

Laporan Proyek

Tugas Besar

Pemrograman Lanjut

30 Desember 2023



Anggota Kelompok

1. Dizhan Damara Wirareksa (202210370311044)
2. Muhammad Syahrul Bachtiar (202210370311046)
3. Wempy Aditya Wiryawan (202210370311058)
4. Andika Nur Islamy (202210370311063)
5. M. Heykal Dhafariel Rajendra (202210370311094)



Latar Belakang

Proyek ini bertujuan untuk mengembangkan aplikasi pencarian film dengan antarmuka pengguna grafis menggunakan JavaFX. Aplikasi ini memanfaatkan layanan The Movie Database (TMDb) melalui API mereka untuk memberikan informasi tentang film yang dicari oleh pengguna.

Tujuan Proyek:

1. Membangun aplikasi pencarian film yang memungkinkan pengguna mencari informasi tentang film berdasarkan judul.
2. Menyediakan antarmuka pengguna yang intuitif dan menarik menggunakan JavaFX.
3. Mengintegrasikan API The Movie Database (TMDb) untuk mendapatkan data film secara real-time.

Tujuan Proyek

Proyek ini bertujuan untuk menciptakan aplikasi pencarian film yang komprehensif dan interaktif dengan menggunakan teknologi Java dan JavaFX. Tujuan utama proyek ini adalah:

1. Membangun Antarmuka Pengguna yang Menarik:
 - a. Mengembangkan antarmuka pengguna grafis (GUI) yang menarik dan mudah digunakan menggunakan JavaFX.
 - b. Menyediakan elemen-elemen UI seperti kotak pencarian, tampilan hasil pencarian, dan tombol navigasi.
2. Integrasi dengan The Movie Database (TMDb):
 - a. Mengintegrasikan aplikasi dengan API The Movie Database (TMDb) untuk mendapatkan data film secara real-time.
 - b. Memastikan aplikasi mampu melakukan pencarian film berdasarkan judul dengan akurasi tinggi.
3. Penyimpanan Riwayat Pencarian:
 - a. Menyediakan fitur penyimpanan riwayat pencarian pengguna ke dalam file lokal.
 - b. Memungkinkan pengguna mengakses dan mengelola riwayat pencarian mereka dengan mudah.
4. Fitur Detail Film:
 - a. Menambahkan kemampuan untuk melihat detail lengkap dari setiap film, termasuk sinopsis, gambar, dan informasi terkait lainnya.
5. Penggunaan Konsep Pemrograman Lanjut:
 - a. Menerapkan konsep-konsep pemrograman lanjut seperti integrasi API eksternal, manajemen file, dan pengembangan antarmuka pengguna menggunakan JavaFX.
 - b. Menyediakan solusi yang efisien dan terstruktur.
6. Peningkatan Pengalaman Pengguna:
 - a. Memberikan pengalaman pencarian film yang cepat, efisien, dan menyenangkan.
 - b. Memastikan antarmuka pengguna dirancang agar mudah dipahami dan diakses oleh berbagai tingkat pengguna.

Metodologi

Metodologi pengembangan proyek ini didasarkan pada pendekatan langkah-demi-langkah yang mencakup tahapan analisis, desain, implementasi, dan pengujian. Berikut adalah detail dari masing-masing fase:

1. Analisis Kebutuhan:
 - Mengidentifikasi kebutuhan dan persyaratan pengguna untuk aplikasi pencarian film.
 - Menganalisis fitur yang diinginkan, termasuk pencarian film, tampilan hasil, penyimpanan histori, dan fitur detail film.
2. Desain Antarmuka Pengguna:
 - Membuat sketsa dan wireframe untuk antarmuka pengguna.
 - Menentukan tata letak, warna, dan elemen-elemen UI untuk mencapai desain yang menarik dan intuitif.
3. Integrasi dengan API TMDb:
 - Mempelajari dokumentasi API The Movie Database (TMDb) untuk memahami endpoint-endpoint yang diperlukan.
 - Menerapkan logika pengambilan data dari TMDb ke dalam aplikasi.
4. Implementasi Backend:
 - Membuat kelas backend (MovieSearchBackend.java) untuk menangani koneksi ke API dan pemrosesan data.
 - Menyusun logika untuk penyimpanan riwayat pencarian ke dalam file lokal.
5. Implementasi Antarmuka Pengguna (Frontend):
 - Menggunakan JavaFX untuk mengimplementasikan desain antarmuka pengguna.
 - Menyusun kelas kontroler dan FXML untuk mengatur interaksi antara frontend dan backend.
6. Pengujian Fungsional:
 - Melakukan pengujian fungsional untuk memastikan bahwa setiap fitur berfungsi seperti yang diharapkan.
 - Menjalankan uji coba untuk menangkap dan menangani situasi kesalahan atau bug.
7. Optimisasi dan Pemeliharaan:
 - Mengoptimalkan performa aplikasi dan mengidentifikasi area-area yang memerlukan pemeliharaan.
 - Menerapkan perbaikan dan pembaruan yang diperlukan setelah fase pengujian.
8. Dokumentasi Kode:



- Membuat dokumentasi yang lengkap untuk setiap kelas, metode, dan logika implementasi.
- Menambahkan komentar yang jelas di dalam kode untuk memudahkan pemahaman dan pengembangan berkelanjutan.

Implementasi

1. Modul 1 - PROGRAM CORRECTNESS

Dalam program ini mengimplementasikan fitur error handling dimana terdapat try dan catch untuk menangani error ketika program di jalankan.

```
....
try (FileWriter writer = new FileWriter(fileName, true)) {
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
    String timestamp = dateFormat.format(new Date());

    writer.write(timestamp + "," + title + "," + imageUrl + "," +
description + "\n");
} catch (IOException e) {
    e.printStackTrace();
    // Handle kesalahan penyimpanan ke file di sini
    MovieSearchController.showAlert("Error", "Gagal dalam menyimpan
file!");
}
....
```

```
public static List<Movie> fetchMoviesFromApi(String title) {
    List<Movie> results = new ArrayList<>();
    try {
        ....
    } catch (Exception e) {
        e.printStackTrace();
        // Handle kesalahan koneksi atau pemrosesan respons di sini
        MovieSearchController.showAlert("Error", "Gagal dalam mengambil
data, cek koneksi anda, dan coba lagi!");
    }
}
```

2. Modul 3 - MODERN PROGRAMMING ENVIRONMENT AND DOCUMENTATION STYLE

Dalam program ini terdapat dokumentasi di setiap fitur/function/bagian pada code nya sehingga dapat memudahkan programmer dalam melakukan maintenance program, dan

juga terdapat file readme.md sebagai dokumentasi/instruksi program untuk programmer lain ketika ingin meng cloning proyek/program dari github.

```
# Aplikasi Pencarian Film
Aplikasi Pencarian Film adalah proyek pengembangan aplikasi sederhana
menggunakan Java dan JavaFX. Aplikasi ini memungkinkan pengguna mencari
informasi tentang film, melihat detail film, dan menyimpan riwayat
pencarian.
## Fitur
- **Pencarian Film:** Pengguna dapat mencari film berdasarkan judul.
- **Tampilan Hasil Pencarian:** Hasil pencarian ditampilkan dalam tata
letak grid yang menarik.
- **Detail Film:** Pengguna dapat melihat detail lengkap dari setiap film.
- **Riwayat Pencarian:** Aplikasi menyimpan riwayat pencarian pengguna.
## Teknologi yang Digunakan
- Java
- JavaFX
- The Movie Database (TMDb) API
## Cara Menjalankan Aplikasi
1. Pastikan Anda memiliki Java diinstal di komputer Anda.
2. Unduh proyek ini atau clone repositori ini ke komputer Anda.
3. Buka proyek menggunakan IDE Java favorit Anda.
4. Jalankan aplikasi dari kelas utama: `MovieSearchApplication.java`.
## Kontribusi
Jika Anda ingin berkontribusi pada proyek ini, silakan fork repositori ini
dan kirim pull request.
## Lisensi
Proyek ini dilisensikan di bawah Lisensi MIT - lihat berkas
[LICENSE.md](LICENSE.md) untuk detail lebih lanjut.
Happy coding!
```

3. Modul 4 - API (APPLICATION PROGRAMMING INTERFACE)

Dalam program ini menggunakan free open source API yang disediakan oleh TMDB untuk data film yang telah release. Jadi program ini dapat mendapatkan sumber data melalui api, dan menampilkan seluruh data yang di dapatkan di tampilan aplikasi.

```
public static List<Movie> fetchMoviesFromApi(String title) {
    List<Movie> results = new ArrayList<>();

    try {
```



```

// Ganti URL dengan endpoint pencarian film TMDb
String apiUrl = "https://api.themoviedb.org/3/search/movie";
String apiKeyParam = "?api_key=" + API_KEY;
String queryParams = "&query=" + title;

// Membuat URL untuk pencarian film
URL url = new URL(apiUrl + apiKeyParam + queryParams);
URLConnection connection = (URLConnection)
url.openConnection();

// Set metode HTTP yang diinginkan (GET)
connection.setRequestMethod("GET");

// Baca respons dari API
BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
StringBuilder response = new StringBuilder();
String line;

while ((line = reader.readLine()) != null) {
    response.append(line);
}

reader.close();

// Proses respons JSON menggunakan pustaka Gson
JsonParser parser = new JsonParser();
JsonObject jsonResponse =
parser.parse(response.toString()).getAsJsonObject();

// Dapatkan array hasil pencarian film dari respons JSON
JsonArray resultsArray =
jsonResponse.getAsJsonArray("results");

// Ambil informasi judul dan URL gambar dari setiap objek hasil
pencarian
for (int i = 0; i < resultsArray.size(); i++) {
    JsonObject movieObject =
resultsArray.get(i).getAsJsonObject();
    String movieTitle = movieObject.get("title").getString();
    String imageUrl = "https://image.tmdb.org/t/p/w500" +

```

```

movieObject.get("poster_path").getAsString();
        String movieDesc =
movieObject.get("overview").getAsString();

        results.add(new Movie(movieTitle, imageUrl, movieDesc));
    }

    // Tutup koneksi
    connection.disconnect();

    } catch (Exception e) {
        e.printStackTrace();
        // Handle kesalahan koneksi atau pemrosesan respons di sini
        MovieSearchController.showAlert("Error", "Gagal dalam mengambil
data, cek koneksi anda, dan coba lagi!");
    }

    return results;
}

```

4. Modul 5 - FILE HANDLING

Dalam program ini juga terdapat file handler yang dimana untuk menangani ketika program telah berhasil mendapatkan data, maka data akan di simpan ke dalam file .txt. yang dimana data tersebut berguna sebagai histori, sehingga dapat dipanggil kembali data yang telah disimpan.

```

public static void saveToFile(String title, String imageUrl, String
description, String fileName) {
    try (FileWriter writer = new FileWriter(fileName, true)) {
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
        String timestamp = dateFormat.format(new Date());

        writer.write(timestamp + "," + title + "," + imageUrl + "," +
description + "\n");
    } catch (IOException e) {
        e.printStackTrace();
        // Handle kesalahan penyimpanan ke file di sini
        MovieSearchController.showAlert("Error", "Gagal dalam menyimpan

```

```

file!");
    }
}

    public static List<String[]> readAllFromFile(String fileName) {
        List<String[]> entries = new ArrayList<>();

        try (BufferedReader reader = new BufferedReader(new
FileReader(fileName))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                entries.add(parts);
            }
        } catch (IOException e) {
            e.printStackTrace();
            // Handle kesalahan membaca dari file di sini
            MovieSearchController.showAlert("Error", "Gagal dalam membuka
file!");
        }

        return entries;
    }
}

```

5. Modul 6 - GUI (GRAPHICAL USER INTERFACE) USING JAVA FX

Program ini menggunakan library javafx sebagai library GUI, sehingga program ini memiliki tampilan UI untuk memanjakan para pengguna. Sehingga program ini dapat lebih mudah digunakan karena pengguna dapat berinteraksi langsung dengan UI/UX / di sisi front end.

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.ScrollPane?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.StackPane?>

```

```

<?import javafx.scene.layout.VBox?>

<StackPane prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/17" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.tubes_proglan.MovieSearchController">
    <VBox alignment="CENTER" spacing="10">

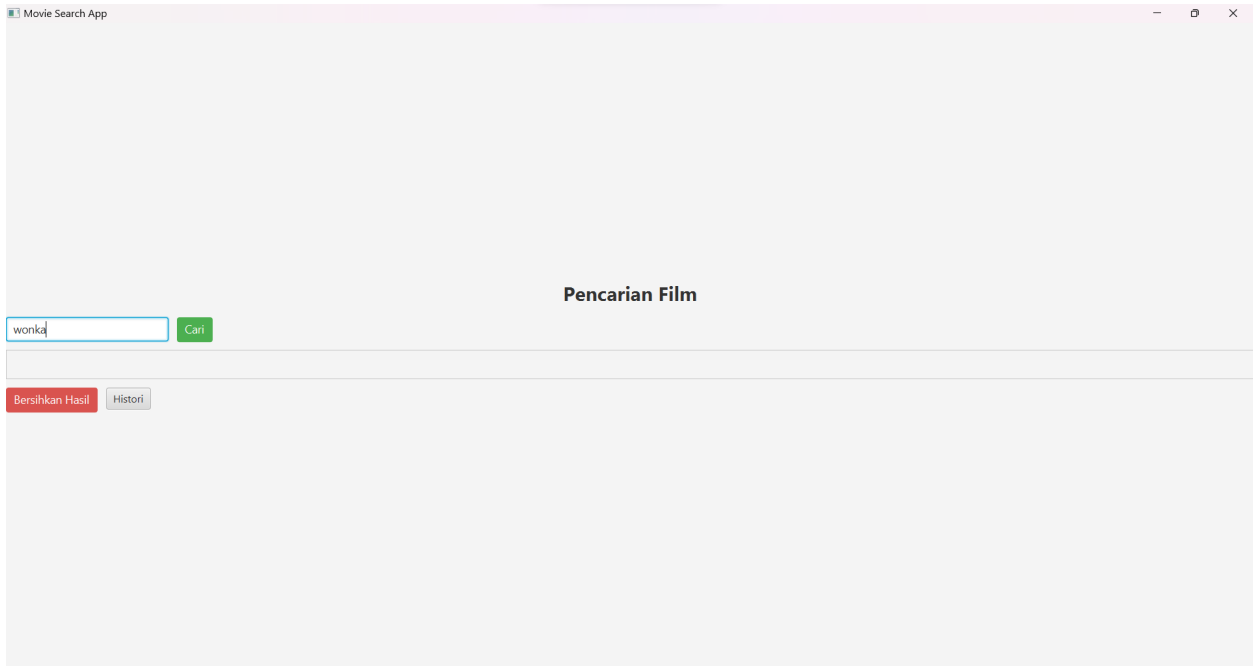
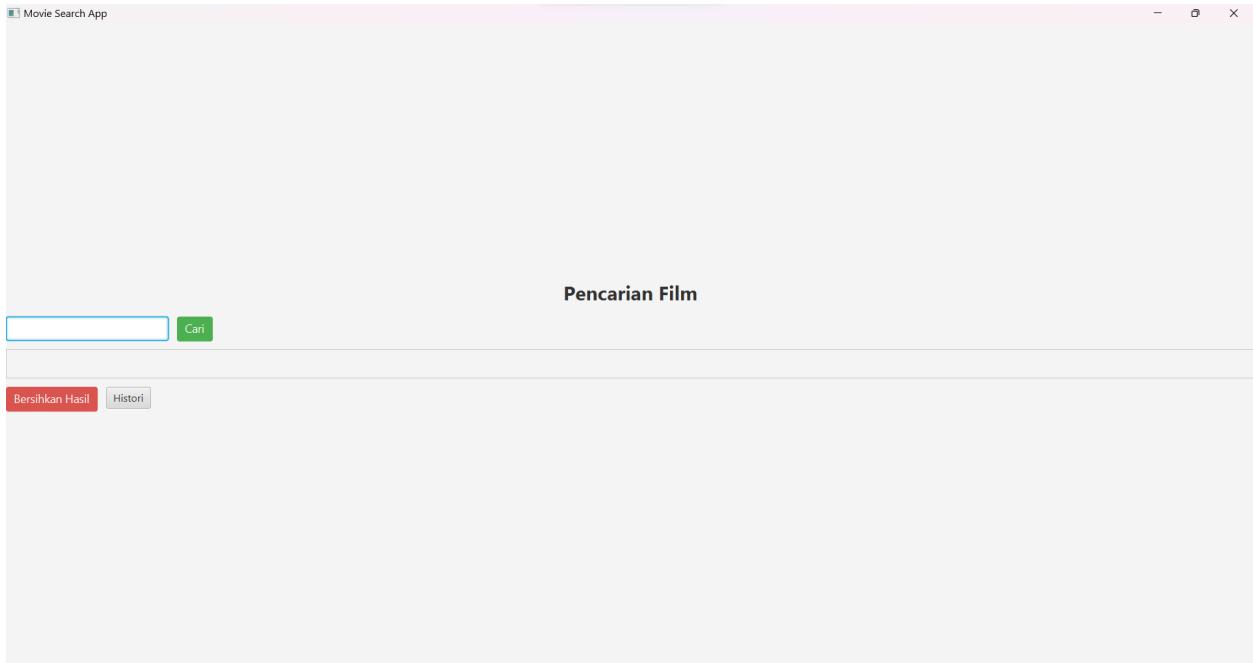
        <Label text="Pencarian Film" styleClass="title-label" />

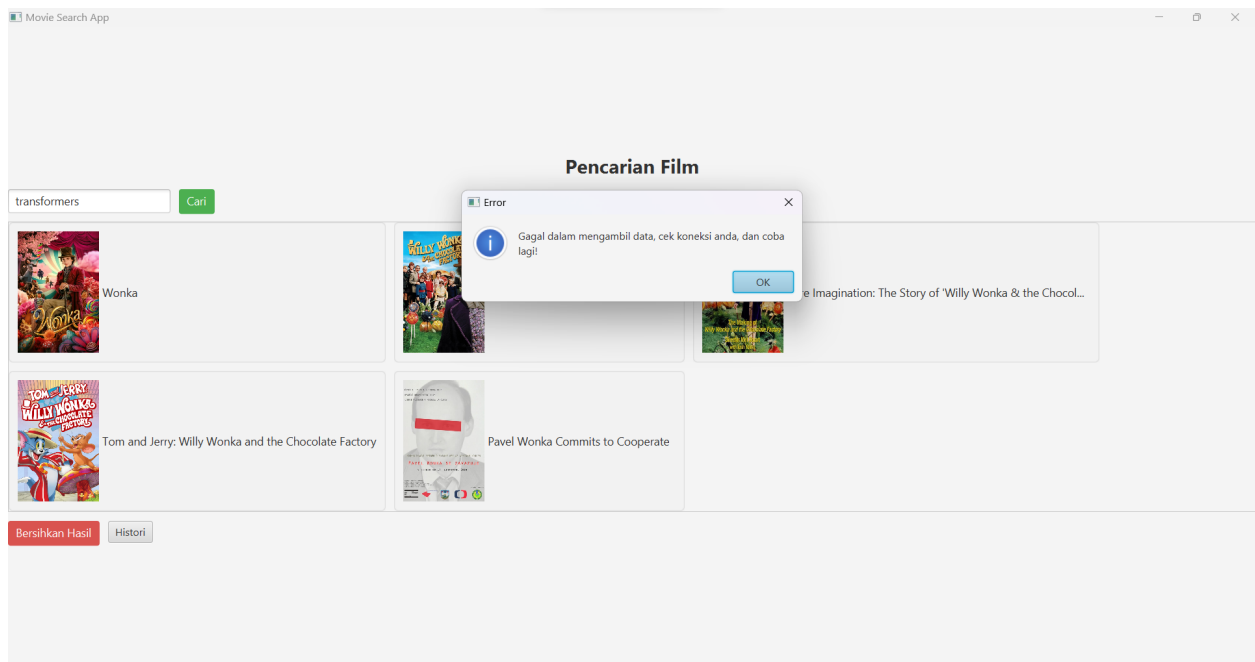
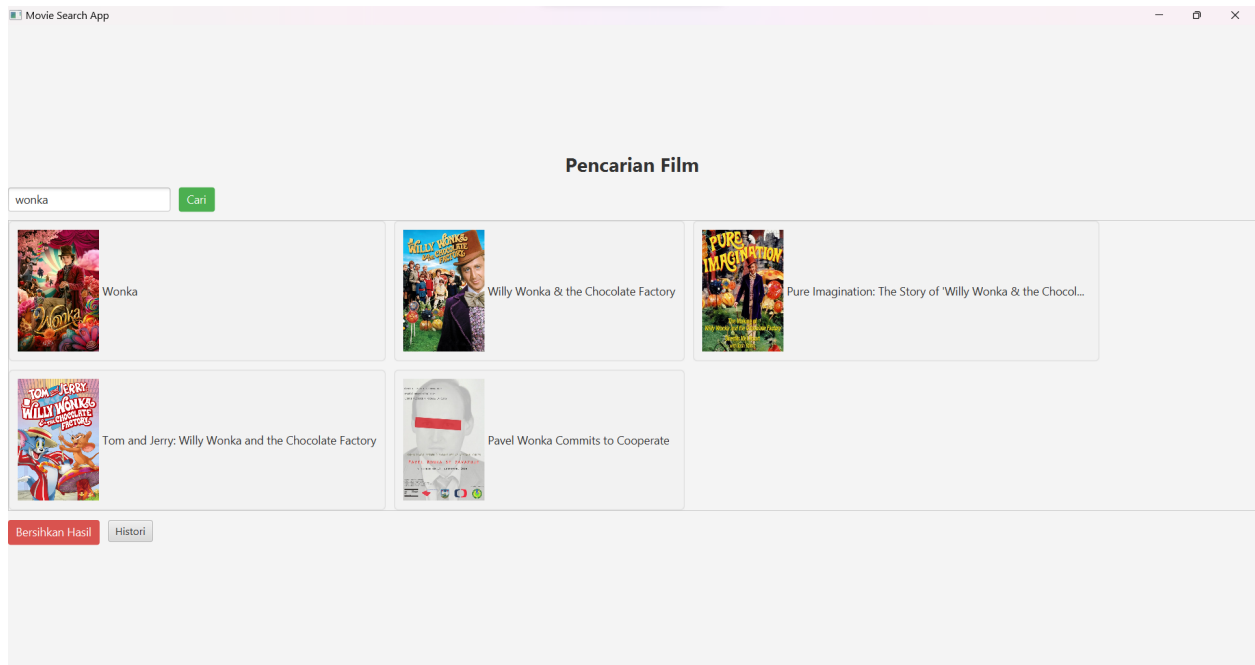
        <!-- VBox untuk tata letak Text Field dan Tombol Pencarian -->
        <VBox spacing="10">
            <HBox spacing="10">
                <TextField fx:id="searchField" promptText="Masukkan judul
film" styleClass="search-field" />
                <Button text="Cari" onAction="#searchButtonClicked"
styleClass="search-button" />
            </HBox>
        </VBox>

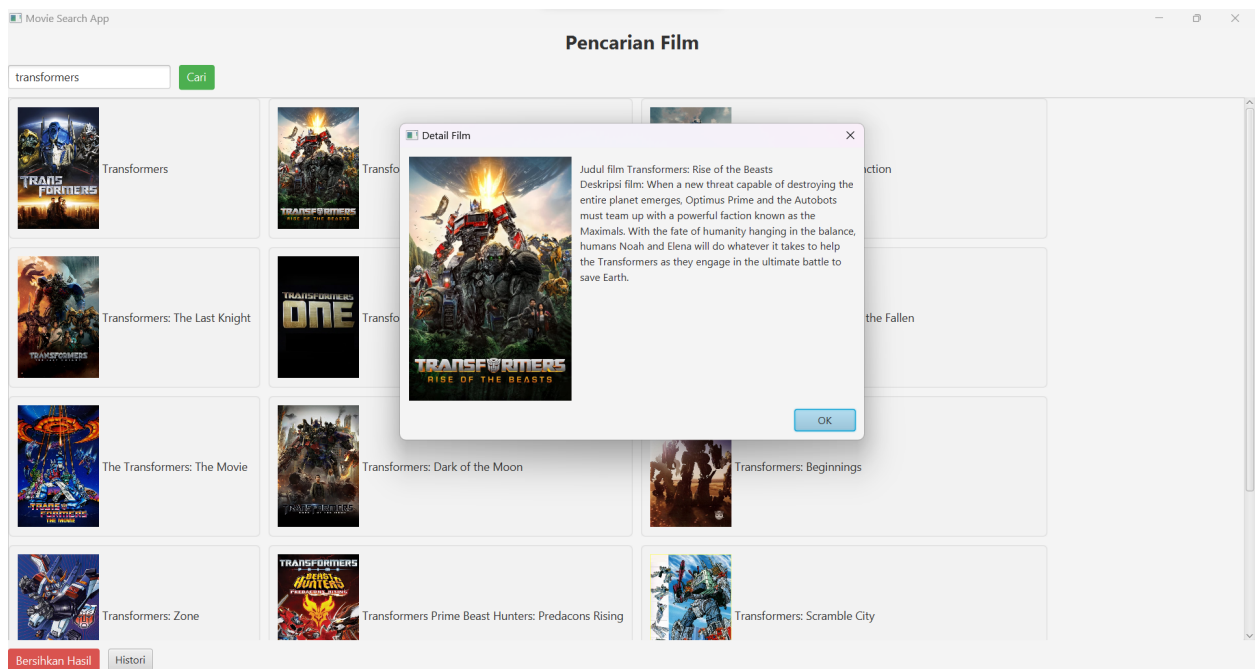
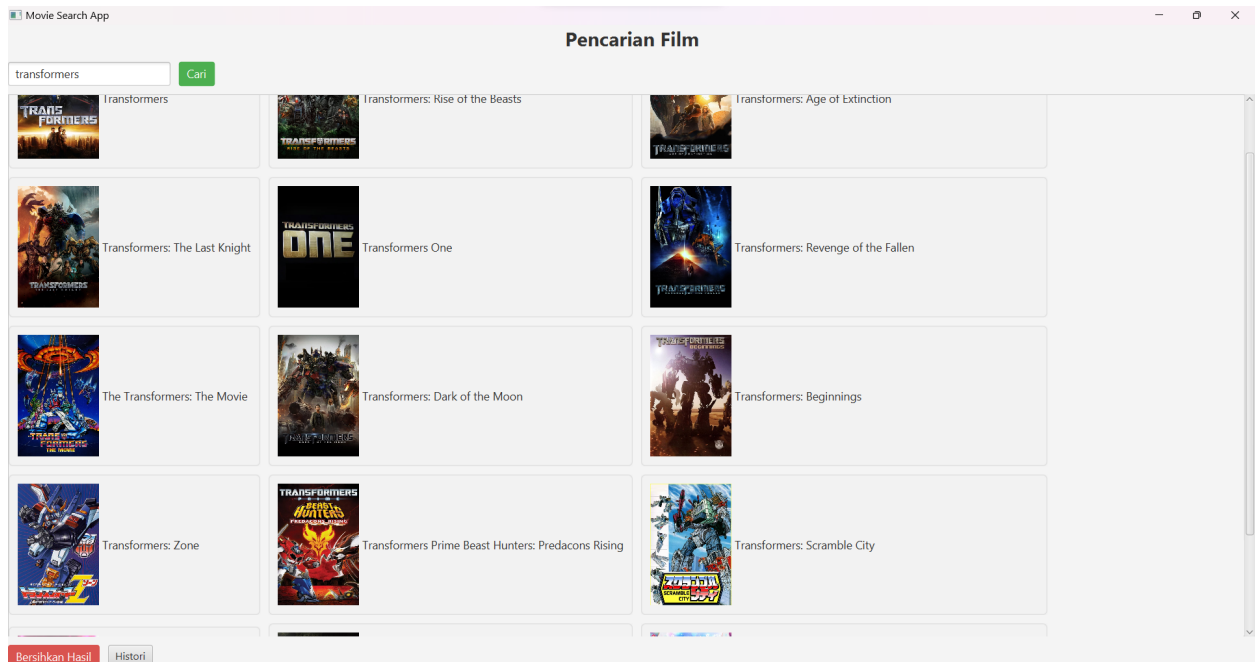
        <!-- Gunakan ScrollPane di sekitar resultGrid -->
        <ScrollPane fitToHeight="true" fitToWidth="true">
            <GridPane fx:id="resultGrid" styleClass="result-grid" hgap="10"
vgap="10" />
        </ScrollPane>

        <HBox spacing="10">
            <Button text="Bersihkan Hasil" onAction="#clearResults"
styleClass="clear-button" />
            <Button text="Histori" onAction="#showHistory"/>
        </HBox>
    </VBox>
</StackPane>

```







Pencarian Film

Cari

Bersihkan Hasil

Histori

Pencarian Film

Masukkan judul film

Cari

Bersihkan Hasil


Histori

Movie Search App


Pencarian Film

Masukkan judul film


Cari




Wonka




Willy Wonka & the Chocolate Factory




Pure Imagination: The Story of 'Willy Wonka & the Chocol...




Tom and Jerry: Willy Wonka and the Chocolate Factory




Pavel Wonka Commits to Cooperate




Transformers



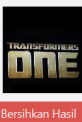
Transformers: Rise of the Beasts




Transformers: Age of Extinction




Transformers: The Last Knight



Transformers One



Transformers: Revenge of the Fallen



The Transformers: The Movie

Bersihkan Hasil

Histori


```

2023-12-29 21:41:14,Stand by Me Doraemon,https://image.tmbd.org/t/p/w500/wc7XQbfx6EIqCuvM8Mt3a1sb2Y.jpg,Sewashi and Doraemon find themselves way bac
2023-12-29 21:41:14,Stand by Me Doraemon 2,https://image.tmbd.org/t/p/w500/vBv8i0FPLnXmtELUjCfC70KHsR4.jpg,Nobita travels to the future to show his beloved gran
2023-12-29 21:41:14,Doraemon: Nobita's Sky Utopia,https://image.tmbd.org/t/p/w500/uux6M8z3hXL0kq8LXSzq8528mrq.jpg,Doraemon, Nobita and his friends go to find Utt
2023-12-29 21:41:14,Doraemon: Nobita's New Dinosaur,https://image.tmbd.org/t/p/w500/wxCZmXRJa8hSv1Tpih8TBSR4o6b.jpg,Nobita accidentally found a fossil dinosaur e
2023-12-29 21:41:14,Doraemon: New Nobita's Great Demon - Peko and the Exploration Party of Five,https://image.tmbd.org/t/p/w500/xKGXQF6ca4nKc0308aZ6nWz06c.jpg,h
2023-12-29 21:41:14,Doraemon: Nobita's Great Adventure in the Antarctic Kachi Kochi,https://image.tmbd.org/t/p/w500/dWxEck1aqWF7Z90v0IzhNwL91w7.jpg,Doraemon and
2023-12-29 21:41:14,Doraemon: Nobita's Little Star Wars 2021,https://image.tmbd.org/t/p/w500/48gKZioIDeU0I0afbYv3kh9u9RQ.jpg,One day during summer vacation, a pa
2023-12-29 21:41:14,Doraemon: Nobita's Dinosaur,https://image.tmbd.org/t/p/w500/g0D94CrNm8bTCgHdpURTMr46s1Q.jpg,After bringing a fossilized egg back to life with
2023-12-29 21:41:14,Doraemon: Nobita's Treasure Island,https://image.tmbd.org/t/p/w500/xiLRCLQmKSVAbiu6rgCRzNQjCSX.jpg,Upon hearing about the story of Treasure I
2023-12-29 21:41:14,Doraemon: Nobita's New Great Adventure Into the Underworld - The Seven Magic Users,https://image.tmbd.org/t/p/w500/kg0Nypy6zEPpE3FUFcrD6egbSt
2023-12-29 21:41:14,Doraemon: Nobita's Chronicle of the Moon Exploration,https://image.tmbd.org/t/p/w500/zAZ43208e643rB3pcx8jSLqc689.jpg,During a Japanese roboti
2023-12-29 21:41:14,Doraemon: Nobita and the Space Heroes,https://image.tmbd.org/t/p/w500/pYVJgcuCU9saw76EujygsRdkSQd.jpg,One day Doraemon, Nobita, Shizuka, Giar
2023-12-29 21:41:14,Doraemon: Nobita's Great Battle of the Mermaid King,https://image.tmbd.org/t/p/w500/lvi0MjGJvWPAAQqaxCwwMslqB68.jpg,When Nobita feels like di
2023-12-29 21:41:14,Doraemon: Nobita and the Birth of Japan,https://image.tmbd.org/t/p/w500/yF6pCSsRKKn5RzeU010zX0zTyVQ.jpg,Doraemon and his friends travel to ar
2023-12-29 21:41:14,Doraemon: Nobita and the Galaxy Super-express,https://image.tmbd.org/t/p/w500/hrIyZ16i9CpXioi0KhpK4U9M6Me.jpg,Doraemon went missing for 3 day
2023-12-29 21:41:14,Doraemon: Nobita and the New Steel Troops: Winged Angels,https://image.tmbd.org/t/p/w500/57IghJfJjXpRwg7VeQIGN6nf40ZH.jpg,The title character,
2023-12-29 21:41:14,Doraemon: Nobita and the Robot Kingdom,https://image.tmbd.org/t/p/w500/qsR4Hcdh2XWtUDQumj1ML5c1rNW.jpg,Doraemon and friends travels into anot
2023-12-29 21:41:14,Doraemon: Nobita and the Island of Miracles - Animal Adventure,https://image.tmbd.org/t/p/w500/mUJGtID4Uy06nLE0Ztqq48y4Hdw.jpg,Nobita and Dor
2023-12-29 21:41:14,Doraemon: Nobita and the Haunts of Evil,https://image.tmbd.org/t/p/w500/TTEW2NYtf0tTaHqY2id2.jpg,Nothing is impossible for Doraemon's
2023-12-29 21:41:14,Doraemon: Nobita and the Windmasters,https://image.tmbd.org/t/p/w500/ko6A0xiYQIexLv26D0zM0FUL3V.jpg,One day, Nobita runs into a creature-lik
2023-12-29 21:52:30,Wonka,https://image.tmbd.org/t/p/w500/qhb1q0ilapbapxWQn9jtRCMwXJF.jpg,Willy Wonka - chock-full of ideas and determined to change the world or
2023-12-29 21:52:30,Willy Wonka & the Chocolate Factory,https://image.tmbd.org/t/p/w500/vmpsZkrs4UvKp9r1atL8B3frA63.jpg,When eccentric candy man Willy Wonka prom
2023-12-29 21:52:30,Pure Imagination: The Story of 'Willy Wonka & the Chocolate Factory',https://image.tmbd.org/t/p/w500/nxxZ6mE456c5ujVrkV2B42mmD4.jpg,Retrospe
2023-12-29 21:52:30,Tom and Jerry: Willy Wonka and the Chocolate Factory,https://image.tmbd.org/t/p/w500/vh51YhJ80l1inSdep61sF8gIS8m.jpg,The classic Roald Dahl t
2023-12-29 21:52:30,Pavel Wonka Commits to Cooperate,https://image.tmbd.org/t/p/w500/AbFQpeeLEHn7QyBQVxF7h5ZMD78.jpg,
2023-12-29 22:08:08,Transformers,https://image.tmbd.org/t/p/w500/lkZ9gqCEjzX85LKR6Jjdiu6AXNp.jpg,Teenage Sam Witwicky becomes involved in the ancient struggle be
2023-12-29 22:08:08,Transformers: Rise of the Beasts,https://image.tmbd.org/t/p/w500/gPbM0MK8CP8A174rmUwGsADNYKD.jpg,When a new threat capable of destroying the
2023-12-29 22:08:08,Transformers: Age of Extinction,https://image.tmbd.org/t/p/w500/jyzrxfX2WaeY60kYZpPYepSjGz4S.jpg,As humanity picks up the pieces, following th
2023-12-29 22:08:08,Transformers: The Last Knight,https://image.tmbd.org/t/p/w500/s5HQf26b3LI02cRcFwNL9sn1o1o.jpg,Autobots and Decepticons are at war, with humar
2023-12-29 22:08:08,Transformers One,https://image.tmbd.org/t/p/w500/nUAIob1PYi69b6cMxKfT7FRRsv.jpg,The story of how a young Optimus Prime and Megatron went fr

```

Dalam konteks Java, "search result" merujuk pada hasil dari operasi pencarian di dalam suatu kumpulan data. Pencarian ini dapat dilakukan untuk menemukan elemen tertentu dalam struktur data seperti array, daftar, atau koleksi lainnya. Hasil pencarian tersebut dapat berupa indeks di mana elemen ditemukan atau informasi lain yang menunjukkan keberadaan atau ketidakberadaan elemen dalam kumpulan data.



Penutup

Kesimpulannya, aplikasi yang dikembangkan menggunakan Java memiliki struktur kelas yang terorganisir dengan baik untuk menangani berbagai aspek aplikasi. Kelas-kelas seperti `FileHandler`, `HelloController`, `Movie`, `MovieSearchApp`, `MovieSearchBackend`, dan `MovieSearchController` masing-masing memiliki peran dan tanggung jawabnya sendiri.

`FileHandler` berfungsi untuk menangani operasi file terkait video, `HelloController` bertanggung jawab menangani permintaan pengguna, khususnya dalam pencarian film. Kelas `Movie` digunakan untuk menyimpan informasi film, sementara `MovieSearchApp` mengelola aplikasi pencarian film dan `MovieSearchBackend` menangani proses pencarian dengan mengeksekusi query dan mengambil data yang diperlukan. `MovieSearchController`, pada gilirannya, mengatur permintaan pencarian dari pengguna.

Dengan struktur kelas ini, pengguna dapat dengan mudah mencari film berdasarkan kriteria tertentu dan menikmati video yang tersedia. Setiap kelas berperan penting dalam mengelola berbagai aspek aplikasi, seperti pengelolaan file, interaksi pengguna, dan proses pencarian film. Desain ini menciptakan aplikasi yang terstruktur, modular, dan efisien dalam menangani tugas-tugasnya.

