



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA



DIKTISAINTEK
BERDAMPAK

KAMPUS
INOVASI

Review Inheritance, Enkapsulasi & Polimorfism

PBO

Yunia Ikawati

Teknik Informatika-PENS





Overview

1. Inheritance (Pewarisan)
2. Enkapsulasi
3. Polimorfisme

Inheritance

- **Inheritance adalah** konsep OOP di mana sebuah class anak (subclass) dapat mewarisi property/data/atribut dan metode/perilaku objek dari class induk (superclass).
- Kita bisa membuat class baru berdasarkan class yang sudah ada, tanpa harus menulis ulang semua kode.
- **Manfaat :**
 - Reusabilitas: Kode yang sudah ada bisa digunakan kembali.
 - Modularitas: Kode lebih terstruktur dan mudah dikembangkan.
 - Ekstensi: Bisa menambahkan atau mengubah perilaku class turunan.

Contoh Inheritance

```
Mobil.java x
Source History
1 // Superclass
2 public class Mobil {
3     protected String merk;
4     protected int tahun;
5
6     public Mobil(String merk, int tahun) {
7         this.merk = merk;
8         this.tahun = tahun;
9     }
10
11     public void nyalakanMesin() {
12         System.out.println(merk + " tahun " + tahun + " mesin dinyalakan.");
13     }
14 }
```

```
Main.java x
Source History
1 public class Main {
2     public static void main(String[] args) {
3         // Membuat objek dari superclass Mobil
4         Mobil mobilBiasa = new Mobil("Honda Brio", 2018);
5         mobilBiasa.nyalakanMesin();
6
7         // Membuat objek dari subclass MobilBalap
8         MobilBalap mobilBalap = new MobilBalap("Ferrari F8", 2022, 340);
9         mobilBalap.nyalakanMesin(); // Method diwarisi dari Mobil
10        mobilBalap.tampilkanKecepatan(); // Method khusus MobilBalap
11    }
12 }
```

```
MobilBalap.java x
Source History
1 // Subclass
2 public class MobilBalap extends Mobil {
3     private int kecepatanMaks;
4
5     public MobilBalap(String merk, int tahun, int kecepatanMaks) {
6         super(merk, tahun);
7         this.kecepatanMaks = kecepatanMaks;
8     }
9
10    public void tampilkanKecepatan() {
11        System.out.println("Kecepatan maksimum: " + kecepatanMaks + " km/jam");
12    }
13 }
```

```
Output - Inheritance_MobilApp (run) x
run:
Honda Brio tahun 2018 mesin dinyalakan.
Ferrari F8 tahun 2022 mesin dinyalakan.
Kecepatan maksimum: 340 km/jam
BUILD SUCCESSFUL (total time: 0 seconds)
```

Enkapsulasi

- **Enkapsulasi** adalah salah satu prinsip utama dalam pemrograman berorientasi objek (OOP) yang bertujuan untuk melindungi data dan mengontrol akses terhadapnya.
- **Enkapsulasi** adalah teknik menyembunyikan detail internal dari sebuah objek dan hanya memperbolehkan akses melalui metode yang telah ditentukan, sehingga data menjadi lebih aman dan terkontrol.
- **Manfaat**
 - **Keamanan Data:** Mencegah manipulasi langsung terhadap atribut.
 - **Validasi & Kontrol Akses:** Data hanya bisa diubah melalui method yang telah ditentukan (setter).
 - **Kemudahan Pemeliharaan:** Perubahan internal tidak memengaruhi kode luar class.

Contoh Enkapsulasi

```
Mobil.java x
Source History
1 public class Mobil {
2     private String merk;
3     private int tahun;
4
5     public Mobil(String merk, int tahun) {
6         this.merk = merk;
7         this.tahun = tahun;
8     }
9
10    public String getMerk() {
11        return merk;
12    }
13
14    public void setMerk(String merk) {
15        this.merk = merk;
16    }
17
18    public int getTahun() {
19        return tahun;
20    }
21
22    public void setTahun(int tahun) {
23        this.tahun = tahun;
24    }
25 }
```

```
Main.java x
Source History
1 public class Main {
2     public static void main(String[] args) {
3         // Membuat objek Mobil
4         Mobil mobil1 = new Mobil("Suzuki Ertiga", 2021);
5
6         // Mengakses data melalui getter
7         System.out.println("Merk: " + mobil1.getMerk());
8         System.out.println("Tahun: " + mobil1.getTahun());
9
10        // Mengubah data melalui setter
11        mobil1.setMerk("Suzuki XL7");
12        mobil1.setTahun(2023);
13
14        // Menampilkan data setelah diubah
15        System.out.println("Merk baru: " + mobil1.getMerk());
16        System.out.println("Tahun baru: " + mobil1.getTahun());
17    }
18 }
```

```
Output - Enkapsulasi_mobil (run) x
run:
Merk: Suzuki Ertiga
Tahun: 2021
Merk baru: Suzuki XL7
Tahun baru: 2023
BUILD SUCCESSFUL (total time: 0 seconds)
```

Polimorfisme

- Polimorfisme berasal dari bahasa Yunani: *poly* (banyak) dan *morph* (bentuk).
Dalam OOP, polimorfisme berarti **kemampuan objek untuk merespons metode yang sama dengan cara berbeda**, tergantung tipe objeknya
- **Polimorfisme** adalah salah satu prinsip utama dalam pemrograman berorientasi objek (OOP) yang memungkinkan satu method atau objek memiliki banyak bentuk atau perilaku berbeda, tergantung konteksnya.
- **Manfaat Polimorfisme:**
 - Mempermudah **ekstensi dan pemeliharaan kode**
 - Mendukung **abstraksi dan fleksibilitas**
 - Memungkinkan penggunaan **interface dan inheritance** secara efisien

Contoh Polimorfisme

```
Mobil.java x
Source History
1 // Superclass
2 class Mobil {
3     public void nyalakanMesin() {
4         System.out.println("Mesin mobil dinyalakan.");
5     }
6
7     public void info() {
8         System.out.println("Ini adalah mobil umum.");
9     }
10 }
```

```
Sedan.java x
Source History
1 // Subclass 1
2 class Sedan extends Mobil {
3     @Override
4     public void nyalakanMesin() {
5         System.out.println("Mesin sedan menyala dengan halus.");
6     }
7
8     @Override
9     public void info() {
10        System.out.println("Ini adalah mobil sedan, cocok untuk keluarga.");
11    }
12 }
```

```
Truk.java x
Source History
1 // Subclass 2
2 class Truk extends Mobil {
3     @Override
4     public void nyalakanMesin() {
5         System.out.println("Mesin truk menyala dengan suara keras.");
6     }
7
8     @Override
9     public void info() {
10        System.out.println("Ini adalah truk, digunakan untuk mengangkut barang berat.");
11    }
12 }
```

```
Main.java x
Source History
1
2 public class Main {
3     public static void main(String[] args) {
4         Mobil m1 = new Sedan(); // Polimorfis
5         Mobil m2 = new Truk();  // Polimorfis
6
7         m1.nyalakanMesin(); // Output: Mesin
8         m1.info();         // Output: Ini ad
9
10        m2.nyalakanMesin(); // Output: Mesin
11        m2.info();         // Output: Ini ad
12    }
13 }
```

```
Output x
Debugger Console x Polimorfisme_mobil (run) x
run:
Mesin sedan menyala dengan halus.
Ini adalah mobil sedan, cocok untuk keluarga.
Mesin truk menyala dengan suara keras.
Ini adalah truk, digunakan untuk mengangkut barang berat.
BUILD SUCCESSFUL (total time: 0 seconds)
```




Aturan Umum Project UAS PBO



DIKTISAINTEK
BERDAMPAK



1. Jenis Project:

- Aplikasi berbasis desktop menggunakan bahasa pemrograman **Java**.
- Harus memiliki tampilan grafis (GUI) menggunakan library seperti **Java Swing** atau **JavaFX**.

2. Kelompok:

- Setiap kelompok terdiri dari **3 mahasiswa**.
- Tidak diperkenankan bekerja sendiri atau berkelompok lebih dari 3 orang.

3. Judul Project:

- **Judul aplikasi harus unik** untuk setiap kelompok.
- Tidak boleh ada dua kelompok dengan judul atau tema aplikasi yang sama.
- Judul harus disetujui oleh dosen sebelum minggu ke-10.

4. Isi Laporan Project: Laporan akhir harus disusun dalam format berikut:

- Cover dan Identitas Kelompok
- Latar Belakang: Penjelasan alasan pemilihan judul dan urgensi aplikasi.
- Deskripsi Aplikasi: Gambaran umum fungsi dan fitur aplikasi.
- Desain Sistem:
 - Diagram UML (minimal: use case dan class diagram).
 - Penjelasan alur kerja sistem.
- Hasil Aplikasi:
 - Screenshot tampilan aplikasi.
 - Penjelasan fitur-fitur utama.
- Kesimpulan: Refleksi hasil kerja dan pembelajaran selama project.
- Referensi: Buku, artikel, atau sumber lain yang digunakan.



Aturan Umum Project UAS PBO

5. Pengumpulan:

- Project dan laporan dikumpulkan pada **minggu ke-14**.
- Format pengumpulan:
 - File aplikasi (.jar atau folder project)
 - Laporan dalam format PDF
 - Video demo aplikasi berdurasi maksimal 5 menit
- Semua file dikumpulkan melalui LMS atau media yang ditentukan dosen.

6. Penilaian:

- Kreativitas dan kompleksitas aplikasi
- Kesesuaian dengan prinsip OOP (inheritance, polymorphism, encapsulation)
- Kualitas tampilan dan interaksi pengguna
- Kelengkapan dan kerapian laporan
- Kerja sama dan kontribusi anggota kelompok

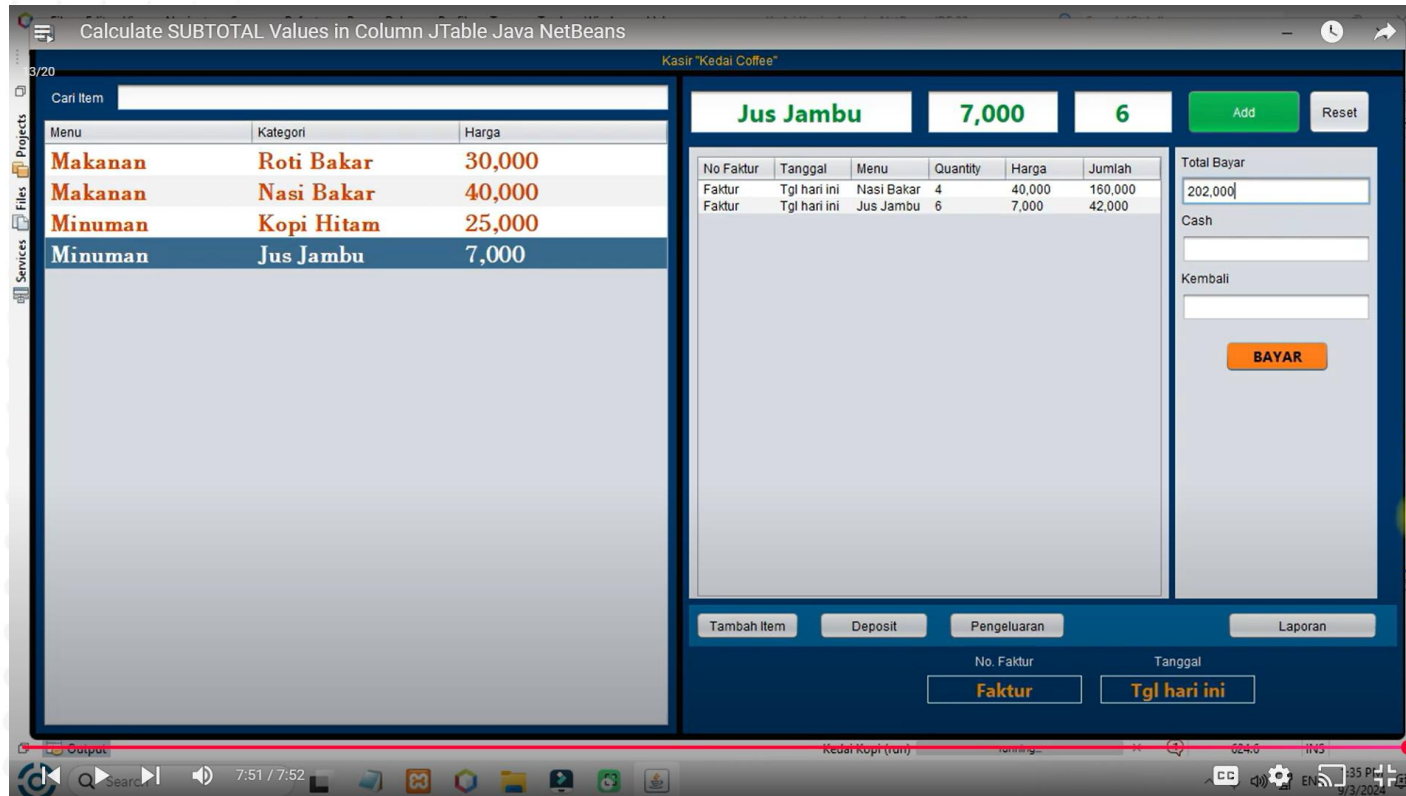
7. Plagiarisme:

- Dilarang keras menyalin project dari internet atau kelompok lain.
- Jika ditemukan plagiarisme, nilai project akan dihapus atau dinyatakan gagal.

Contoh Aplikasi Java Desktop

(Aplikasi Kasir dengan menggunakan Java Netbeans)

- https://www.youtube.com/watch?v=lgjl7zS9nU&list=PLfkCq_npeByCr46UrwanKG_PQdAJy-Kgs&index=13





Tugas Minggu Depan

Diberitahukan bahwa **minggu depan** pengumpulan dokumen awal proyek PBO yang akan dibuat dan dipresentasikan.

Adapun komponen yang wajib dikumpulkan adalah sebagai berikut:

1. Latar Belakang

- Jelaskan alasan pemilihan judul proyek.

- Uraikan urgensi dan manfaat dari aplikasi yang akan dikembangkan.

2. Deskripsi Aplikasi

- Berikan gambaran umum mengenai fungsi utama aplikasi.

- Jelaskan fitur-fitur yang akan tersedia dan bagaimana aplikasi akan digunakan oleh pengguna.

3. Desain Sistem

- Sertakan diagram alur sistem, struktur kelas, dan relasi antar komponen.

- Penjelasan teknis mengenai arsitektur aplikasi (misalnya: MVC, client-server, dll).