*Repository Library Management API* adalah sebuah repository yang berguna untuk mengelola data buku dan penulis. Proyek ini dibuat menggunakan Laravel 11 dan Database MySQL/PostgreSQL

**Pilihan Desain**

**1. RESTful API dan Code Design Pattern**

* **Code Design Pattern**: Proyek menggunakan metode Repository Pattern. Untuk memishkan bisnis logic dan handle data. Ini menjadikan proyek ini dapat lebih mudah di maintenance untuk jangka Panjang. Jadi saya membuat lagi satu folder Repositories yang akan langsung berinteraksi ke database
* **RESTful API**: API yang digunakan menggunakan metode REST yang bertujuan untuk memudahkan endpoint pada frontend nantinya, sehinggan kita hanya perlu mengganti request methodnya.

**2. Eloquent ORM**

* Disini saya menggunakan Eloquent ORM untuk mengolah data dari database. Hal ini dikarenakan lebih mudah untuk mengelola data tersebut untuk ditampilkan, walaupun memang Eloquent ORM ini akan memerlukan waktu proses lebih lama daripada Query Builder Laravel

**3. Validasi**

* Adapun validasi yang saya gunakan adalah menggunakan Request Validasi dari Facade Laravel yaitu Validator. Hal ini memungkin lebih saya lebih mudah menghandle request yang masuk dan lebih mudah untuk saya customize error messagenya

**4. Testing**

* Untuk testing sendiri saya menggunakan phpunit dan bawaan Laravel testing.
* Saya juga menggunakan seeder untuk membuat data untuk melakukan pengetesan pada setiap api yang dibuat

**5. Optimization**

* **Eager loading**: Seperti yang saya jelaskan sebelumnya kalau disini saya menggunakan ORM bawaan Laravel. Kemudian dari data yang saya dapatkan adalah berupa eloquent query, dari hasil tersebut saya dapat melakukan relasi dengan leluasa selama mempunyai id dari relasi table yang diinginkan menggunakan eager loading. Sehingga kita melakukan relasi disaat kita memerlukannya saja.
* **Indexing dan uuid**: Pada migration database saya menggunakan uuid sebagai primary, hal ini saya maksudkan untuk melakukan privasi id pada setiap table. Kemudian saya juga menggunakan indexing kolom yang sering digunakan dalam pencarian, sehingga proses query database lebih cepat.
* **Caching:** Caching sendiri untuk Laravel 11 sudah menyediakan fitur caching, bahkan sudah ada table tersendiri yang dibuat, akan tetapi untuk proyek saat ini belum diterapkan.
* **Pagination:** Untuk mengatasi data yang begitu banyak maka perlu dilakukan pagination setiap data. Jadi data yang dimunculkan tidak harus semuanya terlebih dahulu. Akan tetapi dibuat perhalaman. Sehingga dapat mengatasi pemrosesan server yang berat.