

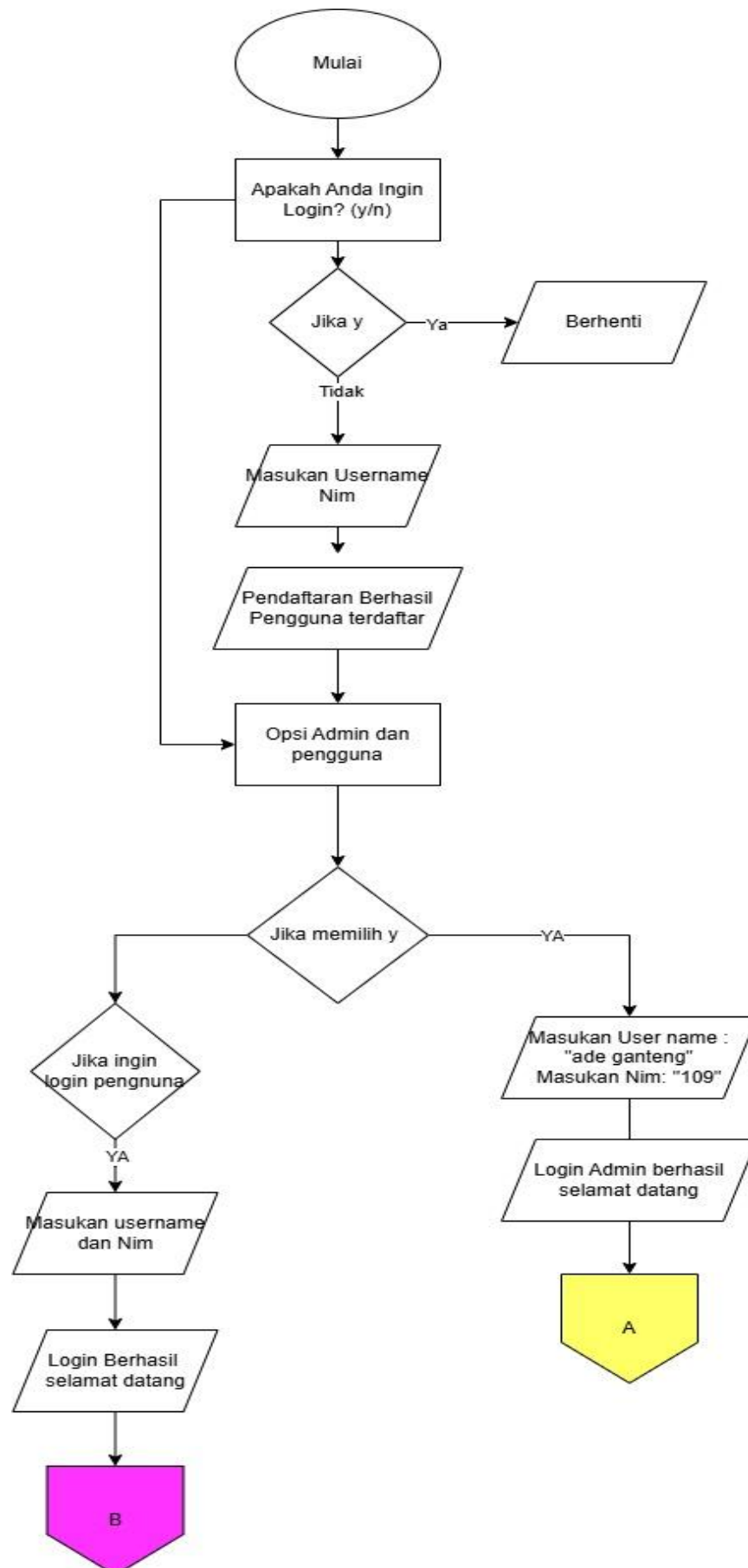
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT

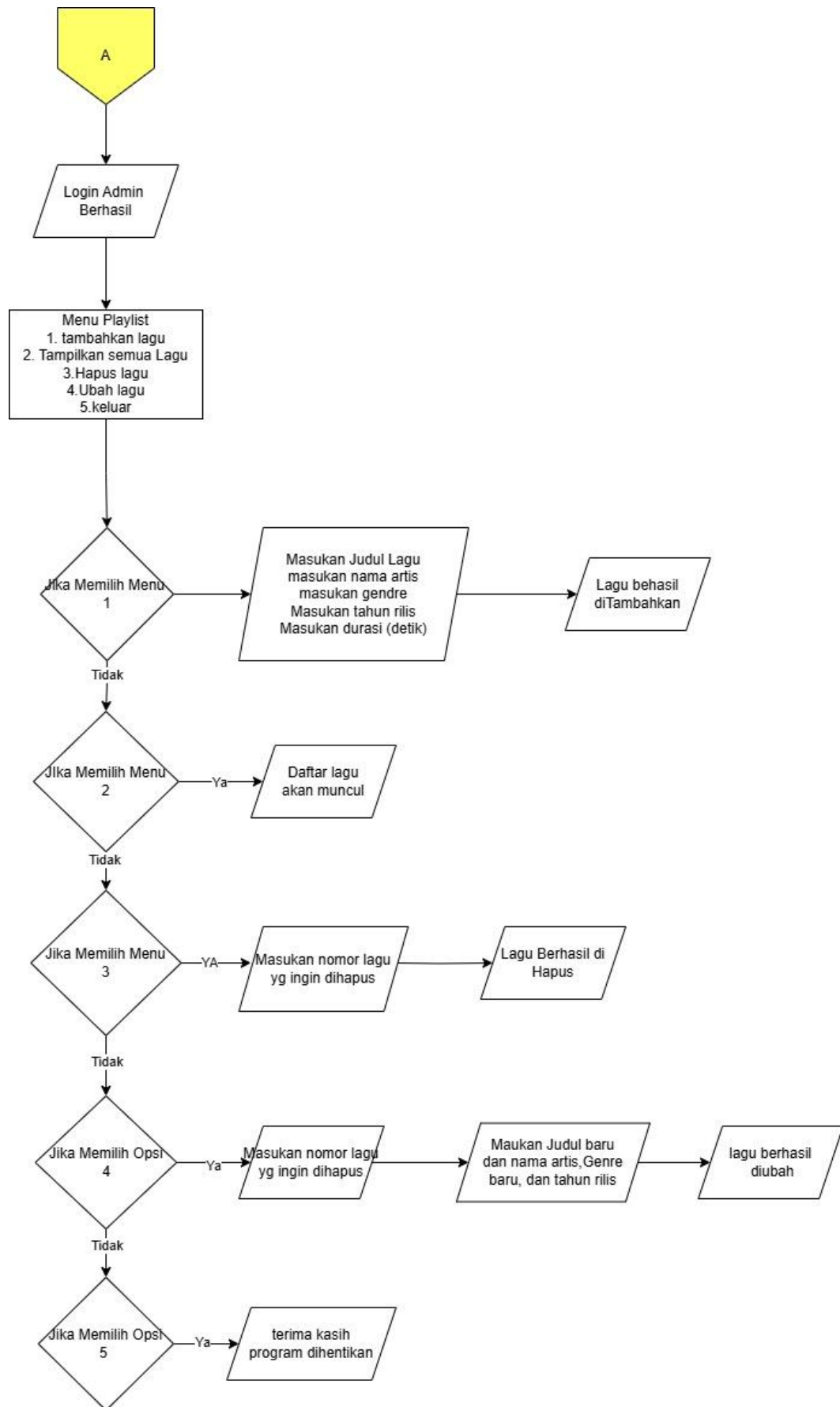


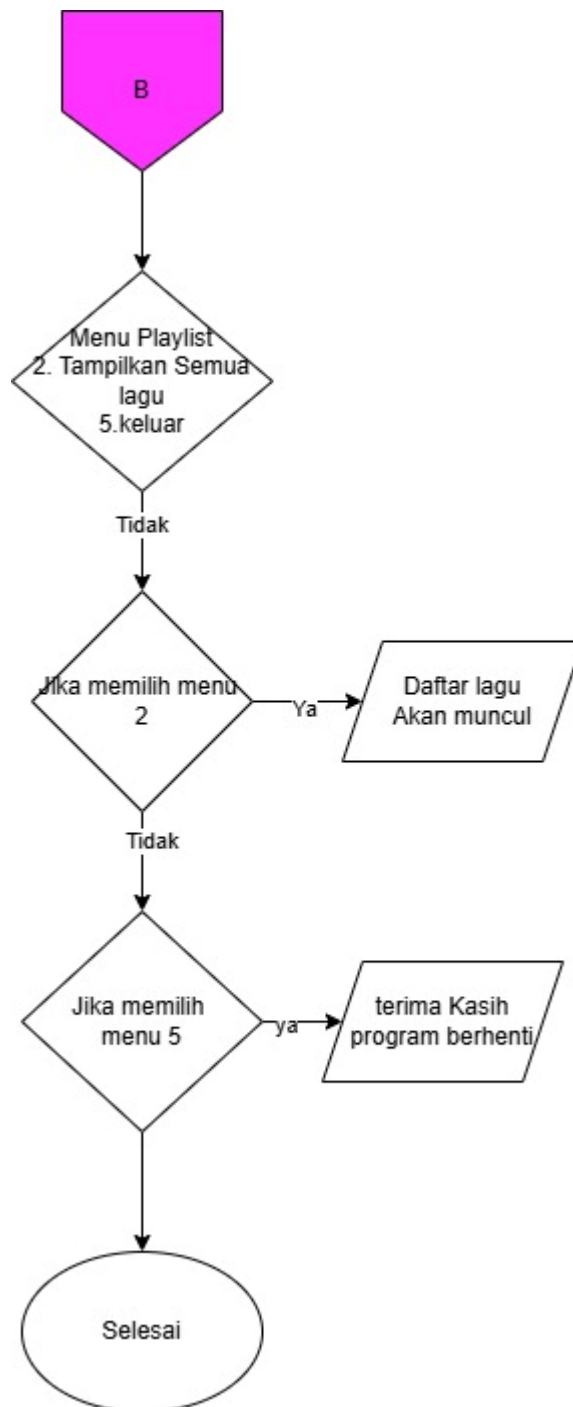
Disusun oleh:
Ade Pasiha Tangke Allo (2409106109)
Kelas (C2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart







2. Analisis Program

2.1 Deskripsi Singkat Program

Dalam aplikasi ini, terdapat dua struktur data utama: Song, yang menyimpan informasi tentang lagu seperti judul dan artis, serta User, yang menyimpan informasi pengguna, termasuk username, NIM, dan status apakah pengguna tersebut adalah admin atau bukan.

Pengguna dapat mendaftar dengan memasukkan username dan NIM. Jika kombinasi tersebut sesuai dengan yang telah ditentukan, pengguna akan diakui sebagai admin. Setelah proses pendaftaran, pengguna dapat melakukan login dengan username dan NIM yang telah didaftarkan. Program membatasi jumlah percobaan login hingga tiga kali untuk mencegah akses yang tidak sah.

Setelah berhasil login, pengguna akan disajikan dengan menu yang berbeda tergantung pada status mereka. Admin memiliki akses untuk menambah, menampilkan, menghapus, dan mengubah lagu dalam playlist, sementara pengguna biasa hanya dapat melihat daftar lagu yang ada. Lagu-lagu tersebut disimpan dalam array, dan informasi tambahan seperti genre, tahun rilis, dan durasi juga disertakan.

Ketika admin menambah atau mengubah lagu, mereka diminta untuk memasukkan informasi yang relevan, dan program akan menampilkan daftar lagu dengan format yang rapi, mencakup judul, artis, genre, tahun rilis, dan durasi. Pengguna dapat keluar dari program kapan saja, dan program juga membatasi jumlah lagu dan pengguna yang dapat didaftarkan hingga maksimal 100. Dengan demikian, aplikasi ini memberikan antarmuka yang sederhana namun efektif untuk mengelola playlist lagu dengan kontrol akses berdasarkan status pengguna.

1. Penjelasan Block Code

1. Library dan Konstanta

```
2. #include <iostream>
3. #include <string>
4. #include <iomanip>
5.
6. using namespace std;
7.
8. const int MAX_SONGS = 100; // Maksimal jumlah lagu
9. const int MAX_USERS = 100; // Maksimal jumlah pengguna
```

- Library digunakan untuk fungsi input/output (iostream), string (string), dan formatting output (iomanip).
- Konstanta MAX_SONGS dan MAX_USERS menentukan jumlah maksimal lagu dan pengguna.

2. Struktur Data

```
3. struct Song {
4.     string title; // Judul lagu
5.     string artist; // Nama artis
6. };
7.
8. // Struktur untuk menyimpan informasi pengguna
9. struct User {
10.    string username; // Nama pengguna
11.    string nim;      // Nomor Induk Mahasiswa
12.    int isAdmin;
```

- Struct song menyimpan informasi lagu : Title (judul) & artist (penyayi)
- Struct User menyimpan informasi pengguna username, nim dan admin untuk menentukan hak akses (Admin atau bukan).

3. Inisialisasi Data 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"},
};

```

- Array song berisi daftar lagu awal.

```

string genres[MAX_SONGS] = {
    "Pop", "Dance", "Rock", "Pop", "Rock", "Pop", "Pop", "Rock",
    "Alternative"
};
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
};

```

- Tambah data untuk setiap lagu: genre, tahun rilis, dan durasi dalam detik.

4. Registrasi Pengguna

```

int userCount = 0; // Jumlah pengguna terdaftar
User users[MAX_USERS];

```

- Array users menyimpan daftar pengguna.
- userCount menyimpan jumlah pengguna yang telah mendaftar

```

char registerChoice;
cout << "Apakah Anda ingin mendaftar? (y/n): ";
cin >> registerChoice;
cin.ignore(); // Mengabaikan newline dari input sebelumnya

```

- Meminta pengguna untuk **mendaftar atau tidak** (y/n).
- cin.ignore() digunakan untuk menghindari masalah saat menggunakan getline().

```
while (registerChoice == 'y' && userCount < MAX_USERS) {
    cout << "Masukkan Username: ";
    getline(cin, users[userCount].username);
    cout << "Masukkan NIM: ";
    getline(cin, users[userCount].nim)
```

- Meminta input username & NIM hingga batas pengguna terpenuhi.

```
if (users[userCount].username == "adeganteng" && users[userCount].nim ==
"109") {
    users[userCount].isAdmin = 1; // Admin
} else {
    users[userCount].isAdmin = 0; // Pengguna biasa
}
```

- Jika username = "adeganteng" dan NIM = "109", pengguna dijadikan admin

```
cout << "Apakah Anda ingin mendaftar pengguna lain? (y/n): ";
cin >> registerChoice;
cin.ignore(); // Mengabaikan newline dari input sebelumnya
```

- Menyimpan apakah ingin mendaftarkan pengguna lain.

5. Login Pengguna

```
string username, nim;
int loginAttempts = 0;
const int MAX_ATTEMPTS = 3; // Maksimal percobaan login
int isLoggedIn = 0; // Menandakan apakah pengguna berhasil login
int isAdmin = 0;
```

- Variabel Login untuk menyimpan input dan batas percobaan login.

```
while (loginAttempts < MAX_ATTEMPTS) {
    cout << "\n=== Login ===" << endl;
    cout << "Masukkan Username: ";
```



```
getline(cin, username);
cout << "Masukkan NIM: ";
getline(cin, nim);
```

- Menyimpan Username dan Nim untuk Login

```
for (int i = 0; i < userCount; i++) {
    if (users[i].username == username && users[i].nim == nim) {
        cout << "Login berhasil! Selamat datang, " << username << "!"
<< endl;

        isLoggedIn = 1; // Login berhasil
        isAdmin = users[i].isAdmin; // Menentukan apakah pengguna
adalah admin
        break;
    }
}
```

- Memeriksa apakah Username & Nim cocok dengan kata yang sudah terdaftar.

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

- Jika gagal login 3 kali maka Program berhenti.

6. Menu Utama

```
int opsi = 0; // Pilihan menu
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; // Pengguna hanya bisa
melihat
```

```

}
cout << "5. Keluar" << endl;
cout << "Pilih menu: ";
cin >> opsi;

```

- Menampilkan menu utama, dengan hak akses berbeda untuk admin dan pengguna biasa.

7. Menampilkan Semua Lagu

```

int opsi = 0; // Pilihan menu
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;
    }
}

```

- Menampilkan daftar lagu dalam format table.

8. Menambahkan Lagu (Admin)

```

if (isAdmin == 1) {
    if (songCount < MAX_SONGS) {
        cout << "Masukkan Judul Lagu: ";
        cin.ignore(); // Mengabaikan newline dari input sebelumnya
        getline(cin, songs[songCount].title);
    }
}

```

- Admin dapat menambahkan lagu.

9. Menghapus Lagu (Admin)

```

if (isAdmin == 1) {
    int indexToDelete;
    cout << "Masukkan nomor lagu yang ingin dihapus (1-" <<
songCount << "): ";
    cin >> indexToDelete;
}

```

- Admin dapat menghapus lagu berdasarkan nomor urut

10. Mengedit Lagu (Admin)

```
11. if (isAdmin == 1) {  
12.     int indexToEdit;  
13.     cout << "Masukkan nomor lagu yang ingin diubah (1-" <<  
    songCount << "): ";  
14.     cin >> indexToEdit;
```

- Admin dapat mengedit Lagu

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-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/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
- 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.