

Nama: Sabrina Azahra

NIM : 20240801113

Mata Kuliah: CSF204 – Struktur Data

🎵 Analisis Sistem Inventaris Alat Musik 🎵

Inventarisasi alat musik merupakan salah satu aspek penting dalam pengelolaan studio, sekolah musik, maupun tempat penyimpanan peralatan seni. Dengan sistem yang baik, data alat musik bisa dicatat, dilacak, dan dikelompokkan dengan rapi sehingga memudahkan proses peminjaman, pengembalian, maupun pengecekan stok.

Di era digital seperti sekarang, pencatatan manual dirasa sudah kurang efektif. Maka dari itu, dibutuhkan sistem berbasis digital yang dapat menangani pengelolaan data secara efisien dan akurat. Proyek ini bertujuan untuk membangun Sistem Inventaris Alat Musik menggunakan framework Laravel yang mengimplementasikan berbagai struktur data seperti *Tree*, *Linked List*, *Array*, *Graph*, dan *Searching* untuk mendukung fungsi-fungsi sistem secara optimal.

Dengan adanya sistem ini, diharapkan pengelolaan alat musik menjadi lebih terstruktur, transparan, dan mempermudah petugas dalam mengambil keputusan berbasis data. Pendekatan menggunakan Laravel dan Filament Admin juga dipilih karena kemampuannya dalam membangun sistem yang cepat dan berskala, serta ramah pengguna. Berikut ini struktur penjelasan didalam studi kasus yang telah diibuat:

1. Struktur Data Tree – Kategori Alat Musik

Untuk mengelompokkan jenis alat musik, digunakan struktur data *Tree* (pohon). Setiap kategori utama seperti "Alat Petik", "Alat Tiup", dan "Alat Pukul" menjadi **parent node**, lalu di bawahnya terdapat **child node** seperti "Gitar", "Ukulele", "Saxophone", "Drum", dan lainnya.

Implementasi ini membantu dalam mengelola hirarki kategori, sehingga saat data ingin ditampilkan di form, admin bisa memilih kategori dan subkategori dengan struktur yang jelas.

Misalnya:

Alat Petik → Gitar, Bass, Ukulele
Alat Tiup → Saxophone, Seruling

Struktur pohon ini diterapkan di tabel `instrument_categories` dengan relasi `parent_id`, lalu dibuatkan Model dan Seeder-nya agar bisa langsung diuji saat development.

2. Struktur Data Linked List – Barang Masuk dan Keluar

Untuk mencatat alur keluar masuk alat musik, digunakan pendekatan *Linked List*. Setiap transaksi masuk atau keluar direkam secara berurutan berdasarkan waktu, seperti rantai yang saling terhubung.

Contoh penggunaannya ada di tabel `instrument_incomings` dan `instrument_outgoings`. Keduanya memiliki relasi ke `instrument_id` dan disusun berdasarkan `created_at`.

Keuntungan:

- Bisa dilihat urutan transaksi alat musik secara real-time
- Memudahkan pelacakan alat yang sering keluar/masuk

3. Struktur Data Array – Stok Alat Musik

Stok alat musik dikelola menggunakan array biasa saat ingin menampilkan data daftar alat dalam bentuk tabel atau form. Array mempermudah dalam memetakan data dari database lalu ditampilkan di frontend (dashboard admin).

Contohnya, data dari model `Instrument` diambil menggunakan query seperti:

php

SalinEdit

```
$alatMusik = Instrument::all();
```

Lalu dikirim ke view dan ditampilkan dalam bentuk tabel di Blade atau Filament Admin.

4. Struktur Data Searching – Pencarian Nama Alat

Fitur pencarian memanfaatkan metode searching sederhana berbasis query SQL, misalnya:

php

SalinEdit

```
$results = Instrument::where('name', 'LIKE', '%keyword%')->get();
```

Dengan ini, admin bisa mencari alat berdasarkan nama, kode, atau kategori. Fitur ini sangat membantu, terutama ketika data alat sudah sangat banyak dan butuh cara cepat untuk mencarinya.

5. Struktur Data Graph – Relasi Antar Alat Musik

Graph digunakan untuk menggambarkan hubungan antar alat musik, terutama alat yang digunakan dalam satu set. Contohnya, "Drum Set" memiliki relasi dengan "Cymbal", "Snare", dan "Stick".

Relasi ini ditanamkan melalui tabel pivot many-to-many antara `Instrument` dan `RelatedInstrument`.

Contoh:

php

SalinEdit

```
public function related()
{
    return $this->belongsToMany(Instrument::class, 'instrument_graph', 'from_id', 'to_id');
}
```

Dengan pendekatan graph, kita bisa melihat keterhubungan antar alat dan mengelola alat sebagai satu grup, bukan hanya item per item.

6. Model-View-Controller dan Filament Admin

Semua struktur data tersebut dibungkus dalam pola kerja MVC:

- **Model:** seperti Instrument, InstrumentCategory, InstrumentIncoming
- **View:** menggunakan Filament Admin untuk antarmuka CRUD
- **Controller:** menangani logika dan alur data dari view ke model

Filament Admin sangat membantu karena memungkinkan kita membuat form, tabel, relasi, dan validasi hanya dengan deklaratif — sangat cepat dan efisien untuk sistem inventaris.

7. Kesimpulan

Dengan memanfaatkan struktur data seperti tree, linked list, array, graph, dan searching, sistem ini menjadi jauh lebih fleksibel dan powerful. Laravel dan Filament Admin memberikan dukungan penuh terhadap arsitektur MVC dan memudahkan pengembangan antarmuka admin.