

Project Pra-UTS Object Oriented Programming

Dosen : **Fikri Hasani**

Kelas : LG01-LEC

Anggota Kelompok :

1. 2440092151 - Daniel Widjaja
 2. 2440036882 - Derrick
 3. 2440030683 - Muhammad Fathariq Dimas Octaviandra
-

Kasus

Kasus project yang ingin kami buat adalah Weeb Tracker yaitu aplikasi untuk mencatat progress menonton Anime atau membaca Manga.

Latarbelakang dari kasus ini karena kami yang menyaksikan anime atau membaca manga seiring waktu lupa dengan anime terakhir yg kita saksikan dan episode berapa yang terakhir kita saksikan.

Kita juga dapat mentracking anime yang kita rencanakan untuk disaksikan kedepannya, ataupun melihat history anime yang sudah pernah kita saksikan.

Tidak hanya untuk anime, kita juga dapat tracking progress pembacaan Manga dan Light Novel.

Class Analysis

Rencana utama dari penyusunan class kami adalah Tracker Inherit → Anime, Manga & Light Novel.

Karena masing-masing Anime, Manga dan Light Novel memiliki komponen yang sama seperti judulnya, status (Planning / On Progress / Finished).

Setelah melakukan eksperimen berkala seiring pengerjaan. Kami menentukan entity sebagai berikut :

Anime	Manga	LightNovel
(properties) id namaSeries startDate status rating genre season totalEpisode currentEpisode	(properties) id namaSeries startDate status rating genre currentVolume currentChapter	(properties) id namaSeries startDate status rating genre currentVolume currentPage
(methods) printDetail()	(methods) printDetail()	(methods) printDetail()

Bisa dilihat ada banyak property dan method yang sama.

Untuk propertynya kita bisa naikkan ke sebuah class parent baru yaitu **Tracker**. Ini adalah salah satu teknik OOP yaitu **Inheritance**.

Untuk methodnya, kita bisa mengimplementasi salah satu prinsip OOP yaitu **Abstraction**. Setiap dari mereka dapat melakukan fungsi yang sama tetapi dengan isi yang berbeda.

Misalnya : **printDetail()** untuk class **Anime** menampilkan progress **totalEpisode** dan **currentEpisode** yg dimana tidak dimiliki oleh **Manga** ataupun **LightNovel**, maka method ini tidak bisa kita gabungkan dalam 1 parent.

Berikut gambaran class setelah memasukkan parent class

Parent Class

Tracker
id namaSeries startDate status rating genre
printDetail()

Child Classes

Inherit from class **Tracker**.

Anime
season
totalEpisode
currentEpisode
printDetail()

Manga
currentVolume
currentChapter
printDetail()

LightNovel
currentVolume
currentPage
printDetail()

Tambahan

Dalam proses pengerjaan ada juga method-method yang membantu dalam proses CRUD. Kami menggunakan **Interface** untuk membuat blueprint aturan method yg harus diimplementasi dalam sebuah model. Contohnya seperti find(), insert() dan delete().

Maka kami membuat interface sebagai berikut :

Controller
printAll() printByStatus() find(id) insert(tracker) delete(tracker)

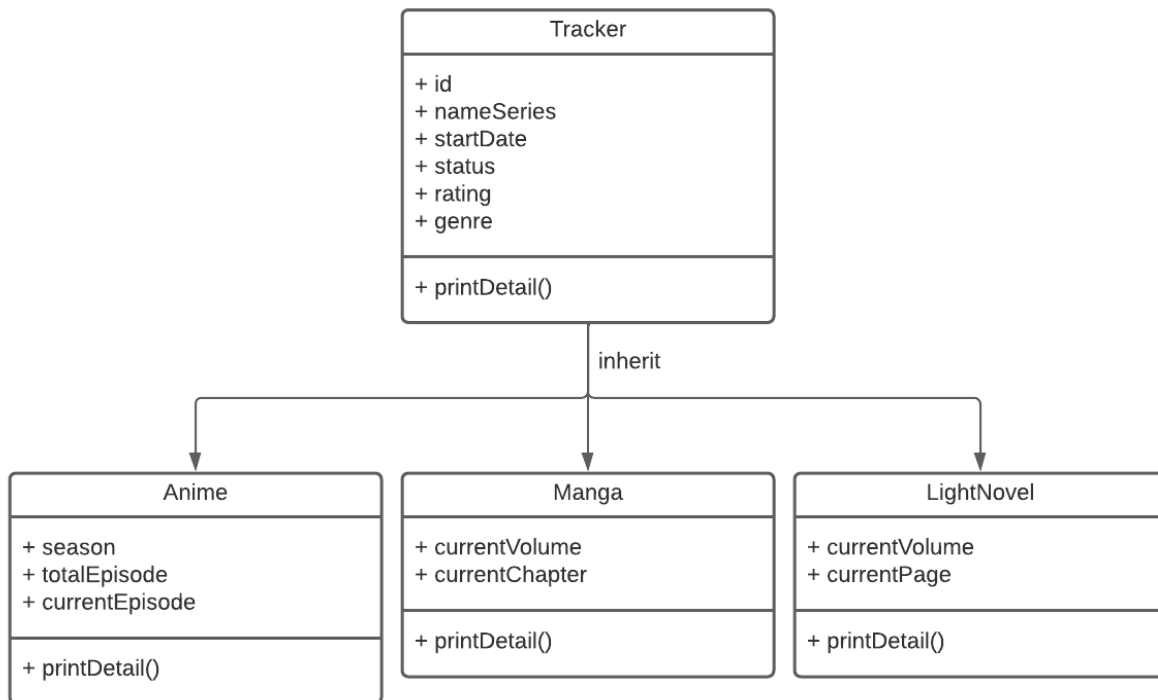
Supaya codingan lebih rapih juga, kami mengimplementasi masing-masing interface ini pada Controller supaya tidak memenuhi method pada Model.

Maka akan dibuat AnimeController, MangaController dan LightNovelController yg bekerja sebagai Service membantu model dalam hal-hal CRUD. Gambaran diagram akan ditampilkan pada **Class Diagram**

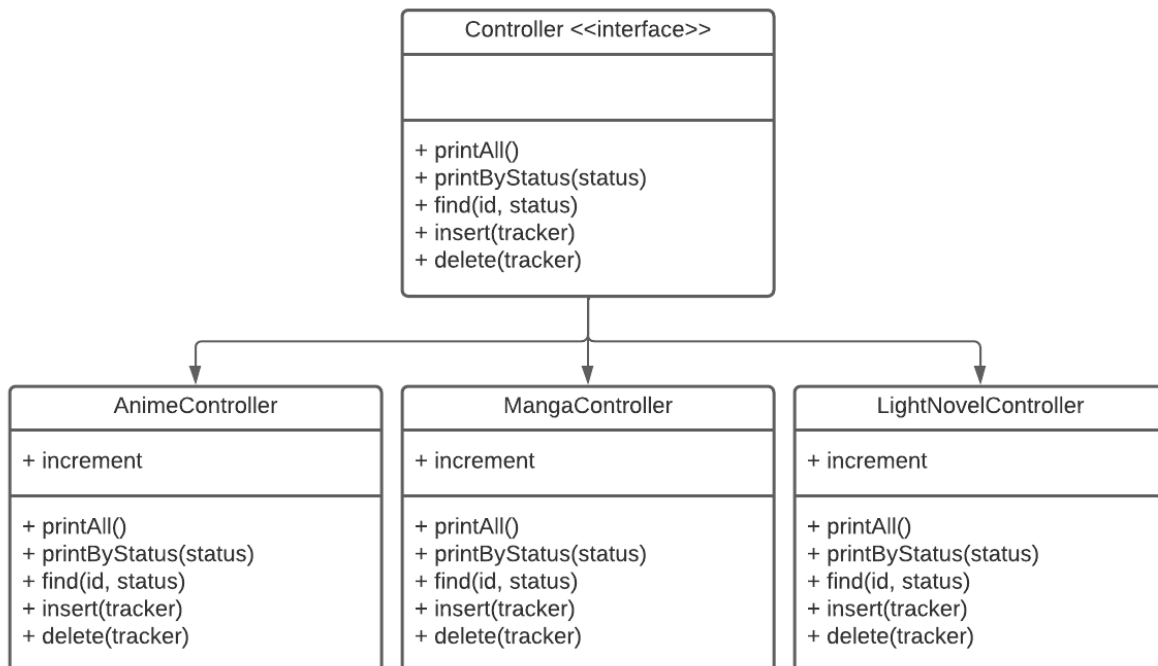
Class Diagram

Berikut class utama dari aplikasi kami

Web Tracker



Kemudian class service atau kami sebut dengan Controller untuk implement **Interface**



Sedikit note, untuk property increment disini gunanya untuk keep track increment id setiap kali kita melakukan insert ke ArrayList kumpulan data kami.

Pembagian Tugas per Anggota

Untuk analisa kasus kami memikirkan bersama, melakukan revisi sesuai dengan kasus yg kami hadapi.

Nama	Tugas
Daniel Widjaja	Pembuatan codingan class sesuai diagram, testing & fix bugs
Derrick	Pembuatan class analysis, class diagram, dan codingan pada bagian Controller.
Muhammad Fathariq Dimas Octaviandra	Pembuatan tampilan menu & menggunakan fungsi-fungsi yang ada pada controller & model.