TUGAS W 04
MANAJEMEN KONFIGURASI DAN EVOLUSI PERANGKAT LUNAK

PRAKTIKUM REFACTORING

DISUSUN OLEH:
KHOLIFAHDINA
2211104004

SE 06 01

S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
2025

Refactoring Song.java

1. Identifikasi Bad Smell dalam Song.java

    a. Feature Envy

    Dalam Song.java, metode setGenre terlalu bergantung pada kelas Genre.

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

    b. Long Method

    Metode printDetails() dalam Song.java menangani beberapa tugas sekaligus

```
public void printDetails() {
    System.out.println("Title: " + title + ",
Artist: " + artist + ", Duration: " + duration + "
seconds");
}
```

    c. Improper Dependency

    Kelas Song secara langsung mengakses atribut dari kelas lain seperti Album dan Artist, sehingga perlu dipisahkan agar lebih modular.

```
public class Song {
    private String title;
    private String artist;
    private String releaseYear;
    private String url;
}
```

2. Refactoring Song.java

    a. Memisahkan Tanggung Jawab (SRP - Single Responsibility Principle)

    Tanggung jawab dalam Song.java terlalu banyak, sehingga dipisahkan menjadi beberapa kelas seperti Album.java dan Artist.java.

```
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;
    }
```

b. Penggunaan Getter dan Setter

Enkapsulasi diterapkan dengan mengganti akses langsung ke atribut menggunakan metode getter dan setter.

```
public String getTitle() {
    return title;
}

public Artist getArtist() {
    return artist;
}

public Album getAlbum() {
    return album;
}
```

c. Membuat Helper Method

Metode printInfo() dipecah menjadi metode yang lebih kecil agar lebih modular.

```
public void printInfo(DetailLevel detailLevel) {
        System.out.println("Song title: " + title);
        System.out.println("Release year: " +
releaseYear);
        if (genre != Genre.UNDEFINED) {
            System.out.println("Genre: " + genre);
        }

        if (detailLevel == DetailLevel.ARTIST ||
detailLevel == DetailLevel.FULL) {
            if (artist != null) {
                artist.printInfo();
            }
        }

        if (detailLevel == DetailLevel.ALBUM ||
```

```
detailLevel == DetailLevel.FULL) {
            if (album != null) {
                album.printInfo();
            }
        }
    }
```

3. Perbandingan Kode Sebelum dan Sesudah

   Kode Sebelum:

```
public package Assignment;


public class Song {

    private String id;
    private String title;
    private String releaseYear;
    private String musicFileURL;
    private int genre;

    private String albumName;
    private String albumCoverURL;

    private String artistName;
    private String artistAlias;
    private String artistImageURL;

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

     public void setAlbum(String albumName, String albumCoverURL)
{
        this.albumName = albumName;
        this.albumCoverURL = albumCoverURL;
    }
```

```
     public void setArtist(String artistName, String artistAlias,
String artistImageURL) {
        this.artistName = artistName;
        this.artistAlias = artistAlias;
        this.artistImageURL = artistImageURL;
    }

    /**
     * Set the genre of this song
     *
     * 0 = undefined
     * 1 = pop
     * 2 = rock
     * 3 = hip hop
     * 4 = RnB
     * 5 = jazz
     * 6 = instrumentals
     * 7 = clowncore
     *
     * @param genre
     */
    public void setGenre(int genre) {
        this.genre = genre;
    }

    /**
     * Print info of the song based on desired detail level
     *
     * 0 = song info only
     * 1 = song info and artist info
     * 2 = song info and album info
     * 3 = song, artist, and album info
     *
     * @param genre
     */
    public void printInfo(int detailLevel) {
        if (detailLevel == 0) {
            System.out.println("song title: " + title);
            System.out.println("release year: " + releaseYear);
            if (genre > 0) {
                System.out.println("genre: " + genre);
            }
        }else if(detailLevel == 1) {
```

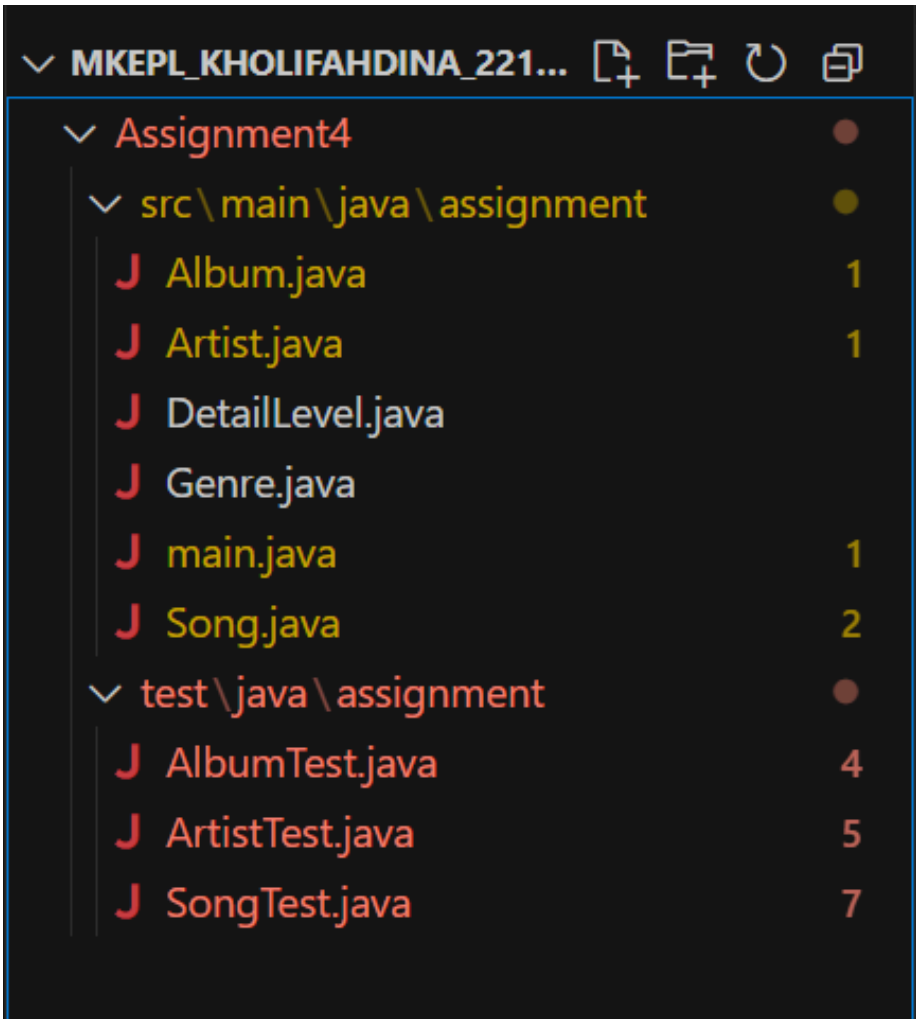
```
            System.out.println("song title: " + title);
            System.out.println("release year: " + releaseYear);
            if (genre > 0) {
                System.out.println("genre: " + genre);
            }
            if (!artistName.equals("")) {
                System.out.println("artist name: " + artistName);
            }
            if (!artistAlias.equals("")) {
                    System.out.println("artist also known as: " +
artistAlias);
            }
        }else if (detailLevel == 2) {
            System.out.println("song title: " + title);
            System.out.println("release year: " + releaseYear);
            if (genre > 0) {
                System.out.println("genre: " + genre);
            }
            if (!albumName.equals("")) {
                System.out.println("album title: " + albumName);
            }
        }else if (detailLevel == 3) {
            System.out.println("song title: " + title);
            System.out.println("release year: " + releaseYear);
            if (genre > 0) {
                System.out.println("genre: " + genre);
            }
            if (!artistName.equals("")) {
                System.out.println("artist name: " + artistName);
            }
            if (!artistAlias.equals("")) {
                    System.out.println("artist also known as: " +
artistAlias);
            }
            if (!albumName.equals("")) {
                System.out.println("album title: " + albumName);
            }
        }
    }

} {

}
```

Kode Setelah:

Modular filenya:



Kodenya:

- 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 void printInfo() {
        System.out.println("Album title: " + name);
    }
}
```

- Artist.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 void printInfo() {
        System.out.println("Artist name: " + name);
        if (!alias.isEmpty()) {
            System.out.println("Also known as: " + alias);
        }
    }
}
```

- DetailLevel.java

```
package assignment;

public enum DetailLevel {
    SONG,       // Hanya info lagu
    ARTIST,     // Info lagu + artis
    ALBUM,      // Info lagu + album
    FULL        // Semua informasi
}
```

- Genre.java

```
package assignment;
```

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

- main.java

```
package assignment;


public class main {
    public static void main(String[] args) {
        // Buat objek Song
          Song song = new Song("Genre", "My Song", "2025",
"http://example.com/music.mp3");
        song.setGenre(Genre.POP); // Set genre Pop

        // Buat objek Artist
            Artist artist = new Artist("John Doe", "JD",
"http://example.com/john.jpg");
        song.setArtist(artist);

        // Buat objek Album
                  Album album = new Album("Best Hits",
"http://example.com/album.jpg");
        song.setAlbum(album);

        // Cetak informasi lagu
           song.printInfo(DetailLevel.SONG); // Menampilkan
info lagu, artis, dan album
    }
}
```

- Song.java

```
package assignment;
```

```java
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 setGenre(Genre genre) {
        this.genre = genre;
    }

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

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

    public void printInfo(DetailLevel detailLevel) {
        System.out.println("Song title: " + title);
        System.out.println("Release year: " + releaseYear);
        if (genre != Genre.UNDEFINED) {
            System.out.println("Genre: " + genre);
        }

        if (detailLevel == DetailLevel.ARTIST || detailLevel
== DetailLevel.FULL) {
            if (artist != null) {
                artist.printInfo();
            }
        }
```

```
        if (detailLevel == DetailLevel.ALBUM || detailLevel
== DetailLevel.FULL) {
            if (album != null) {
                album.printInfo();
            }
        }
    }
}
```

- AlbumTest.java

```
package assignment;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class AlbumTest {
    @Test
    public void testAlbumCreation() {
                    Album album = new Album("Imagine",
"https://example.com/album-cover.jpg");
        assertEquals("Imagine", album.getName());
    }
}
```

- ArtistTest.java

```
package assignment;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class ArtistTest {
    @Test
    public void testArtistCreation() {
            Artist artist = new Artist("John Lennon", "The
Dreamer", "https://example.com/john-lennon.jpg");
        assertEquals("John Lennon", artist.getName());
        assertEquals("The Dreamer", artist.getAlias());
    }
}
```

- SongTest.java

```java
package assignment;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class SongTest {
    @Test
    public void testSongCreation() {
            Song song = new Song("1", "Imagine", "1971",
"https://example.com/imagine.mp3");
        assertEquals("Imagine", song.getTitle());
        assertEquals("1971", song.getReleaseYear());
    }

    @Test
    public void testSetGenre() {
            Song song = new Song("1", "Imagine", "1971",
"https://example.com/imagine.mp3");
        song.setGenre(Genre.ROCK);
        assertEquals(Genre.ROCK, song.getGenre());
    }
}
```

Berikut Link Repository Github: Link