LAPORAN TUGAS W04

MANAJEMEN KONFIGURASI DAN EVOLUSI PERANGKAT LUNAK

PRAKTIKUM REFACTORING

Disusun Oleh:

Nadia Putri Rahmaniar / 2211104012

S1 SE-06-01

Dosen Pengampu:

Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

TUGAS PENDAHULUAN

1. Pendahuluan

Dalam pengembangan perangkat lunak, penyusunan kode yang terstruktur dengan baik sangat krusial untuk meningkatkan keterbacaan, kemudahan pemeliharaan, dan skalabilitas. Dalam tugas ini, dilakukan refaktorisasi pada class Song.java agar lebih modular, sesuai dengan prinsip Single Responsibility Principle (SRP), serta menghilangkan bad smell dalam kode.

2. Identifikasi Masalah dalam Kode Asli

Beberapa masalah yang ditemukan dalam kode asli Song.java:

a. **God Class**: Class Song memiliki terlalu banyak tanggung jawab, termasuk menyimpan informasi tentang album dan artis.

b. **Magic Number**: Genre direpresentasikan dengan angka (0-7), sehingga sulit dipahami dan dapat membingungkan.

c. **Long Method**: Fungsi printInfo(int detailLevel) terlalu kompleks karena mengandung banyak percabangan if-else, yang mengurangi keterbacaan dan pemeliharaan kode.

3. Langkah Refaktorisasi

a. Pemisahan Class

- Membuat class Album untuk menyimpan informasi terkait album.
- Membuat class Artist untuk menyimpan data tentang artis.
- Menggunakan enum Genre untuk menggantikan angka dalam representasi genre, sehingga lebih mudah dipahami dan dikelola.

b. Implementasi Class Baru

- Class Album.java

```java
package assignment;

public class Album {
    private String name;
    private String coverURL;

    public Album(String name, String coverURL) {
        this.name = name;
        this.coverURL = coverURL;
    }

    public String getName() {
        return name;
    }

    public String getCoverURL() {
        return coverURL;
    }
}
```

- Class Artist.java

```java
package assignment;

public class Artist {
    private String name;
    private String alias;
    private String imageURL;

    public Artist(String name, String alias, String imageURL) {
        this.name = name;
        this.alias = alias;
        this.imageURL = imageURL;
    }

    public String getName() {
        return name;
    }

    public String getAlias() {
        return alias;
    }

    public String getImageURL() {
        return imageURL;
    }
}

```

- Enum Genre.java

```java
package assignment;

public enum Genre {
    UNDEFINED, POP, ROCK, HIPHOP, RNB, JAZZ,
    INSTRUMENTALS, CLOWNCORE
}
```

c. Perubahan pada Class Song.java

```java
package assignment;

public class Song {
    private String id;
    private String title;
    private String releaseYear;
    private String musicFileURL;
    private Genre genre;
    private Album album;
    private Artist artist;

    public Song(String id, String title, String releaseYear, String musicFileURL) {
        this.id = id;
        this.title = title;
        this.releaseYear = releaseYear;
        this.musicFileURL = musicFileURL;
        this.genre = Genre.UNDEFINED;
    }

    public void setAlbum(Album album) {
        this.album = album;
    }

    public void setArtist(Artist artist) {
        this.artist = artist;
    }

    public void setGenre(Genre genre) {
        this.genre = genre;
    }

    public void printInfo() {
        System.out.println("Song title: " + title);
        System.out.println("Release year: " + releaseYear);
        System.out.println("Genre: " + genre);
        if (artist != null) {
            System.out.println("Artist: " + artist.getName() + " (aka " + artist.getAlias() + ")");
```

```
38          }
39          if (album != null) {
40              System.out.println("Album: " + album.getName());
41          }
42      }
43  }
```

d. Pembuatan Class Main.java untuk Testing

```
1   package assignment;
2
3   public class Main {
4       public static void main(String[] args) {
5           Song song = new Song("2", "Hotel California", "1976", "hotel_california.mp3");
6           song.setGenre(Genre.ROCK);
7
8           Album album = new Album("Hotel California", "eagles_cover.jpg");
9           song.setAlbum(album);
0
1           Artist artist = new Artist("Eagles", "Don Henley", "eagles.jpg");
2           song.setArtist(artist);
3
4           song.printInfo();
5       }
6   }
```

4. Pengujian Kode

a. Kompilasi semua file Java

javac -d bin -sourcepath src src/Assignment/*.java

b. Jalankan program

java -cp bin assignment.Main

c. Output yang diharapkan:

```
PS D:\School\MKEPL\04_Praktick_Refaktoring\refactor_song> javac -d bin -sourcepath src src/Assignment/*.java
PS D:\School\MKEPL\04_Praktick_Refaktoring\refactor_song> java -cp bin assignment.Main
Song title: Hotel California
Release year: 1976
Genre: ROCK
Artist: Eagles (aka Don Henley)
Album: Hotel California
```

d. Github Code:

Link code

5. Kesimpulan

a. Proses refaktorisasi membuat kode lebih modular dan mudah dibaca.

b. Genre kini direpresentasikan menggunakan enum, menggantikan penggunaan magic number.

c. Informasi mengenai album dan artis dipisahkan ke dalam class khusus.

d. Metode printInfo() disederhanakan agar lebih jelas dan mudah dipahami.