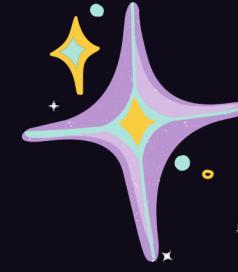


TUGAS PROJEK

Pemrograman Berorientasi Objek





Anggota Kelompok :

- 1. Anjar Tejakusumah**
- 2. Muhammad Hafizh Albar**
- 3. Muhammad Rizal Nurfirdaus**



Pewarisan (Inheritance)

Pewarisan (inheritance) adalah konsep dalam Pemrograman Berorientasi Objek (PBO) di mana sebuah kelas (subclass) mewarisi atribut dan metode dari kelas lain (superclass). Ini memungkinkan pengurangan duplikasi kode, peningkatan reusabilitas, dan pembuatan struktur hierarki yang logis. Dalam program yang Anda berikan, kelas `Laptop` dan `Handphone` mewarisi properti dan metode dari kelas `Electronik`, memungkinkan mereka untuk memperluas fungsionalitas dasar dari `Electronik` dengan menambahkan spesifikasi khusus masing-masing, seperti ukuran layar, CPU, RAM, dan lainnya, sambil tetap memanfaatkan kode yang ada di `Electronik`.

Polymorphism

Polymorphism adalah konsep dalam Pemrograman Berorientasi Objek (PBO) yang memungkinkan metode yang sama untuk berperilaku berbeda berdasarkan objek yang memanggilnya, baik melalui method overriding (runtime polymorphism) maupun method overloading (compile-time polymorphism). Dalam program Anda, ini terlihat saat metode `informasi()` di-override di kelas `Laptop` dan `Handphone`, sehingga saat dipanggil melalui referensi kelas `Electronik`, implementasi spesifik dari kelas `Laptop` atau `Handphone` yang dijalankan. Polymorphism memberikan fleksibilitas, reusabilitas, dan kemudahan pemeliharaan dengan memungkinkan satu metode untuk bekerja dengan berbagai tipe objek.

Overloading Method

Overloading method adalah konsep dalam Pemrograman Berorientasi Objek (PBO) di mana beberapa metode dalam satu kelas memiliki nama yang sama tetapi berbeda dalam parameter (jumlah atau tipe parameter). Ini memungkinkan metode untuk berfungsi dengan cara yang berbeda tergantung pada argumen yang diterima. Overloading method meningkatkan fleksibilitas kode dan membuatnya lebih intuitif, karena pengguna dapat memanggil metode dengan berbagai cara tanpa perlu mengingat nama metode yang berbeda. Sebagai contoh, dalam kelas `Electronik`, metode `informasi()` di-overload dengan satu versi yang tidak menerima parameter dan versi lain yang menerima parameter `String prefix`, memungkinkan pengguna untuk memilih bagaimana mereka ingin menampilkan informasi.

Overriding Method

Overriding method adalah konsep dalam Pemrograman Berorientasi Objek (PBO) di mana subclass menyediakan implementasi spesifik untuk metode yang sudah didefinisikan di superclassnya. Ini memungkinkan subclass untuk menyesuaikan atau memperluas perilaku metode yang diwarisi dari superclass. Method overriding meningkatkan polymorphism, karena memungkinkan objek dari subclass untuk menjalankan metode yang sesuai dengan implementasinya sendiri saat dipanggil melalui referensi superclass. Misalnya, dalam program Anda, kelas `Laptop` dan `Handphone` meng-override metode `informasi()` dari kelas `Electronik`, sehingga masing-masing kelas menampilkan informasi yang spesifik untuk jenis perangkat tersebut saat metode tersebut dipanggil.

Constructor

Constructor adalah metode khusus dalam sebuah kelas yang digunakan untuk menginisialisasi objek baru dari kelas tersebut. Constructor dipanggil secara otomatis ketika sebuah objek dibuat, dan biasanya digunakan untuk menetapkan nilai awal pada atribut objek atau menjalankan inisialisasi yang diperlukan. Constructor memiliki nama yang sama dengan namakelasnya dan tidak memiliki tipe kembalian (return type). Dalam program Anda, constructor di kelas `Electronik` menerima parameter `merek`, `model`, dan `harga` untuk menginisialisasi atribut tersebut, sementara subclass seperti `Laptop` dan `Handphone` memiliki constructor yang memanggil constructor superclass menggunakan keyword `super` untuk memastikan bahwa atribut dari superclass juga diinisialisasi dengan benar.

Berikut kode programnya

rizal > src > main > java > com > example > projek >  Electronik.java >  Electronik

```
1 package com.example.projek;
2
3 public class Electronik {
4     private String merek, model;
5     private int harga;
6
7     public Electronik(String merek, String model, int harga) {
8         this.merek = merek;
9         this.model = model;
10        this.harga = harga;
11    }
12
13    public void informasi() {
14        System.out.println("Merek : " + merek);
15        System.out.println("Model : " + model);
16        System.out.println("Harga : " + harga);
17    }
18
19    public void informasi(String prefix) {
20        System.out.println(prefix + "Merek : " + merek);
21        System.out.println(prefix + "Model : " + model);
22        System.out.println(prefix + "Harga : " + harga);
23    }
24}
```

Analisis Kode Programnya

1. Enkapsulasi:

- Atribut merek, model, dan harga disembunyikan (**private**) dan hanya bisa diakses melalui metode kelas (**informasi**).

2. Konstruktor:

- Konstruktor digunakan untuk menginisialisasi objek dengan nilai merek, model, dan harga.

3. Polimorfisme (Overloading):

- Metode informasi di-overload dengan dua versi: satu tanpa parameter dan satu dengan parameter String (**prefix**).

Implementasi Konsep PBO

- Enkapsulasi: Data disembunyikan dengan **private** dan diakses melalui metode **publik**.
- Overloading: Metode informasi menyediakan dua versi untuk fleksibilitas dalam mencetak informasi.

Berikut kode programnya

The screenshot shows a Java code editor with the following details:

- File Tabs:** Electronik.java, Laptop.java (highlighted with an orange border), Handphone.java, Main.java.
- Path Bar:** rizal > src > main > java > com > example > projek > Laptop.java
- Code Content (Laptop.java):**

```
1 package com.example.projek;
2
3 class Laptop extends Electronik {
4     private double ukuranLayar;
5     private String CPU;
6     private String VGA;
7     private int RAM, storage;
8     public Laptop(String merek, String model, int harga, double ukuranLayar, String CPU, String VGA, int RAM, int storage) {
9         super(merek, model, harga);
10        this.ukuranLayar = ukuranLayar;
11        this.CPU = CPU;
12        this.VGA = VGA;
13        this.RAM = RAM;
14        this.storage = storage;
15    }
16    public void informasi() {
17        super.informasi();
18        System.out.println("Ukuran Layar : " + ukuranLayar + " Inci");
19        System.out.println("CPU : " + CPU);
20        System.out.println("VGA : " + VGA);
21        System.out.println("RAM : " + RAM + " GB");
22        System.out.println("Storage : " + storage + " GB\n");
23    }
24    public void informasi(String prefix) {
25        super.informasi(prefix);
26        System.out.println(prefix + "Ukuran Layar : " + ukuranLayar + " Inci");
27        System.out.println(prefix + "CPU : " + CPU);
28        System.out.println(prefix + "VGA : " + VGA);
29        System.out.println(prefix + "RAM : " + RAM + " GB");
30        System.out.println(prefix + "Storage : " + storage + " GB\n");
31    }
32 }
```

Analisis Kode Programnya

Program ini mengilustrasikan kelas `Laptop` yang merupakan turunan dari kelas `Electronik`, menunjukkan konsep pewarisan dan overloading dalam Java. Kelas `Laptop` memiliki atribut tambahan seperti `ukuranLayar`, `CPU`, `VGA`, `RAM`, dan `storage`, yang diinisialisasi melalui konstruktor. Konstruktor tersebut juga memanggil konstruktor superclass untuk menginisialisasi atribut yang diwarisi. Terdapat dua metode `informasi` yang diimplementasikan: satu tanpa parameter dan satu dengan parameter `prefix`. Keduanya memanggil metode `informasi` dari superclass untuk mencetak informasi dasar, kemudian menambahkan informasi spesifik laptop. Penggunaan `super` memastikan atribut dan metode dari kelas induk digunakan dengan benar, sehingga menggambarkan penerapan konsep inheritance dan method overloading untuk memperluas fungsionalitas secara terstruktur dan modular.

Berikut kode programnya

The screenshot shows a Java code editor interface with the following details:

- File Tabs:** Electronik.java, Laptop.java, Handphone.java (highlighted with an orange border), Main.java.
- Project Path:** rizal > src > main > java > com > example > projek > Handphone.java
- Code Content:** The code defines a class Handphone that extends Electronik. It has private fields for layar (double), CPU (String), baterai, RAM, and storage (int). A constructor initializes these fields. The class contains two methods: informasi() and informasi(String prefix) which both print the device's specifications to the console using System.out.println.

```
1 package com.example.projek;
2
3 class Handphone extends Electronik {
4     private double layar;
5     private String CPU;
6     private int baterai, RAM, storage;
7
8     public Handphone(String merek, String model, int harga, double layar, String CPU, int baterai, int RAM, int storage) {
9         super(merek, model, harga);
10        this.layar = layar;
11        this.CPU = CPU;
12        this.baterai = baterai;
13        this.RAM = RAM;
14        this.storage = storage;
15    }
16
17    public void informasi() {
18        super.informasi();
19        System.out.println("Ukuran Layar : " + layar + " Inci");
20        System.out.println("CPU : " + CPU);
21        System.out.println("Baterai : " + baterai + " mAh");
22        System.out.println("RAM : " + RAM + " GB");
23        System.out.println("Storage : " + storage + " GB\n");
24    }
25
26    public void informasi(String prefix) {
27        super.informasi(prefix);
28        System.out.println(prefix + "Ukuran Layar : " + layar + " Inci");
29        System.out.println(prefix + "CPU : " + CPU);
30        System.out.println(prefix + "Baterai : " + baterai + " mAh");
31        System.out.println(prefix + "RAM : " + RAM + " GB");
32        System.out.println(prefix + "Storage : " + storage + " GB\n");
33    }
}
```

Analisis Kode Programnya

Kode program ini merupakan implementasi sederhana dari konsep pewarisan (inheritance) dan overloading method dalam pemrograman berorientasi objek menggunakan bahasa pemrograman Java.

1. **Pewarisan (Inheritance):** Terlihat dari penulisan class `Handphone extends Electronik`, yang berarti `Handphone` adalah subkelas dari `Electronik`. Ini berarti bahwa `Handphone` mewarisi semua properti dan metode yang dimiliki oleh kelas `Electronik`.
2. **Konstruktor:** Terdapat konstruktor `public Handphone` yang digunakan untuk membuat objek `Handphone`. Konstruktor ini menerima beberapa parameter seperti merek, model, harga, ukuran layar, CPU, baterai, RAM, dan storage. Konstruktor ini juga memanggil konstruktor dari kelas induk (`Electronik`) menggunakan `super(merek, model, harga)`.
3. **Overloading Method:** Terdapat dua metode `informasi()`. Salah satunya adalah metode tanpa parameter, dan yang lainnya adalah metode dengan parameter `String prefix`. Ini disebut `overloading method`, di mana kedua metode memiliki nama yang sama tetapi parameter yang berbeda.
4. **Penyimpanan Informasi:** Metode `informasi()` dan `informasi(String prefix)` dipanggil untuk menampilkan informasi tentang handphone. Informasi yang ditampilkan mencakup merek, model, harga, ukuran layar, CPU, baterai, RAM, dan storage.
5. **Encapsulation:** Variabel-variabel seperti layar, CPU, baterai, RAM, dan storage dienkapsulasi sebagai `private`, artinya hanya dapat diakses dan diubah oleh metode-metode dalam kelas `Handphone` itu sendiri.

Berikut kode programnya

The screenshot shows a Java code editor with the file `Main.java` selected in the tab bar. The code implements a `Main` class with a `main` method. The `main` method uses `JOptionPane.showInputDialog` to get user input for various laptop and handphone specifications, then creates `Laptop` and `Handphone` objects and prints their details.

```
1 package com.example.projek;
2
3 import javax.swing.JOptionPane;
4
5 public class Main {
6     Run | Debug
7     public static void main(String[] args) {
8
9         String merekLaptop = JOptionPane.showInputDialog("Masukkan merek laptop:");
10        String modelLaptop = JOptionPane.showInputDialog("Masukkan model laptop:");
11        int hargaLaptop = Integer.parseInt(JOptionPane.showInputDialog("Masukkan harga laptop:"));
12        double ukuranLayar = Double.parseDouble(JOptionPane.showInputDialog("Masukkan ukuran layar laptop:"));
13        String CPULaptop = JOptionPane.showInputDialog("Masukkan CPU laptop:");
14        String VGALaptop = JOptionPane.showInputDialog("Masukkan VGA laptop:");
15        int RAMLaptop = Integer.parseInt(JOptionPane.showInputDialog("Masukkan RAM laptop (GB):"));
16        int storageLaptop = Integer.parseInt(JOptionPane.showInputDialog("Masukkan storage laptop (GB):"));
17
18        Laptop laptop = new Laptop(merekLaptop, modelLaptop, hargaLaptop, ukuranLayar, CPULaptop, VGALaptop, RAMLaptop, storageLaptop);
19
20        String merekHandphone = JOptionPane.showInputDialog("Masukkan merek handphone:");
21        String modelHandphone = JOptionPane.showInputDialog("Masukkan model handphone:");
22        int hargaHandphone = Integer.parseInt(JOptionPane.showInputDialog("Masukkan harga handphone:"));
23        double layar = Double.parseDouble(JOptionPane.showInputDialog("Masukkan ukuran layar handphone:"));
24        String CPUHandphone = JOptionPane.showInputDialog("Masukkan CPU handphone:");
25        int baterai = Integer.parseInt(JOptionPane.showInputDialog("Masukkan kapasitas baterai handphone (mAh):"));
26        int RAMHandphone = Integer.parseInt(JOptionPane.showInputDialog("Masukkan RAM handphone (GB):"));
27        int storageHandphone = Integer.parseInt(JOptionPane.showInputDialog("Masukkan storage handphone (GB):"));
28
29        Handphone handphone = new Handphone(merekHandphone, modelHandphone, hargaHandphone, layar, CPUHandphone, baterai, RAMHandphone, storageHandphone);
30
31        System.out.println(x: "\n      Informasi Laptop :");
32        System.out.println(x: "=====");
33        laptop.informasi(prefix: "Laptop: ");
34
35        System.out.println(x: "      Informasi Handphone :");
36        System.out.println(x: "=====");
37        handphone.informasi(prefix: "Handphone: ");
38    }
```

Analisis Kode Programnya

1. Menerima Informasi Pengguna:

- Program menggunakan kelas `JOptionPane` dari paket `javax.swing` untuk membuat dialog pop-up yang meminta pengguna memasukkan informasi tentang laptop dan handphone.
- Informasi yang diminta termasuk merek, model, harga, ukuran layar, CPU, VGA (hanya untuk laptop), RAM, dan storage. Informasi ini dimasukkan oleh pengguna melalui dialog.

2. Membuat Objek:

- Setelah mendapatkan semua informasi yang diperlukan dari pengguna, program membuat objek `Laptop` dan `Handphone` menggunakan nilai-nilai yang dimasukkan oleh pengguna.
- Objek-objek ini dibuat dengan menggunakan konstruktor yang telah didefinisikan sebelumnya di kelas `Laptop` dan `Handphone`.

3. Menampilkan Informasi:

- Setelah objek `Laptop` dan `Handphone` dibuat, program menampilkan informasi tentang keduanya.
- Metode `informasi(String prefix)` dipanggil untuk menampilkan informasi yang sesuai dengan jenis perangkat. Prefiks "`Laptop:`" digunakan untuk informasi tentang laptop, sedangkan "`Handphone:`" digunakan untuk informasi tentang handphone.
- Informasi yang ditampilkan mencakup merek, model, harga, ukuran layar, CPU, VGA (hanya untuk laptop), RAM, dan storage.

4. Output di Konsol:

- Informasi tentang laptop dan handphone ditampilkan di konsol menggunakan perintah `System.out.println`.
- Sebelum menampilkan informasi tentang handphone, terdapat teks pembatas berupa garis horizontal untuk memisahkan informasi tentang laptop dan handphone.

5. Penanganan Input:

- Program menggunakan pemrosesan string dan konversi tipe data untuk memastikan bahwa input pengguna yang dimasukkan melalui dialog pop-up dikonversi ke tipe data yang sesuai sebelum digunakan untuk membuat objek.

Run kode programnya

PROBLEMS 29 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Run: Main + - ⏪ ⏴ ⏵ ⏹ ⏷ ⏸ ⏹ ⏻

Informasi Laptop :

```
=====
```

Laptop: Merek : Lenovo
Laptop: Model : LOQ
Laptop: Harga : 13000000
Laptop: Ukuran Layar : 15.6 Inci
Laptop: CPU : i5-12450HX
Laptop: VGA : RTX 3050
Laptop: RAM : 12 GB
Laptop: Storage : 1024 GB

Informasi Handphone :

```
=====
```

Handphone: Merek : Samsung
Handphone: Model : S24 Ultra
Handphone: Harga : 24000000
Handphone: Ukuran Layar : 6.5 Inci
Handphone: CPU : Snapdragon 8 gen 3
Handphone: Baterai : 5000 mAh
Handphone: RAM : 12 GB
Handphone: Storage : 512 GB

PS C:\bpo>

TerimaKasih

