

MANAJEMEN KONFIGURASI DAN EVOLUSI PERANGKAT LUNAK

**TUGAS W04
PRAKTIKUM REFACTORING**



**Disusun Oleh :
Atika Aji Hadiyani
2211104003
SE-06-01**

**Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING FAKULTAS
INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

TUGAS REFACTORING

1. Identifikasi Bad Smell

a. God Object

- Kelas Song menangani terlalu banyak tanggung jawab sekaligus, termasuk informasi lagu, artis, dan album.
- Seharusnya informasi terkait artis dan album dikelola dalam kelas terpisah untuk meningkatkan modularitas.

b. Primitive Obsession

- Genre direpresentasikan menggunakan tipe data int, yang membuat kode kurang jelas dan berpotensi menyebabkan error.
- Penggunaan angka sebagai representasi genre membuat kode sulit dibaca dan tidak fleksibel.

c. Long Method

- Metode printInfo memiliki banyak percabangan yang membuatnya sulit dipahami dan dirawat.
- Menggunakan angka untuk menentukan tingkat detail informasi (detailLevel) bukanlah pendekatan yang optimal.

d. Feature Envy

- Informasi album dan artis tidak seharusnya dikelola langsung dalam kelas Song, tetapi dalam kelas terpisah.
- Pemisahan ini akan meningkatkan keterbacaan dan keterpisahan logika bisnis.

2. Refactoring yang Dilakukan

Untuk meningkatkan struktur dan keterbacaan kode, dilakukan beberapa perubahan berikut:

a. Membuat Enum Genre

- Menggunakan enum Genre untuk menggantikan penggunaan angka dalam menentukan genre lagu.
- Enum memberikan representasi yang lebih jelas dan mengurangi risiko kesalahan.

```
// Enum untuk Genre
enum Genre {
    |   undefined, POP, ROCK, HIP_HOP, RNB, JAZZ, INSTRUMENTALS, CLOWNCORE;
}
```

b. Membuat Kelas Artist dan Album

- Kelas Artist menyimpan informasi tentang artis, termasuk nama, alias, dan gambar.
- Kelas Album menyimpan informasi album, termasuk nama album dan cover.
- Pemisahan ini meningkatkan prinsip Single Responsibility (SRP) sehingga setiap kelas hanya memiliki satu tanggung jawab utama.

```
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 getArtistInfo() {  
        return "Artist: " + name + (alias.isEmpty() ? "" : " (AKA: " + alias + ")");  
    }  
}
```

```
class Album {  
  
    private String name;  
    private String coverUrl;  
  
    public Album(String name, String coverUrl) {  
        this.name = name;  
        this.coverUrl = coverUrl;  
    }  
  
    public String getAlbumInfo() {  
        return "Album: " + name;  
    }  
}
```

c. Menyederhanakan Metode printInfo

- Menghilangkan percabangan yang kompleks dengan memanfaatkan metode pada kelas Artist dan Album.
- Detail informasi ditampilkan sesuai dengan detailLevel yang diberikan.

```
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) {  
        System.out.println(artist.getArtistInfo());  
    }  
    if (detailLevel >= 2 && album != null) {  
        System.out.println(album.getAlbumInfo());  
    }  
}
```

3. Hasil Run

```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 6\MKEPL\Assignment> javac Song.java  
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 6\MKEPL\Assignment> cd ..  
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 6\MKEPL> java Assignment.Song  
Song Title: Love Story  
Release Year: 2008  
Genre: POP  
Artist: Taylor Swift  
Album: Fearless
```

Kode yang sudah di refactoring bisa dilihat di repository GitHub berikut:

[Link Github](#)

4. Kesimpulan

Dengan refactoring ini, kode menjadi lebih modular, mudah diperluas, dan lebih mudah dibaca.

- Modularisasi: Dengan pemisahan kelas Artist dan Album, kode lebih terorganisir dan mudah dikelola.
- Peningkatan Keterbacaan: Penggunaan enum Genre menghilangkan kebutuhan akan angka sebagai indikator genre.
- Maintainability: Metode printInfo lebih sederhana, membuatnya lebih mudah diperbarui dan diperluas.