

Nama : Hanifa Salsabila

NIM : 2100018466

Kelas : I

URL Git :

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)• <u>tampilPc()</u> : 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)• <u>tampilLaptop()</u>: 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)• <u>tampilMac()</u> : 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)• <u>tampilWindows()</u> : void

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

```
public class Main {}_ public class Komputer {}_
```

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

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

- Method

- o Method berupa Function

```
4 usages
static void decor() {
    System.out.println("++++++++++++++++++++++++++++++++++++++++");
}

System.out.println("*****Data Laptop*****");
Komputer kom = new Komputer();
kom.decor();
```

- o Method berupa Procedure

```
2 usages
public void tampilData(){
1 usage
public void tampilPc(){

2 usages
public void tampilLaptop(){
1 usage
public void tampilWindows(){

1 usage
public void tampilMac(){
```

- o Method berupa Setter

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

```
setUkuranMonitor(24);
```

- Method berupa Getter

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

```
System.out.println("Ukuran monitor : "+getUkuranMonitor());
```

- Method berupa Boolean properti

- Method berupa non-boolean properti

- Method dengan parameter

```
2 usages
public Komputer(String merk, int kecProsesor, int sizeMemory, String jnsProsesor){
}

1 usage
public Pc(String merk, int kecProsesor, int sizeMemory, String jnsProsesor){
}
```

- Method dengan Argument Lists

- Method static

```
public static void main(String[] args) {
```

```
4 usages
static void decor() {
```

- Method final

```
public final void tampilMac(){
    System.out.println("=====Laptop Mac=====");
    super.tampilLaptop();
    System.out.println("Security          : "+security);
}
```

Konstruktor

- Constructor

2 usages

```
public Laptop(){}
```

1 usage

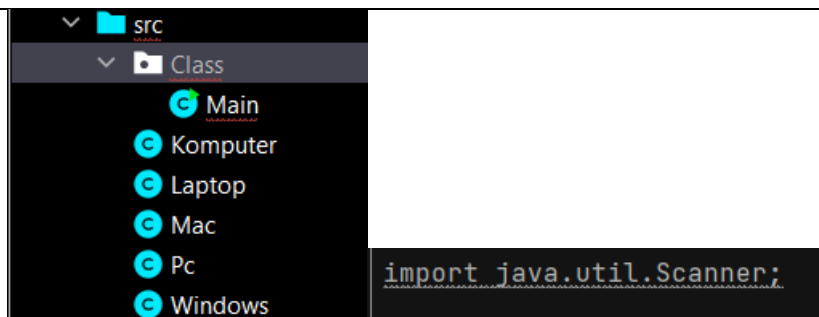
```
public Mac(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, String jnsBatrei, String security)
{
    super(merk, kecProsesor, sizeMemory, jnsProsesor, jnsBatrei);
    this.security=security;
}
```

- Constructor Overloading

- Instansiasi dengan menggunakan dua jenis konstruktor di kelas main

Enkapsulasi dan access modifier

- Paket



- Praktek yang menunjukkan enkapsulasi dengan private di kelas main

- Praktek yang menunjukkan enkapsulasi dengan protected di kelas main

Pewarisan

- Pembuatan perwarisan

```
2 usages
public class Pc extends Komputer {
    2 usages

public class Laptop extends Komputer{
    2 usages

public class Windows extends Laptop{
    2 usages

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

Polymorphisme

- Method Overloading

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

1 usage
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);
}
```

- 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)

--