

LAPORAN TAHAP 2: 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 memboroskan 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 2: 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 FITUR

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).

IMPLEMENTASI ADT

Berdasarkan analisis masalah dan perencanaan fitur, struktur data yang dipilih adalah sebagai berikut:

1. Doubly Linked List (DLL) sebagai penyimpanan library lagu karena mendukung penambahan, penghapusan, dan update data di posisi mana pun dan dua arah yang memudahkan pencarian next/prev efisien untuk data berurutan (library lagu). Struktur record lagu:
 - a. ID Lagu (integer).
 - b. Judul (string).
 - c. Penyanyi (string).
 - d. Genre (string).
 - e. Tahun (integer).

LAPORAN TAHAP 2: TUGAS BESAR STRUKTUR DATA

2. Multi Linked List (MLL) playlist user karena playlist tidak menduplikasi data lagu playlist hanya menyimpan pointer ke node DLL (library) jika data lagu di library berubah maka playlist ikut berubah otomatis dan jika lagu dihapus admin maka referensi playlist menjadi aman dihapus. Struktur playlistnya:
 - a. song (adrLagu).
 - b. next (adrPlay).
3. Queue antrian lagu untuk simulasi tambahkan ke antrian pemutaran berikutnya. Mendukung enqueue (tambah ke belakang) dan dequeue (ambil dari depan).
4. Stack riwayat lagu untuk menyimpan lagu yang sudah diputar dengan fitur kembali ke lagu sebelumnya (undo play) dengan operasi push (tambah ke top) dan pop (kembali/undo).
5. Rekomendasi lagu berdasar aturan kemiripan untuk fitur Next di luar playlist dengan memperhatikan prioritas dari penyanyi yang sama atau genre yang sama dan fallback (lagu pertama di library).

IMPLEMENTASI FITUR

Berikut fitur lengkap yang telah selesai dibuat berdasarkan kode yang sudah disusun:

1. Fitur Peran Admin
 - a. Menambah lagu baru ke library. Lagu disimpan sebagai node baru DLL.
 - b. Melihat semua lagu DLL ditelusuri dari first hingga last untuk menampilkan seluruh lagu.
 - c. Mengubah data lagu dengan mengupdate record pada node DLL seperti judul, artis, genre, dan tahun. Karena playlist menyimpan pointer ke DLL maka playlist akan ikut berubah otomatis.
 - d. Menghapus lagu dengan menghapus node DLL jika lagu dihapus maka node playlist yang menunjuk ke lagu itu juga terhapus agar tidak ada pointer menggantung (dangling pointer).
2. Fitur Peran User:
 - a. Mencari lagu berdasarkan ID Pencarian dilakukan dengan traversal DLL. Hasilnya disimpan sebagai “current song”.
 - b. Memutar lagu / menghentikan lagu simulasi melalui pesan “Memutar lagu: Judul - Artis”.
 - c. Next lagu jika user tidak berada di playlist akan mencari lagu yang mirip berdasarkan artis jika tidak ada maka mencari berdasarkan genre jika tetap tidak ada maka fallback ke lagu pertama.
 - d. Next/Prev dalam playlist jika sedang di playlist, traversal MLL digunakan. (Prev playlist belum diterapkan, masuk fitur lanjutan).
 - e. Menambah lagu ke playlist dengan hanya menyimpan pointer ke node lagu di DLL.
 - f. Menghapus lagu dari playlist node MLL yang memuat pointer ke lagu tersebut dihapus.
 - g. Melihat isi playlist untuk menampilkan seluruh lagu yang ada di playlist.
3. Struktur Menu Main program menyediakan dua mode yaitu Admin Mode dan User Mode. Setiap mode memiliki menu masing-masing sesuai fitur.

LAPORAN TAHAP 2: TUGAS BESAR STRUKTUR DATA

KENDALA DAN SOLUSI

1. Kendala playlist harus mengikuti update library jika lagu diubah pada library, playlist tidak boleh menyimpan kopi data. Solusinya dengan playlist menyimpan pointer (adrLagu) ke node DLL maka playlist langsung berubah otomatis tanpa perlu diperbarui.
2. Kendala penghapusan lagu mengakibatkan pointer menggantung jika Admin menghapus lagu dari DLL, playlist yang menyimpan pointer tersebut menjadi invalid. Solusinya saat penghapusan lagu dilakukan, playlist dicari node yang menunjuk ke lagu tersebut node playlist ikut dihapus menghindari dangling pointer.
3. Kendala implementasi rekomendasi lagu tanpa menggunakan library search. Solusinya dibuat traversal manual dengan loop pertama mencari artis sama, loop kedua mencari genre sama, dan fallback ke first.

LAMPIRAN

[Link GITHUB](#)