

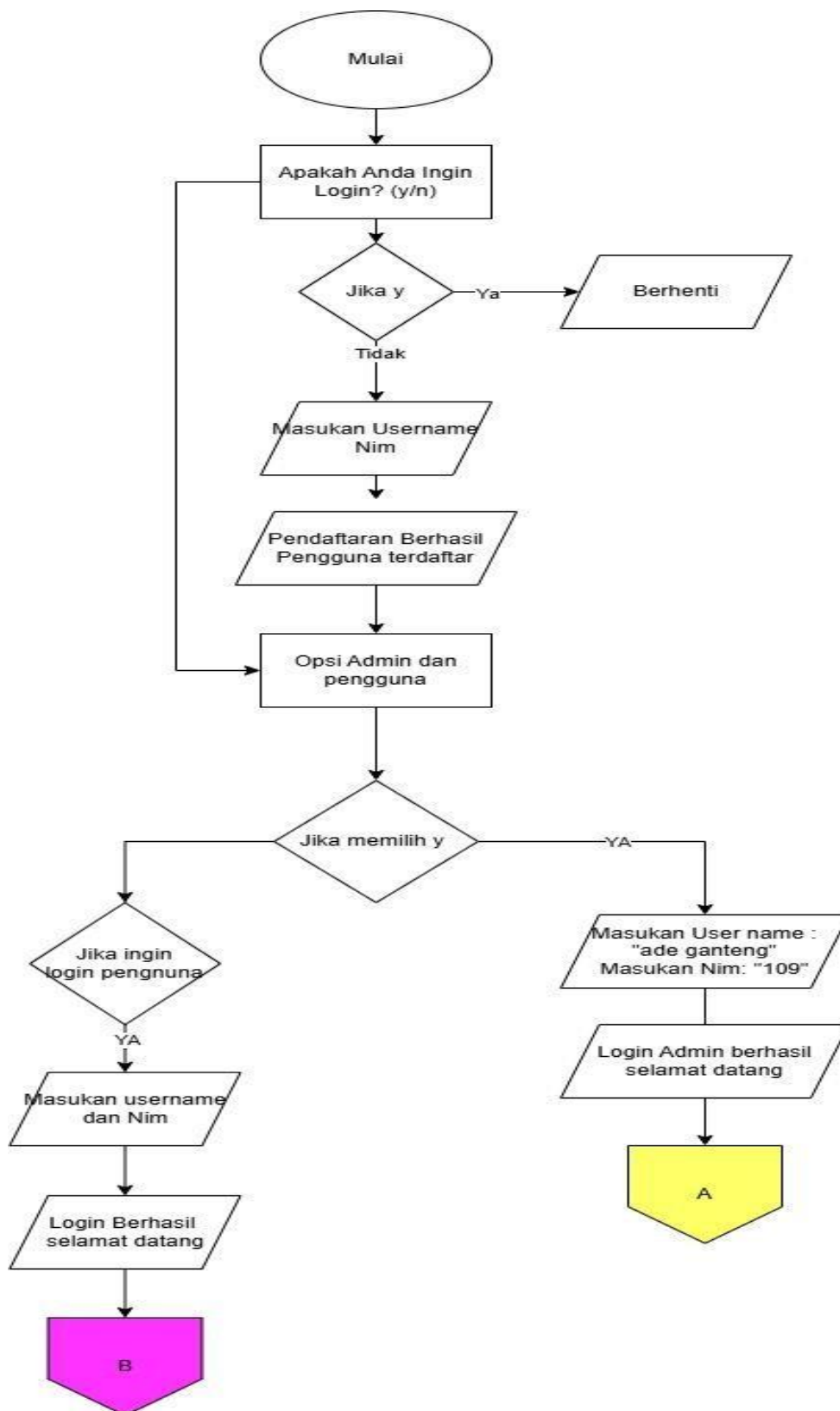
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT

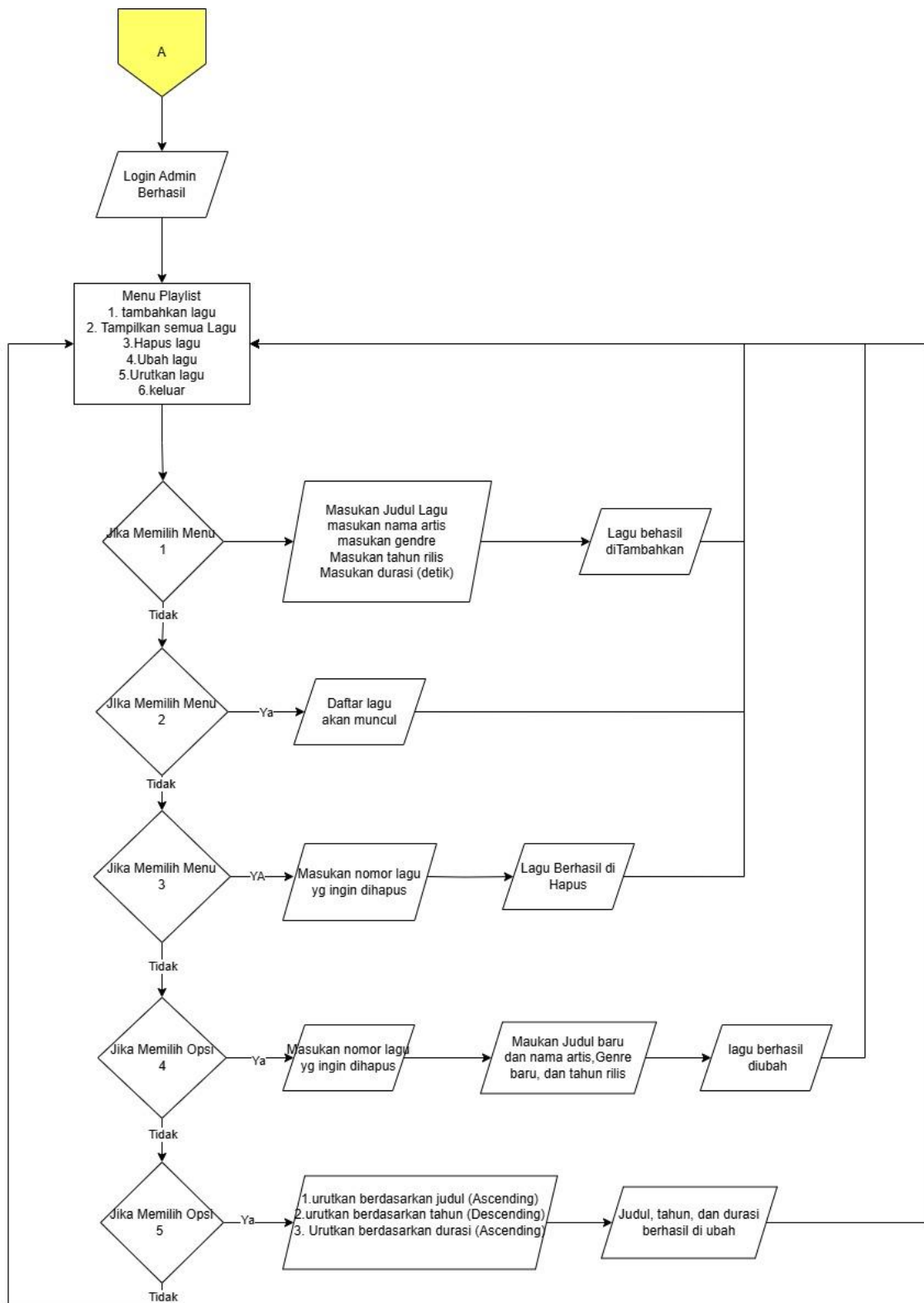


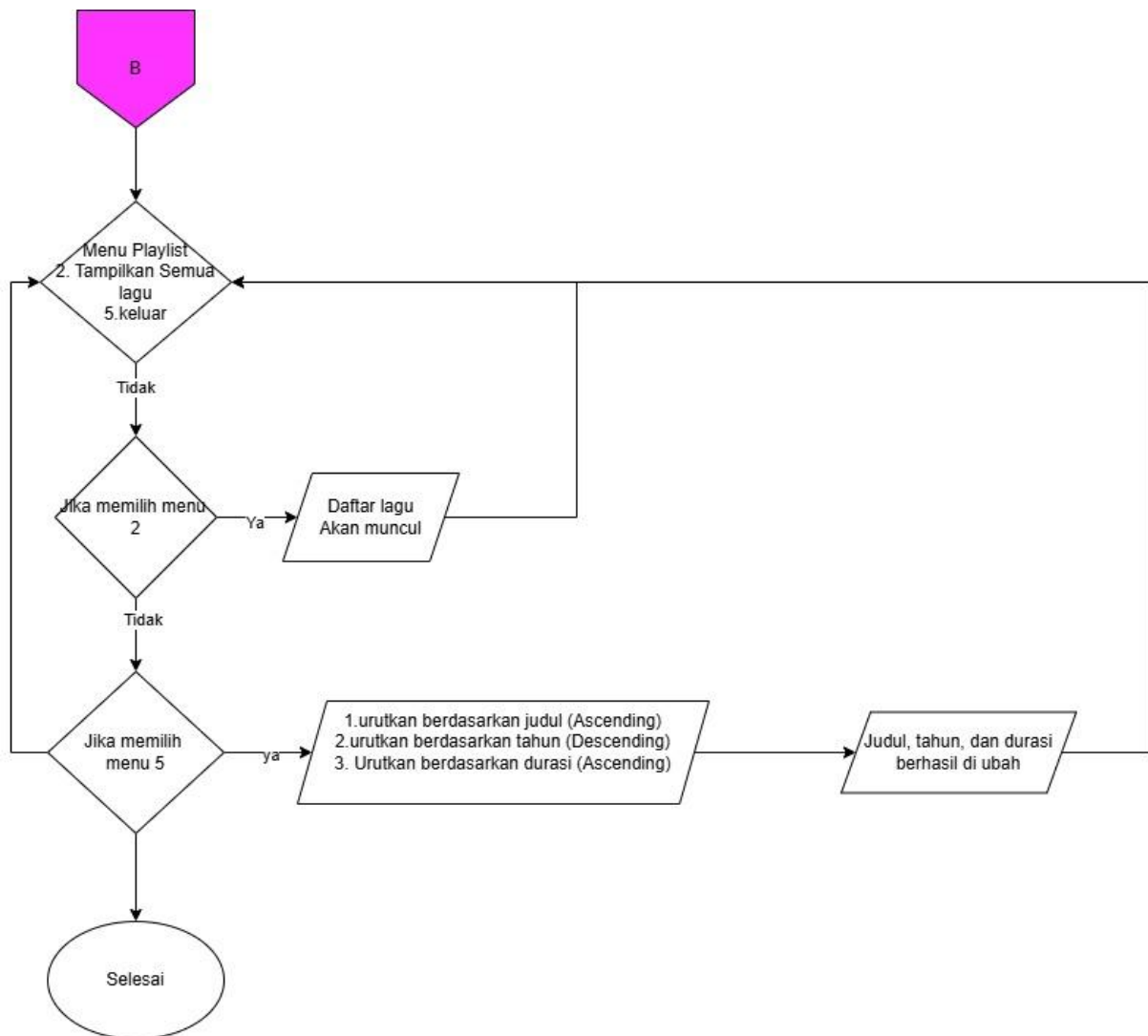
Disusun oleh:
Ade Pasiha Tangke Allo (2409106109)
Kelas (C2 '24)

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.

1. Source Code

1. Struktur Program

```
2. struct Song {  
3.     string title;  
4.     string artist;  
5. };  
6.
```

- Mewakili data lagu: Judul dan Artis.

```
struct User {  
    string username;  
    string nim;  
    int isAdmin;  
};
```

- Mewakili data user dengan:
- Username

- Nim (sebagai password atau ID)
- isAdmin (1 untuk admin, 0 untuk user biasa)

2. Fungsi dan Prosedur

1. TampilkanLaguRekursif

```
2. void tampilkanLaguRekursif(Song* songs, string* genres, int*
   releaseYears, int* durations, int index, int total) {
3.     if (index >= total) return;
4.     cout << left << setw(30) << songs[index].title
5.         << setw(30) << songs[index].artist
6.         << setw(20) << genres[index]
7.         << setw(15) << releaseYears[index]
8.         << setw(10) << durations[index] << endl;
9.     tampilkanLaguRekursif(songs, genres, releaseYears, durations,
   index + 1, total);
10.}
11.
```

- Menampilkan lagu secara rekursif (dipanggil berulang dengan index+1)
- Menggunakan pointer untuk mengakses data dari array struct dan array lainnya

2. TampilkanSemuaLagu

```
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
}
```

- Header tabel + garis pemisah
- Panggil tampilkanLaguRekursif() untuk menampilkan semua lagu satu per satu

3. tambahlagu

```
void tambahLagu(Song* songs, string* genres, int* releaseYears, int* durations,
int* songCount) {
    if (*songCount < MAX_SONGS) {
        cout << "Masukkan Judul Lagu: ";
        cin.ignore();
        getline(cin, songs[*songCount].title);
        cout << "Masukkan Nama Artis: ";
        getline(cin, songs[*songCount].artist);
        cout << "Masukkan Genre: ";
        getline(cin, genres[*songCount]);
        cout << "Masukkan Tahun Rilis: ";
        cin >> releaseYears[*songCount];
        cout << "Masukkan Durasi (dalam detik): ";
        cin >> durations[*songCount];

        *songCount = *songCount + 1;
        cout << "Lagu berhasil ditambahkan!!!" << endl;
    } else {
        cout << "Playlist sudah penuh!!!" << endl;
    }
}
```

- Input data Baru dari pengguna
- Disimpan dari Index terakhir (songCount)
- Menambahkan Jumlah lagu dengan pointer *songCount += 1
- Cek apakah Playlist penuh (MAX_SONGS)

4.HapusLagu

```
void hapusLagu(Song* songs, string* genres, int* releaseYears, int* durations,
int* songCount) {
    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];
        genres[i] = genres[i + 1];
        releaseYears[i] = releaseYears[i + 1];
        durations[i] = durations[i + 1];
    }
    *songCount = *songCount - 1;
    cout << "Lagu berhasil dihapus!" << endl;
} else {
    cout << "Nomor lagu tidak valid!" << endl;
}
}

```

- Menghapus lagu berdasarkan nomor lagu yang dihapus kedepan (1-based index)
- Geser semua data setelah lagu yang dihapus kedepan
- Kurangi jumlah lagu *songCount -= 1

5. ubahLagu

```

6. void ubahLagu(Song* songs, string* genres, int* releaseYears, int*
   durations, int songCount) {
7.     int indexToEdit;
8.     cout << "Masukkan nomor lagu yang ingin diubah (1-" << songCount
   << "): ";
9.     cin >> indexToEdit;
10.    if (indexToEdit > 0 && indexToEdit <= songCount) {
11.        cin.ignore();
12.        cout << "Masukkan Judul Baru: ";
13.        getline(cin, songs[indexToEdit - 1].title);
14.        cout << "Masukkan Nama Artis Baru: ";
15.        getline(cin, songs[indexToEdit - 1].artist);
16.        cout << "Masukkan Genre Baru: ";
17.        getline(cin, genres[indexToEdit - 1]);
18.        cout << "Masukkan Tahun Rilis Baru: ";
19.        cin >> releaseYears[indexToEdit - 1];
20.        cout << "Masukkan Durasi Baru (dalam detik): ";
21.        cin >> durations[indexToEdit - 1];
22.
23.        cout << "Lagu berhasil diubah!" << endl;
24.    } else {
25.        cout << "Nomor lagu tidak valid!" << endl;

```



```

26.     }
27.}

```

- Edit data lagu tertentu Berdasarkan nomor
- Tidak pakai pointer Untuk songCount karena hanya membaca

6. Login

```

7. int login(User* users, int userCount, string username, string nim,
   int* isAdmin) {
8.     int i;
9.     for (i = 0; i < userCount; i = i + 1) {
10.        if (users[i].username == username && users[i].nim == nim) {
11.            *isAdmin = users[i].isAdmin;
12.            return 1; // berhasil login
13.        }
14.    }
15.    return 0; // gagal login
16.}

```

- Cek Username dan Nim
- Jika cocok, set *isAdmin dari data user
- Return 1 (berhasil) atau 0 (gagal)

7. Fungsi

1. Inisialisasi Data

```

2. int main() {
3.     Song songs[MAX_SONGS] = {
4.         {"Kaulah hatiku", "Pashmina"},
5.         {"Blue", "Eiffel 65"},
6.         {"Sempurna", "Gigi"},
7.         {"Pica Pica", "Krisdayanti"},
8.         {"Kangen", "Dewa 19"},
9.         {"Cold", "Maroon 5"},
10.        {"Cry", "James Blunt"},
11.        {"December", "Gigi"},
12.        {"Yellow", "ColdPlay"},
13.    };
14.
15.    string genres[MAX_SONGS] = {
16.        "Pop", "Dance", "Rock", "Pop", "Rock", "Pop", "Pop",
        "Rock", "Alternative"

```

```

17.     };
18.     int releaseYears[MAX_SONGS] = {
19.         2020, 1999, 2000, 2001, 1999, 2017, 2004, 2000, 2000
20.     };
21.     int durations[MAX_SONGS] = {
22.         210, 180, 240, 200, 220, 210, 240, 180, 210
23.     };
24.

```

- Data awal playlist lagu sudah tersedia
- Total lagu awal: songCount = 9

2. Pendaftaran User

```

3. while (registerChoice == 'y' && userCount < MAX_USERS) {
4.     cout << "Masukkan Username: ";
5.     getline(cin, users[userCount].username);
6.     cout << "Masukkan NIM: ";
7.     getline(cin, users[userCount].nim);
8.     if (users[userCount].username == "adeganteng" &&
        users[userCount].nim == "109") {
9.         users[userCount].isAdmin = 1;
10.    } else {
11.        users[userCount].isAdmin = 0;
12.    }
13.    userCount = userCount + 1;
14.    cout << "Pendaftaran berhasil! Pengguna terdaftar: " <<
        users[userCount - 1].username << endl;
15.    cout << "Apakah Anda ingin mendaftar pengguna lain? (y/n): ";
16.    cin >> registerChoice;
17.    cin.ignore();
18. }
19.

```

- User bisa mendaftar (maks MAX_USERS)
- Jika username adeganteng dan nim: 109, akan di anggap admin
- Selain itu dianggap User biasa

3. Login

```
4. while (loginAttempts < MAX_ATTEMPTS) {
5.     cout << "\n=== Login ===" << endl;
6.     cout << "Masukkan Username: ";
7.     getline(cin, username);
8.     cout << "Masukkan NIM: ";
9.     getline(cin, nim);
10.
11.     if (login(users, userCount, username, nim, &isAdmin) == 1) {
12.         cout << "Login berhasil! Selamat datang, " << username <<
            "!" << endl;
13.         isLoggedIn = 1;
14.         break;
15.     } else {
16.         cout << "Username atau NIM salah. Coba lagi." << endl;
17.         loginAttempts = loginAttempts + 1;
18.     }
19. }
20.
21. if (isLoggedIn == 0) {
22.     cout << "Terlalu banyak percobaan. Program dihentikan." << endl;
23.     return 0;
24. }
25.
```

- Max tiga kali Login
- Setelah berhasil login, disimpan isAdmin untuk control hak akses

4. Menu Playlist

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

```

- Admin: bisa tambah, ubah, hapus, lihat lagu
- User bisa: hanya bisa melihat lagu
- Validasi akses berdasarkan isAdmin
- Keluar jika memilih opsi 5

8. Bubble Sort: berdasarkan judul (Ascending)

```

9. void bubbleSortTitleAsc(Song* songs, string* genres, int* releaseYears,
    int* durations, int songCount) {
10.     for (int i = 0; i < songCount - 1; i++) {
11.         for (int j = 0; j < songCount - i - 1; j++) {
12.             if (songs[j].title > songs[j + 1].title) {
13.                 swapSongs(songs, genres, releaseYears, durations, j, 1);
14.             }
15.         }
16.     }
17. }

```

- Mengurutkan lagu berdasarkan Songs[i]. Title (judul lagu) dari A-Z (Ascending)

```

if (songs[j].title > songs[j + 1].title) {
    swapSongs(songs, genres, releaseYears, durations, j, j + 1);
}

```

- Dibandingkan dua judul lagu yang berurutan
- Jika title [j] lebih besar dari title [j+1] (secara alfabet) maka dikukar sebagai swapSong().

- Proses ini dilakukan berulang kali sehingga lagu dengan judul terkecil naik ke atas (mirip gelembung naik-buble).

9. Selection Sort: Berdasarkan Tahun (Descending)

Fungsi:

```
void selectionSortYearDesc(Song* songs, string* genres, int* releaseYears, int*
durations, int songCount) {
    for (int i = 0; i < songCount - 1; i++) {
        int maxIdx = i;
        for (int j = i + 1; j < songCount; j++) {
            if (releaseYears[j] > releaseYears[maxIdx]) {
                maxIdx = j;
            }
        }
    }
}
```

- Mengurutkan berdasarkan releaseYears[i] dari baru ke lama (Descending)

Baris Penting:

```
if (maxIdx != i) {
    swapSongs(songs, genres, releaseYears, durations, i, maxIdx);
}
}
```

- Untuk setiap posisi i, cari indeks lagu dengan tahun rilis terbesar (maxIdx) di sisa array.
- Tukar lagu di indeks i dengan lagu tahun terbesar (maxIdx).
- Ini adalah ciri khas **selection sort**, yaitu selalu memilih nilai maksimum dan menempatkannya ke depan

10. Insertion Sort: Berdasarkan Durasi (Ascending Fungsi:

```
void insertionSortDurationAsc(Song* songs, string* genres, int* releaseYears,
int* durations, int songCount) {
    for (int i = 1; i < songCount; i++) {
        Song keySong = songs[i];
        string keyGenre = genres[i];
        int keyYear = releaseYears[i];
        int keyDuration = durations[i];
        int j = i - 1;
```

- Mengurutkan berdasarkan durations[i] dari terpendek ke terpanjang (ascending).

Baris Penting:

```
while (j >= 0 && durations[j] > keyDuration) {
    songs[j + 1] = songs[j];
    genres[j + 1] = genres[j];
    releaseYears[j + 1] = releaseYears[j];
    durations[j + 1] = durations[j];
    j--;
}
songs[j + 1] = keySong;
genres[j + 1] = keyGenre;
releaseYears[j + 1] = keyYear;
durations[j + 1] = keyDuration;
}
```

- Ambil lagu ke-i dan simpan sebagai "key".
- Bandingkan durasinya ke kiri, geser lagu yang durasinya lebih besar ke kanan.
- Setelah menemukan posisi yang sesuai, tempatkan key di sana.

- Proses ini mirip dengan menyisipkan kartu dalam urutan (insertion).

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

10. Urutkan Lagu

```
=== Menu Urutkan Lagu ===
1. Urutkan berdasarkan Judul (Ascending)
2. Urutkan berdasarkan Tahun (Descending)
3. Urutkan berdasarkan Durasi (Ascending)
Pilihan: 1
```

```
=== Daftar Lagu ===
```

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

Gambar 10 urutkan judul lagu (Ascending)

```

=== Menu Urutkan Lagu ===
1. Urutkan berdasarkan Judul (Ascending)
2. Urutkan berdasarkan Tahun (Descending)
3. Urutkan berdasarkan Durasi (Ascending)
Pilihan: 2

```

```

=== Daftar Lagu ===

```

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

Gambar 10 Urutkan Berdasarkan tahun (Descending)

```

=== Menu Urutkan Lagu ===
1. Urutkan berdasarkan Judul (Ascending)
2. Urutkan berdasarkan Tahun (Descending)
3. Urutkan berdasarkan Durasi (Ascending)
Pilihan: 3

```

```

=== Daftar Lagu ===

```

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

Gambar 10 Urutan berdasarkan durasi (Ascending)

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"

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