

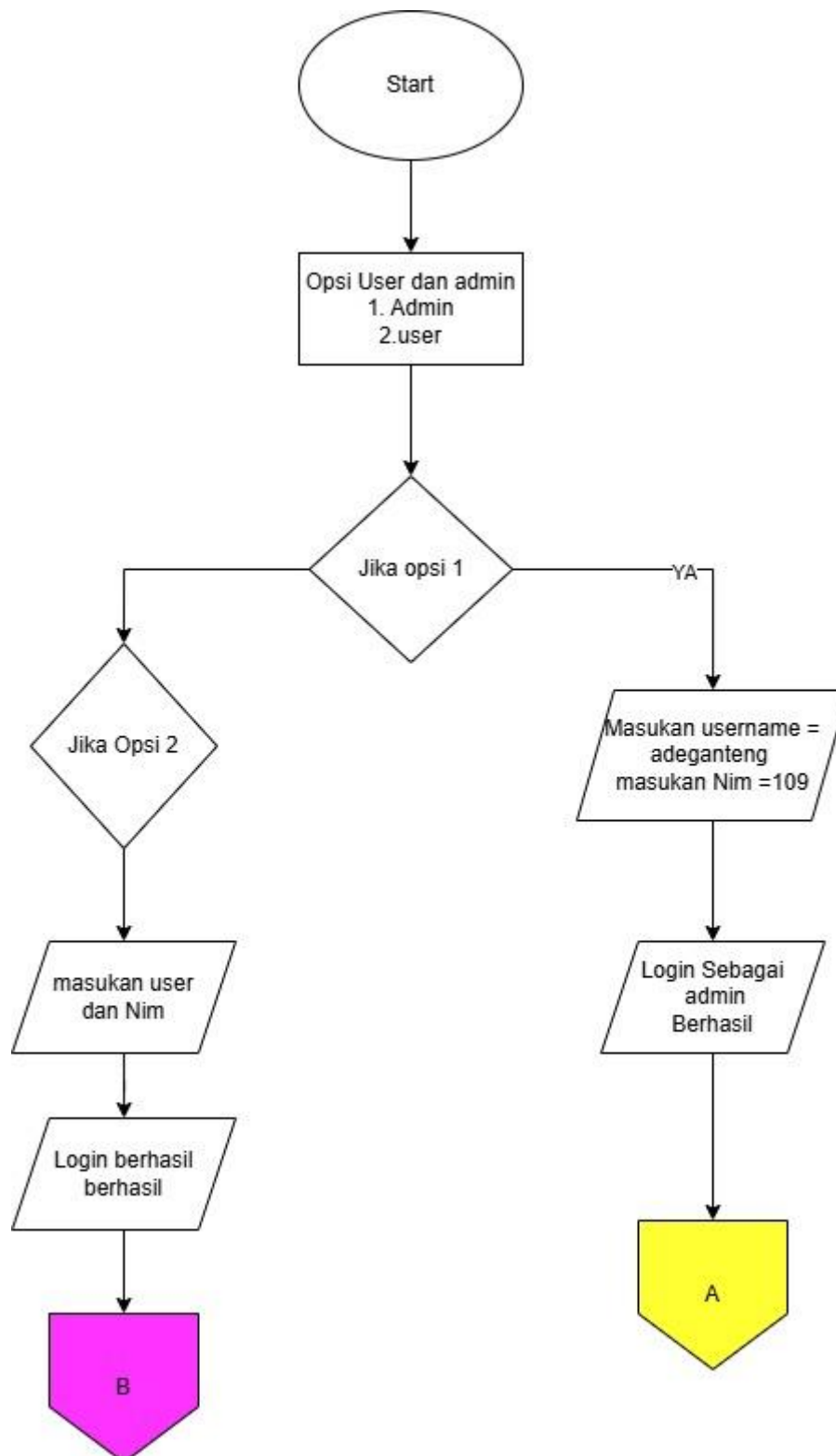
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT

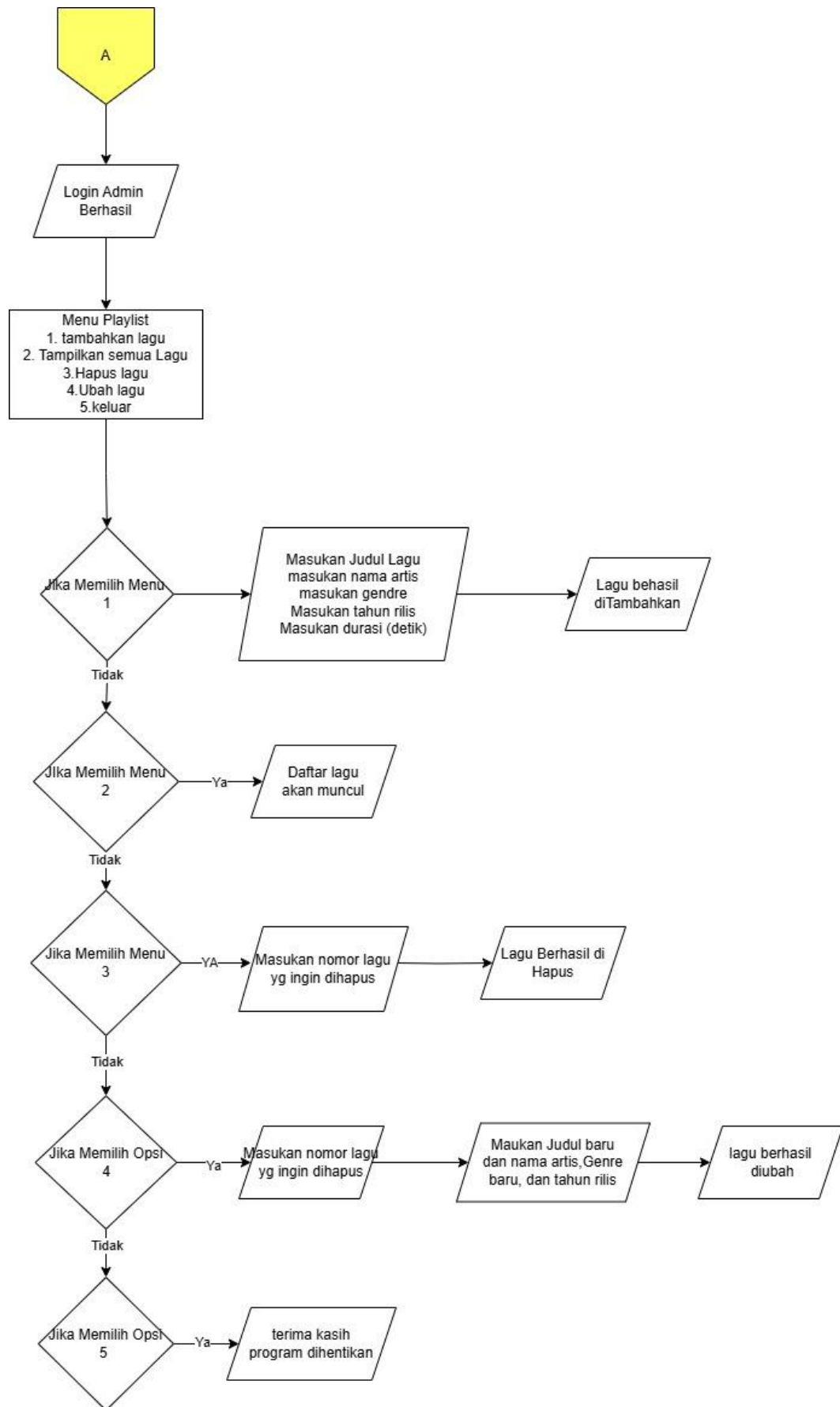


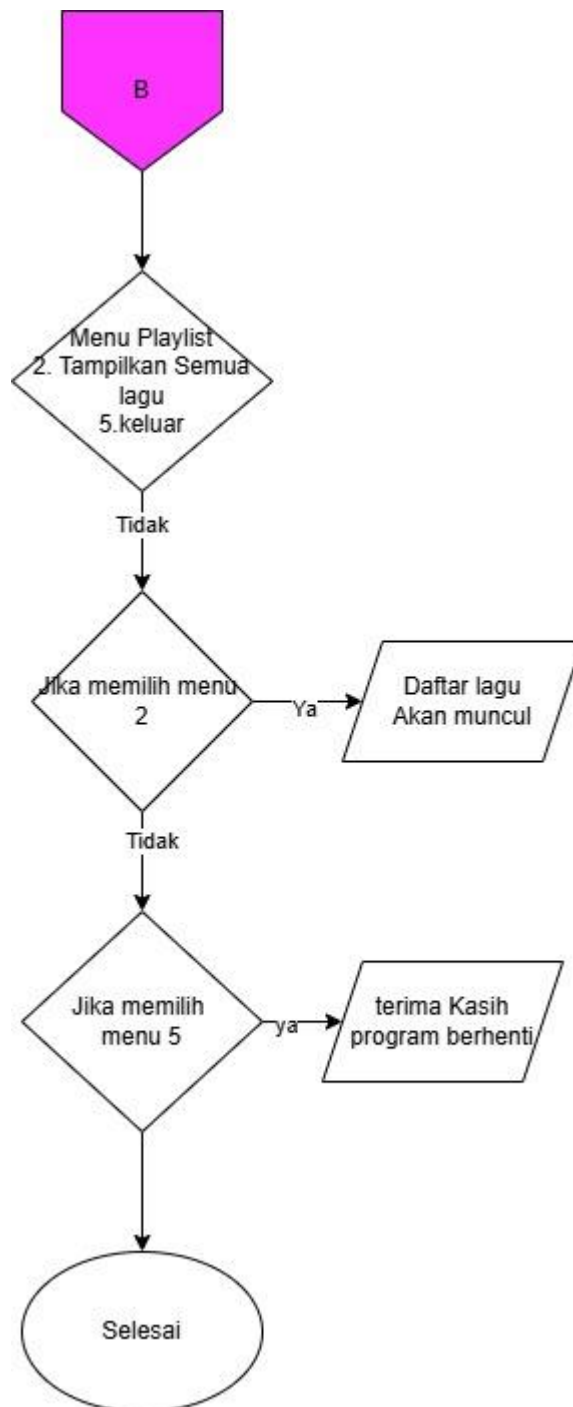
Disusun oleh:
Ade Pasiha Tangke Allo (2409106209)
Kelas (C2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart







2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini adalah sistem manajemen playlist lagu berbasis C++ yang memungkinkan pengguna untuk login sebagai admin atau pengguna biasa. Admin dapat menambahkan, menghapus, dan mengedit lagu, sedangkan pengguna hanya bisa melihat daftar lagu. Data lagu disimpan dalam array dengan batas maksimal 100 lagu. Program berjalan dalam loop hingga pengguna memilih keluar, dengan tampilan daftar lagu dalam format tabel yang rapi.

2.2. Penjelasan Alur & Algoritma

1. Algoritma

1. Deklarasi Awal

- Header file `#include <iostream>`, `#include <string>`, `#include <iomanip>` digunakan untuk input-output, manipulasi string, dan format tampilan tabel.
- Konstanta `MAX_SONGS` ditetapkan sebagai batas maksimum jumlah lagu (100).
- Struct `Song` digunakan untuk menyimpan informasi setiap lagu, yaitu title (judul lagu) dan artist (nama artis).

2. Login Sistem

a. Pemilihan Jenis Login

- Program menampilkan pilihan login:
 - a) 1: Login sebagai Admin
 - b) 2: Login sebagai Pengguna
- User memasukkan pilihan login (`loginChoice`).

b. Proses Login

- Pengguna memiliki maksimal 3 kali percobaan login (`MAX_ATTEMPTS = 3`).

- Admin harus memasukkan username = "adeganteng" dan NIM = "109". Jika sesuai, isAdmin diset ke true.
- Pengguna biasa dapat login dengan username dan NIM bebas.
- Jika gagal login dalam 3 kali percobaan, program akan dihentikan.

3. Menu Playlist Lagu

Setelah login, program menampilkan menu interaktif untuk mengelola playlist lagu:

a. Opsi Menu untuk Admin & Pengguna

Opsi	Menu	Hak Akses
1	Tambah Lagu	Hanya Admin
2	Tampilkan Semua Lagu	Admin & Pengguna
3	Hapus Lagu	Hanya Admin
4	Ubah Lagu	Hanya Admin
5	Keluar	Admin & Pengguna

4. Implementasi Menu

a. Tambah Lagu (Admin Saja)

- Cek apakah jumlah lagu sudah mencapai MAX_SONGS.
- Jika belum penuh:
 - a) Input judul lagu dan nama artis menggunakan getline(cin, ...).
 - b) Lagu baru ditambahkan ke dalam array songs.
 - c) songCount bertambah satu.

b. Tampilkan Semua Lagu

- Menampilkan daftar lagu dalam bentuk tabel dengan format rapi (setw(30)).

- Menggunakan loop for untuk mencetak semua lagu yang tersimpan.

c. Hapus Lagu (Admin Saja)

- Admin memasukkan nomor lagu yang ingin dihapus.
- Jika nomor valid:
 - a) Lagu pada posisi tersebut akan digeser ke atas untuk menutupi lagu yang dihapus.
 - b) songCount berkurang satu.
- Jika nomor tidak valid, muncul pesan error.

d. Ubah Lagu (Admin Saja)

- Admin memasukkan nomor lagu yang ingin diubah.
- Jika valid:
 - a) Input judul baru dan artis baru.
 - b) Data lagu diperbarui dalam array songs.
- Jika nomor tidak valid, muncul pesan error.

e. Keluar

- Jika pengguna memilih opsi 5, program akan mengakhiri eksekusi dengan pesan "Terima kasih! Program dihentikan."

5. Skenario Penggunaan

1. Admin Login

- a) Jika berhasil login, admin dapat menambah, menghapus, mengedit, dan melihat lagu.
- b) Jika gagal login 3 kali, program berhenti.

2. Pengguna Login

- a) Pengguna hanya bisa melihat daftar lagu.

b) Tidak bisa menambah, menghapus, atau mengedit lagu.

3. Menjalankan Menu

a) Program berjalan dalam looping while sampai pengguna memilih opsi keluar (5).

6. Kompleksitas Waktu

- Login: $O(1)$ → Hanya memeriksa username & NIM.
- Tambah Lagu: $O(1)$ → Menambahkan elemen ke array.
- Tampilkan Lagu: $O(n)$ → Iterasi sebanyak songCount.
- Hapus Lagu: $O(n)$ → Perlu menggeser elemen setelah penghapusan.
- Ubah Lagu: $O(1)$ → Hanya mengganti nilai elemen array.

4. Penjelasan Block Code

1. Header dan Deklarasi Konstanta

```
#include <iostream>
#include <string>
#include <iomanip>

using namespace std;

const int MAX_SONGS = 100; // Maksimal jumlah lagu
```

- **#include <iostream>**: Digunakan untuk input dan output (cin, cout).
- **#include <string>**: Digunakan untuk menangani string.
- **#include <iomanip>**: Digunakan untuk formatting output (setw, left).
- **using namespace std;**: Agar tidak perlu mengetik std:: di setiap penggunaan fungsi standar.
- **const int MAX_SONGS = 100;**: Konstanta yang menentukan jumlah maksimum lagu dalam playlist

2. Struktur Data untuk Lagu

```
struct Song {
    string title; // Judul lagu
    string artist; // Nama artis
```

- Struktur Song : struktur untuk informasi lagu, terdiri dari
 - Title untuk menyimpan judul lagu.
 - Artist untuk menyimpan nama artis.

3. Login Pengguna

```
string username, nim;
int loginAttempts = 0;
const int MAX_ATTEMPTS = 3; // Maksimal percobaan login
bool isAdmin = false; // Menandakan apakah pengguna adalah admin
```

- Username dan Nim; menyimpan input pengguna.
- LoginAttempts : menyimpan jumlah percobaan login.
- MAX_ATTEMPTS: Konstanta untuk batas percobaan login.
- isAdmin: Menandakan apakah pengguna masuk sebagai admin.

```
cout << "=== Pilihan Login ===" << endl;
cout << "1. Login sebagai Admin" << endl;
cout << "2. Login sebagai Pengguna" << endl;
cout << "Pilih jenis login (1/2): ";
int loginChoice;
cin >> loginChoice;
```

- Menampilkan Pilihan Login (admin atau pengguna biasa).
- Loginchoise : menyimpan pilihan pengguna.

Proses Login dengan validasi:

```
while (loginAttempts < MAX_ATTEMPTS) {
    cout << "Masukkan Username: ";
    cin.ignore(); // Mengabaikan newline dari input sebelumnya
    getline(cin, username); // Menggunakan getline untuk username
    cout << "Masukkan NIM: ";
    cin >> nim;

    // Validasi login
    if (loginChoice == 1 && username == "adeganteng" && nim == "109") {
        cout << "Login sebagai Admin berhasil!" << endl;
        isAdmin = true; // Menandakan bahwa pengguna adalah admin
        break;
    } else if (loginChoice == 2) {
        cout << "Login sebagai Pengguna berhasil!" << endl;
        break; // Pengguna biasa dapat login dengan username dan NIM apa
saja
    } else {
        cout << "Username atau NIM salah. Coba lagi." << endl;
        loginAttempts++;
    }
}
```

- Admin hanya bisa login dengan username = "adeganteng" dan Nim = "109"

- Pengguna biasa bisa login dengan Username dan Nim apa aja,
- Jika gagal Login sebanyak 3 kali ,program akan keluar.

```

• if (loginAttempts == MAX_ATTEMPTS) {
•     cout << "Terlalu banyak percobaan. Program dihentikan." <<
    endl;
•     return 0;

```

4. Inisialisasi daftar Lagu

```

• Song songs[MAX_SONGS] = {
•     {"Kaulah hatiku", "Pashmina"},
•     {"Blue", "Eiffel 65"},
•     {"Sempurna", "Gigi"},
•     {"Pica Pica", "Krisdayanti"},
•     {"Kangen", "Dewa 19"},
•     {"Cold", "Maroon 5"},
•     {"Cry", "James Blunt"},
•     {"December", "Gigi"},
•     {"Yellow", "ColdPlay"},
•
•     };

```

- **songs[MAX_SONGS]**: Array untuk menyimpan lagu-lagu.
- **songCount = 8**: Jumlah lagu yang ada dalam daftar

5. Menu Utama

```

6. while (opsi != 5) {
7.     cout << "\n=== Menu Playlist ===" << endl;
8.     if (isAdmin) {
9.         cout << "1. Tambah Lagu" << endl;
10.        cout << "2. Tampilkan Semua Lagu" << endl;
11.        cout << "3. Hapus Lagu" << endl;
12.        cout << "4. Ubah Lagu" << endl;
13.    } else {
14.        cout << "2. Tampilkan Semua Lagu" << endl; // Pengguna
        hanya bisa melihat
15.    }
16.    cout << "5. Keluar" << endl;
17.    cout << "Pilih menu: ";
18.    cin >> opsi;

```

- Admin bisa Menambah, Menghapus, Mengubah, dan Melihat lagu.
- Pengguna bisa hanya melihat lagu.
- Menu berjalan dalam loop sehingga pengguna melihat opsi 5 / keluar.

6. Menangani Pilihan menu

a) Tambah lagu (Hanya untuk admin)

```

b)     if (opsi == 1) { // Tambah Lagu (hanya untuk admin)
c)         if (isAdmin) {
d)             if (songCount < MAX_SONGS) {
e)                 cout << "Masukkan Judul Lagu: ";
f)                 cin.ignore(); // Mengabaikan newline dari input
sebelumnya
g)                     getline(cin, songs[songCount].title);
h)                     cout << "Masukkan Nama Artis: ";
i)                     getline(cin, songs[songCount].artist);
j)                     songCount++;
k)                     cout << "Lagu berhasil ditambahkan!" << endl;
l)             } else {
m)                 cout << "Playlist sudah penuh!" << endl;
n)             }
o)         } else {
p)             cout << "Akses ditolak! Hanya admin yang dapat
menambah lagu." << endl;
q)         }

```

- Admin dapat menambahkan lagu baru jika Playlist belum penuh.

b) Tampilan semua lagu

```

} else if (opsi == 2) { // Tampilkan Semua Lagu
    cout << "\n=== Daftar Lagu ===" << endl;
    cout << left << setw(30) << "Judul" << setw(30) << "Artis" <<
endl;

    cout << string(60, '-') << endl;
    for (int i = 0; i < songCount; i++) {
        cout << left << setw(30) << songs[i].title << setw(30) <<
songs[i].artist << endl;
    }
}

```

- Menampilkan daftar lagu yang ada dalam playlist

- Menggunakan set () agar tampilan lebih rapi

c) Hapus Lagu

```
else if (opsi == 3) { // Hapus Lagu (hanya untuk admin)
    if (isAdmin) {
        int indexToDelete;
        cout << "Masukkan nomor lagu yang ingin dihapus (1-" <<
songCount << "): ";
        cin >> indexToDelete;
        if (indexToDelete > 0 && indexToDelete <= songCount) {
            for (int i = indexToDelete - 1; i < songCount - 1; i++) {
                songs[i] = songs[i + 1];
            }
            songCount--;
            cout << "Lagu berhasil dihapus!" << endl;
        } else {
            cout << "Nomor lagu tidak valid!" << endl;
        }
    } else {
        cout << "Akses ditolak! Hanya admin yang dapat menghapus
lagu." << endl;
    }
}
```

- Admin dapat menghapus lagu berdasarkan nomor urut.
- Lagu setelah digeser kekiri akan mengisi celah.

d) Ubah lagu (hanya untuk Admin)

```
else if (opsi == 4) { // Ubah Lagu (hanya untuk admin)
    if (isAdmin) {
        int indexToEdit;
        cout << "Masukkan nomor lagu yang ingin diubah (1-" <<
songCount << "): ";
        cin >> indexToEdit;
        if (indexToEdit > 0 && indexToEdit <= songCount) {
            cout << "Masukkan Judul Baru: ";
            cin.ignore(); // Mengabaikan newline dari input sebelumnya
            getline(cin, songs[indexToEdit - 1].title);
            cout << "Masukkan Nama Artis Baru: ";
            getline(cin, songs[indexToEdit - 1].artist);
            cout << "Lagu berhasil diubah!" << endl;
        } else {
            cout << "Nomor lagu tidak valid!" << endl;
        }
    }
}
```

```
    }  
  } else {  
    cout << "Akses ditolak! Hanya admin yang dapat mengubah lagu."  
<< endl;  
  }
```

- Admin dapat mengedit judul dan artis lagu yang sudah ada

4. Uji Coba dan Hasil Output

4.1 Uji Coba

A. 1. Uji Login

- Admin Login
 - a. Input: 1 (Pilih Admin)
 - b. Input Username: adeganteng
 - c. Input NIM: 109
 - d. Ekspektasi Hasil: "Login sebagai Admin berhasil!"
- Pengguna Biasa Login
 - a. Input: 2 (Pilih Pengguna)
 - b. Input Username: user123
 - c. Input NIM: 123456
 - d. Ekspektasi Hasil: "Login sebagai Pengguna berhasil!"
- Gagal Login (Salah Input 3 Kali)
 - a. Input: 1
 - b. Input Username/NIM salah sebanyak 3 kali
 - c. Ekspektasi Hasil: "Terlalu banyak percobaan. Program dihentikan."

2. Uji Menu Admin

- Menambahkan Lagu Baru
 - a. Pilih menu 1
 - b. Input judul lagu: Perfect
 - c. Input artis: Ed Sheeran
 - d. Ekspektasi Hasil: "Lagu berhasil ditambahkan!"

- Menampilkan Semua Lagu
 - a. Pilih menu 2
 - b. Ekspektasi Hasil: Tabel daftar lagu tampil di layar
- Menghapus Lagu
 - a. Pilih menu 3
 - b. Input nomor lagu yang valid
 - c. Ekspektasi Hasil: "Lagu berhasil dihapus!"
- Mengubah Lagu
 - a. Pilih menu 4
 - b. Input nomor lagu yang valid
 - c. Masukkan judul/artis baru
 - d. Ekspektasi Hasil: "Lagu berhasil diubah!"

3. Uji Menu Pengguna

- Menampilkan Semua Lagu
 - a. Pilih menu 2
 - b. Ekspektasi Hasil: Daftar lagu ditampilkan
- Mencoba Menambah, Menghapus, atau Mengubah Lagu
 - a. Pilih menu 1, 3, atau 4
 - b. Ekspektasi Hasil: "Akses ditolak! Hanya admin yang dapat melakukan tindakan ini."

4. Keluar Program

- Pilih menu 5
- Ekspektasi Hasil: "Terima kasih! Program dihentikan."

4.2 Hasil Output

A. Tampilan Pilihan Login

```
=== Pilihan Login ===  
1. Login sebagai Admin  
2. Login sebagai Pengguna  
Pilih jenis login (1/2): 1
```

Gambar 4.2 Tampilan Pilihan Login

B. Tampilan Login Sebagai Admin

```
Masukkan Username: adeganteng  
Masukkan NIM: 109  
Login sebagai Admin berhasil!  
  
=== Menu Playlist ===  
1. Tambah Lagu  
2. Tampilkan Semua Lagu  
3. Hapus Lagu  
4. Ubah Lagu  
5. Keluar  
Pilih menu:
```

Gambar 4.2 Tampilan Login Sebagai Admin

C. Tampilan Menu Playlist

```
=== Menu Playlist ===  
1. Tambah Lagu  
2. Tampilkan Semua Lagu  
3. Hapus Lagu  
4. Ubah Lagu  
5. Keluar  
Pilih menu: 1  
Masukkan Judul Lagu:
```

Gambar 4.2 Tampilan Menu Playlist

D. Menambahkan lagu

```
=== Menu Playlist ===
1. Tambah Lagu
2. Tampilkan Semua Lagu
3. Hapus Lagu
4. Ubah Lagu
5. Keluar
Pilih menu: 1
Masukkan Judul Lagu: sd
Masukkan Nama Artis: 12
Masukkan Genre: pop
Masukkan Tahun Rilis: 2017
Masukkan Durasi (dalam detik): 200
Lagu berhasil ditambahkan!

=== Menu Playlist ===
1. Tambah Lagu
2. Tampilkan Semua Lagu
3. Hapus Lagu
4. Ubah Lagu
5. Keluar
Pilih menu: 2

=== Daftar Lagu ===
```

Judul	Artis	Genre	Tahun	Durasi
Kaulah hatiku	Pashmina	Pop	2020	210
Blue	Eiffel 65	Dance	1999	180
Sempurna	Gigi	Rock	2000	240
Pica Pica	Krisdayanti	Pop	2001	200
Kangen	Dewa 19	Rock	1999	220
Cold	Maroon 5	Pop	2017	210
Cry	James Blunt	Pop	2004	240
December	Gigi	Rock	2000	180
sd	12	pop	2017	200

Gambar 4.2 Menambahkan Lagu

E. Menampilkan Semua Lagu

```
Selamat datang Cantik Ganteng rina!
>>>>> SILAHKAN PILIH YA CANTIK DAN GANTENG! <<<<<
1. Tambah lagu baru ke playlist
2. Lihat playlist
3. Edit lagu di playlist
4. Hapus lagu dari playlist
5. Logout
Pilih opsi: 4
Playlist kamu:
1. Hare With Me - d4vd
2. Somboddy's Pleasure - Aziz Hendra
3. Sempurna - Andra & The backbone
Lagu nomor berapa yang ingin dihapus Cantik Dan Ganteng!: 1
Apa kamu yakin ingin menghapus lagu ini?
1. Iya
2. Tidak
Pilih: 1
Lagu yang kamu pilih sudah dihapus dari playlist!

Selamat datang Cantik Ganteng rina!
>>>>> SILAHKAN PILIH YA CANTIK DAN GANTENG! <<<<<
1. Tambah lagu baru ke playlist
2. Lihat playlist
3. Edit lagu di playlist
4. Hapus lagu dari playlist
5. Logout
Pilih opsi: 2
Playlist kamu:
1. Somboddy's Pleasure - Aziz Hendra
2. Sempurna - Andra & The backbone
```

Gambar 4.2 Menampilkan semua Lagu

F. Hapus Lagu

```
=== Menu Playlist ===
1. Tambah Lagu
2. Tampilkan Semua Lagu
3. Hapus Lagu
4. Ubah Lagu
5. Keluar
Pilih menu: 3
Masukkan nomor lagu yang ingin dihapus (1-9): 5
Lagu berhasil dihapus!

=== Menu Playlist ===
1. Tambah Lagu
2. Tampilkan Semua Lagu
3. Hapus Lagu
4. Ubah Lagu
5. Keluar
Pilih menu: 2

=== Daftar Lagu ===
```

Judul	Artis	Genre	Tahun	Durasi
Kaulah hatiku	Pashmina	Pop	2020	210
Blue	Eiffel 65	Dance	1999	180
Sempurna	Gigi	Rock	2000	240
Pica Pica	Krisdayanti	Pop	2001	200
Cold	Maroon 5	Pop	2017	210
Cry	James Blunt	Pop	2004	240
December	Gigi	Rock	2000	180
sd	12	pop	2017	200

Gambar 4.2 Hapus Lagu

G. Mengubah Lagu

```
=== Menu Playlist ===
1. Tambah Lagu
2. Tampilkan Semua Lagu
3. Hapus Lagu
4. Ubah Lagu
5. Keluar
Pilih menu: 4
Masukkan nomor lagu yang ingin diubah (1-8): 4
Masukkan Judul Baru: Dj Aloy
Masukkan Nama Artis Baru: Aloy
Masukkan Genre Baru: Dj
Masukkan Tahun Rilis Baru: 2025
Masukkan Durasi Baru (dalam detik): 200
Lagu berhasil diubah!

=== Menu Playlist ===
1. Tambah Lagu
2. Tampilkan Semua Lagu
3. Hapus Lagu
4. Ubah Lagu
5. Keluar
Pilih menu: 2

=== Daftar Lagu ===
```

Judul	Artis	Genre	Tahun	Durasi
Kaulah hatiku	Pashmina	Pop	2020	210
Blue	Eiffel 65	Dance	1999	180
Sempurna	Gigi	Rock	2000	240
Dj Aloy	Aloy	Dj	2025	200
Cold	Maroon 5	Pop	2017	210
Cry	James Blunt	Pop	2004	240
December	Gigi	Rock	2000	180
sd	12	pop	2017	200

Gambar 4.2 Mengubah Lagu

H. Keluar

```
=== Menu Playlist ===
1. Tambah Lagu
2. Tampilkan Semua Lagu
3. Hapus Lagu
4. Ubah Lagu
5. Keluar
Pilih menu: 5
Terima kasih! Program dihentikan.
PS D:\GitHub\Praktikum-APL\Post-Test\Post-Test-2>
```

Gambar 4.2 Keluar

5. Git

```
hp@LAPTOP-QCIVAFNM MINGW64 /d/GitHub/Praktikum-APL
$ git config --global user.email "pasihaade@gmail.com"

hp@LAPTOP-QCIVAFNM MINGW64 /d/GitHub/Praktikum-APL
$ git init
Initialized empty Git repository in D:/GitHub/Praktikum-APL/.git/

hp@LAPTOP-QCIVAFNM MINGW64 /d/GitHub/Praktikum-APL (master)
$ git add .

hp@LAPTOP-QCIVAFNM MINGW64 /d/GitHub/Praktikum-APL (master)
$ git branch -M main

hp@LAPTOP-QCIVAFNM MINGW64 /d/GitHub/Praktikum-APL (main)
$ git remote add origin https://github.com/adepasiha/Praktikum-Apl.git

hp@LAPTOP-QCIVAFNM MINGW64 /d/GitHub/Praktikum-APL (main)
$ git commit -m "Ade ganteng"
[main (root-commit) 207074d] Ade ganteng
1 file changed, 74 insertions(+)
create mode 100644 Post-Test/Pots-Test-1/2409106109-AdePasihaTAngkeAllo-PT-1.cpp

hp@LAPTOP-QCIVAFNM MINGW64 /d/GitHub/Praktikum-APL (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 1.04 KiB | 533.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/adepasiha/Praktikum-Apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

hp@LAPTOP-QCIVAFNM MINGW64 /d/GitHub/Praktikum-APL (main)
$ |
```

Gambar 5 Tampilan Git Bash

1. Mengatur Email Global Untuk Git

\$ git config --global user "pasihaade@gmail.com"

-Perintah ini digunakan untuk mengatur email global Git, yang akan digunakan dalam setiap commit yang dibuat oleh pengguna.

2. Menginstal Repository Git

\$ Git init

-Peintah ini menginisialisasi (membuat) Repository Git dalam folder Praktikum APL

Jika sudah ada repository Git folder tersebut maka Git hanya akan mengingatkan bahwa repository sudah ada

3. Menambahkan File ke Staging Area

\$ git add.

- Perintah ini menambahkan semua file yang ada didalam folder ke staging area
- Staging area adalah tempat sementara sebuah file dikomit kedalam repository

4. Menambahkan Remote Repository (Gagal Karena Sudah ada)

\$ git Remote add origin <https://Github.com/adelesiha/praltikum-apl>

- perintah ini digunakan untuk menambahkan repository remote dengan nama origin
- Error: “remote origin already exists”, ini terjadi karena sebelumnya sudah ada repository remote yang sudah dikaitkan dengan nama origin.

5. Membuat Commit dengan pesan “Update”

\$ git commit -m “Update”

- Perintah ini menyimpan perubahan dalam repository dengan commit dan pesan “Update”.

-File yang dicommit:

- Post-test/Post-test-1/2409106109-AdePasihaTangkeAllo-PT-2.cpp
- Post-test/Post-test-1/2409106109-AdePasihaTangkeAllo-PT-2.exe

6. Mendorong (Push) Perubahan ke Repository Remote

\$ git push -u origin main

- Perintah ini mengunggah (push) perubahan ke repository remote pada branch main
- Karena ini adalah push pertama, flag -u digunakan untuk mengatur branch local main agar terhubung dengan branch main di remote repository.
- Proses ini terjadi:
 - Menghitung objek (Counting objects: 6).
 - Mengeompresi objek sebelum mengunggahnya.
 - Menulis (mengunggah) objek ke github.
 - Menampilkan informasi bahwa branch main sekarang dilacak oleh remote repository origin/main.