

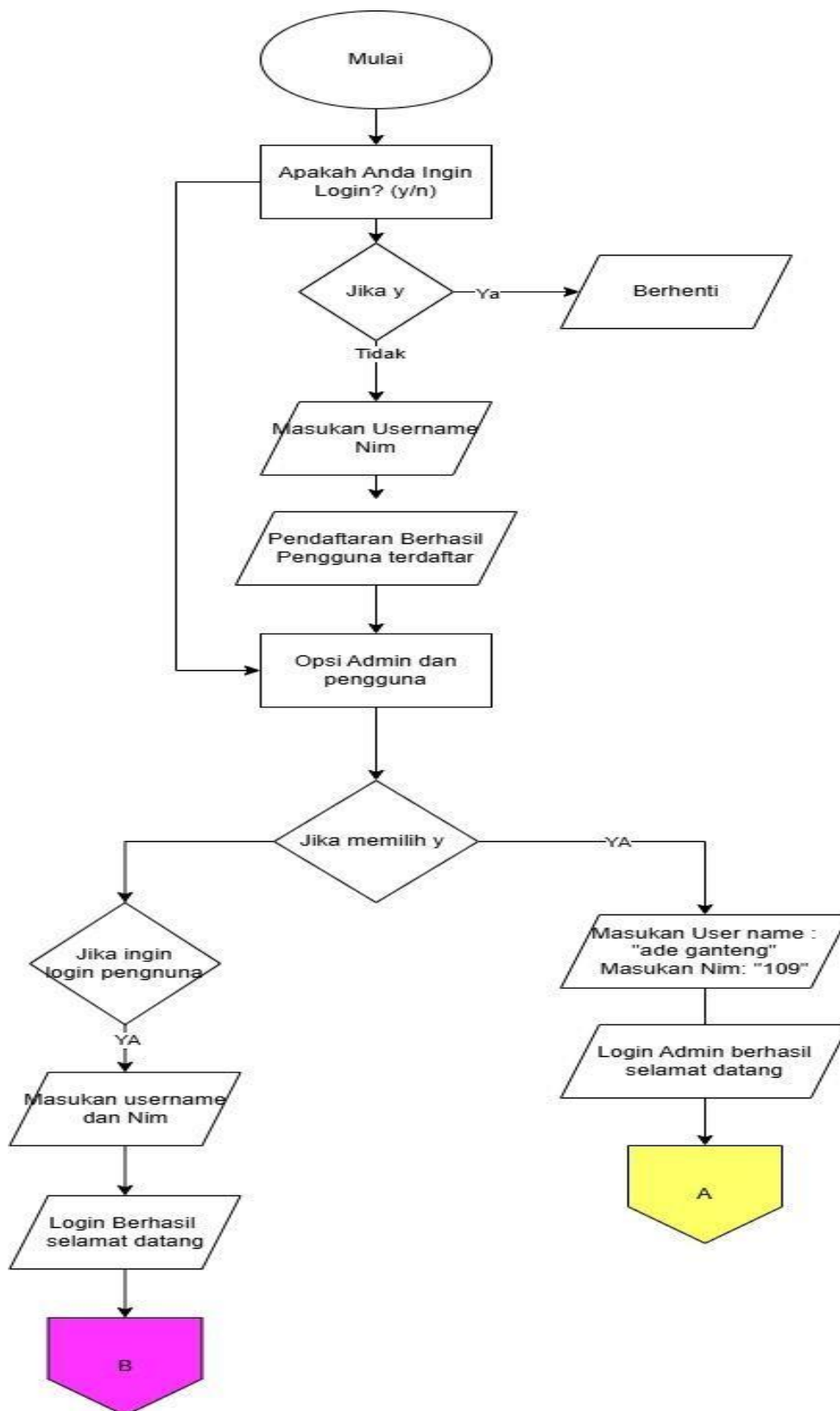
**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**

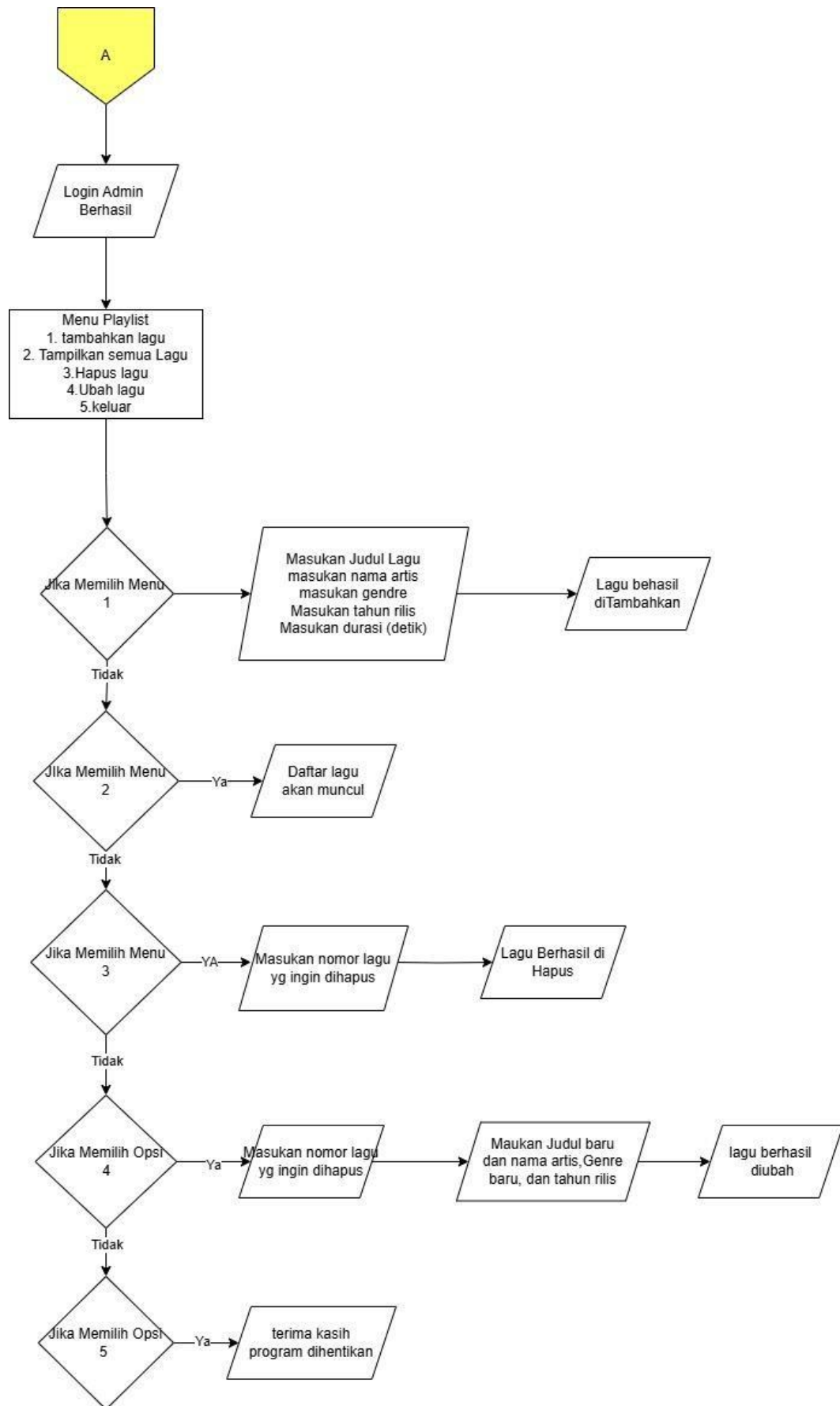


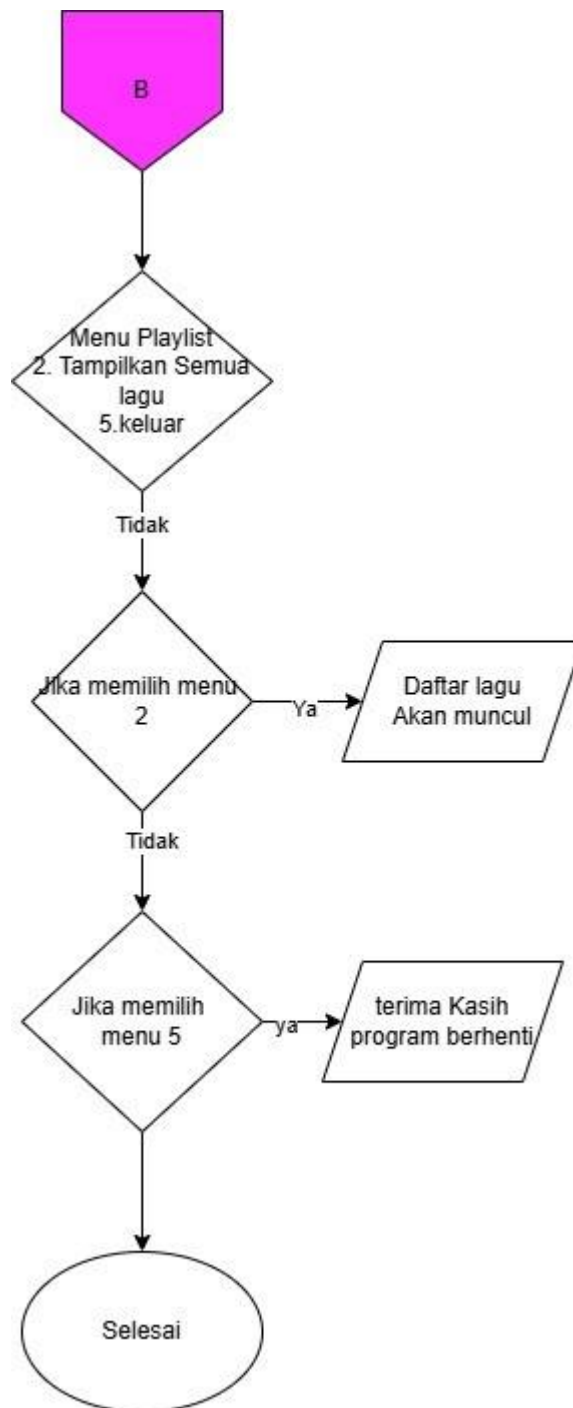
**Disusun oleh:**  
**Ade Pasiha Tangke Allo (2409106109)**  
**Kelas (C2 '24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA 2025**

## 1. Flowchart







## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini saya buat untuk mengelola playlist lagu, di mana pengguna bisa melihat daftar lagu, dan kalau dia login sebagai admin, bisa juga menambah, menghapus, atau mengubah data lagu. Setiap user harus login dulu, dan kalau username-nya "adeganteng" dengan NIM "109", dia otomatis jadi admin. Data lagu disimpan dalam array, dan ditampilkan dalam bentuk tabel. Untuk menampilkan semua lagu, saya pakai rekursi biar lebih menarik. Program ini juga punya fitur registrasi user, pembatasan jumlah lagu dan user, serta validasi login maksimal 3 kali. Intinya, program ini semacam sistem playlist sederhana dengan perbedaan hak akses antara user biasa dan admin.

## 3. Source Code

### 1. Librari dan Konstanta

```
2. #include <iostream>
3. #include <string>
4. #include <iomanip>
5.
```

- #include untuk mengimpor Pustaka C++ standar.
- Iostream input/output (misal; cin, cout).
- String untuk menggunakan tipe data string
- Iomanip untuk mengatur format output (setw untuk lebar kolom).

```
const int MAX_SONGS = 100;
const int MAX_USERS = 100;
```

- Konstanta maksimum jumlah lagu dan pengguna.

## 2. Struct (Struktur Data)

```
struct Song {  
    string title;  
    string artist;  
};
```

- Struct Song untuk menyimpan data lagu (judul dan artis)

```
struct Song {  
    string title;  
    string artist;  
};
```

- Struct User menyimpan info pengguna, termasuk isAdmin (0 = biasa, 1 = admin).

## 3. Rekursif untuk Menampilkan Lagu

```
void tampilkanLaguRekursif(Song songs[], string genres[], int releaseYears[],  
int durations[], int index, int total) {  
    if (index >= total) return;  
    cout << left << setw(30) << songs[index].title  
        << setw(30) << songs[index].artist  
        << setw(20) << genres[index]  
        << setw(15) << releaseYears[index]  
        << setw(10) << durations[index] << endl;  
    tampilkanLaguRekursif(songs, genres, releaseYears, durations, index + 1,  
total);  
}
```

- Menggunakan daftar lagu secara rekursif
- Digunakan untuk menampilkan lagu satu per satu berdasarkan indeks.

#### 4. Prosedur menampilkan lagu

```
void tampilkanSemuaLagu(Song songs[], string genres[], int releaseYears[], int
durations[], int songCount) {
    cout << "\n=== Daftar Lagu ===" << endl;
    cout << left << setw(30) << "Judul" << setw(30) << "Artis" << setw(20) <<
"Genre" << setw(15) << "Tahun" << setw(10) << "Durasi" << endl;
    cout << string(105, '-') << endl;
    tampilkanLaguRekursif(songs, genres, releaseYears, durations, 0,
songCount); // Gunakan rekursi
}
```

- Menampilkan header table lagu..
- Memanggil TampilkaLaguRekursif

#### 5. Prosedur menambahkan lagu

```
6. void tambahLagu(Song songs[], string genres[], int releaseYears[], int
durations[], int &songCount) {
7.     if (songCount < MAX_SONGS) {
8.         cout << "Masukkan Judul Lagu: ";
9.         cin.ignore();
10.        getline(cin, songs[songCount].title);
11.        cout << "Masukkan Nama Artis: ";
12.        getline(cin, songs[songCount].artist);
13.        cout << "Masukkan Genre: ";
14.        getline(cin, genres[songCount]);
15.        cout << "Masukkan Tahun Rilis: ";
16.        cin >> releaseYears[songCount];
17.        cout << "Masukkan Durasi (dalam detik): ";
18.        cin >> durations[songCount];
19.
20.        songCount++;
21.        cout << "Lagu berhasil ditambahkan!" << endl;
22.    } else {
23.        cout << "Playlist sudah penuh!" << endl;
24.    }
25.}
```

- Menambahkan lagu baruke array songs, beserta genre, tahun, dan durasinya

- Cek batas maksimum dengan songCount.

## 6. Prosedur Menghapus lagu

```

7. void hapusLagu(Song songs[], string genres[], int releaseYears[], int
   durations[], int &songCount) {
8.     int indexToDelete;
9.     cout << "Masukkan nomor lagu yang ingin dihapus (1-" << songCount <<
   "): ";
10.    cin >> indexToDelete;
11.    if (indexToDelete > 0 && indexToDelete <= songCount) {
12.        for (int i = indexToDelete - 1; i < songCount - 1; i++) {
13.            songs[i] = songs[i + 1];
14.            genres[i] = genres[i + 1];
15.            releaseYears[i] = releaseYears[i + 1];
16.            durations[i] = durations[i + 1];
17.        }
18.        songCount--;
19.        cout << "Lagu berhasil dihapus!" << endl;
20.    } else {
21.        cout << "Nomor lagu tidak valid!" << endl;
22.    }
23.}

```

- Menghapus lagu berdasarkan nomor urut (input user)
- Menggeser elemen setelah lagu yang dihapus agar array tetap rapat.
- Update songCount

## 7. Prosedur Mengubah lagu

```

void ubahLagu(Song songs[], string genres[], int releaseYears[], int
durations[], int songCount) {
    int indexToEdit;
    cout << "Masukkan nomor lagu yang ingin diubah (1-" << songCount << "): ";
    cin >> indexToEdit;
    if (indexToEdit > 0 && indexToEdit <= songCount) {
        cin.ignore();
        cout << "Masukkan Judul Baru: ";
        getline(cin, songs[indexToEdit - 1].title);
    }
}

```



```

        cout << "Masukkan Nama Artis Baru: ";
        getline(cin, songs[indexToEdit - 1].artist);
        cout << "Masukkan Genre Baru: ";
        getline(cin, genres[indexToEdit - 1]);
        cout << "Masukkan Tahun Rilis Baru: ";
        cin >> releaseYears[indexToEdit - 1];
        cout << "Masukkan Durasi Baru (dalam detik): ";
        cin >> durations[indexToEdit - 1];

        cout << "Lagu berhasil diubah!" << endl;
    } else {
        cout << "Nomor lagu tidak valid!" << endl;
    }
}

```

- Mengubah data lagu berdasarkan nomor lagu.
- Update data langsung pada array sesuai input baru dari user.

## 8. fungsi Login

```

bool login(User users[], int userCount, string username, string nim, int
&isAdmin) {
    for (int i = 0; i < userCount; i++) {
        if (users[i].username == username && users[i].nim == nim) {
            isAdmin = users[i].isAdmin;
            return true;
        }
    }
    return false;
}

```

- Mengecek apakah Username dan Nim cocok dengan yang ada di array user.
- Jika cocok isAdmin diisi dengan user dan fungsi mengembalikan true.

## 9. Inisialisasi Lagu

```
int releaseYears[MAX_SONGS] = {
    2020, 1999, 2000, 2001, 1999, 2017, 2004, 2000, 2000
};
int durations[MAX_SONGS] = {
    210, 180, 240, 200, 220, 210, 240, 180, 210
};

int songCount = 9;
```

- Data awal 9 lagu dimasukan langsung ke array.

## 10. Registrasi Pengguna

```
char registerChoice;
cout << "Apakah Anda ingin mendaftar? (y/n): ";
cin >> registerChoice;
cin.ignore();

while (registerChoice == 'y' && userCount < MAX_USERS) {
    cout << "Masukkan Username: ";
    getline(cin, users[userCount].username);
    cout << "Masukkan NIM: ";
    getline(cin, users[userCount].nim);
    users[userCount].isAdmin = (users[userCount].username == "adeganteng"
    && users[userCount].nim == "109") ? 1 : 0;
    userCount++;
    cout << "Pendaftaran berhasil! Pengguna terdaftar: " << users[userCount
    - 1].username << endl;
    cout << "Apakah Anda ingin mendaftar pengguna lain? (y/n): ";
    cin >> registerChoice;
    cin.ignore();
}
```

- Mentediakan fitur pendaftaran User.
- Jika username ‘adegantang’ dan nim “109”, otomatis jadi admin.

## 11. Login Pengguna

```
while (loginAttempts < MAX_ATTEMPTS) {
    cout << "\n=== Login ===" << endl;
    cout << "Masukkan Username: ";
    getline(cin, username);
    cout << "Masukkan NIM: ";
    getline(cin, nim);

    if (login(users, userCount, username, nim, isAdmin)) {
        cout << "Login berhasil! Selamat datang, " << username << "!" <<
endl;
        isLoggedIn = true;
        break;
    } else {
        cout << "Username atau NIM salah. Coba lagi." << endl;
        loginAttempts++;
    }
}
```

- Maksimum 3 Percobaan login.
- Jika berhasil, isLoggedIn = true.

## 12. Menu Interaktif

```
while (opsi != 5) {
    cout << "\n=== Menu Playlist ===" << endl;
    if (isAdmin == 1) {
        cout << "1. Tambah Lagu" << endl;
        cout << "2. Tampilkan Semua Lagu" << endl;
        cout << "3. Hapus Lagu" << endl;
        cout << "4. Ubah Lagu" << endl;
    } else {
        cout << "2. Tampilkan Semua Lagu" << endl;
    }
}
```

- Tampilan Menu sesuai role:
  - Admin bisa menambah, menghapus, mengubah, dan melihat lagu.
  - User bisa hanya bisa melihat lagu

- Input pilihan menu → progress berdasarkan opsi

### 13. Keluar Program

```
while (opsi != 5) {  
    cout << "\n=== Menu Playlist ===" << endl;  
    if (isAdmin == 1) {  
        cout << "1. Tambah Lagu" << endl;  
        cout << "2. Tampilkan Semua Lagu" << endl;  
        cout << "3. Hapus Lagu" << endl;  
        cout << "4. Ubah Lagu" << endl;  
    } else {  
        cout << "2. Tampilkan Semua Lagu" << endl;  
    }  
}
```

- Keluar Program

## 4. Hasil Output

### 4.1 Hasil Output

#### 1. Login Admin dan Menu Playlist

```
Apakah Anda ingin mendaftar? (y/n): y
Masukkan Username: adeganteng
Masukkan NIM: 109
Pendaftaran berhasil! Pengguna terdaftar: adeganteng
Apakah Anda ingin mendaftar pengguna lain? (y/n): n

=== Login ===
Masukkan Username: adeganteng
Masukkan NIM: 109
Login berhasil! Selamat datang, adeganteng!

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

Gambar 1 Tampilan Admin dan Menu Playlist

#### 2. Tambahkan Lagu

```
=== Menu Playlist ===
1. Tambah Lagu
2. Tampilkan Semua Lagu
3. Hapus Lagu
4. Ubah Lagu
5. Keluar
Pilih menu: 1
Masukkan Judul Lagu: replay
Masukkan Nama Artis: adez
Masukkan Genre: pop
Masukkan Tahun Rilis: 2016
Masukkan Durasi (dalam detik): 200
Lagu berhasil ditambahkan!
```

Gambar 2 Tambahkan Lagu

### 3. Tampilan semua lagu

```
=== 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
Yellow	ColdPlay	Alternative	2000	210
replay	adez	pop	2016	200

Gambar 3 Tampilan Semua Lagu

### 4. Hapus Lagu

```
Pilih menu: 3
Masukkan nomor lagu yang ingin dihapus (1-10): 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
Yellow	ColdPlay	Alternative	2000	210
replay	adez	pop	2016	200

Gambar 4 Tampilan Hapus Lagu

## 5. Ubah Lagu

```
Pilih menu: 4
Masukkan nomor lagu yang ingin diubah (1-9): 2
Masukkan Judul Baru: love me
Masukkan Nama Artis Baru: Lil Weine
Masukkan Genre Baru: pop
Masukkan Tahun Rilis Baru: 2017
Masukkan Durasi Baru (dalam detik): 300
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
love me	Lil Weine	pop	2017	300
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
Yellow	ColdPlay	Alternative	2000	210
replay	adez	pop	2016	200

Gambar 5 Tampilan Ubah lagu

## 6. Keluar Program

```
Pilih menu: 5
Terima kasih! Program dihentikan.
PS D:\GitHub\Praktikum-APL\Post-Test\Post-Test-3>
```

Gambar 6 Tampilan Keluar Program

## 7. Login Pengguna

```
Apakah Anda ingin mendaftar? (y/n): y
Masukkan Username: claudia
Masukkan NIM: 111
Pendaftaran berhasil! Pengguna terdaftar: claudia
Apakah Anda ingin mendaftar pengguna lain? (y/n): n

=== Login ===
Masukkan Username: claudia
Masukkan NIM: 111
Login berhasil! Selamat datang, claudia!
```

Gambar 7 Tampilan Pengguna

## 8. Tampilan semua Lagu

```
=== Menu Playlist ===
2. Tampilkan Semua 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
Yellow	ColdPlay	Alternative	2000	210

Gambar 8 Tampilan semua Lagu

## 9. Keluar Program

```
Pilih menu: 5
Terima kasih! Program dihentikan.
PS D:\GitHub\Praktikum-APL\Post-Test\Post-Test-3>
```

Gambar 9 Keluar dari Program



## 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-Ap1.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-Ap1.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](mailto: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

-Perintah 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/adepasiha/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

-Kerena 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 (Rnumering objects: 6).
- Mengeompresi objek sebelum mengunggahnya.
- Menulis (mengunggah) objek ke gothub.
- Menampilkan informasi bahwa branch main sekarang dilacak oleh remote repository origin/main.