

Penerapan Rest API Menggunakan Retrofit Untuk Sistem Informasi Film Berbasis Android (Studi Kasus: Sinopsis Film)

Robbie Christover ¹, Sugiyatno ^{1,*}, Herlawati ¹

* Korespondensi: e-mail: sugiyatno@dsn.ubharajaya.ac.id

¹ Informatika; Universitas Bhayangkara Jakarta Raya; Jl. Raya Perjuangan Bekasi Utara, Kota Bekasi, 88955882 / 88955871; e-mail: robbie.christover18@mhs.ubharajaya.ac.id, sugiyatno@dsn.ubharajaya.ac.id, herlawati@dsn.ubharajaya.ac.id

Submitted : **13 Agustus 2022**
Revised : **16 September 2022**
Accepted : **19 Oktober 2022**
Published : **30 November 2022**

DOI:
<https://doi.org/10.31599/jsracs.v3i2.1393>

Abstract

Looking for film information is something that is often done before watching, but not infrequently many people waste time and money because the films they watch are not what they want, due to the lack of information on the films they want to watch. From this problem, the author proposes an application by implementing Rest Api Using Retrofit for an Android-Based Film Information System so that it can search for film synopsis, by utilizing an open-source API. The result of this research is an application that will provide information on the synopsis of the film you are looking for to watch.

Keywords: *Android, Film Information, Film Synopsis, Rest API, Retrofit*

Abstrak

Mencari Informasi Film merupakan hal yang sering dilakukan sebelum menonton, namun tak jarang juga banyak yang menghabiskan waktu dan uang karena film ditonton tidak sesuai dengan yang diinginkan, karena kurangnya Informasi film yang diinginkan Ditonton. Dari permasalahan ini Peneliti Mengusulkan aplikasi dengan menerapkan Rest Api Menggunakan Retrofit Untuk Sistem Informasi Film Berbasis Android sehingga dapat melakukan pencarian Sinopsis film, dengan memanfaatkan API yang bersifat *Open Source*. Hasil dari penelitian ini adalah sebuah aplikasi yang akan memberikan informasi Sinopsis film yang dicari untuk ditonton.

Kata kunci: *Sinopsis Film, Rest API, Android, Informasi Film, Retrofit*

1. Pendahuluan

Di Era Revolusi Industri 4.0, Perkembangan teknologi informasi semakin pesat, hampir seluruh kegiatan dan aktivitas kehidupan manusia tidak terlepas dari *Smartphone*, baik di bidang pendidikan, hiburan, dan lain sebagainya. begitu pula dengan aktivitas mencari informasi sinopsis film untuk ditonton bioskop. Berdasarkan informasi yang peneliti dapat, Bekraf bersama Rumah Sinema melakukan survei terhadap dua ribu responden terkait segmentasi dan pengambilan keputusan masyarakat untuk menonton film. Dari survei tersebut, 59% penonton merupakan mahasiswa jenjang pendidikan Strata-1 (S1) dan 33% siswa Sekolah Menengah Atas (SMA). Lalu, 43% penonton di bioskop pendapatan berpendapatan Rp 1 juta-Rp 5 juta. Kemudian, 85% responden mengaku menonton di bioskop hingga dua kali sebulan. Bahkan, 1% responden menonton di bioskop hingga enam kali dalam sebulan (Azizah, 2019), namun tak jarang juga yang berkunjung ke bioskop menghabiskan waktu dan uang, karena film yang ingin ditonton tidak sesuai dengan apa yang diinginkan, karena kurangnya informasi Sinopsis film yang didapatkan.

Dari permasalahan tersebut peneliti ingin berkontribusi untuk meneliti dan mengembangkan aplikasi dengan menerapkan REST API menggunakan Retrofit yang dapat melakukan Pencarian Informasi Sinopsis Film, dengan memanfaatkan API pada website <https://www.themoviedb.org/> (*The Movie Database (TMDb)*, n.d.). Sehingga dengan ini, aplikasi tersebut dapat menunjukan Informasi Sinopsis Film. Android merupakan sistem operasi Linux yang bersifat *Open source* untuk perangkat *mobile* atau telepon genggam. Pada tahun 2005, Google membeli Android dengan ditandai juga didirikannya *Open Handset Alliance (OHA)* yang bertujuan untuk melakukan pengembangan standar terbuka bagi perangkat seluler. Android menawarkan penekanan yang menyeluruh dalam pengembangan aplikasinya. Sehingga aplikasi yang dibangun dapat berjalan pada berbagai perangkat sistem operasi android seperti *smartphone*, *smartwatch*, *table* dan lainnya (Budiman, 2019).

Kotlin merupakan bahasa pemrograman yang pragmatis untuk android yang mengkombinasikan *object oriented* (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman Java. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis *desktop*, *web* dan bahkan untuk *backend* (Sibarani et al., 2018). REST API adalah API berbasis *website* yang menggunakan teknologi REST dan menggunakan format JSON (*JavaScript Object Notation*), yaitu sebuah format pertukaran data yang menggunakan metode http GET, POST, PUT, dan DELETE untuk menerima, membuat, memperbarui dan menghapus resource (Ramadan & Handaga, 2019). API merupakan cara untuk berinteraksi secara terprogram dengan komponen perangkat lunak atau sumber daya yang terpisah. Adanya API membuat produk atau layanan yang kamu gunakan terhubung dengan produk dan layanan lainnya tanpa harus tahu bagaimana cara penerapannya (Ramadan & Handaga, 2019).

2. Metode Penelitian

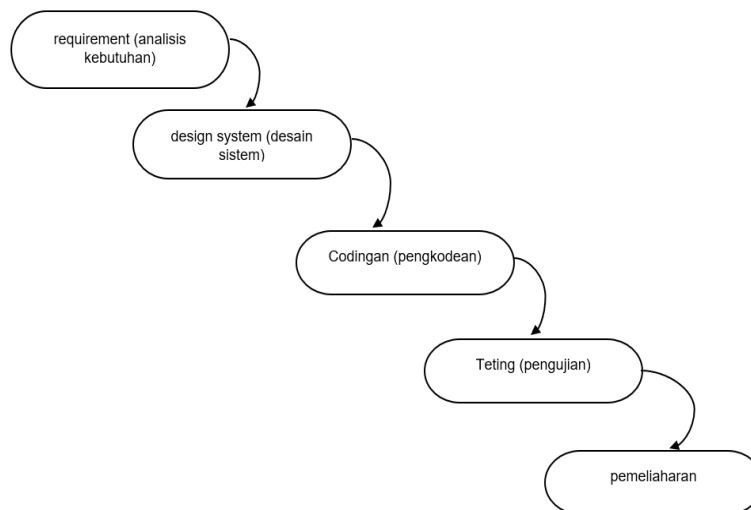
Untuk mendapatkan data yang aktual agar dapat digunakan untuk memecahkan suatu masalah, maka dibutuhkan sebuah cara ilmiah dalam penelitian.

2.1. Pengumpulan Data

Metode pengumpulan data digunakan dengan tujuan untuk memperoleh data yang sesuai harapan penelitian. Pengumpulan data menggunakan 3(tiga) jenis teknik yaitu sebagai berikut: (a) Studi pustaka berguna untuk menentukan metode yang tepat dalam menyelesaikan masalah penelitian, serta sebagai acuan untuk memperoleh informasi yang berkaitan dengan topik pembahasan; (b) Observasi yaitu pengamatan fenomena masalah yang terjadi melalui artikel berita nasional, untuk menyusun pertanyaan yang akan diajukan kepada narasumber saat wawancara; (c) Wawancara untuk memperkuat fenomena yang terjadi di internet, peneliti mewawancarai beberapa pihak yang sering berkunjung ke Bioskop.

2.2. Pengembangan Sistem

Waterfall merupakan salah satu metode SDLC (*Software Development Life Cycle*) atau Siklus Hidup pengembangan sistem dan rekayasa perangkat lunak, yang digunakan untuk proses pengembangan dan perubahan sistem.



Sumber: Hasil Penelitian (2022)

Gambar 1. Tahapan *Waterfall*

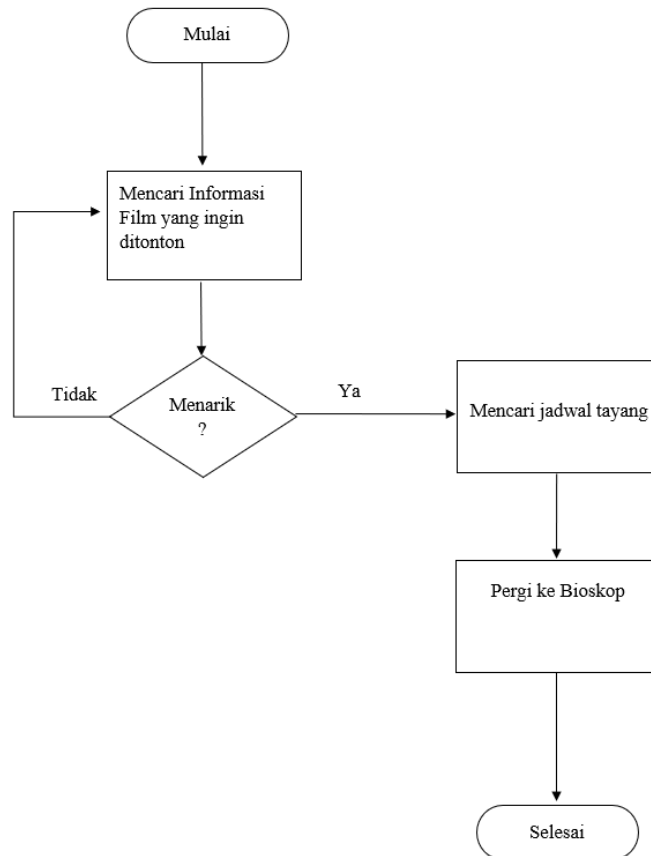
Waterfall adalah metode yang bersifat sistematis dan berurutan dalam mengembangkan perangkat lunak, mulai dari *requirement* (analisis kebutuhan), *design system* (desain sistem), *codingan* (pengkodean), *testing* (pengujian), penerapan program, dan pemeliharaan (Putri, 2022).

3. Hasil dan Pembahasan

3.1. Sistem Berjalan

Dalam mengembangkan sebuah sistem yang digunakan untuk memecahkan masalah, tentunya diperlukan Informasi mengenai sistem yang sudah ada, supaya hasil sistem baru yang

dibuat dapat sepenuhnya menyelesaikan masalah. *Flowchart* adalah suatu penyajian yang sistematis tentang proses dan logika dari kegiatan penganganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. *Flowchart* memecahkan masalah kedalam segmen-segmen kecil dan menolong dalam analisis alternatif-alternatif pengoperasian (Nurdin & Putra, 2018). Adapun hasil yang di dapat setelah melakukan penelitian terhadap sejumlah responden dapat digambarkan melalui *Flowchart* Gambar 2.



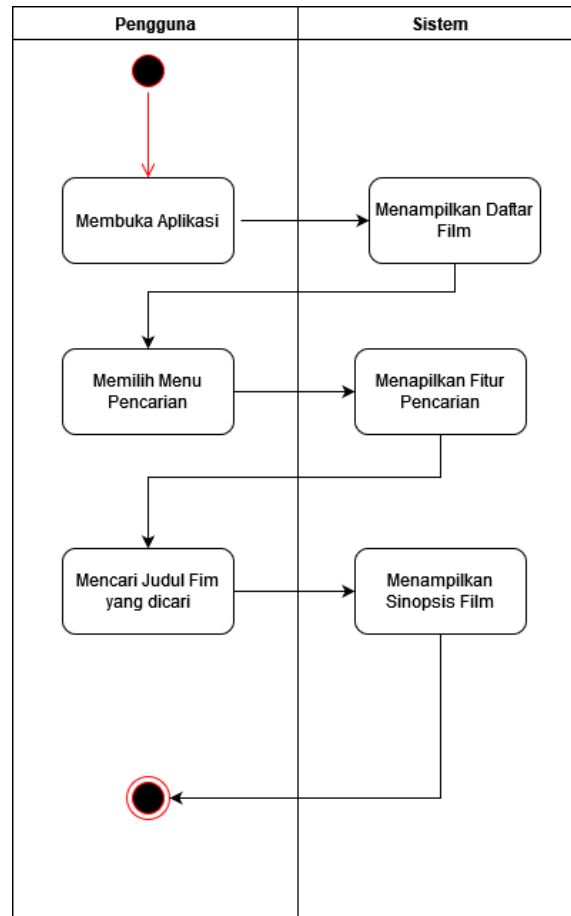
Sumber: Hasil Penelitian (2022)

Gambar 2. *Flowchart* Proses Mencari Sinopsis Film Bioskop

Keterangan Alur *Flowchart* pada proses mencari rekomendasi film untuk ditonton dibioskop adalah sebagai berikut: (a) Mulai, artinya saat pertama kali proses akan dijalankan; (b) Mencari Informasi Film yang akan ditonton; (c) Apabila informasi film tersebut menarik, Maka akan lanjut untuk mencari Jadwal tayang dibioskop, jika tidak sesuai, proses pencarian informasi film yang lain akan dilakukan kembali; (d) Pergi ke Bioskop untuk menonton film yang ingin ditonton.

3.2. Sistem Usulan

Setelah peneliti menemukan permasalahan yang terjadi, maka peneliti mengusulkan beberapa pemecahan masalah yang diterapkan kedalam aplikasi android untuk sistem Informasi film yang menerapkan Rest API menggunakan Retrofit sebagaimana Gambar 3.



Sumber: Hasil Penelitian (2022)

Gambar 3. Activity Diagram Sistem Usulan

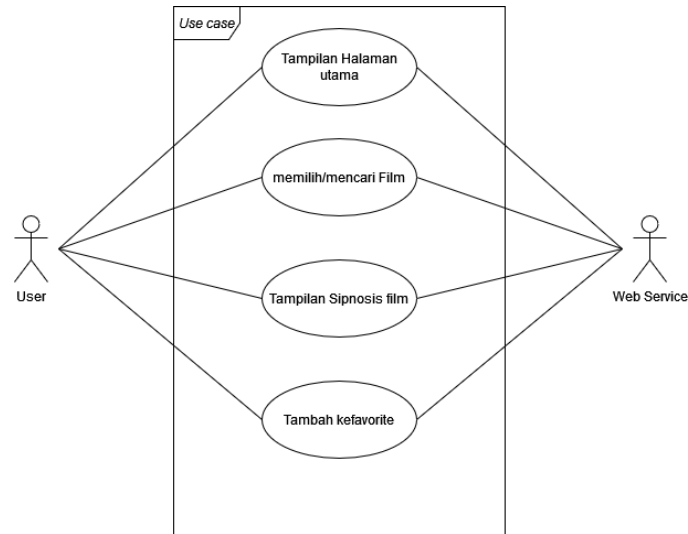
Berikut ini penjelasan dari *Activity Diagram* Sistem Usulan: (a) *User* membuka aplikasi; (b) Sistem akan menampilkan list film; (c) *User* memilih fitur pencarian film; (d) Sistem menampilkan fitur pencarian film; (e) *User* memasukkan judul film yang ingin dicari; (f) Sistem akan menampilkan sinopsis film yang telah dicari.

3.3. Desain Sistem

Dalam perancangan program atau aplikasi ini menggunakan alat bantu standar permodelan sistem berupa UML (*Unified Modeling Language*) dan dengan memanfaatkan API yang ada pada website <https://www.themoviedb.org/> sebagai database Sistem ini. UML yang digunakan antara lain:

a. Use Case Diagram

Use Case diagram Mendeskripsikan interaksi, antara sistem yang akan dikembangkan (Julianto & Setiawan, 2019). Berikut Gambar 4 merupakan gambaran *Use Case Diagram* kegiatan interaksi antar pengguna dengan sistem. Use Case Diagram Aplikasi: (a) *User* membuka aplikasi dan sistem akan menampilkan halaman utama; (b) *User* memilih atau mencari film; (c) Sistem akan menampilkan sinopsis film; (d) *User* menambahkan film ke *favorite*.

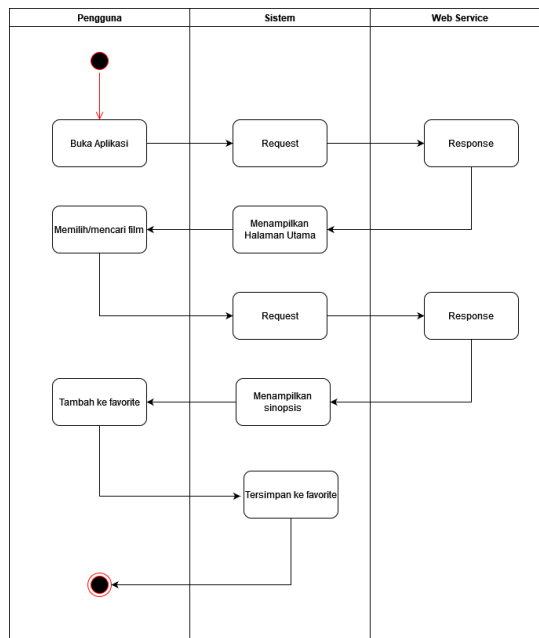


Sumber: Hasil Penelitian (2022)

Gambar 4. Use Case Diagram Aplikasi

b. Activity Diagram

Activity diagram menggambarkan sebuah alur aktivitas dari sebuah sistem (Heriyanto, 2018). Berikut Gambar 5 merupakan Activity Diagram yang memodelkan alur kerja berdasarkan urutan aktivitas sistem. Activity Diagram Alur Kerja Sistem: (a) User membuka aplikasi dan sistem akan menampilkan halaman utama; (b) User memilih atau mencari film; (c) Sistem akan menampilkan sinopsis film; (d) User menambahkan film ke favorite; (e) Sistem akan menyimpan informasi Sinopsis film di fitur favorite.

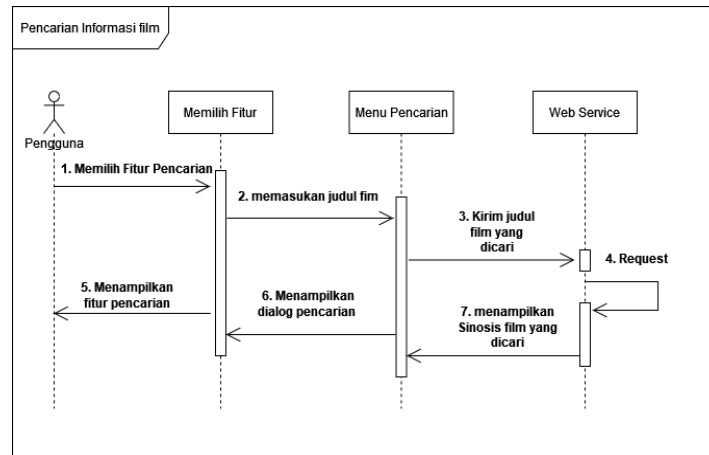


Sumber: Hasil Penelitian (2022)

Gambar 5. Diagram Activity Alur Kerja Sistem

c. *Sequence Diagram*

Sequence diagram menggambarkan perilaku sebuah object pada suatu use case dengan mendeskripsikan life time dan pesan yang dikirim maupun diterima antar object (Julianto & Setiawan, 2019). *Sequence Diagram* Pencarian Informasi film: (a) *User* Memilih fitur pencarian; (b) Sistem menampilkan fitur pencarian; (c) *User* Memasukan judul film di dialog pencarian; (d) Sistem melakukan pengiriman judul film; (e) Sistem akan menerima dan menampilkan sinopsis film.

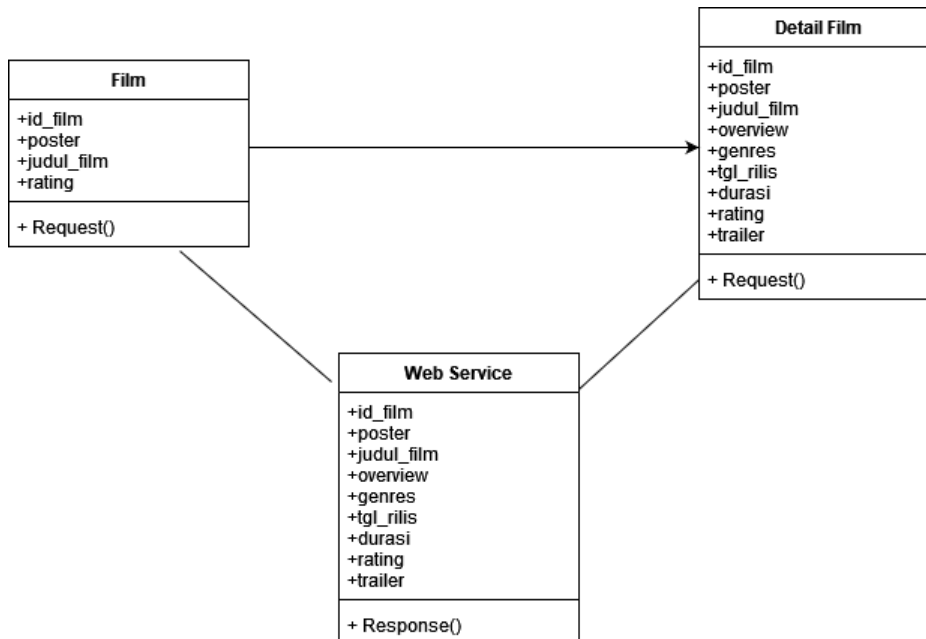


Sumber: Hasil Penelitian (2022)

Gambar 6. *Sequence Diagram* Pencarian Informasi Film

d. *Class Diagram*

Berikut Gambar 7 merupakan *Class Diagram* yang menunjukan atribut atau properti dalam sistem.



Sumber: Hasil Penelitian (2022)

Gambar 7. *Class Diagram* Layanan Sistem

3.4. Implementasi

Proses pengambilan data API pada website <https://www.themoviedb.org/>. berformat JSON.



Sumber: Hasil Penelitian (2022)

Gambar 8. Proses Saat Mengkonsumsi API

Aplikasi akan membuat Request ke Rest API, Request yang dibuat memiliki Body. Body adalah parameter untuk melakukan Request dalam bentuk *Object* POJO Berformat JSON. Kemudian Rest API memberikan respons dalam format JSON yang akan dijadikan *Object* POJO ke dalam aplikasi.

3.5. Implementasi Rest Api Menggunakan Retrofit

Retrofit adalah pustaka klien HTTP untuk aplikasi android dan java. Retrofit digunakan sebagai REST klien. Dengan adanya retrofit tidak perlu membuat metode-metode sendiri untuk terhubung ke layanan web REST. Fungsi utama retrofit adalah untuk mempermudah proses pertukaran data dari format JSON atau XML dan untuk menangani proses transmisi data (permintaan) melalui jaringan (network)(Ramadan & Handaga, 2019).

Pada tahap ini peneliti melakukan pengkodean program menggunakan bahasa pemrograman kotlin, peran retrofit dilakukan pada tahap ini. Agar bisa saling terhubung antara aplikasi android dengan API, maka peneliti membuat sebuah Interface *MovieApiService*. Interface ini bertujuan untuk mendefinisikan metode GET dan Alamat URL menuju lokasi JSON. Berikut pada Gambar 9 merupakan Interface *MovieApiService*:

```
1 import com.robby.findmovies.data.network.model.response.movie.Movie
2 import com.robby.findmovies.data.network.model.response.movie.detail.MovieDetailResponse
3 import com.robby.findmovies.utils.Constant
4 import okhttp3.OkHttpClient
5 import retrofit2.Retrofit
6 import retrofit2.converter.gson.GsonConverterFactory
7 import retrofit2.http.GET
8 import retrofit2.http.Path
9 import retrofit2.http.Query
10 import java.util.concurrent.TimeUnit
11
12 interface MovieApiService {
13     @GET("discover/movie")
14     suspend fun getMovies(
15         @Query("api_key") apiKey: String = Constant.API_KEY
16     ): BaseMovieResponse<List<Movie>>
17
18     @GET("movie/{id}")
19     suspend fun getMovieDetail(
20         @Path("id") id: String,
21         @Query("api_key") apiKey: String = Constant.API_KEY,
22         @Query("append_to_response") response: String = Constant.VIDEOS
23     ): MovieDetailResponse
24
25     @GET("search/movie")
26     suspend fun searchMovies(
27         @Query("query") query: String,
28         @Query("api_key") apiKey: String = Constant.API_KEY
29     ): BaseMovieResponse<List<Movie>>
30 }
```

Sumber: Hasil Penelitian (2022)

Gambar 9. Interface *MovieApiService*

3.6. Implementasi Antarmuka

Tampilan antarmuka dan langkah-langkah dalam penggunaan aplikasi Informasi film dijelaskan sebagai berikut.

Tampilan *Splashscreen*

Tampilan pada Gambar 10 merupakan tampilan awal pada saat pengguna menjalankan aplikasi.

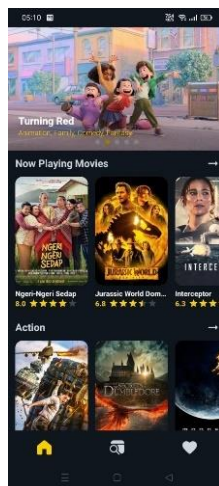


Sumber: Hasil Penelitian (2022)

Gambar 10. Tampilan *Splashscreen*

Tampilan Halaman Utama

Tampilan pada Gambar 11 merupakan tahap awal bagi pengguna untuk Memilih rekomendasi film sesuai Genre yang diinginkan.



Sumber: Hasil Penelitian (2022)

Gambar 11. Tampilan Halaman Utama

Tampilan Genre Film

Tampilan pada Gambar 12 akan menampilkan list film sesuai dengan genre yang telah dipilih oleh pengguna.



Sumber: Hasil Penelitian (2022)

Gambar 12. Tampilan Halaman Utama

Tampilan Sinopsis Film

Tampilan pada Gambar 13 menampilkan sinopsis dan trailer salah satu film yang telah dipilih oleh pengguna.



Sumber: Hasil Penelitian (2022)

Gambar 13. Tampilan Sinopsis Film

Tampilan Search Movies

Tampilan pada Gambar 14, pengguna dapat melakukan pencarian film.



Sumber: Hasil Penelitian (2022)

Gambar 14. Tampilan Search Movies

Tampilan Favorite Movies

Pada tampilan Gambar 15, pengguna dapat melihat film yang telah ditambahkan ke favorite film oleh pengguna sendiri.



Sumber: Hasil Penelitian (2022)

Gambar 15. Tampilan Favorite

3.4 Pengujian *Black box Testing*

Pengujian *Black Box Testing* merupakan pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Agung & Lestari, 2021). Pengujian ini dilakukan menggunakan Black box Testing guna untuk menguji fungsionalitas dari kinerja aplikasi yang telah dirancang. Pada Tabel 1 hasil dari pengujian yang telah dilakukan.

Tabel 1. Pengujian *Black Box Testing*

No.	Menu / Fungsi	Skenario Pengujian	Hasil
1	Halaman Utama	Menampilkan Film berdasarkan <i>genre</i>	Berhasil
2	<i>Genre</i> Film	Menampilkan List Genre Film	Berhasil
3	Detail Film	Menampilkan Sinopsis film	Berhasil
4	<i>Search Movies</i>	Menampilkan film berdasarkan kata yang dicari	Berhasil
5	<i>Favorite movies</i>	Menampilkan film yang telah ditambahkan pengguna ke <i>favorite</i>	Berhasil

Sumber: Hasil Penelitian (2022)

4. Kesimpulan

Berdasarkan uraian yang telah disampaikan peneliti, maka dapat disimpulkan dengan penerapan REST API menggunakan Retrofit pada sistem Informasi film dapat mempermudah proses pengambilan data API berformat JSON, karena metode yang cukup sederhana. Sistem informasi yang telah dibuat dapat membantu untuk mencari informasi sinopsis film yang ingin ditonton.

Daftar Pustaka

- Azizah, N. (2019). *Bekraf: Film Indonesia Tumbuh Pesat*. Www.Republika.Co.Id. <https://www.republika.co.id/berita/pye9ux463/bekraf-film-indonesia-tumbuh-pesat>
- Agung, R. B., & Lestari, T. D. (2021). Aplikasi Monitoring Produktivitas Proses Injection Moulding Menggunakan Algoritma Selection Sort Berbasis Website (Studi Kasus : PT Mada Wikri Tunggal). *Informatics, Science and Technologies Journal*, 11(1).
- Budiman, E. (2019). *Mobile Programming for Student*.
- Heriyanto, Y. (2018). Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT. APM Rent Car. *Jurnal Intra-Tech*, 2(2), 64–77.
- Julianto, S., & Setiawan, S. (2019). Perancangan Sistem Informasi Pemesanan Tiket Bus Pada Po. Handoyo Berbasis Online. *Simatupang, Julianto Sianturi, Setiawan*, 3(2), 11–25. <https://journal.amikmahaputra.ac.id/index.php/JIT/article/view/56/48>
- Nurdin, N., & Putra, S. A. W. E. (2018). Implementasi Sistem Informasi Simpan Pinjam Berbasis Multi User. *JJESIK: Jurnal Elektronik Sistem Informasi dan Komputer*, 4(1), 35–46.
- Putri, A. (2022). *Implementasi Media Penjualan Online Berbasis Web Menggunakan Metode System Development Life Cycle (SDLC)*. <http://openjournal.unpam.ac.id/index.php/JOAIIA/index53>
- Ramadan, S. R., & Handaga, B. (2019). Forlap Scraper menggunakan Aplikasi Android. *Emitor: Jurnal Teknik Elektro*, 20(1), 19–25. <https://doi.org/10.23917/emitor.v20i1.8434>
- Sibarani, N. S., Munawar, G., & Wisnuadhi, B. (2018). Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin. *9th Industrial Research Workshop and National Seminar (IRONS), Juli*, 319–324.
- The Movie Database (TMDB)*. (n.d.). Retrieved July 4, 2022, from <https://www.themoviedb.org/>