

LAPORAN PRATIUM IV
PEMOGRAMAN BERBASI OBJEK

“Pilar OOP (Encapsulation, Inheritance, Polymorphism)”



Disusun oleh :

Naufal Dira Agustian - 22115330011

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

1. Pendahuluan

Terdapat empat prinsip dasar atau pilar dari Object Oriented Programming (OOP), yaitu Encapsulation, Inheritance, Polymorphism, dan Abstraction. Prinsip-prinsip ini memberikan keuntungan seperti modularity, *code reusability*, *extensibility*, *data hiding*, dan sebagainya.

- **Encapsulation:** Mekanisme untuk menyembunyikan detail implementasi internal suatu objek dan mengontrol akses terhadap data serta perilaku di dalamnya, dengan membungkus/mengelompokkan data dan *method* dalam satu unit (*class*).
- **Inheritance:** Mekanisme di mana sebuah *class* dapat mewarisi properti (*attribute*) dan perilaku (*method*) dari *class* lain.
- **Polymorphism:** Konsep yang memungkinkan satu antarmuka atau *method* untuk memiliki banyak bentuk berbeda, sehingga objek dapat berperilaku berbeda meskipun dipanggil melalui cara yang sama.

2. Tujuan Praktikum

- Mahasiswa mampu memahami dan menerapkan prinsip **Encapsulation** dalam Java.
- Mahasiswa mampu menggunakan konsep **Inheritance** untuk membangun hierarki kelas.
- Mahasiswa mampu menerapkan **Polymorphism**

Proses Pengerjaan (Implementasi Kode)

Studi kasus praktikum adalah pembangunan sistem manajemen perpustakaan yang melibatkan berbagai jenis buku dan pengguna, yang harus dibangun menggunakan konsep Object Oriented Programming.

- Langkah 1: Penerapan Encapsulation pada Class Book

Kelas Book memiliki atribut *private* (title, author, isAvailable) dan menyediakan *getter* serta *method* publik untuk mengakses/memodifikasi data.

File: Book.java

```
package pratikum4;

public class book {

    private String title;
```

```
private String author;

private boolean isAvailable;

// Constructor

public book(String title, String author) {

    this.title = title;

    this.author = author;

    this.isAvailable = true;

}

// Getter methods (for accessing private data) [cite: 100]

public String getTitle() {

    return title;

}

public String getAuthor() {

    return author;

}

public boolean isAvailable() {

    return isAvailable;

}

// Methods to modify private data (like setters) [cite: 100]

public void borrowBook() {

    this.isAvailable = false;

}

public void returnBook() {
```

```
this.isAvailable = true;

}

}
```

- Langkah 2: Penerapan Inheritance (Class Novel, Magazine, Textbook)

Kelas Novel, Magazine, dan Textbook mewarisi properti dari Book (*superclass*).

File: Novel.java

```
package pratikum4;

public class Novel extends book {

    private String genre;

    public Novel(String title, String author, String genre) {

        super(title, author);

        this.genre = genre;

    }

    public String getGenre() {

        return genre;

    }

}
```

File: Magazine.java

```
package pratikum4;

public class Magazine extends book {

    private String category;

    public Magazine(String title, String author, String category) {

        super(title, author);

    }

}
```

```

this.category = category;

}

public String getCategory() {

return category;

}

}

```

File: Textbook.java

```

package pratikum4;

public class TextBook extends book {

private String studyField;

public TextBook(String title, String author, String studyField) {

super(title, author);

this.studyField = studyField;

}

public String getStudyField() {

return studyField;

}

}

```

- Langkah 3: Penerapan Polymorphism pada Class User

Kelas User memiliki metode viewBookDetails(Book book) yang menunjukkan Polymorphism dengan menggunakan instanceof untuk menyesuaikan tampilan detail sesuai tipe buku.

File: User.java

```

package pratikum4;

public class User {

```

```
private String name;

public User() {

    this.name = "Pengguna Umum";

}

public User(String name) {

    this.name = name;

}

public void borrowBook(book book) {

    if (book.isAvailable()) {

        book.borrowBook();

        System.out.println("Buku \"" + book.getTitle() + "\" berhasil dipinjam oleh "
+ this.name); // [cite: 158, 159]

    } else {

        System.out.println("Maaf " + this.name + ", buku \"" + book.getTitle() + "\"
sedang tidak tersedia."); // [cite: 161]

    }

}

public void returnBook(book book) {

    if (!book.isAvailable()) {

        book.returnBook();

        System.out.println("Buku \"" + book.getTitle() + "\" berhasil dikembalikan.");

    } else {

        System.out.println("Buku \"" + book.getTitle() + "\" sudah tersedia.");

    }

}
```

```

}

public void viewBookDetails(book book) {

    System.out.println("Judul: " + book.getTitle());

    System.out.println("Penulis: " + book.getAuthor());

    System.out.println("Tersedia: " + (book.isAvailable() ? "Ya" : "Tidak"));

    if (book instanceof Novel) {

        Novel novel = (Novel) book;

        System.out.println("Genre: " + novel.getGenre());

    } else if (book instanceof Magazine) {

        Magazine magazine = (Magazine) book;

        System.out.println("Kategori: " + magazine.getCategory());

    } else if (book instanceof TextBook) {

        TextBook textbook = (TextBook) book;

        System.out.println("Bidang Studi: " + textbook.getStudyField());

    }

}

}

}

```

- **Langkah 4: Program Utama (Main.java)**

Program utama membuat objek buku, objek pengguna, dan menjalankan fungsi peminjaman/pengembalian.

File: Main.java

```

package pratikum4;

public class Main {

```

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

    book novel = new Novel("Laskar Pelangi", "Andrea Hirata", "Drama");

    book magazine = new Magazine("National Geographic", "Various Authors",
    "Science");

    book textbook = new TextBook("Pemrograman Java", "Anonymous",
    "Informatika");

    User user = new User("Superman");

    System.out.println("=== Detail Buku ===");

    user.viewBookDetails(novel);

    System.out.println();

    user.viewBookDetails(magazine);

    System.out.println();

    user.viewBookDetails(textbook);

    System.out.println();

    System.out.println("=== Proses Peminjaman Buku ===");

    user.borrowBook(novel);

    user.borrowBook(magazine);

    System.out.println("\nStatus Buku Setelah Dipinjam:");

    System.out.println(novel.getTitle() + " tersedia: " + novel.isAvailable());

    System.out.println(magazine.getTitle() + " tersedia: " +
    magazine.isAvailable());

    System.out.println("\n=== Proses Pengembalian Buku ===");

    user.returnBook(novel);

    System.out.println("\nStatus Buku Setelah Dikembalikan:");

    System.out.println(novel.getTitle() + " tersedia: " + novel.isAvailable());
```



```

}

}

```

3. Hasil Akhir (Output)

```

1 package pratikum4;
2

```

```

<terminated> Main [Java Application] C:\Users\Asus\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full\jre\bin\java.exe
=== Detail Buku ===
Judul: Laskar Pelangi
Penulis: Andrea Hirata
Tersedia: Ya
Genre: Drama

Judul: National Geographic
Penulis: Various Authors
Tersedia: Ya
Kategori: Science

Judul: Pemrograman Java
Penulis: Anonimous
Tersedia: Ya
Bidang Studi: Informatika

=== Proses Peminjaman Buku ===
Buku "Laskar Pelangi" berhasil dipinjam oleh Superman
Buku "National Geographic" berhasil dipinjam oleh Superman

Status Buku Setelah Dipinjam:
Laskar Pelangi tersedia: false
National Geographic tersedia: false

=== Proses Pengembalian Buku ===
Buku "Laskar Pelangi" berhasil dikembalikan.

Status Buku Setelah Dikembalikan:
Laskar Pelangi tersedia: true

```

Program dimulai dengan menampilkan Detail Buku untuk tiga objek buku yang berbeda (Novel, Magazine, dan Textbook) menggunakan *method* `viewBookDetails()` pada objek user. Karena adanya Polymorphism dan pemeriksaan tipe objek saat *runtime* (*instanceof*), *output* dapat secara spesifik menampilkan atribut unik masing-masing *subclass* : Laskar Pelangi menampilkan Genre: Drama , National Geographic menampilkan Kategori: Science , dan Pemrograman Java menampilkan Bidang Studi: Informatika. Awalnya, semua buku berstatus Tersedia: Ya. Selanjutnya, dalam Proses Peminjaman Buku , objek user meminjam Laskar Pelangi dan National Geographic , dan sistem mencetak pesan sukses untuk peminjaman oleh "Superman". Status ketersediaan kedua buku tersebut kemudian diperiksa dan dikonfirmasi menjadi tersedia: false. Terakhir, pada Proses Pengembalian Buku, buku Laskar Pelangi dikembalikan , menghasilkan pesan sukses pengembalian , dan statusnya kembali menjadi tersedia: true.