

MANAJEMEN KONFIGURASI DAN EVOLUSI PERANGKAT LUNAK

TUGAS W04 - PRAKTIKUM REFACTORING



Disusun Oleh:

Maria Nathasya Desfera Pangestu

2211104008

SE0601

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

Berdasarkan class Song.Java yang tercantum pada assignment ini lakukan hal-hal berikut ini:

1. Identifikasi bad smell yang ada pada class tersebut!
2. Lakukan refactoring terhadap class Song.Java sesuai dengan bad smell yang dilakukan, silahkan buat method dan class baru jika dibutuhkan!

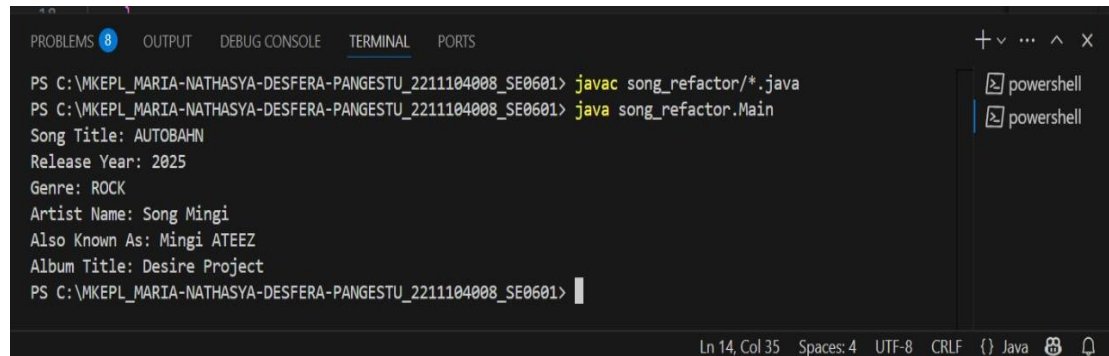
Laporkan bad smell yang ditemukan dan refactoring yang dilakukan dalam sebuah file pdf dan link [github](#).

Jawab:

1. Identifikasi bad smell pada class tersebut adalah:
 - Large class/God object: Class song memiliki terlalu banyak kategori dalam satu class seperti menyimpan data lagu, informasi album, artis, dan genre.
 - Primitive obsession: Genre dipresentasikan sebagai integer, dimana ini membuat rawan kesalahan seperti angka yang tidak valid atau tidak bisa digunakan.
 - Duplicate code: Pada printInfo() terlalu banyak pengulangan kode untuk mencetak lagu dan juga harus mengecek satu persatu.
 - Feature envy: Class song hanya mengakses atribut seperti artistName, artisAlias, albumName, dan lainnya secara langsung.
2. Refactoring terhadap class Song.Java sesuai dengan bad smell tersebut yaitu:
 - Memisahkan kelas Artist dan Album menjadi kelas sendiri-sendiri agar menangani data masing-masing, sehingga kelas Song hanya menyimpan objek Artist dan Album saja.
 - Mengubah Genre menjadi enum Genre agar hanya nilai valid yang bisa digunakan, sehingga mencegah kesalahan input. Contohnya Genre.Rock, dan lain sebagainya.
 - Memakai metode khusus di kelas Artist dan Album untuk mencetak informasi mereka sendiri. printInfo() hanya untuk memanggil printArtistInfo() dan printAlbumInfo().
 - Menambah metode printArtistInfo() di Artist dan printAlbumInfo() di Album, sehingga pada Song hanya perlu memanggilnya saja. Kelas Song hanya focus pada data lagu.
 - Setelah itu lakukan uji kode dengan menjalankan perintah pada terminal yaitu sebagai berikut:

Mengetikkan perintah “javac song_refactor/*.java” untuk melakukan kompilasi pada seluruh class. Kemudian ketikkan “java song_refactor.Main” untuk menjalankan program.

Hasil Outputnya akan seperti gambar berikut.



The screenshot shows a terminal window with the following output:

```
PS C:\MKEPL_MARIA-NATHASYA-DESFERA-PANGESTU_2211104008_SE0601> javac song_refactor/*.java
PS C:\MKEPL_MARIA-NATHASYA-DESFERA-PANGESTU_2211104008_SE0601> java song_refactor.Main
Song Title: AUTOBAHN
Release Year: 2025
Genre: ROCK
Artist Name: Song Mingi
Also Known As: Mingi ATEEZ
Album Title: Desire Project
PS C:\MKEPL_MARIA-NATHASYA-DESFERA-PANGESTU_2211104008_SE0601>
```

The terminal window has tabs for PROBLEMS (8), OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The status bar at the bottom indicates 'Ln 14, Col 35', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Java'.

3. Link github:

https://github.com/Marianaferaz/MKEPL_MARIA-NATHASYA-DESFERA-PANGESTU_2211104008_SE0601.git

Screenshot hasil setelah di refactoring

Class Song

```
1 // Source code is decompiled from a .class file using FernFlower decompiler.
2 package song_refactor;
3
4 public class Song {
5     private String id;
6     private String title;
7     private String releaseYear;
8     private String musicFileURL;
9     private Genre genre;
10    private Album album;
11    private Artist artist;
12
13    public Song(String var1, String var2, String var3, String var4) {
14        this.id = var1;
15        this.title = var2;
16        this.releaseYear = var3;
17        this.musicFileURL = var4;
18        this.genre = Genre.UNDEFINED;
19    }
20
21    public void setAlbum(Album var1) {
22        this.album = var1;
23    }
24
25    public void setArtist(Artist var1) {
26        this.artist = var1;
27    }
28
29    public void setGenre(Genre var1) {
30        this.genre = var1;
31    }
32
33    public void printInfo(int var1) {
34        System.out.println("Song Title: " + this.title);
35        System.out.println("Release Year: " + this.releaseYear);
36        System.out.println("Genre: " + String.valueOf(this.genre));
37        if (var1 >= 1 && this.artist != null) {
38            this.artist.printArtistInfo();
39        }
40
41        if (var1 >= 2 && this.album != null) {
42            this.album.printAlbumInfo();
43        }
44    }
45 }
46 }
47
```

Class Album

```
1 // Source code is decompiled from a .class file using FernFlow decompiler.
2 package song_refactor;
3
4 public class Album {
5     private String name;
6     private String coverURL;
7
8     public Album(String var1, String var2) {
9         this.name = var1;
10        this.coverURL = var2;
11    }
12
13    public void printAlbumInfo() {
14        System.out.println("Album Title: " + this.name);
15    }
16 }
17
```

Class Artist

```
1 // Source code is decompiled from a .class file using FernFlow decompiler.
2 package song_refactor;
3
4 public class Artist {
5     private String name;
6     private String alias;
7     private String imageURL;
8
9     public Artist(String var1, String var2, String var3) {
10        this.name = var1;
11        this.alias = var2;
12        this.imageURL = var3;
13    }
14
15    public void printArtistInfo() {
16        System.out.println("Artist Name: " + this.name);
17        if (!this.alias.isEmpty()) {
18            System.out.println("Also Known As: " + this.alias);
19        }
20    }
21 }
22 }
23
```

Class Genre

```
1 // Source code is decompiled from a .class file using FernFlower decompiler.
2 package song_refactor;
3
4 public enum Genre {
5     UNDEFINED,
6     POP,
7     ROCK,
8     HIP_HOP,
9     RNB,
10    JAZZ,
11    INSTRUMENTALS,
12    CLOWNCORE;
13
14    private Genre() {
15    }
16 }
17
```

Class Main

```
1 // Source code is decompiled from a .class file using FernFlower decompiler.
2 package song_refactor;
3
4 public class Main {
5     public Main() {
6     }
7
8     public static void main(String[] var0) {
9         Song var1 = new Song("2", "AUTOBAHN", "2025", "autobahn.mp3");
10        Artist var2 = new Artist("Song Mingi", "Mingi ATEEZ", "image.jpg");
11        var1.setArtist(var2);
12        Album var3 = new Album("Desire Project", "cover.jpg");
13        var1.setAlbum(var3);
14        var1.setGenre(Genre.ROCK);
15        var1.printInfo(3);
16    }
17 }
18
```