**Tugas W04 - Praktikum Refactoring**

**Nama : Ahmad Junaidii**

**NIM : 2211104002**

**Kelas : S1SE0601**

1. **Identifikasi bad smell yang ada pada class tersebut!**
2. God Class

* Class Song memiliki terlalu banyak tanggung jawab.
* Informasi artist dan album sebaiknya dipisahkan ke dalam class tersendiri.

1. Primitive Obsession

* Genre menggunakan integer (int genre) dengan angka 0-7, yang dapat menyebabkan kesalahan interpretasi.
* Solusi: Gunakan enum untuk menggantikan angka genre.

1. Long Method (printInfo)

* Method printInfo(int detailLevel) memiliki banyak struktur if-else yang kompleks.
* Solusi: Gunakan polymorphism atau enum dengan strategi cetak informasi.

1. Feature Envy

* Method setAlbum dan setArtist menunjukkan bahwa atribut album dan artist lebih cocok berada dalam class terpisah.

1. **Lakukan refactoring terhadap class Song.Java sesuai dengan bad smell yang dilakukan, silahkan buat method dan class baru jika dibutuhkan!**
2. Song.java (Class utama untuk lagu)

package Assignment;

public class Song {

    private final String id;

    private final String title;

    private final String releaseYear;

    private final String musicFileURL;

    private Genre genre;

    private Artist artist;

    private Album album;

    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(int detailLevel) {

        System.out.println("Song Title: " + title);

        System.out.println("Release Year: " + releaseYear);

        if (genre != Genre.UNDEFINED) {

            System.out.println("Genre: " + genre);

        }

        if (detailLevel >= 1 && artist != null) {

            artist.printArtistInfo();

        }

        if (detailLevel >= 2 && album != null) {

            album.printAlbumInfo();

        }

    }

    public String getId() {

        return id;

    }

    public String getTitle() {

        return title;

    }

    public String getReleaseYear() {

        return releaseYear;

    }

    public String getMusicFileURL() {

        return musicFileURL;

    }

}

1. Artist.java (Class untuk data artis)

package Assignment;

public class Artist {

    private final String name;

    private final String alias;

    private final String imageURL;

    public Artist(String name, String alias, String imageURL) {

        this.name = name;

        this.alias = alias;

        this.imageURL = imageURL;

    }

    public void printArtistInfo() {

        System.out.println("Artist Name: " + name);

        if (!alias.isEmpty()) {

            System.out.println("Also Known As: " + alias);

        }

    }

    public String getName() {

        return name;

    }

    public String getAlias() {

        return alias;

    }

    public String getImageURL() {

        return imageURL;

    }

}

1. Album.java (Class untuk data album)

package Assignment;

public class Artist {

    private final String name;

    private final String alias;

    private final String imageURL;

    public Artist(String name, String alias, String imageURL) {

        this.name = name;

        this.alias = alias;

        this.imageURL = imageURL;

    }

    public void printArtistInfo() {

        System.out.println("Artist Name: " + name);

        if (!alias.isEmpty()) {

            System.out.println("Also Known As: " + alias);

        }

    }

    public String getName() {

        return name;

    }

    public String getAlias() {

        return alias;

    }

    public String getImageURL() {

        return imageURL;

    }

}

1. Genre.java (Enum untuk genre)

package Assignment;

public enum Genre {

    UNDEFINED, POP, ROCK, HIPHOP, RNB, JAZZ, INSTRUMENTALS, CLOWNCORE

}

Perubahan yang dilakukan:

* Memisahkan class Artist dan Album

Menghilangkan God Class, karena Song.java sebelumnya memiliki terlalu banyak tanggung jawab.

* Menggunakan Genre sebagai Enum

Menghilangkan Primitive Obsession, karena sebelumnya menggunakan angka sebagai kode genre.

* Refactor method printInfo()

Menghindari Long Method dengan membiarkan Artist dan Album menangani pencetakan informasinya sendiri.

* Kode lebih modular & mudah dikembangkan

Jika ada perubahan pada Artist atau Album, kita tidak perlu mengubah Song.java.