

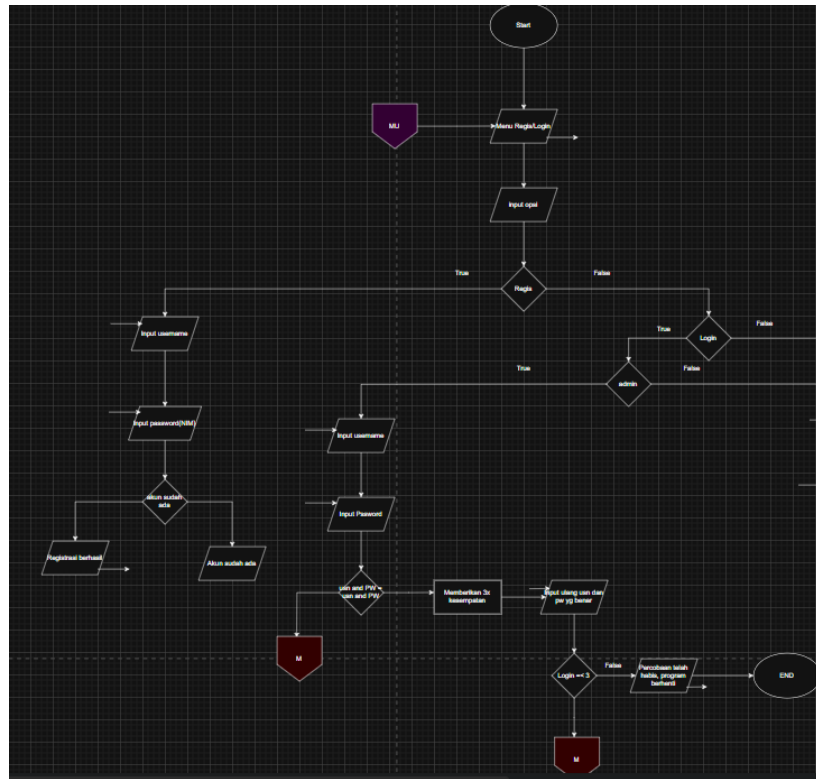
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT



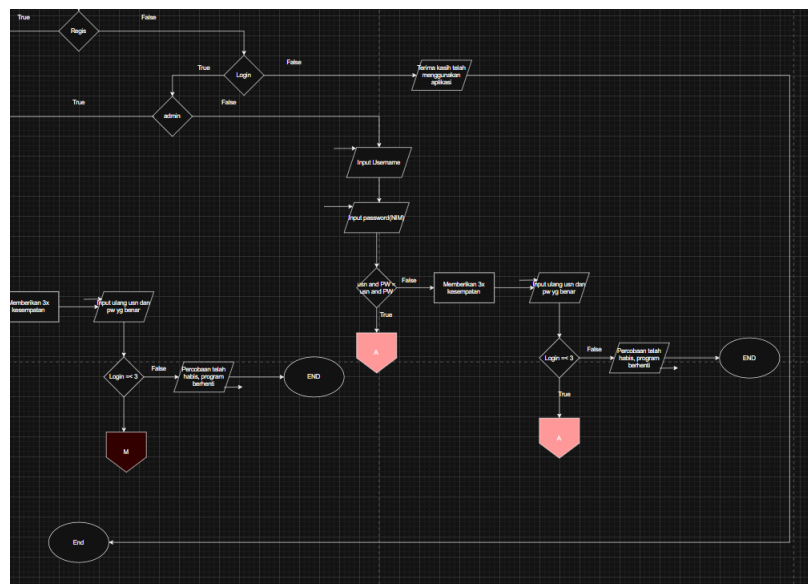
Disusun oleh:
Danendra Hazzel Putra Wahana 2409106096
Kelas C1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

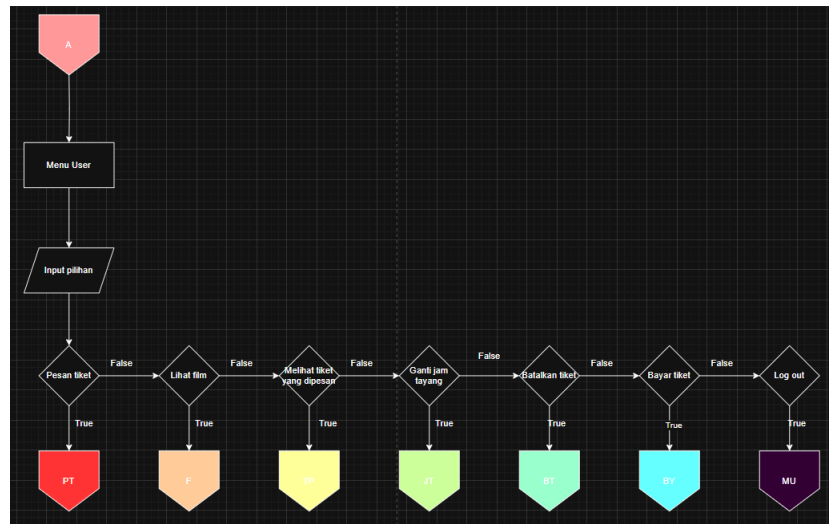
1. Flowchart



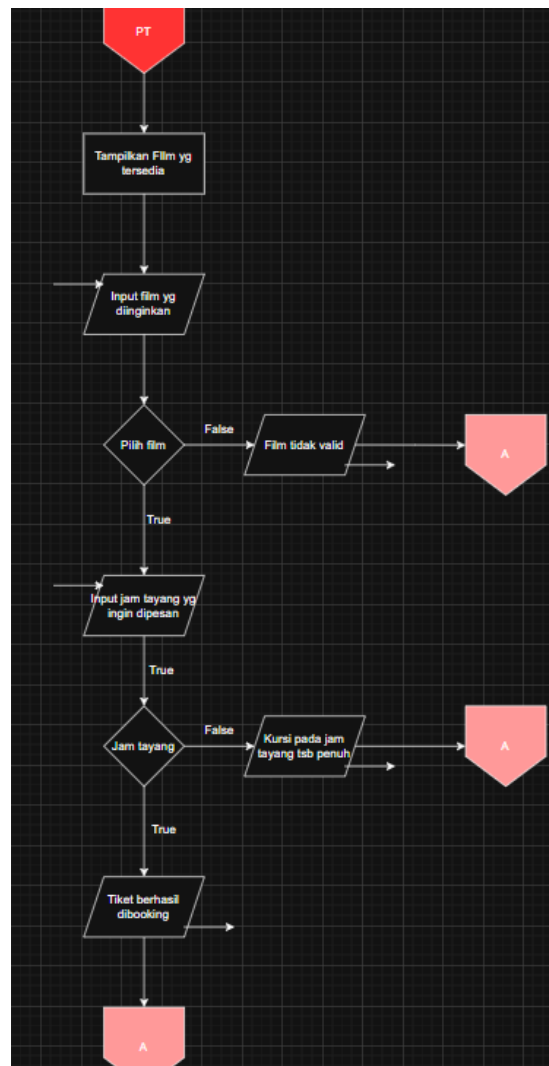
Gambar 1.1 Bagian menu regis



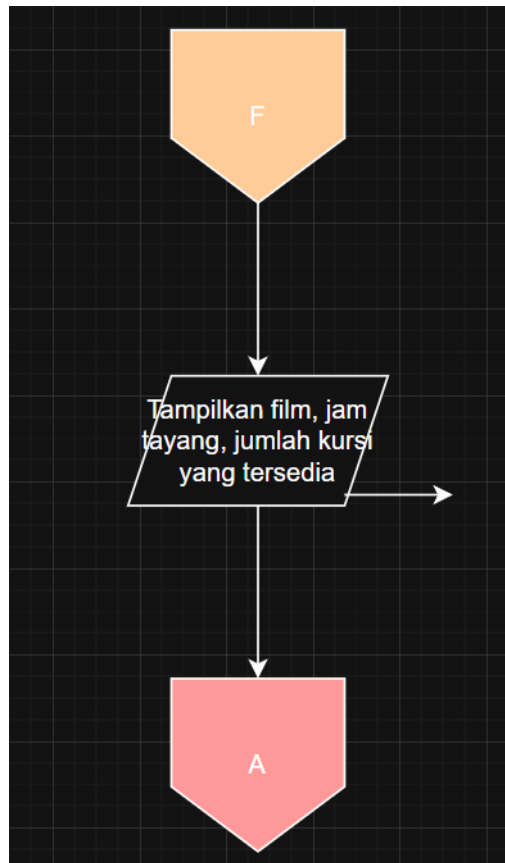
Gambar 1.2 Bagian menu regis bagian 2



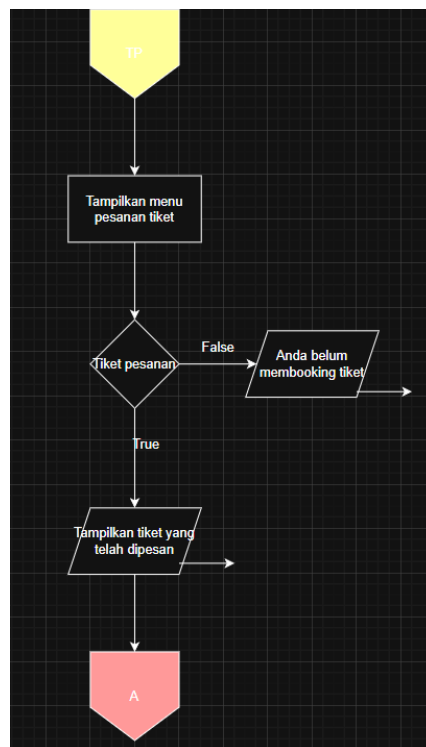
Gambar 1.3 Bagian menu user



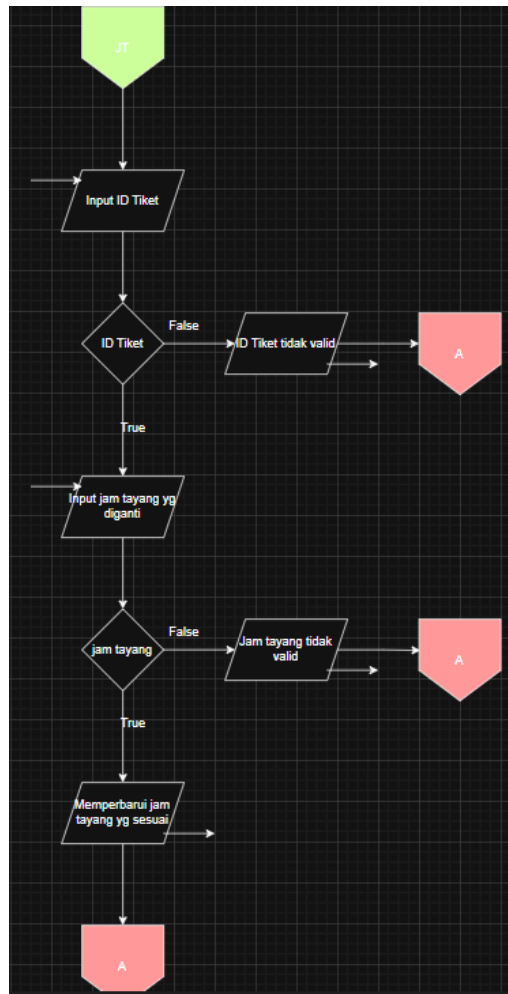
Gambar 1.4 Bagian pesan tiket



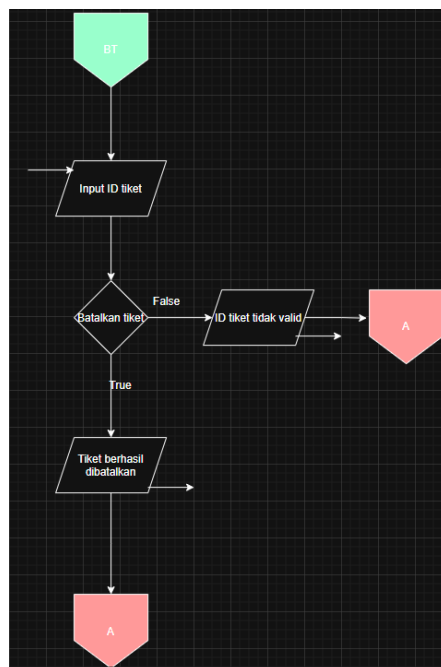
Gambar 1.5 Bagian menampilkan film yang tersedia



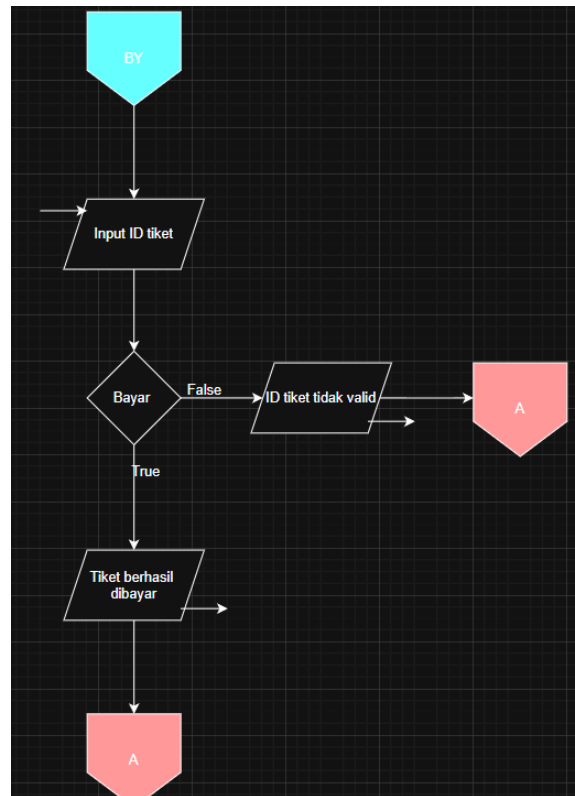
Gambar 1.6 Bagian menampilkan tiket yang sudah dibeli.



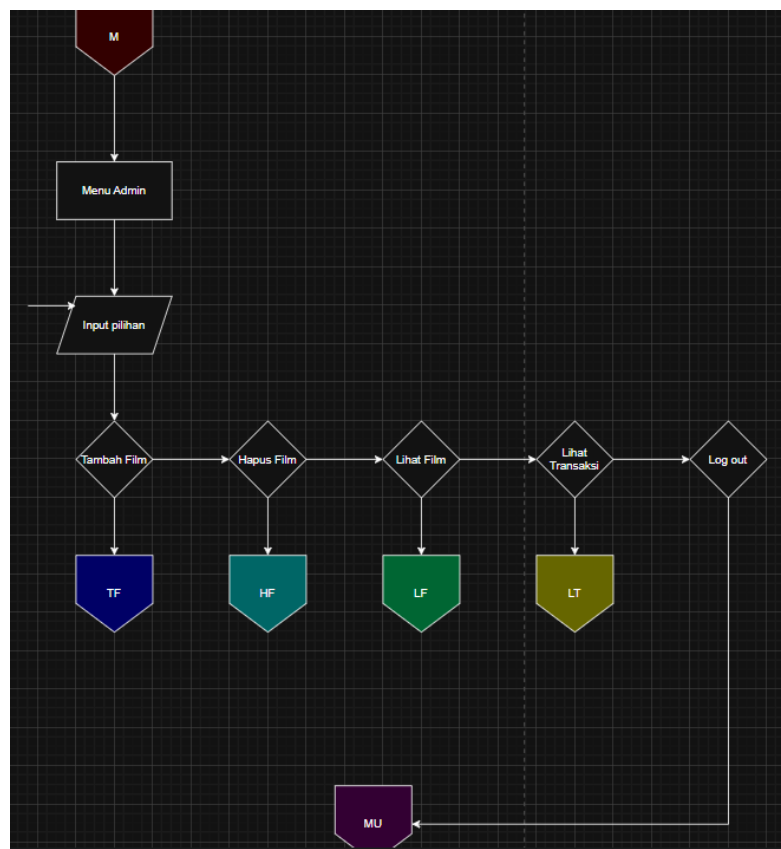
Gambar 1.7 Bagian mengganti jam tayang



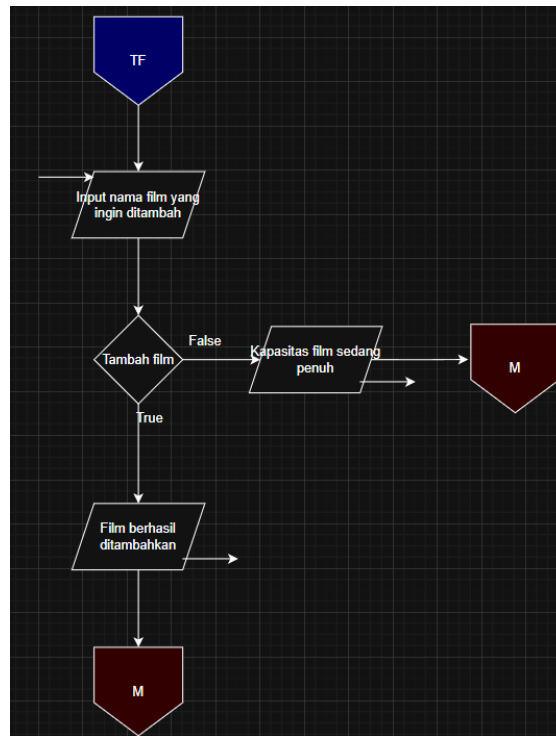
Gambar 1.8 Bagian pembatalan tiket



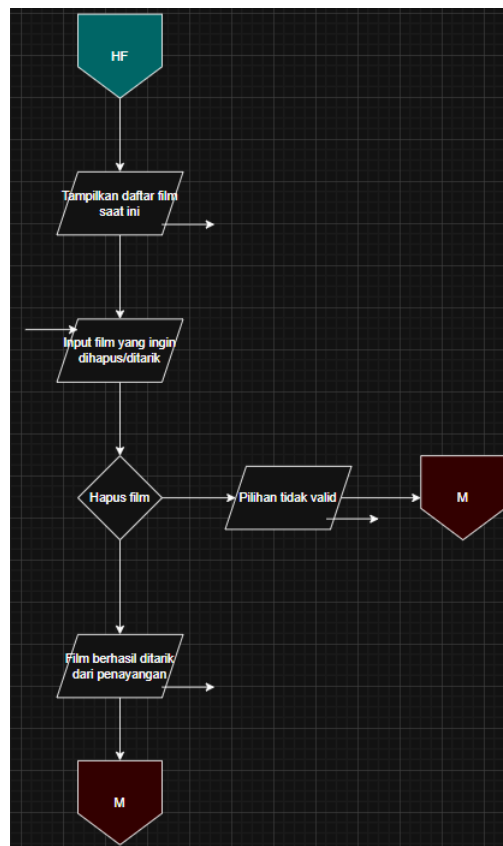
Gambar 1.9 Bagian bayar tiket



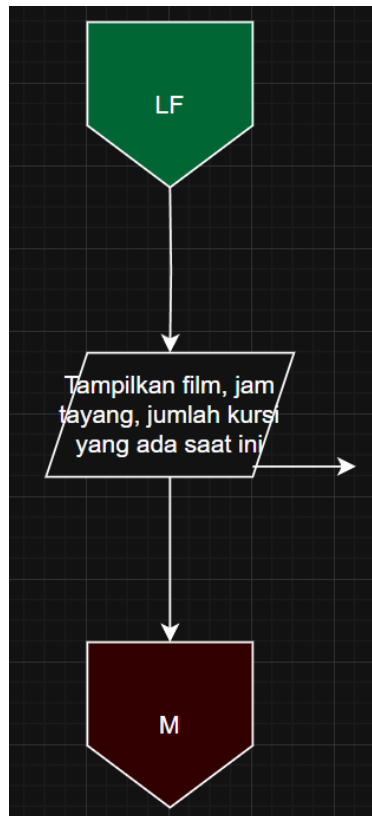
Gambar 1.10 Bagian menu admin



Gambar 1.11 Bagian menambah film



Gambar 1.12 Bagian penarikan film



Gambar 1.13 Bagian melihat film saat ini



Gambar 1.14 Bagian informasi pemesanan tiket user

2. Analisis Program

Program ini adalah sebuah pemesanan tiket online sederhana yang memungkinkan Pengguna dapat login sebagai pembeli maupun admin. Tentu fitur admin maupun user memiliki perbedaannya masing-masing. Berikut adalah penjelasan fitur dari admin maupun pembeli :

- a) Admin dapat menambahkan atau menghapus film, melihat daftar film yang tersedia, serta melihat transaksi pemesanan tiket yang dilakukan beberapa user.
- b) User dapat memesan tiket, melihat film yang tersedia, melihat tiket yang sudah dipesan, mengganti jam tayang, membatalkan tiket, dan melakukan pembayaran.

Program ini bertujuan untuk memudahkan para user untuk memesan/membooking tiket dengan mudah secara online yang dapat dilakukan kapan saja dan dimana saja serta admin yang dapat mengkoordinasi pemesanan dan informasi pada bioskop saat ini dengan mudah hanya melalui dari aplikasi.

3. Source Code

```
1 #include <iostream>
2 #include <string>
3 #include <iomanip>
4 using namespace std;
5
6 // Nested struct untuk menyimpan data jam tayang dan kursi terisi
7 struct JamTayang {
8     string jam;
9     int kursiTerisi;
10 };
11
12 // Struct untuk menyimpan data film
13 struct FilmData {
14     string namaFilm;
15     JamTayang jamTayang[5];
16 };
17
18 // Struct untuk menyimpan data user
19 struct User {
20     string username;
21     string password;
22     bool isAdmin;
23 };
24
25 // Struct untuk menyimpan data booking
26 struct Booking {
27     int filmId;
28     int jamId;
29     bool isPaid;
30     int userId; // Menyimpan ID user yang melakukan booking
31 };
32
33 // Deklarasi fungsi dan prosedur dengan pointer
34 void tampilkanMenuUtama();
35 void registrasiUser(User *users, int *userCount);
36 bool login(User *users, int userCount, bool *isAdmin, int *currentUserId);
37 bool menuAdmin(FilmData *films, int *filmCount, Booking *bookings, int *bookingCount, User *users, int userCount, const int *maksKursi);
38 bool menuUser(FilmData *films, int filmCount, Booking *bookings, int *bookingCount, int *currentUserId, User *users, const int *maksKursi);
39 void tambahFilm(FilmData *films, int *filmCount);
40 void hapusFilm(FilmData *films, int *filmCount);
41 void lihatFilm(const FilmData *films, int filmCount, const int *maksKursi);
42 void lihatTransaksi(const Booking *bookings, int bookingCount, const FilmData *films, const User *users);
43 void pesanTiket(FilmData *films, int filmCount, Booking *bookings, int *bookingCount, int *currentUserId, const int *maksKursi);
44 void lihatTiketUser(const Booking *bookings, int bookingCount, const FilmData *films, int *currentUserId);
45 void gantiJamTayang(FilmData *films, Booking *bookings, int bookingCount, int *currentUserId, const int *maksKursi);
46 void batalkanTiket(FilmData *films, Booking *bookings, int *bookingCount);
47 void bayarTiket(Booking *bookings, int *bookingCount, const FilmData *films);
48 void tampilkanFilmTersedia(const FilmData *films, int filmCount, int index, const int *maksKursi); // Fungsi rekursif
49 void tampilkanJamTayang(const JamTayang *jamTayang, int index, const int *maksKursi); // Fungsi overLoading
50 void tampilkanJamTayang(const JamTayang &jamTayang); // Fungsi overLoading dengan reference
51
52 int main() {
```

```

1  int main() {
2      // Array of struct untuk menyimpan data film
3      FilmData films[10] = {
4          {"Kimi No Nawa", {"10:00", 0}, {"12:00", 0}, {"15:00", 0}, {"17:00", 0}, {"19:00", 0}},
5          {"Godzilla : Minus one", {"10:00", 0}, {"12:00", 0}, {"15:00", 0}, {"17:00", 0}, {"19:00", 0}},
6          {"Avengers : secret wars", {"10:00", 0}, {"12:00", 0}, {"15:00", 0}, {"17:00", 0}, {"19:00", 0}},
7          {"Your Lie In April", {"10:00", 0}, {"12:00", 0}, {"15:00", 0}, {"17:00", 0}, {"19:00", 0}},
8          {"Cek Toko Sebelah", {"10:00", 0}, {"12:00", 0}, {"15:00", 0}, {"17:00", 0}, {"19:00", 0}}
9      };
10
11     int filmCount = 5; // Jumlah film awal
12
13     // Array untuk menyimpan data user
14     User users[10];
15     int userCount = 0;
16
17     // Array untuk menyimpan booking
18     Booking bookings[10];
19     int bookingCount = 0;
20
21     // Kapasitas maksimum per film dan jam
22     const int maksKursi = 20;
23
24     // Inisialisasi admin dan user
25     users[userCount++] = {"admin", "selalubentar", true}; // Admin
26     users[userCount++] = {"Danendra Hazzel PW", "2409106096", false}; // User biasa
27
28     int pilihan = 0;
29     bool isRunning = true;
30     int loginAttempts = 0; // Menghitung percobaan Login
31     bool isAdmin = false;
32     int currentUserId = -1;
33
34     // Menu utama
35     while (isRunning) {
36         tampilkanMenuUtama();
37         cin >> pilihan;
38
39         switch (pilihan) {
40             case 1: {
41                 registrasiUser(users, &userCount);
42                 break;
43             }
44             case 2: {
45                 if (login(users, userCount, &isAdmin, &currentUserId)) {
46                     cout << "Login berhasil!\n";
47                     cout << "Selamat Datang " << (isAdmin ? "Bos" : "Tuan") << " " << users[currentUserId].username << endl;
48
49                     // Menu setelah Login
50                     bool isLoggedIn = true;
51                     while (isLoggedIn) {

```

```

1         if (isAdmin) {
2             isLoggedIn = menuAdmin(films, &filmCount, bookings, &bookingCount, users, userCount, &maksKursi);
3         } else {
4             isLoggedIn = menuUser(films, filmCount, bookings, &bookingCount, &currentUserId, users, &maksKursi);
5         }
6     }
7 } else {
8     loginAttempts++;
9     cout << "Login gagal, Username atau Password salah! Sisa percobaan anda sebanyak " << 3 - loginAttempts << endl;
10
11     if (loginAttempts >= 3) {
12         cout << "Kesempatan anda telah habis, silahkan mencoba lagi dalam beberapa saat.\n";
13         isRunning = false;
14     }
15 }
16 break;
17 }
18 case 3: {
19     cout << "Terima kasih telah menggunakan aplikasi ini, enjoy the movie!\n";
20     isRunning = false;
21     break;
22 }
23 default: {
24     cout << "Pilihan tidak valid!\n";
25     break;
26 }
27 }
28 }
29
30 return 0;
31 }
32
33 // Implementasi fungsi dan prosedur dengan pointer
34 void tampilkanMenuUtama() {
35     cout << "\n==== MENU UTAMA =====\n";
36     cout << "1. Registrasi\n";
37     cout << "2. Login\n";
38     cout << "3. Keluar\n";
39     cout << "Pilihan Anda: ";
40 }
41
42 void registrasiUser(User *users, int *userCount) {
43     if (*userCount >= 10) {
44         cout << "Maaf, kapasitas user sudah penuh!\n";
45         return;
46     }
47
48     cout << "\n==== REGISTRASI =====\n";
49     string username, password;
50     cout << "Masukkan Username: ";
51     cin >> ws;

```

```

1  getline(cin, username);
2  cout << "Masukkan Password: ";
3  cin >> password;
4
5  // Cek apakah username sudah ada
6  bool usernameExists = false;
7  for (int i = 0; i < *userCount; i++) {
8      if (users[i].username == username) {
9          usernameExists = true;
10         break;
11     }
12 }
13
14 if (usernameExists) {
15     cout << "Registrasi gagal! Akun telah tersedia.\n";
16 } else {
17     users[*userCount] = {username, password, false};
18     (*userCount)++;
19     cout << "Registrasi berhasil!\n";
20 }
21 }
22
23 bool login(User *users, int userCount, bool *isAdmin, int *currentUserId) {
24     cout << "\n==== LOGIN =====\n";
25     cout << "Ingin login sebagai apa?\n";
26     cout << "1. Admin\n";
27     cout << "2. User\n";
28     cout << "Pilihan Anda: ";
29     int loginType;
30     cin >> loginType;
31
32     if (loginType < 1 || loginType > 2) {
33         cout << "Pilihan tidak valid!\n";
34         return false;
35     }
36
37     string username, password;
38     cout << "Masukkan Username: ";
39     cin >> ws;
40     getline(cin, username);
41     cout << "Masukkan Password: ";
42     cin >> password;
43
44     for (int i = 0; i < userCount; i++) {
45         if (users[i].username == username && users[i].password == password) {
46             if ((loginType == 1 && users[i].isAdmin) || (loginType == 2 && !users[i].isAdmin)) {
47                 *isAdmin = users[i].isAdmin;
48                 *currentUserId = i;
49                 return true;
50             }
51         }
52     }
53 }

```

```

1     }
2     return false;
3 }
4
5 bool menuAdmin(FilmData *films, int *filmCount, Booking *bookings, int *bookingCount, User *users, int userCount, const int *maksKursi) {
6     cout << "\n===== MENU ADMIN =====\n";
7     cout << "1. Tambah Film\n";
8     cout << "2. Hapus Film\n";
9     cout << "3. Lihat Film\n";
10    cout << "4. Lihat Transaksi\n";
11    cout << "5. Log out\n";
12    cout << "Pilihan Anda: ";
13
14    int pilihan;
15    cin >> pilihan;
16
17    switch (pilihan) {
18        case 1: tambahFilm(films, filmCount); break;
19        case 2: hapusFilm(films, filmCount); break;
20        case 3: lihatFilm(films, *filmCount, maksKursi); break;
21        case 4: lihatTransaksi(bookings, *bookingCount, films, users); break;
22        case 5: {
23            cout << "Kembali ke menu utama\n";
24            return false; // Indikator untuk Logout
25        }
26        default: cout << "Pilihan tidak valid!\n"; break;
27    }
28    return true; // Tetap di menu admin
29 }
30
31 bool menuUser(FilmData *films, int filmCount, Booking *bookings, int *bookingCount, int *currentUserId, User *users, const int *maksKursi) {
32     cout << "\n===== MENU USER =====\n";
33     cout << "1. Pesan Tiket\n";
34     cout << "2. Lihat Film yang Tersedia\n";
35     cout << "3. Lihat Tiket yang Sudah Dipesan\n";
36     cout << "4. Ganti Jam Tayang\n";
37     cout << "5. Batalkan Tiket\n";
38     cout << "6. Bayar Tiket\n";
39     cout << "7. Log out\n";
40     cout << "Pilihan Anda: ";
41
42     int pilihan;
43     cin >> pilihan;
44
45     switch (pilihan) {
46         case 1: pesanTiket(films, filmCount, bookings, bookingCount, currentUserId, maksKursi); break;
47         case 2: lihatFilm(films, filmCount, maksKursi); break;
48         case 3: lihatTiketUser(bookings, *bookingCount, films, currentUserId); break;
49         case 4: gantiJamTayang(films, bookings, *bookingCount, currentUserId, maksKursi); break;
50         case 5: batalkanTiket(films, bookings, bookingCount); break;
51         case 6: bayarTiket(bookings, bookingCount, films); break;
52         case 7: {

```

```

1     case 7: {
2         cout << "Kembali ke menu utama\n";
3         return false; // Indikator untuk logout
4     }
5     default: cout << "Pilihan tidak valid!\n"; break;
6 }
7 return true; // Tetap di menu user
8 }
9
10 void tambahFilm(FilmData *films, int *filmCount) {
11     if (*filmCount >= 10) {
12         cout << "Maaf, kapasitas film sudah penuh!\n";
13         return;
14     }
15
16     cout << "\n===== TAMBAH FILM =====\n";
17     cout << "Masukkan Nama Film: ";
18     cin >> ws;
19     getline(cin, films[*filmCount].namaFilm);
20
21     for (int i = 0; i < 5; i++) {
22         films[*filmCount].jamTayang[i].jam = to_string(10 + i * 2) + ":00";
23         films[*filmCount].jamTayang[i].kursiTerisi = 0;
24     }
25
26     cout << "Film berhasil ditambahkan!\n";
27     (*filmCount)++;
28 }
29
30 void hapusFilm(FilmData *films, int *filmCount) {
31     cout << "\n===== HAPUS FILM =====\n";
32     cout << "Daftar Film:\n";
33     for (int i = 0; i < *filmCount; i++) {
34         cout << i + 1 << ". " << films[i].namaFilm << endl;
35     }
36
37     int filmPilihan;
38     cout << "Pilih film yang ingin ditarik (1-" << *filmCount << "): ";
39     cin >> filmPilihan;
40
41     if (filmPilihan < 1 || filmPilihan > *filmCount) {
42         cout << "Pilihan tidak valid!\n";
43         return;
44     }
45
46     for (int i = filmPilihan - 1; i < *filmCount - 1; i++) {
47         films[i] = films[i + 1];
48     }
49
50     (*filmCount)--;
51     cout << "Film berhasil ditarik dari penayangan!\n";

```



```

1  }
2
3  void lihatFilm(const FilmData *films, int filmCount, const int *maksKursi) {
4      cout << "\n===== DAFTAR FILM =====\n";
5      tampilkanFilmTersedia(films, filmCount, 0, maksKursi);
6  }
7
8  // Fungsi rekursif untuk menampilkan film yang tersedia dengan pointer
9  void tampilkanFilmTersedia(const FilmData *films, int filmCount, int index, const int *maksKursi) {
10     if (index >= filmCount) return;
11
12     cout << index + 1 << ". " << films[index].namaFilm << endl;
13     cout << "    Jam Tayang: ";
14     tampilkanJamTayang(films[index].jamTayang, 0, maksKursi);
15     cout << endl;
16
17     tampilkanFilmTersedia(films, filmCount, index + 1, maksKursi);
18 }
19
20 // Fungsi overloading untuk menampilkan jam tayang (array) dengan pointer
21 void tampilkanJamTayang(const JamTayang *jamTayang, int index, const int *maksKursi) {
22     if (index >= 5) return;
23
24     int kursiTersisa = *maksKursi - jamTayang[index].kursiTerisi;
25     cout << jamTayang[index].jam << " (" << kursiTersisa << " kursi) ";
26
27     tampilkanJamTayang(jamTayang, index + 1, maksKursi);
28 }
29
30 // Fungsi overloading untuk menampilkan jam tayang (single) dengan reference (address-of)
31 void tampilkanJamTayang(const JamTayang &jamTayang) {
32     cout << jamTayang.jam;
33 }
34
35 void lihatTransaksi(const Booking *bookings, int bookingCount, const FilmData *films, const User *users) {
36     cout << "\n===== TRANSAKSI =====\n";
37     for (int i = 0; i < bookingCount; i++) {
38         cout << "ID Booking: " << i << endl;
39         cout << "Film: " << films[bookings[i].filmId].namaFilm << endl;
40         cout << "Jam: " << films[bookings[i].filmId].jamTayang[bookings[i].jamId].jam << endl;
41         cout << "User: " << users[bookings[i].userId].username << endl;
42         cout << "Status: " << (bookings[i].isPaid ? "Sudah dibayar" : "Belum dibayar") << endl;
43         cout << "-----\n";
44     }
45 }
46
47 void pesanTiket(FilmData *films, int filmCount, Booking *bookings, int *bookingCount, int *currentUserId, const int *maksKursi) {
48     cout << "\n===== PESAN TIKET =====\n";
49
50     // Tampilkan film yang tersedia
51     cout << "Film yang tersedia:\n";

```



```

1  for (int i = 0; i < filmCount; i++) {
2      cout << i + 1 << ". " << films[i].namaFilm << endl;
3  }
4
5  int filmPilihan;
6  cout << "Pilih film (1-" << filmCount << "): ";
7  cin >> filmPilihan;
8
9  if (filmPilihan < 1 || filmPilihan > filmCount) {
10     cout << "Film tidak valid!\n";
11     return;
12 }
13
14 // Tampilkan jam tayang
15 cout << "Jam tayang yang tersedia:\n";
16 for (int i = 0; i < 5; i++) {
17     int kursiTersisa = *maksKursi - films[filmPilihan - 1].jamTayang[i].kursiTerisi;
18     cout << i + 1 << ". " << films[filmPilihan - 1].jamTayang[i].jam << " (Kursi tersisa: " << kursiTersisa << ")\n";
19 }
20
21 int jamPilihan;
22 cout << "Pilih jam tayang (1-5): ";
23 cin >> jamPilihan;
24
25 if (jamPilihan < 1 || jamPilihan > 5) {
26     cout << "Jam tayang tidak valid!\n";
27     return;
28 }
29
30 // Cek apakah kursi masih tersedia
31 if (films[filmPilihan - 1].jamTayang[jamPilihan - 1].kursiTerisi >= *maksKursi) {
32     cout << "Maaf, kursi untuk film dan jam tersebut sudah penuh!\n";
33     return;
34 }
35
36 // Simpan booking
37 bookings[*bookingCount].filmId = filmPilihan - 1;
38 bookings[*bookingCount].jamId = jamPilihan - 1;
39 bookings[*bookingCount].isPaid = false;
40 bookings[*bookingCount].userId = *currentUserId;
41
42 // Update kursi terisi
43 films[filmPilihan - 1].jamTayang[jamPilihan - 1].kursiTerisi++;
44
45 cout << "Booking berhasil!\n";
46 cout << "ID Booking Anda: " << *bookingCount << endl;
47 cout << "Film: " << films[filmPilihan - 1].namaFilm << endl;
48 cout << "Jam: " << films[filmPilihan - 1].jamTayang[jamPilihan - 1].jam << endl;
49 cout << "Status: Belum dibayar\n";
50
51 (*bookingCount)++;

```

```

1  }
2
3  void lihatTiketUser(const Booking *bookings, int bookingCount, const FilmData *films, int *currentUserId) {
4      cout << "\n==== TIKET YANG SUDAH DIPESAN =====\n";
5      bool adaTiket = false;
6
7      for (int i = 0; i < bookingCount; i++) {
8          if (bookings[i].filmId != -1 && bookings[i].userId == *currentUserId) {
9              adaTiket = true;
10             cout << "ID Booking: " << i << endl;
11             cout << "Film: " << films[bookings[i].filmId].namaFilm << endl;
12             cout << "Jam: " << films[bookings[i].filmId].jamTayang[bookings[i].jamId].jam << endl;
13             cout << "Status: " << (bookings[i].isPaid ? "Sudah dibayar" : "Belum dibayar") << endl;
14             cout << "-----\n";
15         }
16     }
17
18     if (!adaTiket) {
19         cout << "Belum ada tiket yang dipesan!\n";
20     }
21 }
22
23 void gantiJamTayang(FilmData *films, Booking *bookings, int bookingCount, int *currentUserId, const int *maksKursi) {
24     cout << "\n==== GANTI JAM TAYANG =====\n";
25
26     int idBooking;
27     cout << "Masukkan ID Booking: ";
28     cin >> idBooking;
29
30     if (idBooking < 0 || idBooking >= bookingCount || bookings[idBooking].filmId == -1) {
31         cout << "ID Booking tidak valid!\n";
32         return;
33     }
34
35     if (bookings[idBooking].isPaid) {
36         cout << "Tidak dapat mengubah jam tayang tiket yang sudah dibayar!\n";
37         return;
38     }
39
40     int filmId = bookings[idBooking].filmId;
41     int jamLama = bookings[idBooking].jamId;
42
43     // Tampilkan jam tayang yang tersedia
44     cout << "Jam tayang saat ini: " << films[filmId].jamTayang[jamLama].jam << endl;
45     cout << "Jam tayang baru yang tersedia:\n";
46
47     for (int i = 0; i < 5; i++) {
48         int kursiTersisa = *maksKursi - films[filmId].jamTayang[i].kursiTerisi;
49         cout << i + 1 << ". " << films[filmId].jamTayang[i].jam << " (Kursi tersisa: " << kursiTersisa << ")\n";
50     }
51
52     int jamBaru;
53     cout << "Pilih jam tayang baru (1-5): ";
54     cin >> jamBaru;

```

```

1     int jamBaru;
2     cout << "Pilih jam tayang baru (1-5): ";
3     cin >> jamBaru;
4
5     if (jamBaru < 1 || jamBaru > 5) {
6         cout << "Jam tayang tidak valid!\n";
7         return;
8     }
9
10    if (films[filmId].jamTayang[jamBaru - 1].kursiTerisi >= *maksKursi) {
11        cout << "Maaf, kursi untuk jam tersebut sudah penuh!\n";
12        return;
13    }
14
15    // Update kursi terisi
16    films[filmId].jamTayang[jamLama].kursiTerisi--;
17    films[filmId].jamTayang[jamBaru - 1].kursiTerisi++;
18
19    // Update booking
20    bookings[idBooking].jamId = jamBaru - 1;
21
22    cout << "Jam tayang berhasil diubah!\n";
23    cout << "Film: " << films[filmId].namaFilm << endl;
24    cout << "Jam tayang baru: " << films[filmId].jamTayang[jamBaru - 1].jam << endl;
25 }
26
27 void batalkanTiket(FilmData *films, Booking *bookings, int *bookingCount) {
28     cout << "\n==== BATALKAN TIKET =====\n";
29
30     int idBooking;
31     cout << "Masukkan ID Booking: ";
32     cin >> idBooking;
33
34     if (idBooking < 0 || idBooking >= *bookingCount || bookings[idBooking].filmId == -1) {
35         cout << "ID Booking tidak valid!\n";
36         return;
37     }
38
39     if (bookings[idBooking].isPaid) {
40         cout << "Tidak dapat membatalkan tiket yang sudah dibayar!\n";
41         return;
42     }
43
44     int filmId = bookings[idBooking].filmId;
45     int jamId = bookings[idBooking].jamId;
46
47     // Update kursi terisi
48     films[filmId].jamTayang[jamId].kursiTerisi--;
49
50     // Hapus booking
51     bookings[idBooking].filmId = -1;
52     bookings[idBooking].jamId = -1;
53     bookings[idBooking].isPaid = false;
54
55     cout << "Tiket berhasil dibatalkan!\n";

```

```

1     bookings[idBooking].jamId = -1;
2     bookings[idBooking].isPaid = false;
3
4     cout << "Tiket berhasil dibatalkan!\n";
5 }
6
7 void bayarTiket(Booking *bookings, int *bookingCount, const FilmData *films) {
8     cout << "\n===== BAYAR TIKET =====\n";
9
10    int idBooking;
11    cout << "Masukkan ID Booking: ";
12    cin >> idBooking;
13
14    if (idBooking < 0 || idBooking >= *bookingCount || bookings[idBooking].filmId == -1) {
15        cout << "ID Booking tidak valid!\n";
16        return;
17    }
18
19    if (bookings[idBooking].isPaid) {
20        cout << "Tiket sudah dibayar sebelumnya!\n";
21        return;
22    }
23
24    // Update status pembayaran
25    bookings[idBooking].isPaid = true;
26
27    cout << "Pembayaran berhasil!\n";
28    cout << "Film: " << films[bookings[idBooking].filmId].namaFilm << endl;
29    cout << "Jam: " << films[bookings[idBooking].filmId].jamTayang[bookings[idBooking].jamId].jam << endl;
30    cout << "Status: Sudah dibayar\n";
31 }

```

4. Uji Coba dan Hasil Output

```
PS C:\Users\HazzelPW> cd "c

===== MENU UTAMA =====
1. Registrasi
2. Login
3. Keluar
Pilihan Anda:
```

Gambar 4.1 Output menu utama

```
===== MENU UTAMA =====
1. Registrasi
2. Login
3. Keluar
Pilihan Anda: 1

===== REGISTRASI =====
Masukkan Username: hajel
Masukkan Password: 23
Registrasi berhasil!
```

Gambar 4.2 Output Registrasi

```
===== MENU UTAMA =====
1. Registrasi
2. Login
3. Keluar
Pilihan Anda: 2

===== LOGIN =====
Ingin login sebagai apa?
1. Admin
2. User
Pilihan Anda: 1
Masukkan Username: admin
Masukkan Password: selalubentar
Login berhasil!
Selamat Datang Bos admin

===== MENU ADMIN =====
1. Tambah Film
2. Hapus Film
3. Lihat Film
4. Lihat Transaksi
5. Log out
Pilihan Anda: █
```

Gambar 4.3 Output login sebagai admin dan menu admin

```

===== MENU ADMIN =====
1. Tambah Film
2. Hapus Film
3. Lihat Film
4. Lihat Transaksi
5. Log out
Pilihan Anda: 1

===== TAMBAH FILM =====
Masukkan Nama Film: God of War
Film berhasil ditambahkan!

===== MENU ADMIN =====
1. Tambah Film
2. Hapus Film
3. Lihat Film
4. Lihat Transaksi
5. Log out
Pilihan Anda: 2

===== HAPUS FILM =====
Daftar Film:
1. Kimi No Nawa
2. Godzilla : Minus one
3. Avengers : secret wars
4. Your Lie In April
5. Cek Toko Sebelah
6. God of War
Pilih film yang ingin ditarik (1-6): 6
Film berhasil ditarik dari penayangan!

```

Gambar 4.4 Output menambah dan menghapus film

```

===== MENU ADMIN =====
1. Tambah Film
2. Hapus Film
3. Lihat Film
4. Lihat Transaksi
5. Log out
Pilihan Anda: 3

===== DAFTAR FILM =====
1. Kimi No Nawa
   Jam Tayang: 10:00 (20 kursi) 12:00 (20 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)
2. Godzilla : Minus one
   Jam Tayang: 10:00 (20 kursi) 12:00 (20 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)
3. Avengers : secret wars
   Jam Tayang: 10:00 (20 kursi) 12:00 (20 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)
4. Your Lie In April
   Jam Tayang: 10:00 (20 kursi) 12:00 (20 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)
5. Cek Toko Sebelah
   Jam Tayang: 10:00 (20 kursi) 12:00 (20 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)

===== MENU ADMIN =====
1. Tambah Film
2. Hapus Film
3. Lihat Film
4. Lihat Transaksi
5. Log out
Pilihan Anda: 4

===== TRANSAKSI =====

===== MENU ADMIN =====
1. Tambah Film
2. Hapus Film
3. Lihat Film
4. Lihat Transaksi
5. Log out
Pilihan Anda: 5
Kembali ke menu utama

```

Gambar 4.5 Output melihat film, melihat transaksi dan log out dari akun admin

```

===== MENU UTAMA =====
1. Registrasi
2. Login
3. Keluar
Pilihan Anda: 2

===== LOGIN =====
Ingin login sebagai apa?
1. Admin
2. User
Pilihan Anda: 2
Masukkan Username: Danendra Hazzel PW
Masukkan Password: 2409106096
Login berhasil!
Selamat Datang Tuan Danendra Hazzel PW

===== MENU USER =====
1. Pesan Tiket
2. Lihat Film yang Tersedia
3. Lihat Tiket yang Sudah Dipesan
4. Ganti Jam Tayang
5. Batalkan Tiket
6. Bayar Tiket
7. Log out
Pilihan Anda: 

```

Gambar 4.6 Output login sebagai user dan menu utama user

```

===== MENU USER =====
1. Pesan Tiket
2. Lihat Film yang Tersedia
3. Lihat Tiket yang Sudah Dipesan
4. Ganti Jam Tayang
5. Batalkan Tiket
6. Bayar Tiket
7. Log out
Pilihan Anda: 1

===== PESAN TIKET =====
Film yang tersedia:
1. Kimi No Nawa
2. Godzilla : Minus one
3. Avengers : secret wars
4. Your Lie In April
5. Cek Toko Sebelah
Pilih film (1-5): 4
Jam tayang yang tersedia:
1. 10:00 (Kursi tersisa: 20)
2. 12:00 (Kursi tersisa: 20)
3. 15:00 (Kursi tersisa: 20)
4. 17:00 (Kursi tersisa: 20)
5. 19:00 (Kursi tersisa: 20)
Pilih jam tayang (1-5): 2
Booking berhasil!
ID Booking Anda: 0
Film: Your Lie In April
Jam: 12:00
Status: Belum dibayar

```

Gambar 4.7 Output memesan tiket

```

===== MENU USER =====
1. Pesan Tiket
2. Lihat Film yang Tersedia
3. Lihat Tiket yang Sudah Dipesan
4. Ganti Jam Tayang
5. Batalkan Tiket
6. Bayar Tiket
7. Log out
Pilihan Anda: 2

===== FILM YANG TERSEDIA =====
1. Kimi No Nawa
   Jam Tayang: 10:00 (20 kursi) 12:00 (20 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)
2. Godzilla : Minus one
   Jam Tayang: 10:00 (20 kursi) 12:00 (20 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)
3. Avengers : secret wars
   Jam Tayang: 10:00 (20 kursi) 12:00 (20 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)
4. Your Lie In April
   Jam Tayang: 10:00 (20 kursi) 12:00 (19 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)
5. Cek Toko Sebelah
   Jam Tayang: 10:00 (20 kursi) 12:00 (20 kursi) 15:00 (20 kursi) 17:00 (20 kursi) 19:00 (20 kursi)

===== MENU USER =====
1. Pesan Tiket
2. Lihat Film yang Tersedia
3. Lihat Tiket yang Sudah Dipesan
4. Ganti Jam Tayang
5. Batalkan Tiket
6. Bayar Tiket
7. Log out
Pilihan Anda: 3

===== TIKET YANG SUDAH DIPESAN =====
ID Booking: 0
Film: Your Lie In April
Jam: 12:00
Status: Belum dibayar
-----

```

Gambar 4.8 Output melihat film yang tersedia dan melihat tiket yang telah dipesan

```

===== MENU USER =====
1. Pesan Tiket
2. Lihat Film yang Tersedia
3. Lihat Tiket yang Sudah Dipesan
4. Ganti Jam Tayang
5. Batalkan Tiket
6. Bayar Tiket
7. Log out
Pilihan Anda: 4

===== GANTI JAM TAYANG =====
Masukkan ID Booking: 0
Jam tayang saat ini: 12:00
Jam tayang baru yang tersedia:
1. 10:00 (Kursi tersisa: 20)
2. 12:00 (Kursi tersisa: 19)
3. 15:00 (Kursi tersisa: 20)
4. 17:00 (Kursi tersisa: 20)
5. 19:00 (Kursi tersisa: 20)
Pilih jam tayang baru (1-5): 1
Jam tayang berhasil diubah!
Film: Your Lie In April
Jam tayang baru: 10:00

===== MENU USER =====
1. Pesan Tiket
2. Lihat Film yang Tersedia
3. Lihat Tiket yang Sudah Dipesan
4. Ganti Jam Tayang
5. Batalkan Tiket
6. Bayar Tiket
7. Log out
Pilihan Anda: 5

===== BATALKAN TIKET =====
Masukkan ID Booking: 1
Tiket berhasil dibatalkan!

```

Gambar 4.9 Output ganti jam tayang dan membatalkan tiket


```

===== MENU USER =====
1. Pesan Tiket
2. Lihat Film yang Tersedia
3. Lihat Tiket yang Sudah Dipesan
4. Ganti Jam Tayang
5. Batalkan Tiket
6. Bayar Tiket
7. Log out
Pilihan Anda: 6

===== BAYAR TIKET =====
Masukkan ID Booking: 0
Pembayaran berhasil!
Film: Your Lie In April
Jam: 10:00
Status: Sudah dibayar

===== MENU USER =====
1. Pesan Tiket
2. Lihat Film yang Tersedia
3. Lihat Tiket yang Sudah Dipesan
4. Ganti Jam Tayang
5. Batalkan Tiket
6. Bayar Tiket
7. Log out
Pilihan Anda: 7
Kembali ke menu utama

```

Gambar 4.10 Bayar tiket dan log out dari akun user

```

===== MENU ADMIN =====
1. Tambah Film
2. Hapus Film
3. Lihat Film
4. Lihat Transaksi
5. Log out
Pilihan Anda: 4

===== TRANSAKSI =====
ID Booking: 0
Film: Your Lie In April
Jam: 10:00
User: Danendra Hazzel PW
Status: Sudah dibayar
-----
ID Booking: 1

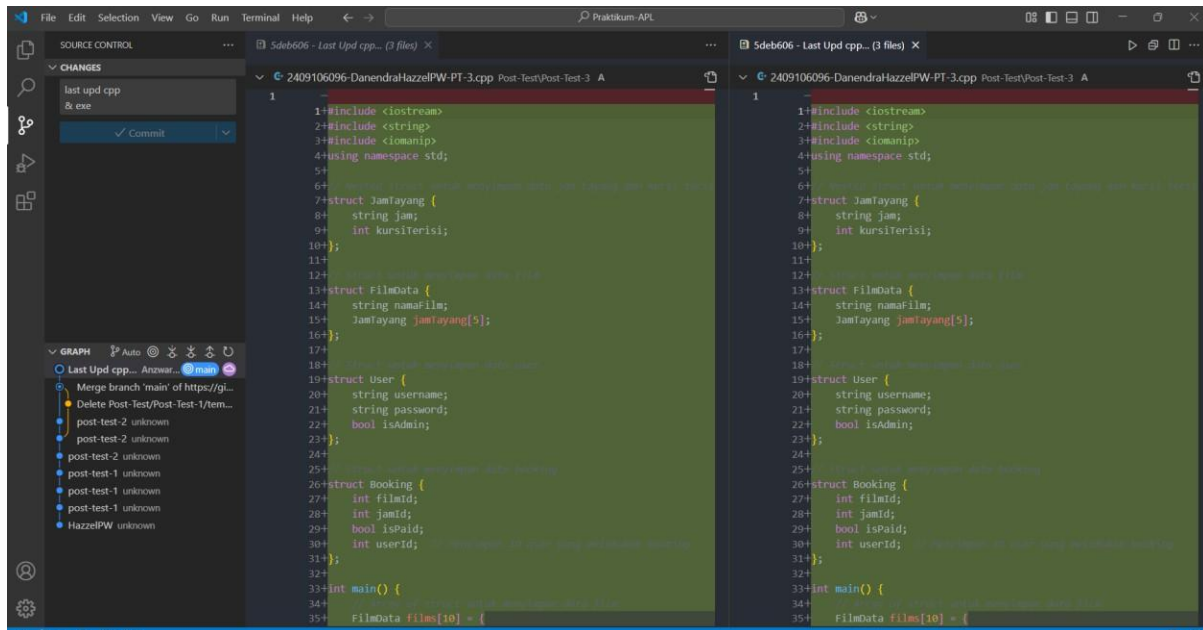
```

Gambar 4.11 Output informasi pada admin jika user melakukan transaksi

```
===== MENU UTAMA =====  
1. Registrasi  
2. Login  
3. Keluar  
Pilihan Anda: 3  
Terima kasih telah menggunakan aplikasi ini, enjoy the movie!  
PS C:\Users\HazzelPW\Downloads>
```

Gambar 4.12 Output keluar dari program

5. Langkah-Langkah Git pada VSCode



1. Pastikan VSCode terhubung dengan github
2. Buka folder yang ingin di commit ke github dengan VSCode
3. Masuk kedalam source control(ctrl+shift+g)
4. Ketik pesan pada folder yang ingin di push, sebagai contoh disini mengetik “Last Update cpp & exe”
5. Tekan “more action” dan pilih bagian “commit & push” yang berguna untuk menyimpan dan mendorong folder kita ke github.
6. Output update akan tampil pada layar dan juga pada “Graph” dibawah