

# LAPORAN TAHAP 1: TUGAS BESAR STRUKTUR DATA

## ‘Music Player’ Console Application

By “ngantuk ya bram nama kelompoknya”  
Zaky Muhammad Fauzi 103052400064  
Brama Hartoyo 103052400030

---

## LATAR BELAKANG

Mata kuliah Struktur Data memiliki tujuan utama untuk memberikan pemahaman mendalam mengenai berbagai jenis penyimpanan dan pengelolaan data, seperti Linked List, Stack, Queue, Tree, dan Graph. Teori-teori ini tidak hanya bersifat konseptual, tetapi diharapkan mampu diterapkan oleh mahasiswa untuk menyelesaikan permasalahan nyata dalam pengembangan perangkat lunak.

Salah satu studi kasus yang relevan dengan pengelolaan data yang kompleks adalah aplikasi pemutar musik modern. Layanan seperti Spotify atau Apple Music menuntut pengelolaan koleksi lagu dalam skala besar, pengaturan playlist yang fleksibel, antrian pemutaran yang dinamis, serta kemampuan rekomendasi lagu. Oleh karena itu, dalam tugas besar ini, kami diminta untuk membangun sebuah replika aplikasi pemutar musik berbasis konsol.

Aplikasi ini akan mensimulasikan peran Admin (pengelola data) dan User (pengguna fitur), di mana efisiensi penyimpanan dan akses data menjadi kunci utama. Melalui proyek ini, kami akan melatih kemampuan analisis dalam memilih struktur data yang paling kompatibel dengan kebutuhan fitur, seperti penggunaan Multi Linked List untuk relasi data atau Queue untuk antrian lagu.

## ANALISIS MASALAH

Berdasarkan deskripsi tugas, kami mengidentifikasi beberapa tantangan dan kebutuhan sistem yang harus diselesaikan:

1. Manajemen data lagu yang kompleks aplikasi harus mampu menyimpan data lagu dengan atribut seperti Judul, Artis, Genre, Album, dan Tahun menggunakan tipe data bentukan (record). Tantangannya adalah bagaimana menyimpan data ini agar mudah ditambah, diubah, dan dihapus oleh Admin, serta dicari dengan cepat oleh User.
2. Relasi antara Library Utama dan Playlist User terdapat ketergantungan data yang ketat. Jika Admin mengubah atau menghapus data lagu di library utama, perubahan tersebut harus tercermin secara otomatis di semua playlist milik User yang memuat lagu tersebut. Ini memerlukan struktur data yang memungkinkan referensi atau pointer yang efisien tanpa menduplikasi data lagu secara fisik yang akan membosankan memori.
3. Logika pemutaran lagu (Next/Prev) Fitur navigasi lagu memiliki logika percabangan:
  - a. Jika sedang dalam mode Playlist, lagu selanjutnya mengikuti urutan playlist.
  - b. Jika sedang dalam mode Library (luar playlist), lagu selanjutnya haruslah lagu yang "mirip" (berdasarkan prioritas kesamaan Artis atau Genre). Hal ini menuntut penggunaan algoritma pencarian atau struktur data relasional (seperti Graph atau Tree) untuk menentukan kemiripan lagu.

## LAPORAN TAHAP 1: TUGAS BESAR STRUKTUR DATA

4. Peran ganda (Admin vs User) sistem harus membedakan akses. Admin memiliki hak mutlak terhadap database lagu, sementara User fokus pada manipulasi playlist pribadi dan pemutaran lagu.

### PERENCANAAN SOLUSI

Untuk menyelesaikan masalah di atas, hasil diskusi dengan asisten praktikum MRR, kelompok kami menyediakan fitur sebagai berikut.

1. Struktur data utama

Fitur	Struktur Data yang Dipilih
Penyimpanan Lagu	Doubly Linked List
Data Playlist User	Multi Linked List
Antrian Lagu	Queue
Riwayat Lagu	Stack
Rekomendasi Lagu Mirip	Graph/Pencarian Array

2. Rancangan tipe data (record/struct)

- a. ID Lagu (integer).
- b. Judul (string).
- c. Penyanyi (string).
- d. Genre (string).
- e. Tahun (integer).

3. Skenario algoritma “mirip” (similarity logic)

Untuk skenario algoritma lagu yang serupa, lagu yang dipilih dengan mengecek dari prioritas sebagai berikut.

- a. Prioritas 1: Lagu dengan nama penyanyi yang sama.
- b. Prioritas 2: Jika tidak ada penyanyi sama, cari lagu dengan genre yang sama.
- c. Fallback: Jika tidak ada keduanya, putar lagu secara acak/urutan ID.

4. Alat pengembangan

Kami akan menggunakan Github sebagai tempat penyimpanan kode, dokumentasi, dan kolaborasi versi (version control).

### LAMPIRAN

## LAPORAN TAHAP 1: TUGAS BESAR STRUKTUR DATA

