>Laptop()</u>• Laptop(merk: String, kecProsesor : int, sizeMemory: int, jnsProsesor : String, jnsBatrai: String)• tampilLaptop(): void
4.	Mac	<ul style="list-style-type: none">• security: String	<ul style="list-style-type: none">• Mac()• Mac(merk: String, kecProsesor : int, sizeMemory: int, jnsProsesor : String, security: String)• tampilMac() : void
5.	Windows	<ul style="list-style-type: none">• fitur: String	<ul style="list-style-type: none">• Windows()• Windows(merk: String, kecProsesor : int, sizeMemory: int, jnsProsesor : String, fitur: String)• tampilWindows() : void

- Pembuatan Class (tidak perlu dengan isi class karena akan dijelaskan masing-masing)

```
public class Main {
    public Komputer() {}
```

- Variable (tipe data, access modifier, nama variable)

```
public String merk, jnsProsesor;
private int kecProsesor, sizeMemory;
private int ukuranMonitor;
public String jnsBatrei;
public String fitur;
private String security;
```

- Method

- o Method berupa Function

```
static void decor() {
    System.out.println("+++++");
}
Komputer kom = new Komputer();
kom.decor();
```

- o Method berupa Procedure

```
public void tampilData()
public void tampilPc()
public void tampilLaptop()
public void tampilWindows()
public final void tampilMac()
```

- o Method berupa Setter

```
public void setUkuranMonitor(int ukuranMonitor) {
    this.ukuranMonitor = ukuranMonitor;
}
```

- o Method berupa Getter

```
public int getUkuranMonitor() {
    return ukuranMonitor;
}
```

- o Method berupa Boolean properti

```
boolean running = true;
int counter = 0;
String jawab;
Scanner scn = new Scanner(System.in);

while( running ) {
    System.out.println("Apakah anda ingin keluar?");
    System.out.print("Jawab [ya/tidak]> ");

    jawab = scn.nextLine();

    // cek jawabannya, kalau ya maka berhenti mengulang
    if( jawab.equalsIgnoreCase("ya") ){
        running = false;
    }
}
```

```

        counter++;
    }

    System.out.println("Anda sudah melakukan perulangan sebanyak " +
        counter + " kali");
}

```

- Method berupa non-boolean properti

- Method dengan parameter

```

public Komputer(String merk, int kecProsesor, int sizeMemory, String
jnsProsesor)
public Pc(String merk, int kecProsesor, int sizeMemory, String jnsProsesor)

public Laptop(String merk, int kecProsesor, int sizeMemory, String
jnsProsesor, String jnsBatrei)
public Mac(String merk, int kecProsesor, int sizeMemory, String
jnsProsesor, String jnsBatrei, String security)

```

- Method dengan Argument Lists

- Method static

```

public static void main(String[] args)
static void decor()

```

- Method final

```

public final void tampilMac()

```

Konstruktor

- Constructor

```

public Komputer(){}
public Laptop(String merk, int kecProsesor, int sizeMemory, String
jnsProsesor, String jnsBatrei){
    super(merk, kecProsesor, sizeMemory, jnsProsesor);
    this.jnsBatrei=jnsBatrei;
}

```

- Constructor Overloading

```

public Windows(String merk, int kecProsesor, int sizeMemory, String
jnsProsesor, String jnsBatrei){
    super(merk, kecProsesor, sizeMemory, jnsProsesor, jnsBatrei);
}

public Windows(String merk, int kecProsesor, int sizeMemory, String
jnsProsesor, String jnsBatrei, String fitur){
    super(merk, kecProsesor, sizeMemory, jnsProsesor, jnsBatrei);
    this.fitur=fitur;
    System.out.println("Fitur          : "+fitur);
}

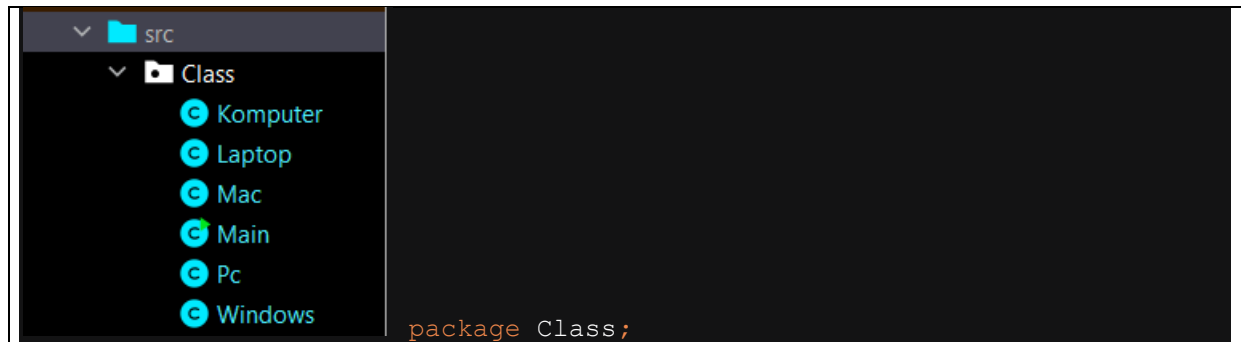
```

```
}
```

- Instansiasi dengan menggunakan dua jenis konstruktor di kelas main

Enkapsulasi dan access modifier

- Paket



- Praktek yang menunjukkan enkapsulasi dengan private di kelas main

```
private int ukuranMonitor;  
p.setUkuranMonitor(24);  
System.out.println("Ukuran monitor : "+p.getUkuranMonitor());
```

- Praktek yang menunjukkan enkapsulasi dengan protected di kelas main

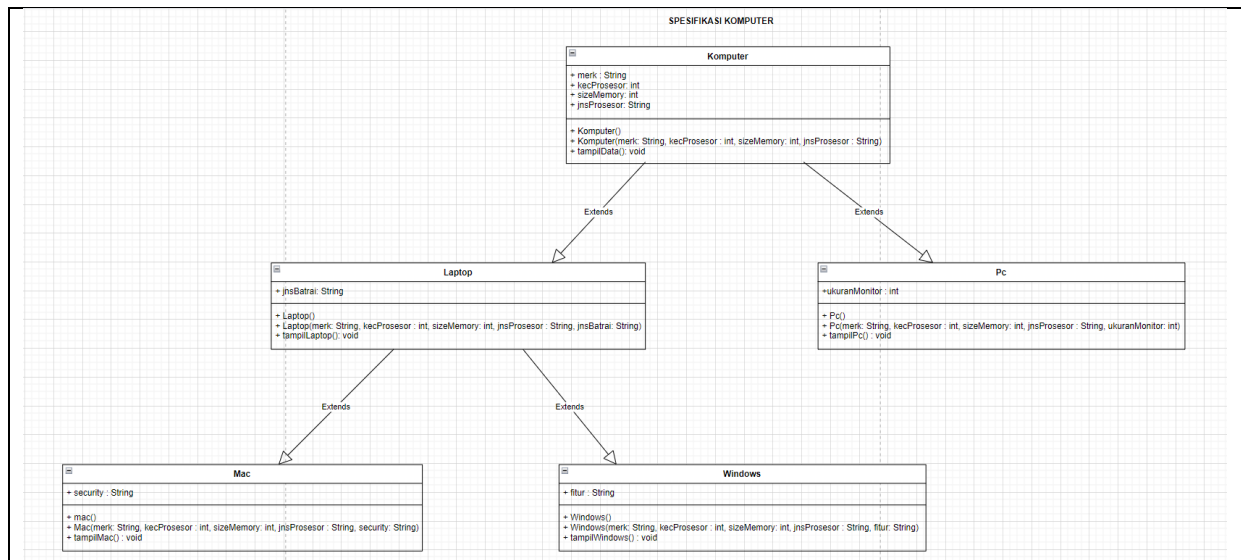
Pewarisan

- Pembuatan perwarisan

```
public class Pc extends Komputer{}  
public class Laptop extends Komputer{}  
public class Windows extends Laptop{}  
public class Mac extends Laptop{}  
}
```

- Pembuatan ancestor class atau kelas yang memiliki banyak kelas orang tua/kelas leluhur

- Praktek yang menunjukkan dan penjelasan mengapa harus dilakukan Pewarisan



Karena komputer memiliki dua pembagian yaitu Pc dan Laptop, sedangkan Laptop memiliki anak yaitu Mac, Windows dan masih banyak lagi pembagiannya

Polymorphisme

- Method Overloading

```

public Windows(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, String jnsBatrai){
    super(merk, kecProsesor, sizeMemory, jnsProsesor, jnsBatrai);
}

public Windows(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, String jnsBatrai, String fitur){
    super(merk, kecProsesor, sizeMemory, jnsProsesor, jnsBatrai);
    this.fitur=fitur;
    System.out.println("Fitur          : "+fitur);
}
  
```

- Method Overriding

- Overloading atau Overriding pada Method static

- Overloading atau Overriding pada Method final

- Overloading atau Overriding dengan perubahan return type

- Overloading atau Overriding dengan perubahan access modifier (public, protected, dan private)

--