

UNIVERSITAS GUNADARMA
FAKULTAS TEKNOLOGI INDUSTRI



PENULISAN ILMIAH

**APLIKASI PENCARIAN INFORMASI FILM BERBASIS WEBSITE
MENGGUNAKAN REACTJS DAN NODEJS**

Nama	: Alvin Faiz Rinaldi
NPM	: 50418609
Jurusan	: Teknik Informatika
Pembimbing	: Dr. Yulia Chalri, S.Kom., MMSI

**Diajukan Guna Melengkapi Sebagian Syarat Dalam Mencapai
Gelar Sarjana Muda**

Depok

2021

PERNYATAAN ORIGINALITAS DAN PUBLIKASI

Saya yang bertanda tangan di bawah ini,

Nama : Alvin Faiz Rinaldi
NPM : 50418609
Judul PI : Aplikasi Pencarian Informasi Film Berbasis Website
Menggunakan ReactJS dan NodeJS
Tanggal Sidang : 9 September 2021
Tanggal Lulus : 9 September 2021

Menyatakan bahwa tulisan ini adalah merupakan hasil karya sendiri dan dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala kutipan dalam bentuk apa pun telah mengikuti kaidah, etika yang berlaku. Mengenai isi dan tulisan merupakan tanggung jawab penulis, bukan Universitas Gunadarma.

Demikian pernyataan ini dibuat dengan sebenarnya dan dengan penuh kesadaran.

Tangerang Selatan, 22 Agustus 2021



(Alvin Faiz Rinaldi)

LEMBAR PENGESAHAN

Judul : Aplikasi Pencarian Informasi Film Berbasis Website
Menggunakan ReactJS dan NodeJS
Nama : Alvin Faiz Rinaldi
NPM : 50418609
Tanggal Sidang : 9 September 2021
Tanggal Lulus : 9 September 2021

Menyetujui,

Pembimbing

Koordinator PI

(Dr. Yulia Chalri, S.Kom., MMSI)

(Dr. Achmad Fahrurrozi S.Si, M.Si)

Ketua Jurusan Teknik Informatika

(Dr. Lintang Yuniar Banowosari, S.Kom., M.Sc.)

ABSTRAKSI

Alvin Faiz Rinaldi 50418609.

Aplikasi Pencarian Informasi Film Berbasis Website Menggunakan ReactJS dan NodeJS.

PI. Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma 2021.

Kata Kunci: Website, Movie, Film, ReactJS, NodeJS.

(x + 49 + Lampiran)

Banyaknya website informasi film di internet membuat para pengguna merasa kebingungan harus mengunjungi website yang mana untuk mendapatkan informasi lengkap dari suatu film yang sedang dicari. Lebih dari itu, informasi yang ditawarkan masih sangat minim, sehingga penggunaan waktu menjadi tidak maksimal. Salah satu solusi untuk masalah ini adalah dengan menyediakan sebuah tempat dimana para penggunanya tidak hanya mendapatkan informasi lengkap mengenai film yang dicari, tetapi juga mempersembahkan rekomendasi film yang baru dirilis saat mengakses halaman utama dari website. Dengan adanya perkembangan teknologi saat ini, aplikasi berhasil dibuat dengan berbasis website. Tidak hanya dapat dinikmati oleh pengguna PC/Laptop, pengguna dengan device smartphone pun dapat merasakan kenyamanan dalam menggunakan aplikasi ini. Proses perancangan aplikasi dibangun berdasarkan Use Case Diagram, Activity Diagram dan Struktur Navigasi. Aplikasi dikembangkan menggunakan IDE (Integrated Development Environment) Visual Studio Code dengan bahasa pemrograman Javascript, library pendukung ReactJS sebagai antarmuka dan NodeJS memungkinkan untuk menjalankan kode Javascript di sisi server. Nantinya akan berhubungan dengan API (Application Programming Interface) yang menyediakan data segala jenis film yang bernama TMDB (The Movie Database) API, ketentuan utama harus terdaftar sebagai anggota untuk mendapatkan API key, dengan begitu akses dapat digunakan. Dalam aplikasi pencarian informasi film terdapat menu Movie untuk melihat daftar genre, now playing, indonesian movies, most popular dan coming soon. Menu Person untuk melihat daftar deretan popular person. Dan menu Search untuk melakukan pencarian berdasarkan judul film, person ataupun tahun rilis dari suatu film. Pengujian aplikasi ini dilakukan dengan menggunakan 2 browser pada platform windows dan 4 perangkat smartphone android dengan spesifikasi yang berbeda. Berdasarkan hasil pengujian, diperoleh bahwa fungsi-fungsi yang ada pada aplikasi berfungsi sesuai dengan yang diharapkan dan aplikasi dapat berjalan pada beberapa smartphone dengan baik.

Daftar Pustaka (2015-2021)

KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadirat Allah SWT, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan Penulisan Ilmiah yang berjudul “Aplikasi Pencarian Informasi Film Berbasis Website Menggunakan ReactJS dan NodeJS”. Penulisan Ilmiah ini disusun guna melengkapi sebagian syarat dalam mencapai gelar setara Sarjana Muda pada Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma. Dalam kesempatan ini, penulis ingin mengucapkan terima kasih dan penghargaan setinggi-tingginya atas segala dukungan, bimbingan dan pengarahan yang diberikan kepada penulis. Ucapan terima kasih penulis sampaikan ke berbagai pihak, terutama kepada :

1. Ibu Prof. Dr. E. S. Margianti, SE, MM., selaku Rektor Universitas Gunadarma.
2. Bapak Prof, Dr.-Ing. Adang Suhendra, S.Si., S.Kom., M.Sc., selaku Dekan Fakultas Teknologi Industri Universitas Gunadarma.
3. Bapak Dr. Achmad Fahruzozi, S.Si., M.Si., selaku Koordinator PI Jurusan Teknik Informatika Universitas Gunadarma.
4. Ibu Dr. Lintang Yuniar Banowosari, S.Kom., M.Sc., selaku Ketua Jurusan Teknik Informatika Universitas Gunadarma.
5. Ibu Dr. Yulia Chalri, S.Kom., MMSI., selaku Dosen Pembimbing yang telah meluangkan waktunya serta memberikan masukan, arahan dan bimbingan kepada penulis dalam menyelesaikan Penulisan Ilmiah ini.
6. Kedua Orang Tua penulis yang selalu mendoakan dengan tulus dan memberikan dukungan secara moril maupun materiil, sehingga penulis dapat menyelesaikan penulisan ini.
7. Semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu-persatu.

Daftar Isi

	Halaman
Cover.....	i
Pernyataan Originalitas dan Publikasi	ii
Lembar Pengesahan	iii
Abstraksi	iv
Kata Pengantar.....	v
Daftar Isi.....	vi
Daftar Tabel.....	viii
Daftar Gambar	ix
1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Batasan Masalah	2
1.3 Tujuan Penelitian.....	3
1.4 Metode Penelitian	3
1.5 Sistematika Penulisan	4
2. TINJAUAN PUSTAKA	5
2.1 Film.....	5
2.2 Visual Studio Code.....	5
2.3 Javascript.....	5
2.4 React JS	6
2.5 Node JS	7
2.6 NPM	8
2.7 Yarn	8
2.8 API.....	8
2.9 Internet	8
2.10 Website.....	9
2.11 HTML	9
2.12 CSS	9

2.13	Hosting	10
2.14	DOM	10
2.15	JSON	11
	2.15.1 Elemen JSON	11
	2.15.2 Kelebihan Menggunakan JSON	11
	2.15.3 Kekurangan Menggunakan JSON.....	12
2.16	UML.....	12
	2.16.1 Use Case Diagram.....	13
	2.16.2 Activity Diagram.....	14
2.17	Struktur Navigasi	15
	2.17.1 Struktur Navigasi Linier	15
	2.17.2 Struktur Navigasi Hirarki	15
	2.17.3 Struktur Navigasi Non Linier	16
	2.17.4 Struktur Navigasi Campuran	16
3.	PEMBAHASAN.....	18
3.1	Gambaran Umum Aplikasi.....	18
3.2	Perancangan.....	19
	3.2.1 Perancangan UML	19
	3.2.2 Struktur Navigasi	20
	3.2.3 Rancangan I/O	21
3.3	Implementasi	27
	3.3.1 Membuat Project	27
	3.3.2 Logika Program.....	32
	3.3.3 Hosting Website	42
3.4	Uji Coba Aplikasi	45
4.	PENUTUP.....	49
4.1	Kesimpulan.....	49
4.2	Saran	49
	Daftar Pustaka	50
	Lampiran Output Program	L-1

Daftar Tabel

	Halaman
Tabel 3.1 Hasil Uji Coba Black Box	45
Tabel 3.2 Hasil Uji Coba Berdasarkan Spesifikasi Smartphone	47

Daftar Gambar

	Halaman
Gambar 2.1 Contoh Struktur Navigasi Linier	15
Gambar 2.2 Contoh Struktur Navigasi Hirarki	16
Gambar 2.3 Contoh Struktur Navigasi Non Linier	16
Gambar 2.4 Contoh Struktur Navigasi Campuran	17
Gambar 3.1 Use Case Diagram Pencarian Informasi Film	19
Gambar 3.2 Activity Diagram Pencarian Informasi Film	20
Gambar 3.3 Struktur Navigasi Pencarian Informasi Film	21
Gambar 3.4 Perancangan Tampilan Halaman Utama	22
Gambar 3.5 Perancangan Tampilan Menu Movie	23
Gambar 3.6 Perancangan Tampilan Menu Person	23
Gambar 3.7 Perancangan Tampilan Menu Search	24
Gambar 3.8 Perancangan Tampilan Detail Person	24
Gambar 3.9 Perancangan Tampilan Detail Search Movie	25
Gambar 3.10 Perancangan Tampilan Detail Search Person	25
Gambar 3.11 Perancangan Tampilan Detail Search Year	26
Gambar 3.12 Perancangan Tampilan Detail Film	27
Gambar 3.13 Halaman Awal Visual Studio Code	27
Gambar 3.14 Pembuatan Folder Baru	28
Gambar 3.15 Pemilihan Folder	28
Gambar 3.16 Halaman Awal setelah pemilihan folder	29
Gambar 3.17 Pembuatan file	29
Gambar 3.18 Halaman TMDB sebelum login	30
Gambar 3.19 Halaman TMDB sesudah login	30
Gambar 3.20 Halaman API	31
Gambar 3.21 File .env	31
Gambar 3.22 Tampilan Halaman Utama	34
Gambar 3.23 Tampilan Menu Movie	35

Gambar 3.24 Tampilan Menu Person	36
Gambar 3.25 Tampilan Menu Search	37
Gambar 3.26 Tampilan Detail Person	38
Gambar 3.27 Tampilan Detail Search Movie	39
Gambar 3.28 Tampilan Detail Search Person	40
Gambar 3.29 Tampilan Detail Search Year	40
Gambar 3.30 Tampilan Detail Film	41
Gambar 3.31 Tampilan Halaman Netlify	42
Gambar 3.32 Tampilan Halaman Netlify setelah login	42
Gambar 3.33 Pembuatan folder build	43
Gambar 3.34 Drag and drop folder build	43
Gambar 3.35 Website sudah online	44
Gambar 3.36 Kustomisasi nama situs	44

BAB I

PENDAHULUAN

1.1 Latar Belakang

Film adalah media komunikasi yang bersifat audio visual. Sebuah film pada umumnya dapat mencakup berbagai pesan, baik itu pesan hiburan maupun pendidikan. Penggunaan teknologi dikatakan dapat mempermudah pengguna untuk mencari dan mengakses berbagai informasi mengenai dunia perfilman. Informasi akan mudah diperoleh dan dapat diakses dimana saja melalui internet dengan dukungan perangkat seperti komputer, laptop serta smartphone. Penyampaian informasi melalui internet menggunakan website dirasa cocok terutama bagi para penggemar film untuk menilai seberapa menarik film yang akan ditonton.

Di sisi lain, pengembangan website mengalami peningkatan yang sangat pesat. Di tahun 2020, StackOverflow merilis survei pengembangan tahunan yang diberi nama StackOverflow Developer Survey 2020. Di mana terdapat bagian Most Popular Technologies, Javascript berada di peringkat pertama sebagai Bahasa Pemrograman terpopuler dengan presentase 67,7% dari 57,378 responden. Sementara itu pada bagian Web Frameworks, library ReactJS menempati posisi kedua terpopuler setelah jQuery dengan presentase 35,9% dari 42,279 responden. Dan pada bagian Other Frameworks, Libraries and Tools, peringkat pertama diduduki oleh NodeJS dengan presentase 51,4% dari 40,314 responden.

Berdasarkan latar belakang di atas, penulis berkeinginan untuk membangun sebuah website pengakses informasi film dengan bahasa pemrograman Javascript yang menggunakan ReactJS sebagai tampilan antarmukanya dan NodeJS sebagai platform pengolah data yang berjalan pada sisi server. Oleh karena itu, penulis memilih topik ini untuk Penulisan Ilmiah dengan judul “Aplikasi Pencarian Informasi Film Berbasis Website Menggunakan ReactJS dan NodeJS”. Website ini dirasa sesuai untuk

membuat masyarakat pencinta film tertarik dan terbantu. Dengan memberikan informasi film yang lengkap dari mulai judul film sampai para pemeran yang bermain dalam film tersebut, penulis harap keingintahuan mereka tentang jalan cerita film dapat terpenuhi baik dari segi penjelasan sinopsis maupun segi visual yang disuguhkan oleh poster film tersebut.

1.2 Batasan Masalah

Dalam mengerjakan suatu penelitian terdapat batasan-batasan yang diperlukan agar tidak menyalahi aturan dari yang telah direncanakan, sehingga tujuannya tercapai seperti selayaknya. Batasan masalah yang ada di dalam penulisan ilmiah ini yaitu aplikasi dapat menampilkan beberapa informasi penting dari film yang berada di posisi atas pada halaman utama web seperti *wallpaper*, judul, *rating*, *review*, *genre*, durasi, tahun rilis, premis dan *trailer* film itu sendiri. Pada posisi tengah dan bawah, halaman utama web menampilkan daftar film yang sedang *trending* saat ini, disertai dengan poster, judul, *rating* dan tahun rilis film sebagai sorotan singkat dilanjut dengan menampilkan daftar pemeran dan sutradara yang sedang *trending* saat ini. Aplikasi ini memiliki menu tambahan selain menu utama, yaitu menu film, menu orang dan menu pencarian. Pada menu film, terdapat 5 tab di dalamnya yaitu *Genre*, *Now Playing*, *Indonesian Movies*, *Most Popular* dan *Coming Soon*. Di mana pada tab *Genre* memiliki label yang dapat dipilih, antara lain *Action*, *Adventure*, *Animation*, *Comedy*, *Crime*, *Documentary*, *Drama*, *Family*, *Fantasy*, *History*, *Horror*, *Music*, *Mystery*, *Romance*, *Sci-Fi*, *TV-Movie*, *Thriller*, *War* dan *Western*. Pada menu orang, aplikasi menampilkan halaman *Popular Person* yang berisi 6 kolom foto dengan baris yang tidak terbatas ke bawah. Pada menu pencarian, aplikasi dapat melakukan pencarian berdasarkan judul film, pemeran serta tahun perilisan film. Saat judul atau poster film dipilih, informasi akan ditampilkan secara rinci yang menambahkan sutradara, anggaran, pendapatan, produksi, pemeran film dan daftar film yang serupa.

1.3 Tujuan Penulisan

Tujuan dari penulisan ilmiah ini adalah membuat aplikasi yang menyediakan sarana informasi yang berkaitan dengan perfilman menggunakan ReactJS dan NodeJS berbasis website yang diharapkan dapat mempermudah para penikmat film untuk mengetahui informasi film secara detail melalui aplikasi.

1.4 Metode Penelitian

Penulis menggunakan SDLC (Software Development Life Cycle) dalam pengembangan sistem bertujuan untuk menyelesaikan permasalahan secara efektif. Model yang dipakai adalah Waterfall Model yang mana pada model ini melibatkan penyelesaian satu tahap secara lengkap sebelum melangkah ke tahap selanjutnya. Berikut merupakan tahapan-tahapan yang harus dilewati :

1. Analisa dan Perencanaan

Pada tahap awal, penulis mengoleksi semua hal yang dibutuhkan website yang akan dibuat.

2. Perancangan

Tahap ini merupakan tahapan dimana rancangan sistem dibuat, seperti penyusunan algoritma program, paket dependensi dari framework yang perlu diinstall, rute penjaluran pada sisi server serta tampilan muka pada sisi client.

3. Implementasi

Setelah aplikasi dirancang, selanjutnya terapkan ke dalam bentuk website. Implementasikan sesuai rancangan yang sudah dibangun.

4. Pengujian

Pada tahap akhir, dilakukan uji coba terhadap perangkat Laptop ASUS TUF FX505DT yang menggunakan sistem operasi Windows 10 64-bit. Tujuan pengujian yaitu untuk memeriksa apakah sistem berjalan sesuai yang direncanakan dari awal atau tidak.

1.5 Sistematika Penulisan

Sistematika penulisan ilmiah ini terdiri dari 4 bab yang diantaranya menerangkan tentang :

Bab I Pendahuluan

Berisi tentang latar belakang masalah, ruang lingkup, tujuan penulisan, metode penulisan dan sistematika penulisan.

Bab II Landasan Teori

Dalam bab ini penulis akan menjelaskan berbagai macam teori pendukung yang memiliki keterkaitan dengan penulisan ini antara lain tentang ReactJS dan NodeJS sebagai media teknologi yang akan digunakan.

Bab III Pembahasan

Dalam bab ini penulis menguraikan tahapan perancangan dan pembuatan aplikasi dengan aspek pendukung pada hardware dan software yang digunakan sampai implementasi serta pengujian terhadap aplikasi yang dibuat.

Bab IV Penutup

Bab ini berisikan kesimpulan dari hasil penulisan yang dibuat disertai dengan saran-saran untuk menjadi bahan evaluasi agar kedepannya dapat dikembangkan dan disempurnakan.

BAB II

TINJAUAN PUSTAKA

2.1 Film

Film adalah media komunikasi yang bersifat audio visual untuk menyampaikan suatu pesan kepada sekelompok orang yang berkumpul di suatu tempat tertentu. (Effendy, 1986: 134). Pesan tersebut menggunakan mekanisme lambang-lambang yang ada pada pikiran manusia berupa tulisan, suara, perkataan, percakapan dan sebagainya. Film juga dianggap sebagai media komunikasi yang ampuh karena dengan gambar dan suara, film mampu bercerita banyak dalam waktu singkat. Film memiliki satu sasaran, yaitu menarik perhatian orang terhadap muatan-muatan masalah yang dikandung. Selain itu, film dapat dirancang untuk melayani keperluan publik terbatas maupun publik yang seluas-luasnya.

2.2 VS Code (Visual Studio Code)

Visual Studio Code adalah source code editor gratis yang dibuat oleh Microsoft untuk Windows, Linux, dan macOS. VS Code memiliki fitur debugging, syntax highlighting, intelligent code complete, snippet, code refactoring, dan embedded Git. Dengan kemampuannya yang baik, tidak heran banyak programmer yang memilih untuk menggunakan VS Code. Salah satu fitur keunggulan VS Code adalah extension yang disediakannya. Extension merupakan kumpulan plugin yang dapat diinstal ke teks editor untuk membuatnya lebih powerful.

2.3 Javascript

JavaScript merupakan bahasa skrip populer yang dipakai untuk menciptakan halaman web yang dapat berinteraksi dengan pengguna dan dapat merespon event yang terjadi pada halaman. JavaScript merupakan perekat yang menyatukan halaman-halaman web. Akan sangat susah menjumpai halaman web komersial yang tidak memuat kode JavaScript.

JavaScript, awalnya dikenal sebagai LiveScript, dikembangkan oleh Brendan Eich di Netscape pada tahun 1995 yang menjadi bagian terintegrasi di dalam Netscape Navigator 2.0. JavaScript merupakan bahasa skrip yang menghidupkan halaman-halaman HTML. JavaScript dapat dijalankan pada hampir semua platform. JavaScript merupakan bahasa sisi-klien yang didesain pada browser komputer, bukan pada server. Ia dibangun secara langsung ke dalam browser, Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera, hampir semua browser. Dalam sintaksis, JavaScript mirip dengan C, Perl, dan Java.

Karena JavaScript terkait dengan browser, ia sangat terintegrasi dengan HTML. Ketika browser memuat sebuah halaman, server akan mengirim konten utuh dari dokumen, termasuk HTML dan statemen statemen JavaScript. Konten HTML kemudian dibaca dan diinterpretasi baris-demi-baris sampai tag pembuka JavaScript dibaca, pada saat itu interpreter JavaScript mengambil alih. Ketika tag penutup JavaScript diraih, pemrosesan HTML berlanjut. Meskipun JavaScript secara umum digunakan sebagai bahasa skrip sisi-klien, ia dapat pula digunakan (pada konteks lain) selain pada browser web. Netscape menciptakan JavaScript sisi-server yang bisa diprogram sebagai bahasa CGI, seperti Python atau Perl.

2.4 React JS

React JS adalah sebuah library bahasa pemrograman JavaScript yang dibuat oleh Facebook. ReactJS merupakan library yang bersifat Open-Source. React JS digunakan untuk membangun sebuah User Interface pada Website, Mobile, dan juga aplikasi Desktop. Pada awalnya, React dicetuskan oleh seorang karyawan Facebook bernama Jordan Walke. Cetak biru yang dihasilkan diberi nama “FaxJS”, inspirasinya datang dari XHP. Sebelumnya, Facebook telah sadar jika web dan aplikasi mereka sudah berkembang cukup kompleks. Penggunaan cara lama dengan JQuery untuk memperbarui kolom dan daftar chat, beranda, dan lain-lainnya secara real-time dirasa memakan banyak biaya.

Diciptakannya React berkontribusi dalam menghasilkan solusi untuk mengatasi manipulasi DOM (Document Object Model) yang membutuhkan sumber besar. React menyajikan Virtual DOM yang diklaim cukup cepat. Pada 2011,

Facebook mulai menggunakan React pada halaman berita. Tahun selanjutnya, React digunakan pada Instagram lalu pada 2013, baru dirilis untuk umum atau bisa diakses secara open source. Para web developer atau software consultant saat ini banyak yang menggunakan React dengan alasan dapat membuat dan mendesain tampilan simpel bagi tiap-tiap tingkatan di dalam aplikasi yang sedang dikembangkan. Selain Facebook, aplikasi besar lainnya, seperti Netflix, Dropbox, Ebay, Airbnb, American Express, WhatsApp, dan lain-lainnya juga mengandalkan React untuk membuat web app.

Keuntungan dari React ini dibangun dengan konsep Reusable Components, dimana kita dapat membuat komponen kecil (bagian kecil dari User Interface, seperti tombol, input, dan sebagainya) yang dapat digunakan secara berulang sehingga memberikan performa yang bagus pada aplikasi yang dibuat, serta dapat membuat proses development menjadi lebih cepat. React juga mengusung konsep Learn Once, Write Anywhere. Seperti yang telah disinggung di atas, tujuan penting penggunaan React adalah agar bisa lebih cepat, sederhana dan tentunya terukur.

2.5 Node JS

Node JS adalah suatu platform atau perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis website dan ditulis didalam bahasa pemrograman Javascript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang berjalan di sisi client atau browser saja, maka Node JS ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server.

Node JS juga dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Node JS memiliki library server HTTP sendiri sehingga memungkinkan untuk menjalankan server web tanpa menggunakan program server web seperti Apache atau Nginx. Untuk mengeksekusi Javascript sebagai bahasa server diperlukan engine yang cepat dan mempunyai performa yang bagus. Engine Javascript dari Google bernama V8 yang dipakai oleh Node JS yang juga merupakan engine yang dipakai oleh browser Google Chrome.

2.6 NPM (Node Package Manager)

NPM adalah aplikasi dan repository untuk mengembangkan dan membagikan kode JavaScript yang merupakan bagian dari Node JS yang mencakup semua yang dibutuhkan untuk menjalankan program yang ditulis dalam JavaScript. JavaScript digunakan menjalankan script pada server untuk menerjemahkan konten sebelum dikirim ke browser website.

2.7 Yarn

Yarn dikembangkan oleh Facebook dalam upaya untuk menyelesaikan beberapa kekurangan NPM. Secara teknis, Yarn bukan pengganti NPM karena bergantung pada modul dari registry NPM. Yarn masih mengandalkan struktur NPM yang sama. Registry itu sendiri tidak berubah, tetapi metode instalasi berbeda. Karena Yarn memberi akses ke paket yang sama dengan NPM, pindah dari NPM ke Yarn tidak harus untuk melakukan perubahan apa pun pada alur kerja development.

2.8 API (Application Programming Interface)

API adalah singkatan dari Application Programming Interface, berfungsi sebagai perantara yang menjembatani dua aplikasi untuk berkomunikasi satu sama lain. API dapat dianalogikan sebagai penerjemah antara dua orang yang tidak berbicara dengan bahasa yang sama, tetapi tetap dapat berkomunikasi untuk memenuhi kebutuhan satu sama lain. Bagi mobile developer atau web developer, API adalah salah satu aspek yang paling penting untuk membantu pekerjaan mereka, karena dapat menyederhanakan program yang sedang mereka buat di dalam sebuah website atau aplikasi.

2.9 Internet

Internet adalah suatu jaringan komunikasi yang memiliki fungsi untuk menghubungkan antara satu media elektronik dengan media elektronik yang lain dengan cepat dan tepat. Jaringan komunikasi tersebut, akan menyampaikan

beberapa informasi yang dikirim melalui transmisi sinyal dengan frekuensi yang telah disesuaikan. Untuk standar global dalam penggunaan jaringan internet sendiri menggunakan TCP / IP (Transmission Control Protocol / Internet Protocol).

Istilah TCP / IP merupakan bentuk protokol pertukaran paket yang digunakan oleh berbagai pengguna global / dunia. Kemudian, proses untuk menghubungkan antara rangkaian internet disebut dengan “internetworking”. Menurut salah satu ahli dalam bidang IT, Onno W. Purbo (2005) menjelaskan bahwa pengertian internet adalah suatu media yang digunakan untuk mengefisienkan proses komunikasi menggunakan aplikasi seperti website, email, atau voip.

2.10 Website

Menurut Bekti (2015:35), Website merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

2.11 HTML (Hypertext Markup Language)

HTML adalah singkatan dari Hypertext Markup Language yaitu bahasa markah standar yang digunakan untuk membuat sebuah halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah web internet (Browser). HTML dapat juga digunakan sebagai link-link antara file-file dalam situs atau dalam komputer dengan menggunakan localhost, atau link yang menghubungkan antar situs dalam dunia internet.

2.12 CSS (Cascading Style Sheet)

CSS adalah singkatan dari Cascading Style Sheet yaitu salah satu bahasa desain web yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda (markup language). Biasanya CSS digunakan untuk

mendesain sebuah halaman HTML dan XHTML. CSS dibuat untuk memisahkan konten utama dengan tampilan dokumen yang meliputi layout, warna dan font.

2.13 Hosting

Hosting adalah rumah dari website. Artinya, hosting akan menjadi tempat untuk menaruh berbagai data yang ada di website. Sebagaimana diketahui, dalam website ada berbagai jenis data mulai dari gambar, tulisan, video, plugin, script dan lainnya. Fungsi hosting adalah sebagai media untuk menyimpan aneka data tersebut. Alhasil, ketika website yang kita kelola dibuka oleh orang lain, maka informasi website akan terpampang dengan jelas tanpa adanya masalah.

Fungsi lain yang perlu diketahui yaitu sebagai tempat penyimpanan data yang akan memudahkan pengguna untuk mengakses data. Dalam hal ini, ada beberapa kapasitas dari hosting mulai dari hitungan MB hingga GB, tergantung harga dan spesifikasinya. Kemudian dapat membuat website terhubung dengan internet. Jika tidak ada masalah pada hosting, akses website akan cenderung cepat dan lancar.

2.14 DOM (Document Object Model)

DOM (Document Object Model) adalah interface yang memungkinkan developer untuk memanipulasi konten, struktur, dan style situs web. Yang mana di dalam object ini bukan hanya ada struktur, namun juga interaksi (behavior) beserta event-nya. Setiap isi dari HTML yang sudah dimuat dalam peramban web (web browser), bisa dimodifikasi melalui DOM.

Pada model DOM ini, setiap elemen html dipandang sebagai sebuah object. Setiap object bisa terdiri dari object-object lain, sama halnya dengan dokumen html yang terdiri dari elemen root (elemen <html>), elemen root terdiri dari elemen <head> dan elemen <body>, elemen <body> boleh jadi terdiri dari elemen <a>, <h1>, <p>, dst. Elemen-elemen pada dokumen html membentuk sebuah object document yang merupakan object dari dokumen html itu sendiri.

2.15 JSON (Javascript Object Notation)

JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer. Format JSON dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data. Hingga saat ini, penggunaan pertukaran JSON sudah sangat populer dalam dunia web development dan sudah merupakan hal yang biasa, bahkan menjadi standar dalam pertukaran data dalam web development.

2.15.1 Elemen JSON

Elemen didalam JSON atau Javascript Object Notation tersusun dari dua struktur, yaitu kumpulan pasangan nilai/nama dan daftar nilai terurutkan. Kedua elemen tersebut akan dijelaskan lebih lengkap berikut ini.

- a. **Kumpulan pasangan nama/nilai**, pada bahasa pemrograman lain, pasangan nama / nilai ini sering disebut sebagai object (objek), record (rekaman), struct (struktur), dictionary (kamus), hash table (tabel hash), keyed list (daftar kunci) atau associative array.
- b. **Daftar nilai terurutkan (an ordered list of value)**, pada bahasa pemrograman lain, ordered list of value ini biasa disebut juga sebagai array (larik), vector (vektor), list (daftar), atau sequence (urutan).

2.15.2 Kelebihan Menggunakan JSON

- a. Kecepatan dalam penguraian yang merupakan proses pengenalan bagian terkecil dari suatu dokumen JSON/XML sehingga membuat kecepatan penguraian pada JSON melampaui XML.
- b. Kemampuan untuk menyimpan data dalam bentuk array yang memungkinkan transfer menjadi lebih mudah.

- c. Berdasar pada JavaScript membuat JSON memiliki sintaks yang kecil dan ringan sehingga lebih responsif terhadap request.
- d. Keunggulan dalam penanganan API untuk aplikasi web ataupun desktop.
- e. Adanya dukungan untuk bahasa pemrograman lain seperti PostgreSQL dan JavaScript.

2.15.3 Kekurangan Menggunakan JSON

- a. Berbeda dengan XML yang memiliki sintaks yang menyerupai HTML, sintaks JSON distruktur dan diformat dengan gaya penulisan yang sulit dipahami.
- b. Bahasa JavaScript rentan terhadap hacking terutama pada website-website yang belum terpercaya.
- c. Tidak adanya penanganan error pada saat request.

2.16 UML (Unified Modelling Language)

Bahasa pemodelan perangkat lunak atau UML, diperkenalkan pertama kali pada tahun 1997. Saat ini telah berkembang menjadi sebuah bahasa pemodelan yang baku dalam pengembangan perangkat lunak yang menggunakan pendekatan berorientasi objek. Intensitas penggunaan UML yang tinggi didukung dengan konsep pemodel yang dirumuskan dalam setiap rilis spesifikasi UML. Rilis spesifikasinya dikeluarkan oleh Object Management Group (OMG). Sampai pada tahun ini, OMG telah merilis 11 versi spesifikasi UML, yang terakhir adalah versi 2.5.1 yang termasuk dalam revisi UML 2.0. Pengembangan alat bantu untuk pemodelan dengan UML berkembang cukup pesat. Bahkan sebagiannya tergolong sebagai free software. Dampaknya adalah ketersediaan banyak pilihan bagi pengembang perangkat lunak untuk menggunakannya. Contohnya Star UML, Argo UML, UML Designer.

UML menyediakan banyak diagram yang diperlukan untuk menjelaskan sistem yang sedang dikembangkan, baik dari aspek statis maupun dinamisnya. Salah satu diagram penting yang digunakan untuk mengilustrasikan kebutuhan dari sistem adalah use case diagram. Use case diagram menjelaskan secara visual konteks dari interaksi antara aktor dengan sistem. Setiap use case menyatakan

spesifikasi perilaku dari sistem yang sedang dijelaskan yang memang dibutuhkan oleh aktor untuk memenuhi tujuannya. Namun demikian, penjelasan detail dari interaksi yang terjadi antara aktor dan sistem, berkaitan dengan sebuah use case tertentu, harus dijelaskan secara deskriptif dalam sebuah use case scenario. Oleh karena itu, use case scenario dan use case diagram, yang dibutuhkan dalam pemodelan UC dari sebuah sistem, harus mampu menjelaskan fungsionalitas sistem secara lengkap dan valid.

Kelebihan UML dibandingkan dengan bahasa pemodelan yang lain, antara lain menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti, menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti, mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu, menyediakan basis formal untuk bahasa pemodelan, memadukan praktek-praktek terbaik di industri perangkat lunak menjadi terminologi dan notasi yang diterima luas, menyediakan kemampuan merepresentasikan semua konsep yang relevan untuk sistem perangkat lunak dan menyediakan fleksibilitas yang diperlukan bagi konsep-konsep perangkat lunak yang baru.

2.16.1 Use Case Diagram

Use Case Diagram adalah proses penggambaran yang dilakukan untuk menunjukkan hubungan antara pengguna dengan sistem yang dirancang. Hasil representasi dari skema tersebut dibuat secara sederhana dan bertujuan untuk memudahkan *user* dalam membaca informasi yang diberikan.

Use Case Diagram terdiri dari :

- 1. Actor**

Actor merupakan setiap hal di luar sistem yang menggunakan komponen system untuk melakukan sesuatu. Actor dapat berupa manusia, perangkat, atau bahkan sistem tersebut yang menjadi peranan dalam keberhasilan sebuah operasi dalam sistem yang dibangun.

- 2. System**

Komponen ini menyatakan batasan dari sistem di dalam relasi yang dilakukan dengan actor yang menggunakannya (di luar sistem). Serta, fitur harus disediakan di dalam sistem tersebut.

3. Use Case

Komponen yang ketiga adalah use case, yang merupakan gambaran umum dari fungsional sebuah sistem. Dengan begitu, pengguna dan konsumen dapat mengetahui setiap fungsi yang dibangun dalam sistem tersebut.

Terdapat juga contoh dari relasi yang digunakan untuk membuat use case diagram, berikut adalah penjelasannya.

1. Association

Association adalah teknik yang digunakan untuk mengidentifikasi sebuah interaksi antara komponen actor dengan use case tertentu. Hal tersebut digambarkan dengan garis penghubung antara aktor dengan use case.

2. Dependency

Dependency relation terbagi menjadi dua jenis, yaitu include dan exclude. Include berfungsi untuk mengidentifikasi hubungan atau relasi antara dua use case, yang mana use case yang satu akan memanggil yang lainnya. Jenis yang kedua atau exclude merupakan jenis yang apabila dilakukan pemanggilan maka memerlukan suatu kondisi tertentu dan akan terjadi dependensi.

3. Generalization

Generalisasi merupakan hubungan antara dua use case atau dua aktor, dimana salah satu meng-inherit dan menambahkan atau melakukan override sifat dari komponen yang lainnya.

2.16.2 Activity Diagram

Activity diagram ialah sesuatu yang menjelaskan tentang alir kegiatan dalam program yang sedang dirancang, bagaimana proses alir berawal, keputusan yang mungkin terjadi, dan bagaimana sistem akan berakhir. Activity diagram juga

dapat menjelaskan metode paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram adalah state diagram khusus, yang mana state ini berfungsi sebagai action dan sebagian besar transisi ditrigger oleh akhir state sebelumnya (internal processing).

2.17 Struktur Navigasi

Struktur navigasi adalah alur yang digunakan dalam aplikasi yang dibuat. Sebelum menyusun aplikasi multimedia kedalam sebuah software, kita harus menentukan terlebih dahulu alur apa yang akan digunakan dalam aplikasi yang dibuat. Bentuk dasar dari struktur navigasi yang biasa digunakan dalam proses pembuatan aplikasi multimedia ada empat macam, yaitu struktur navigasi linier, hirarki, non linier dan campuran.

2.17.1 Struktur Navigasi Linier

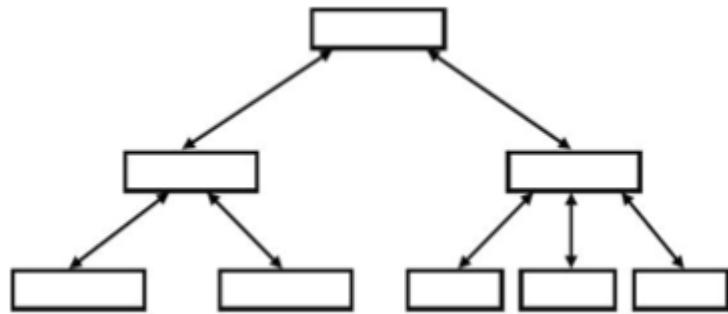
Struktur navigasi linier merupakan struktur yang mempunyai satu rangkaian cerita berurutan. Struktur ini menampilkan satu demi satu tampilan layer secara berurutan menurut aturannya. Struktur navigasi linier dapat dilihat pada **Gambar 2.1**.



Gambar 2.1 Contoh Struktur Navigasi Linier

2.17.2 Struktur Navigasi Hirarki

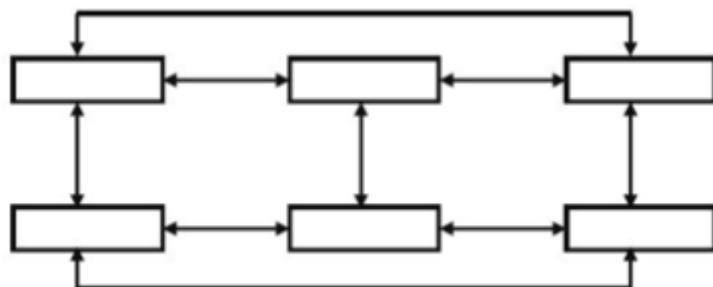
Struktur navigasi hirarki sering disebut struktur navigasi bercabang, yaitu merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data atau gambar pada layer dengan kriteria tertentu. Tampilan pada menu utama disebut master page (halaman utama satu), halaman tersebut mempunyai halaman percabangan yang disebut slave page (halaman pendukung) dan jika dipilih akan menjadi halaman kedua, begitu seterusnya. Struktur navigasi linier dapat dilihat pada **Gambar 2.2**.



Gambar 2.2 Contoh Struktur Navigasi Hirarki

2.17.3 Struktur Navigasi Non Linier

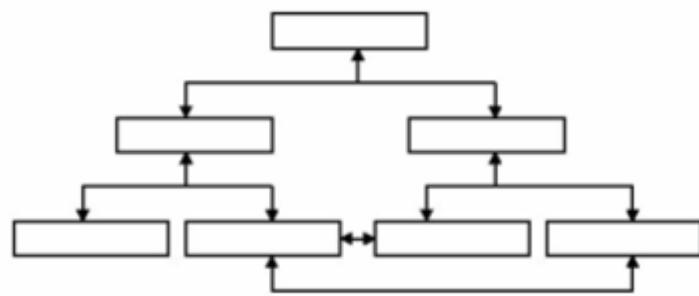
Struktur navigasi non linier (tidak terurut) merupakan pengembangan dari struktur navigasi linier, hanya saja pada struktur ini diperkenankan untuk membuat percabangan. Percabangan pada struktur non linier berbeda dengan percabangan pada struktur hirarki, pada struktur ini kedudukan semua page sama, sehingga tidak dikenal adanya master atau slave page. Struktur navigasi non linier dapat dilihat pada **Gambar 2.3**.



Gambar 2.3 Contoh Struktur Navigasi Non Linier

2.17.4 Struktur Navigasi Campuran

Struktur navigasi campuran (composite) merupakan gabungan dari struktur sebelumnya dan disebut juga struktur navigasi bebas, maksudnya adalah jika suatu tampilan membutuhkan percabangan maka dibuat percabangan. Struktur ini paling banyak digunakan dalam pembuatan aplikasi multimedia. Struktur navigasi campuran dapat dilihat pada **Gambar 2.4**.



Gambar 2.4 Contoh Struktur Navigasi Campuran

BAB III

PEMBAHASAN

3.1 Gambaran Umum Aplikasi

Aplikasi yang dibuat ini untuk mempermudah pengguna dalam mendapatkan informasi film yang berisikan :

1. Rekomendasi film yang berada pada jumbotron atau banner di halaman utama.
2. Tersedia beberapa *genre* film yang dapat dipilih.
3. Terdapat fitur pilihan lainnya seperti *Now Playing*, *Indonesian Movies*, *Most Popular* dan *Coming Soon*.
4. Melakukan pencarian aktor dalam film yang dipilih yang dapat menampilkan berperan sebagai siapa, biografi singkat serta tempat dan tanggal lahir dari aktor tersebut.
5. Pencarian secara spesifik menggunakan input tahun.
6. Data mengenai film diambil dari The Movie Database (TMDb) API dengan bentuk file JSON.
7. Pemberian logika dilakukan dengan bahasa pemrograman Javascript

Halaman utama akan menampilkan beberapa komponen, antara lain:

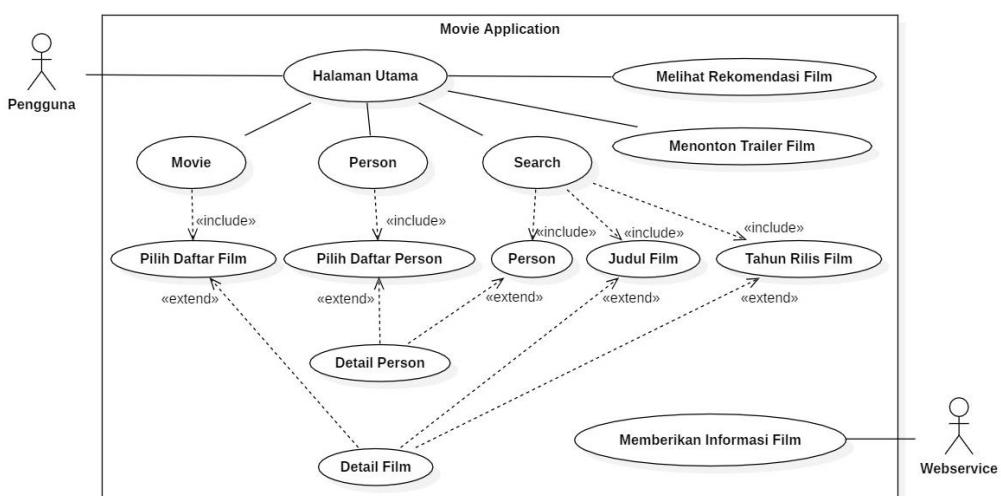
1. Judul, sebagai identitas aplikasi dengan pemberian gambar favicon yang berada pada tab browser.
2. Menu, sebagai opsi pilihan dari menu *movie*, *person* dan *search* sebagai peletakan pencarian informasi film.
3. *Jumbotron* atau *banner*, sebagai rekomendasi film yang berisi judul film, *rating*, jumlah *review*, *genre*, durasi, tahun, premis singkat serta tombol untuk menonton *trailer* dari film tersebut.
4. *Trending Movies*, sebagai film yang sedang trending saat ini.
5. *Trending Person*, sebagai aktor yang sedang trending saat ini.

3.2 Perancangan

Untuk membuat aplikasi pencarian informasi film ini sebelumnya penulis perlu membuat struktur hirarki web. Karena struktur hirarki ini merupakan suatu gambaran secara umum dalam perancangan dan pembuatan web ini. Maka penulis mengurutkan beberapa langkah dalam perancangan dan pembuatan aplikasi pencarian informasi film, yaitu perancangan UML, struktur navigasi, dan rancangan I/O.

3.2.1 Perancangan UML

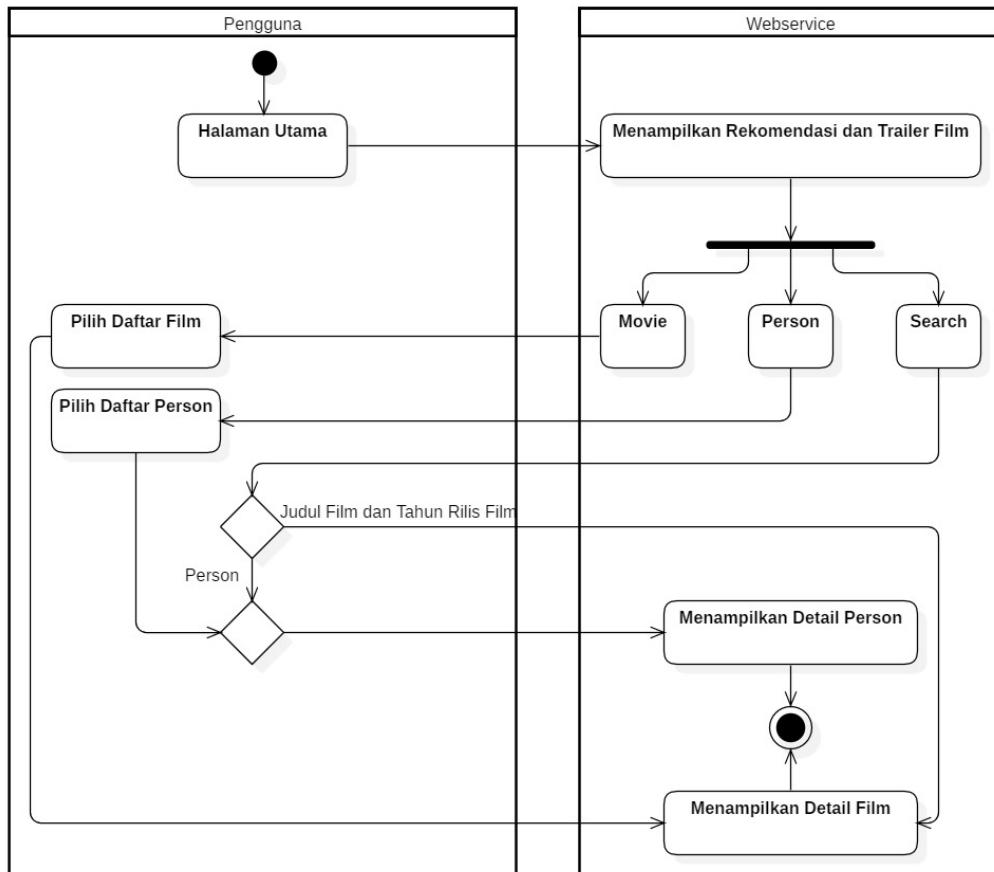
Pada perancangan UML menggunakan Use Case Diagram, pengguna diperuntukkan mengakses website. Pada halaman utama menampilkan rekomendasi film dan trailer film. Menu *Movie*, *Person* dan *Search* memiliki *include* yang harus dijalankan. Terdapat *extend* dari beberapa *use case* sebagai tambahan menu yang tidak terlihat pada menu utama seperti *Detail Person* dan *Detail Film*. Webservice menerima *request* dan menyediakan informasi kepada pengguna. **Gambar 3.1** menyajikan Use Case Diagram dari aplikasi yang dibuat.



Gambar 3.1 Use Case Diagram Pencarian Informasi Film

Pencarian informasi film berdasarkan Use Case Diagram menunjukkan tampilan pada halaman utama. Selanjutnya, akan ditambahkan dengan

menggunakan Activity Diagram. **Gambar 3.2** menyajikan Activity Diagram dari aplikasi yang dibuat.



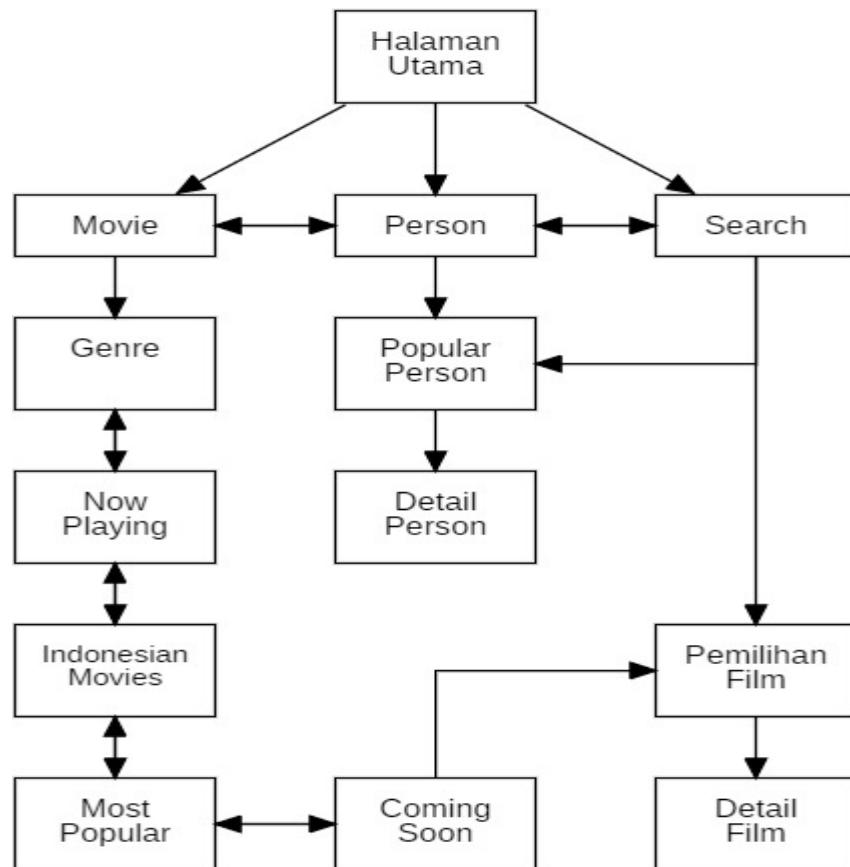
Gambar 3.2 Activity Diagram Pencarian Informasi Film

Berdasarkan Activity Diagram, pengguna mengakses halaman utama. Webservice kemudian menampilkan rekomendasi film dan trailer film dengan 3 menu utama lainnya. Pengguna selanjutnya melakukan pemilihan dengan menu yang tersedia terutama pada menu *Movie* dan *Person*. Pencarian spesifik dapat menggunakan Menu *Search*. *Detail Person* dan *Detail Film* akan ditampilkan oleh Webservice jika pengguna sudah memilih daftar film atau *person* yang tersedia.

3.2.2 Struktur Navigasi

Sebuah struktur navigasi dibutuhkan dalam pembuatan website karena akan mempermudah dalam pembuatan website tersebut, serta langkah-langkah yang

dilakukan lebih terstruktur sehingga dapat menggapai keinginan masing-masing pengguna. Jenis struktur navigasi yang penulis gunakan pada aplikasi ini adalah struktur navigasi campuran, gabungan antara linier dan hirarki. **Gambar 3.3** menyajikan struktur navigasi campuran dari aplikasi yang dibuat.



Gambar 3.3 Struktur Navigasi Pencarian Informasi Film

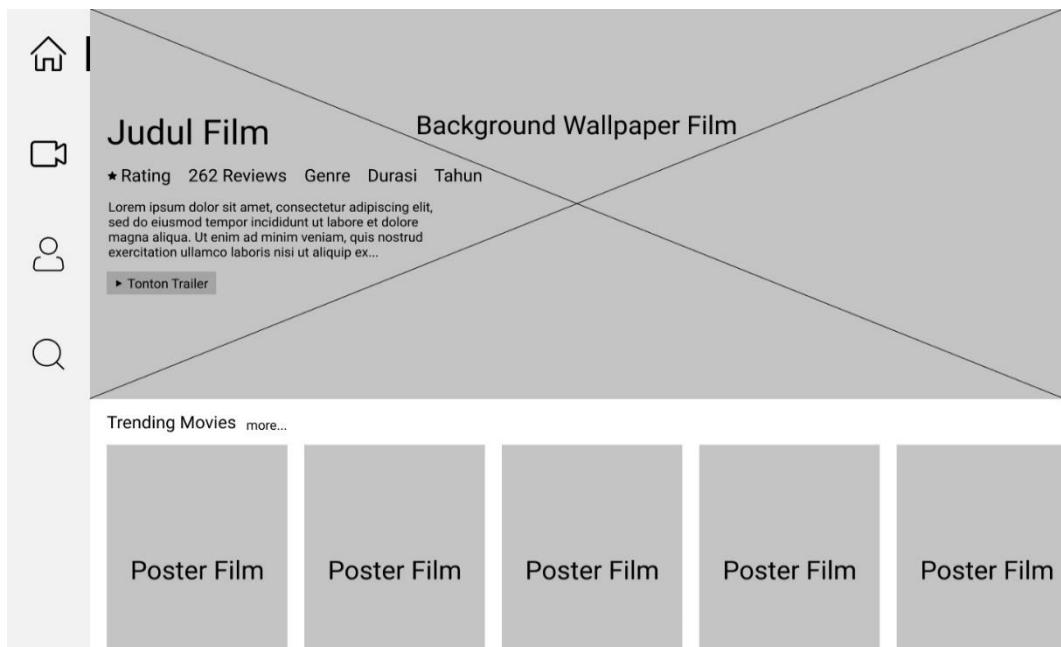
Sebelum membuat website, penulis harus membuat perancangan untuk tampilan yang akan dibuat berdasarkan dari struktur navigasi sebelumnya agar memudahkan dalam pembuatan tampilan website seperti yang diharapkan.

3.2.3 Rancangan I/O

Halaman utama atau menu *home* merupakan halaman yang terdiri dari 2 bagian yaitu *sidebar* menu dan *headline*. Pada *sidebar* menu terdapat 4 buah *button* dengan *icon* yang berfungsi sebagai penghubung keempat menu yaitu *home*, *movie*, *person* dan *search*. Pada *headline* terdapat *jumbotron* atau *banner* di bagian atas

yang berisi judul film, *rating*, jumlah *review*, *genre*, durasi, tahun, premis singkat, *button* untuk menonton *trailer* serta *background wallpaper* film yang ditampilkan secara *landscape* atau melebar ke samping. Di bagian bawah terdapat *trending movies* dimana menampilkan beberapa pilihan film yang sedang trending berikut dengan poster film yang dapat dipilih dan digeser secara horizontal. Opsi *button more* di samping teks *trending movies* dapat dipilih apabila judul film yang ingin dicari tidak berada pada menu *trending movies*. *Button more* menampilkan daftar film yang lebih luas, dengan begitu halaman akan otomatis beralih ke menu *movie*.

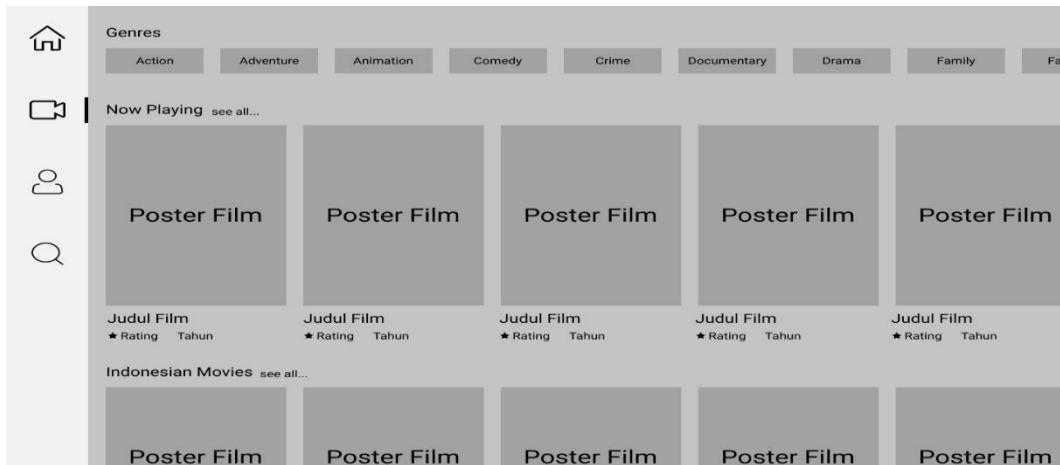
Gambar 3.4 menyajikan rancangan tampilan untuk Halaman Utama.



Gambar 3.4 Perancangan Tampilan Halaman Utama

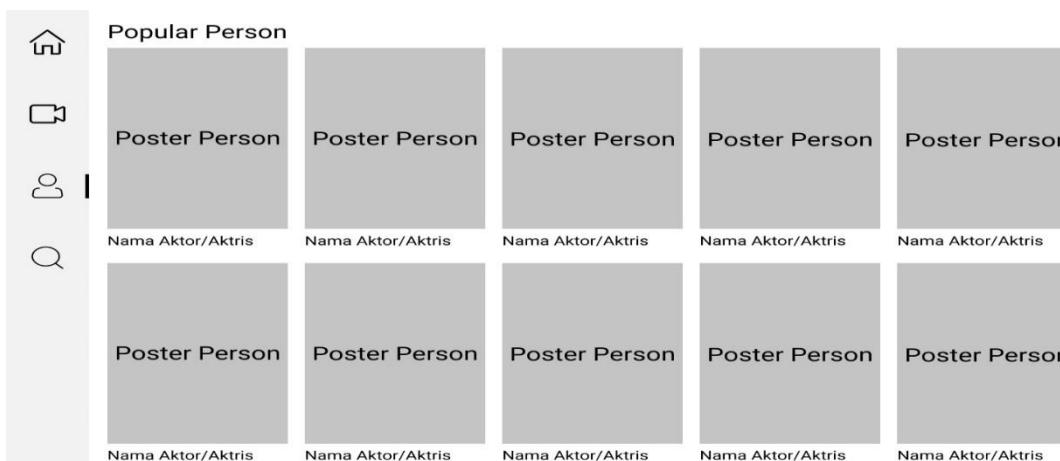
Di menu *movie* terdapat *sidebar* menu dan *header*. Pada *sidebar* menu disini memiliki fungsi yang sama seperti pada menu di halaman sebelumnya, namun posisi garis akan berpindah ke depan *icon button* dari menu *movie*. Pada bagian *header* berfungsi untuk menampilkan daftar *genre* film, begitu juga dengan *now playing* dan *indonesian movies* berisikan judul film, *rating*, tahun dan poster film.

Gambar 3.5 menyajikan rancangan tampilan untuk menu *Movie*.



Gambar 3.5 Perancangan Tampilan Menu *Movie*

Menu *person* menampilkan aktor atau aktris populer yang memiliki peran di film yang dibintanginya. **Gambar 3.6** menyajikan rancangan tampilan untuk menu *Person*.



Gambar 3.6 Perancangan Tampilan Menu *Person*

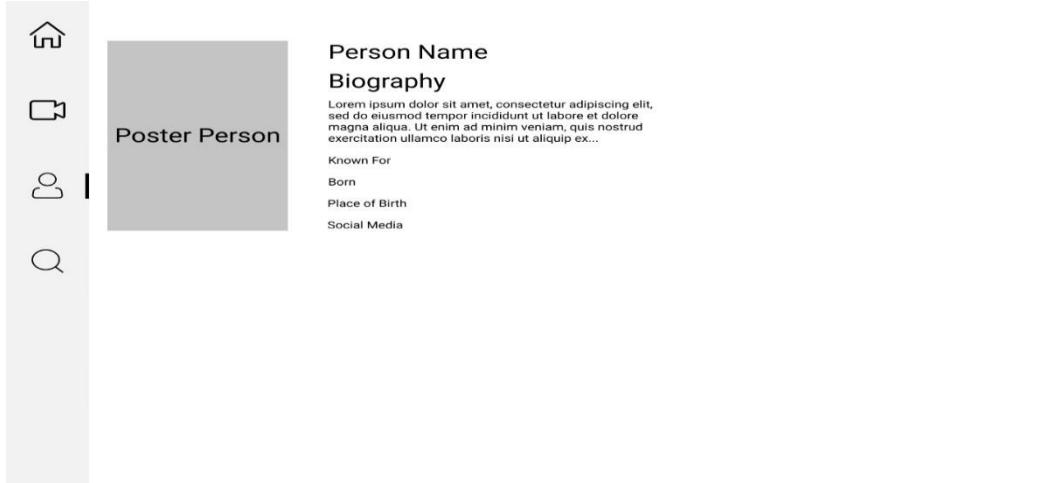
Pada tampilan ini ditujukan untuk melakukan pencarian spesifik berdasarkan judul film, pemain film dan tahun apabila diketahui. **Gambar 3.7** menyajikan rancangan tampilan untuk menu *Search*.



Gambar 3.7 Perancangan Tampilan Menu *Search*

Saat melakukan pemilihan aktor pada menu *person* melalui klik poster *person* ataupun nama aktor, halaman website akan mengarahkan ke tampilan *detail person*. Tampilan *detail person* menyediakan informasi biografi, dikenal sebagai apa, tanggal lahir, tempat lahir dan social media apabila aktor tersebut memiliki.

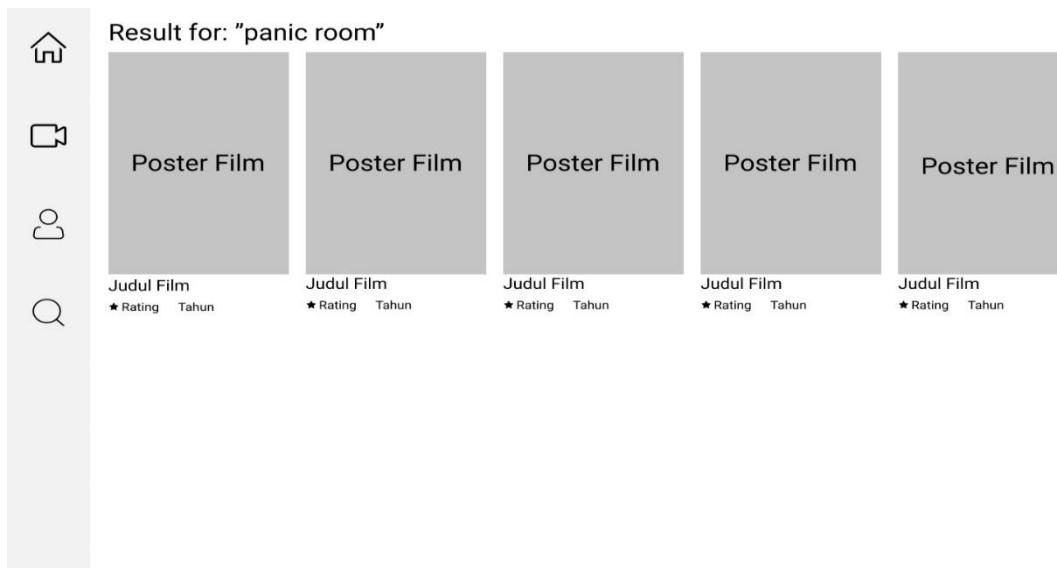
Gambar 3.8 menyajikan rancangan tampilan untuk *Detail Person*.



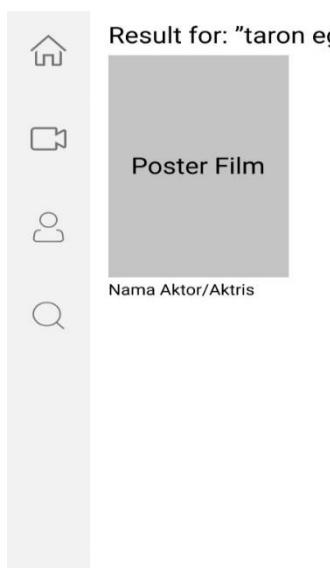
Gambar 3.8 Perancangan Tampilan *Detail Person*

Pada saat pengguna tidak dapat menemukan film maupun aktor melalui menu *movie* dan menu *person*, alternatif lain dapat digunakan yaitu menu *search*. Menu *search* dapat melakukan pencarian berdasarkan input dengan beberapa kata

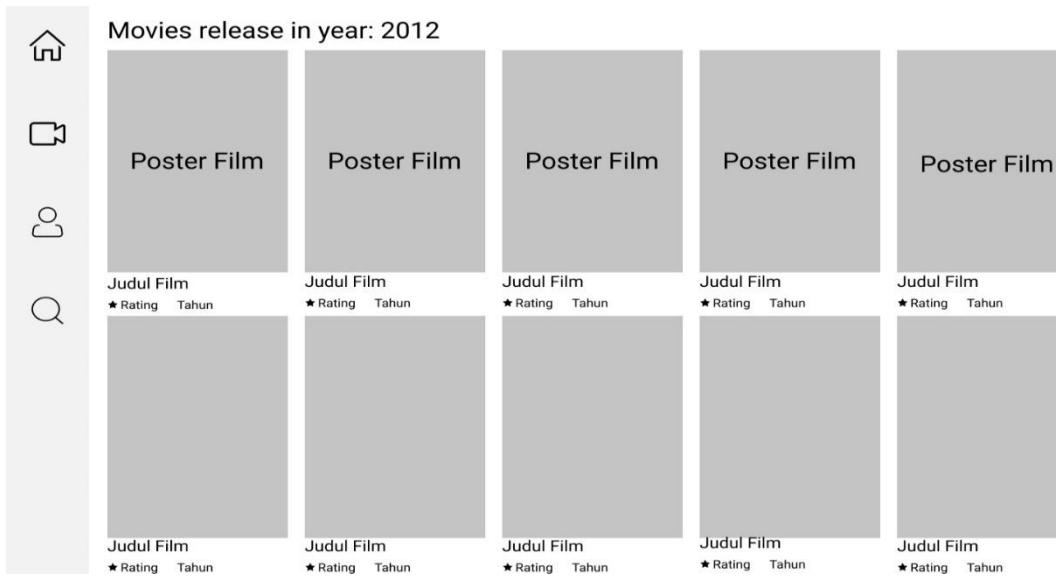
kunci yang disediakan seperti judul film, nama aktor dan tahun rilis film. **Gambar 3.9** sampai **Gambar 3.11** menyajikan rancangan tampilan untuk *Detail Search Movie*, *Detail Seacrh Person* dan *Detail Search Year*.



Gambar 3.9 Perancangan Tampilan *Detail Search Movie*

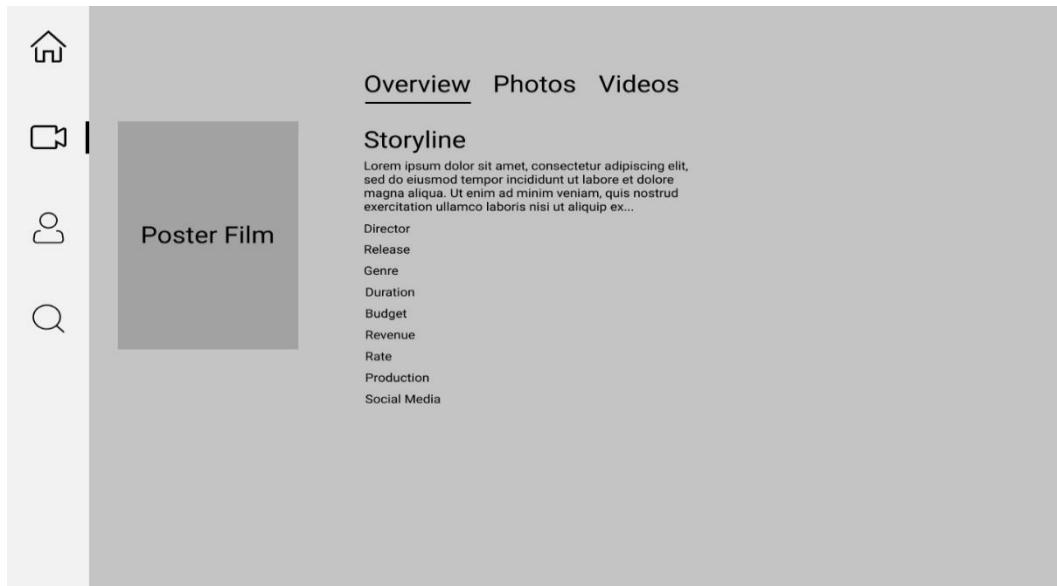


Gambar 3.10 Perancangan Tampilan *Detail Search Person*



Gambar 3.11 Tampilan *Detail Search Year*

Setelah sebelumnya pengguna dapat mencari dan memilih film rekomendasi di halaman utama, menu *movie* dan menu *search*, pengguna akan dialihkan ke tampilan dari *detail film* apabila telah mengklik poster atau judul film yang ada. Pada tampilan ini berisikan poster dari film tersebut, tambahan lainnya juga memiliki jalan cerita film, siapa yang menyutradarai film itu, tanggal rilis filmnya, termasuk ke dalam *genre* apa saja filmnya, berapa lama durasi filmnya, *budget* atau anggaran yang digelontorkan dalam pembuatan filmnya, *revenue* atau pendapatan yang dihasilkan selama film itu ditayangkan, *rate* atau nilai yang didapatkan dari penonton dengan skala 1-10 yang kemudian dirata-ratakan lalu dibagi dengan banyaknya jumlah penonton yang memberi nilai, diproduksi oleh media apa saja dan terdapat *social media* yang dapat diakses oleh pengguna untuk tujuan utama dalam hal promosi film yang berstatus pra-rilis maupun pasca-rilis. **Gambar 3.12** menyajikan rancangan tampilan untuk *Detail Film*.



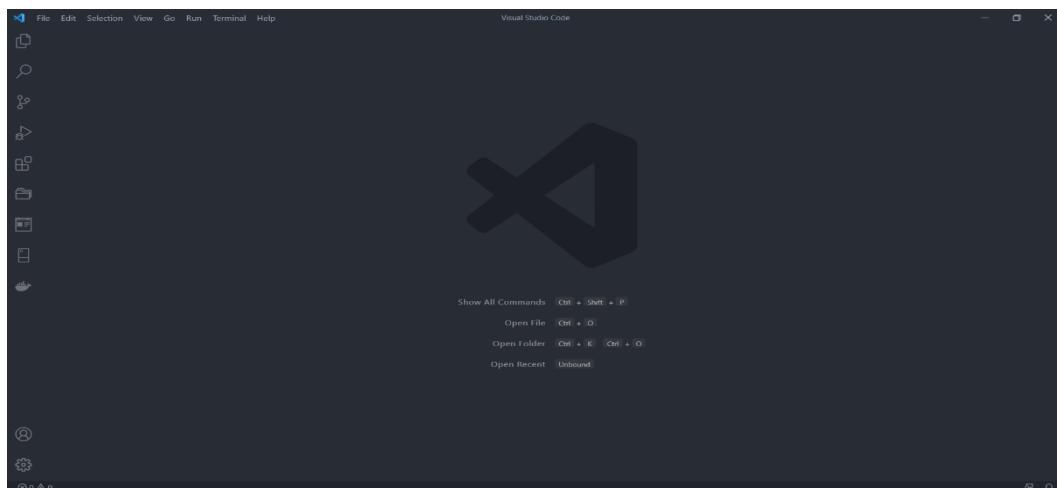
Gambar 3.12 Perancangan Tampilan *Detail Film*

3.3 Implementasi

Pembuatan halaman website pencarian informasi film ini menggunakan bahasa pemrograman Javascript dengan library React Router dan server NodeJS.

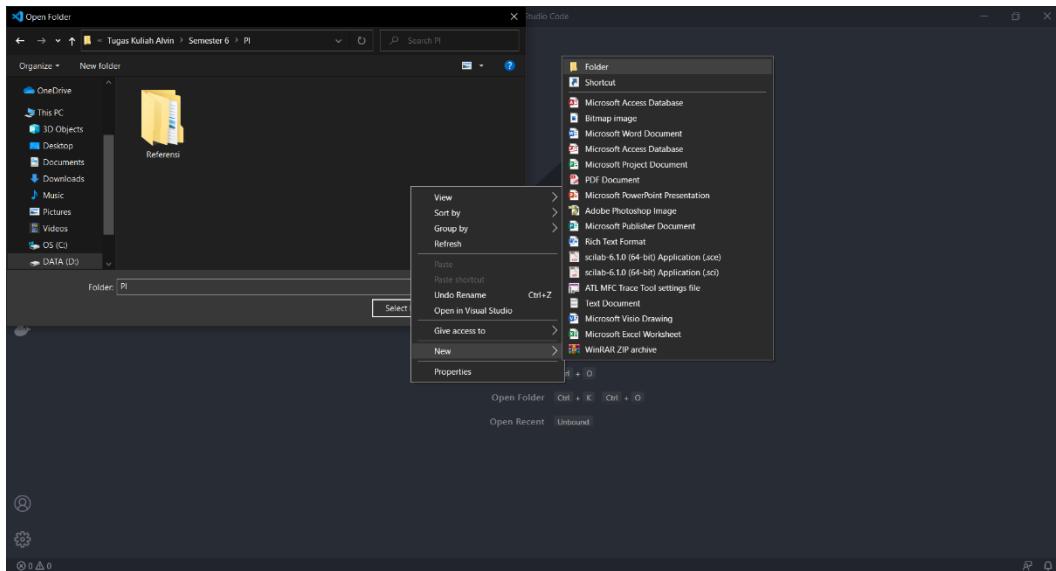
3.3.1 Membuat Project

Dalam pembuatan project disini, penulis menggunakan aplikasi visual studio code. Buka folder terlebih dahulu untuk menempatkan sebuah project. Tampilan halaman awal VS Code disajikan pada **Gambar 3.13** berikut.

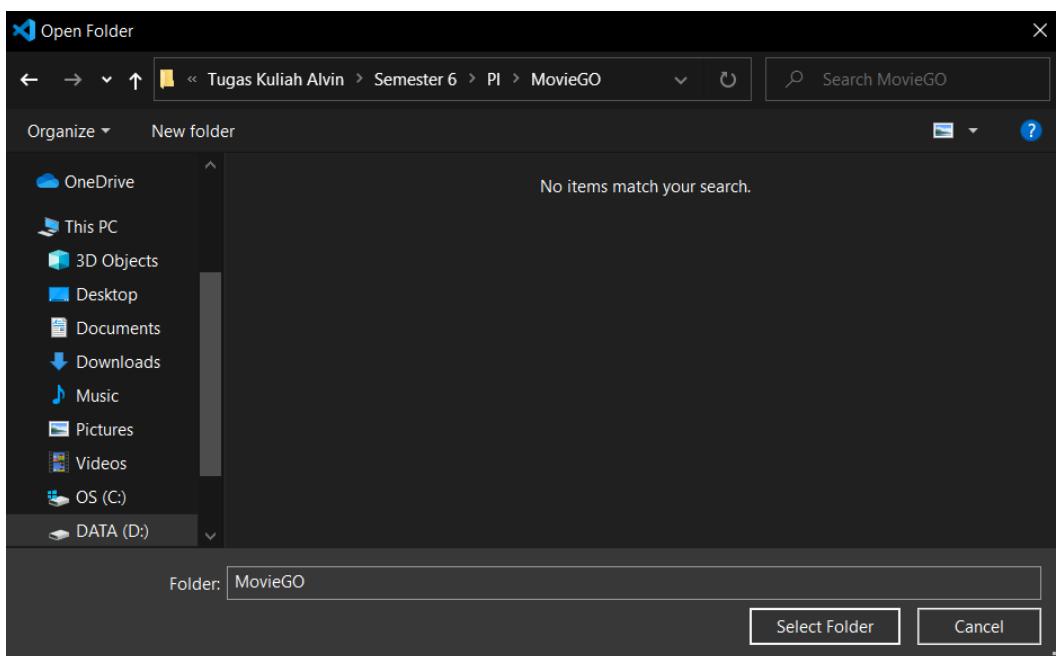


Gambar 3.13 Halaman Awal Visual Studio Code

Penulis akan membuat folder baru pada directory yang ditentukan sebagai project dengan nama **MovieGO**. Tampilan pembuatan folder dan pemilihan folder disajikan pada **Gambar 3.14** dan **Gambar 3.15** berikut.

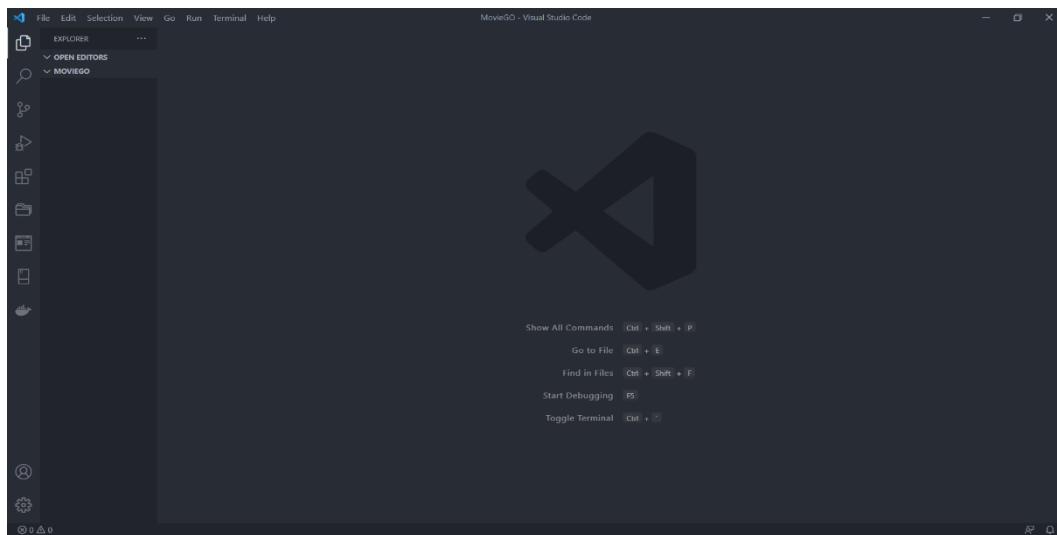


Gambar 3.14 Pembuatan Folder Baru



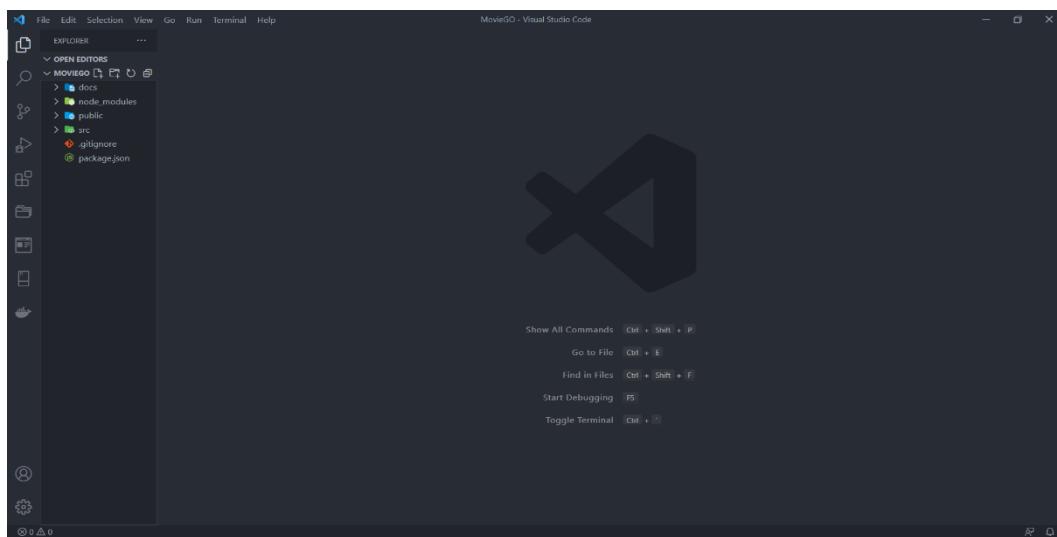
Gambar 3.15 Pemilihan Folder

Setelah membuat dan memilih folder, tampilan akan sama seperti halaman awal dari visual studio code, yang membedakan adalah muncul *sidebar* menu dengan nama folder yang sudah dipilih. File yang ada di dalam folder tersebut masih kosong karena penulis belum menambahkannya. Tampilan folder yang telah dipilih disajikan pada **Gambar 3.16** berikut.



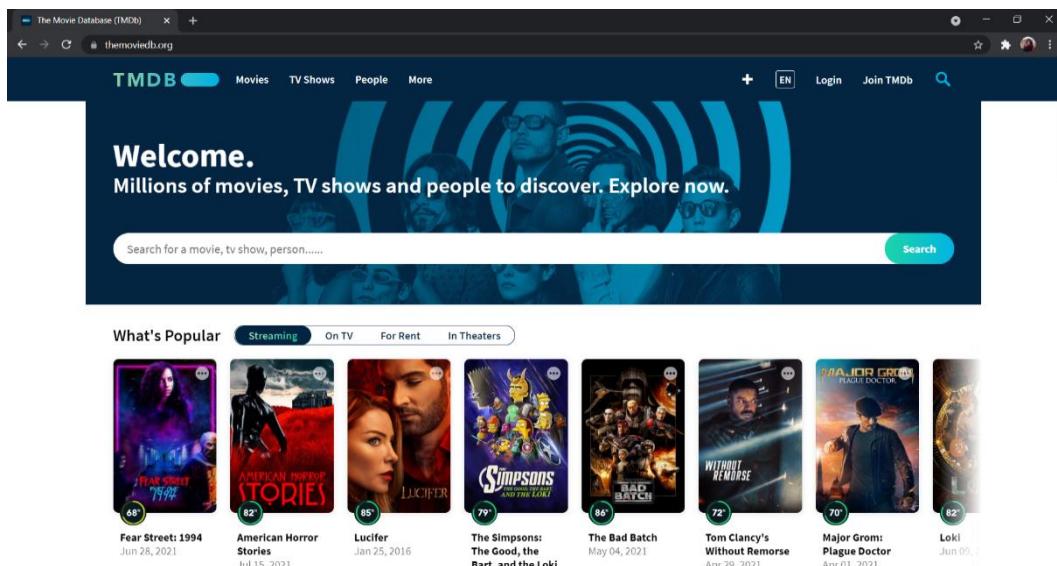
Gambar 3.16 Halaman Awal setelah pemilihan folder

Penulis menambahkan beberapa file yang dibutuhkan untuk dijadikan sebagai jalannya project tampilnya halaman website yang berisi informasi film. Tampilan file project disajikan pada **Gambar 3.17** berikut.

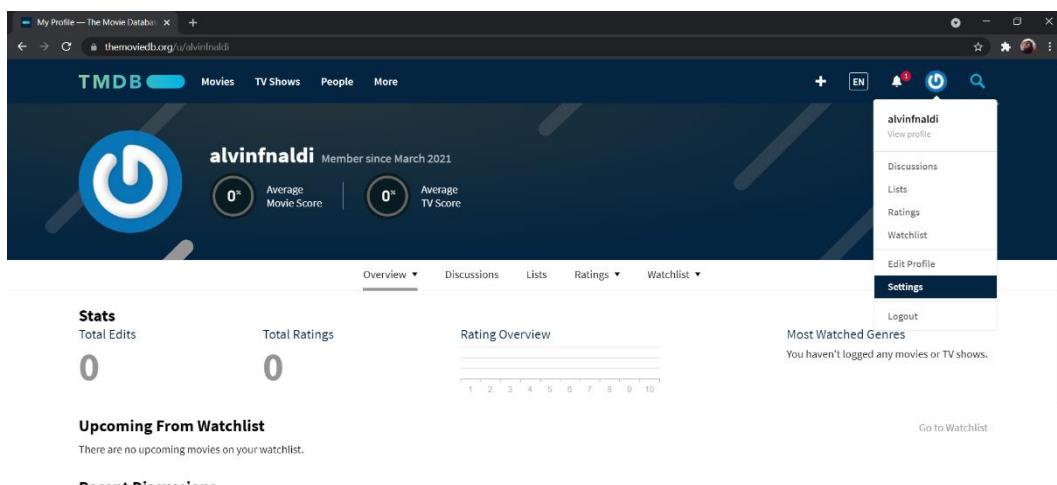


Gambar 3.17 Pembuatan File

Sebelum melanjutkannya, sumber informasi film penulis dapatkan dari website The Movie Database (TMDB). API Key diperlukan sebagai koneksi ke website agar informasi film dapat tampil pada halaman website yang dibuat. Langkah untuk mendapatkan API Key yaitu lakukan *login* ke dalam websitenya, kemudian pilih menu *settings* untuk melihat pengaturan yang tersedia. Tampilan Halaman TMDB sebelum dan sesudah *login* disajikan pada **Gambar 3.18** dan **Gambar 3.19** berikut.

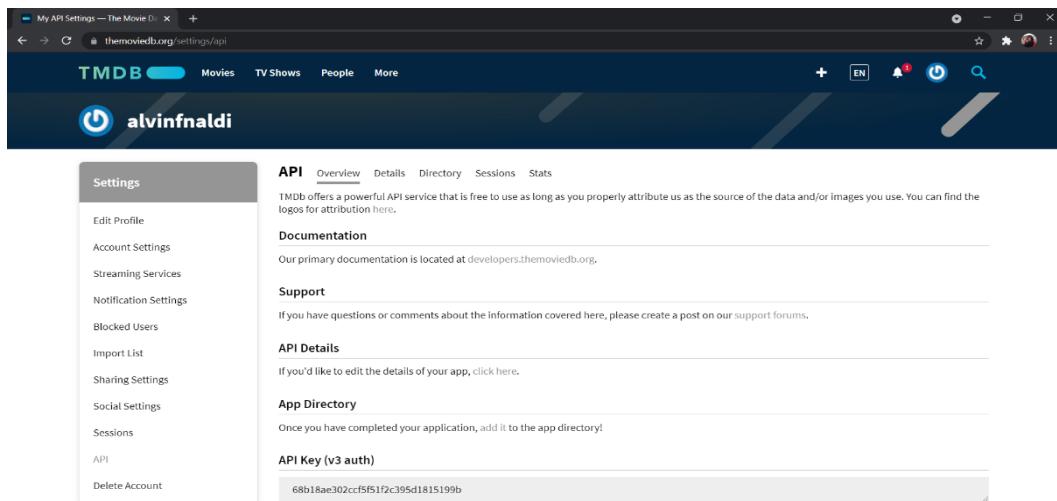


Gambar 3.18 Halaman TMDB sebelum *login*



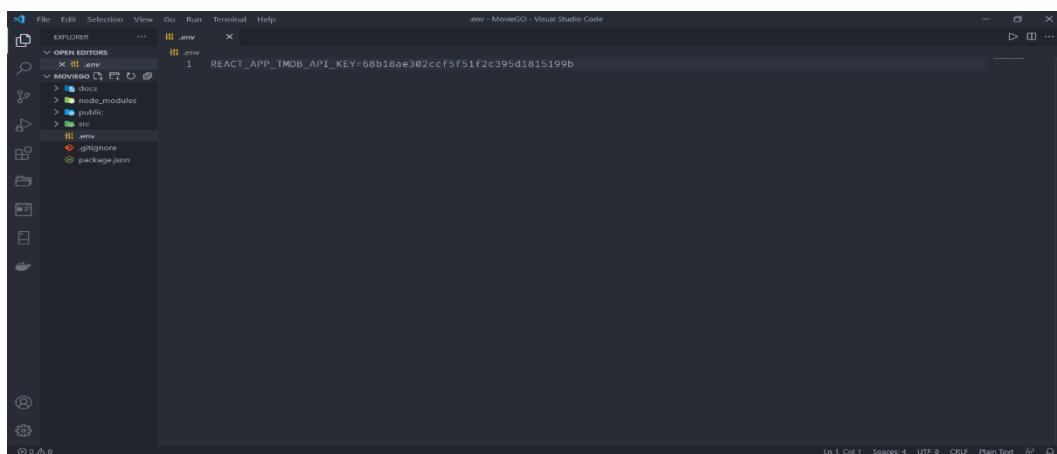
Gambar 3.19 Halaman TMDB sesudah *login*

Pada bagian *settings*, pilih menu API. Maka di bagian bawah terdapat API Key yang dapat dipakai sebagai autentikasi apabila ingin menggunakan berbagai fitur untuk menampilkan halaman tertentu yang disediakan informasinya oleh website The Movie Database (TMDB). Tampilan halaman API disajikan pada **Gambar 3.20** berikut.



Gambar 3.20 Halaman API

Buat file .env lalu masukkan API Key ke REACT_APP_TMDB_API_KEY. File .env digunakan untuk menaruh kode yang bersifat rahasia seperti host, password dan database yang digunakan agar tidak tercampur dengan file utama. Untuk memakai aksesnya hanya perlu import menggunakan metode process.env. Tampilan File .env disajikan pada **Gambar 3.21** berikut.



Gambar 3.21 File .env

3.3.2 Logika Program

Supaya aplikasi dapat berjalan sebagaimana mestinya, diharuskan untuk mengatur logika program aplikasi tersebut.

- Akses API Key

Dalam sintaks ini terdapat 2 *function* yaitu *requestHeader* dan *requestURL*.

Function requestHeader digunakan untuk meminta akses *header* dengan *method* *get* dan *header* menerima dalam bentuk *application/json*. *Function requestURL* digunakan untuk mengakses parameter *endpoint* dan *params* dengan melewati pengkondisian, kemudian *params* dapat menambahkan nilai *key* dan *val*.

```
export function requestHeader() {
    return {
        method: 'GET',
        headers: {
            'Accept': 'application/json'
        }
    }
}

export function requestURL(endpoint, params) {
    if (params === null) {
        params = [];
    }
    params.push({
        key: 'api_key',
        val: process.env.REACT_APP_TMDB_API_KEY
    });
}
```

- Program Halaman Utama

Pada halaman utama ini film dipanggil sebagai tampilan *headline* dengan *function getMovie*, *headline* film yang terpilih ditempatkan pada *jumbotron* atau *banner* di bagian atas. Untuk *trending movie*, data diambil dari *function getTrending* dimana akan menampilkan daftar judul film yang sedang *trending* saat ini berikut dengan *rating*, tahun dan poster film tersebut di bagian bawah.

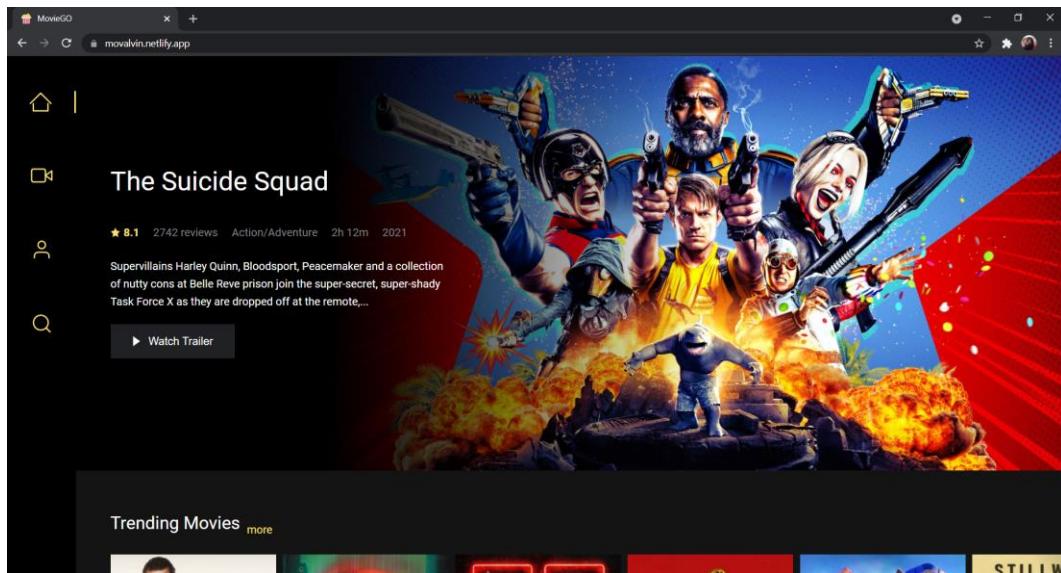
```

async getMovie(id, onSuccess) {
    try {
        const url = `/movie/${id}`;
        const response = await fetch(Lib.requestURL(url));
        const responseJson = await response.json();
        const movie = Lib.filterMovie(responseJson);
        onSuccess(movie);
    } catch (error) {
        console.log(error);
        this.handleError();
    }
}

async getTrending(media, onSuccess) {
    try {
        const response = await fetch(Lib.requestURL(`/trending/${media}/day`))
        const responseJson = await response.json();
        let data = responseJson.results;
        if (media === 'movie') {
            let movies = Lib.filterMovies(data);
            onSuccess(movies);
        }
    } catch (error) {
        console.log(error);
        this.handleError();
    }
}

```

Contoh tampilan Halaman Utama disajikan pada **Gambar 3.22** berikut.



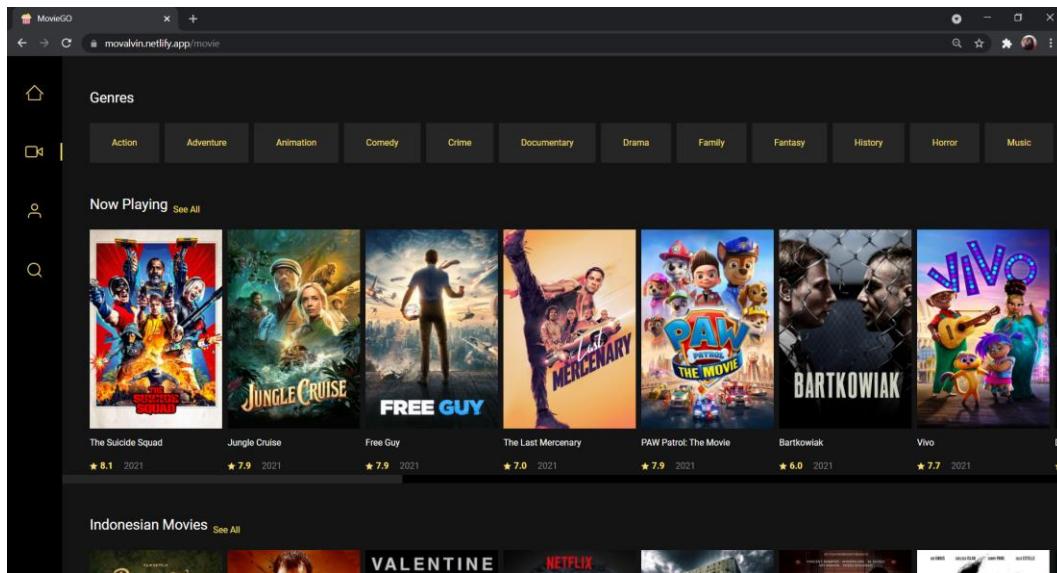
Gambar 3.22 Tampilan Halaman Utama

- Program Menu *Movie*

Pada program ini lebih memilih memecah dari banyaknya film dan memanggil *key* dan *value* yang tersedia.

```
async getMoviesByGenre(genre_id, page, onSuccess) {
    const params = [
        { key: 'with_genres', val: genre_id },
        { key: 'page', val: page }
    ];
}
async getShowing(page, onSuccess) {
    const params = [
        { key: 'primary_release_date.lte', val: Lib.formatDate(Lib.now()) },
        { key: 'sort_by', val: 'popularity.desc' },
    ];
}
async getIndonesianMovies(page, onSuccess) {
    const params = [
        { key: 'region', val: 'ID' },
        { key: 'with_original_language', val: 'id' },
    ];
}
```

Contoh tampilan Menu *Movie* disajikan pada **Gambar 3.23** berikut.



Gambar 3.23 Tampilan Menu *Movie*

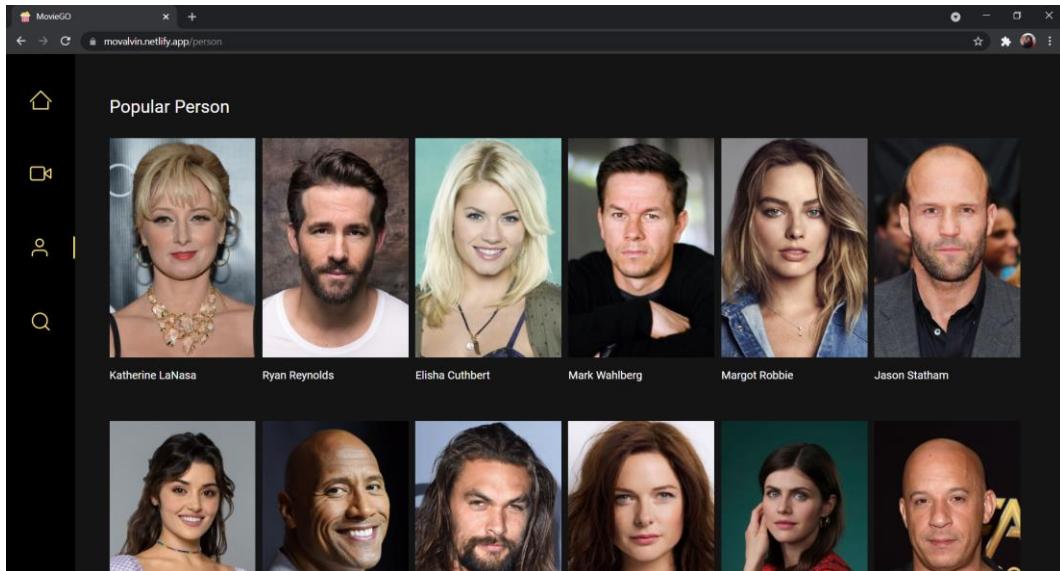
▪ Program Menu *Person*

Pada program ini, data *person* diambil dari URL */person* dengan memunculkan *id person* pada browser, id masing-masing *person* tentunya akan berbeda.

```
async getPerson(id, onSuccess) {
    try {
        const url = `/person/${id}`;
        const response = await fetch(Lib.requestURL(url))
        const responseJson = await response.json();
        const person = Lib.filterPerson(responseJson);
        onSuccess(person);

    } catch(error) {
        console.log(error);
        this.handleError();
    }
}
```

Contoh tampilan Menu *Person* disajikan pada **Gambar 3.24** berikut.



Gambar 3.24 Tampilan Menu *Person*

- Program Menu *Search*

Pada pemilihan film berdasarkan input dari pengguna, logika program ini dapat membedakan 3 inputan *keyword*, diantaranya judul film, *person* dan tahun rilis film menggunakan pengkondisionan.

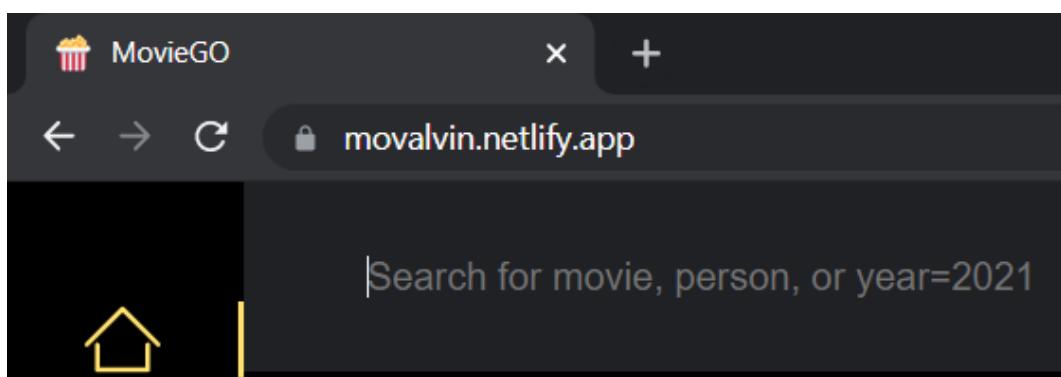
```
async search(keyword, page, onSuccess) {
  try {
    const params = [
      { key: 'query', val: keyword }
    ];
    const response = await fetch(Lib.requestURL('/search/multi', params))
    const responseJson = await response.json();
    let pages = responseJson.total_pages;
    let total = responseJson.total_results;
    let results = Lib.filterSearchResults(responseJson.results);
    onSuccess(results, pages, total);

  } catch(error) {
    console.log(error);
    this.handleError();
  }
}
```

```
search(keyword) {
    this.setState({keyword, loading: true});

    if(this.searchByYear){
        this.movieService.getMoviesByYear(keyword, this.page, (results, pages)
=> {
            const moreResults = this.state.results;
            moreResults.push(...results);
            this.setState({
                results: moreResults,
                loading: false
            });
        });
    }
    else {
        this.movieService.search(keyword, this.page, (results, pages) => {
            const moreResults = this.state.results;
            moreResults.push(...results);
            this.setState({
                results: moreResults,
                loading: false
            });
        });
    }
}
```

Contoh tampilan Menu *Search* disajikan pada **Gambar 3.25** berikut.



Gambar 3.25 Tampilan Menu *Search*

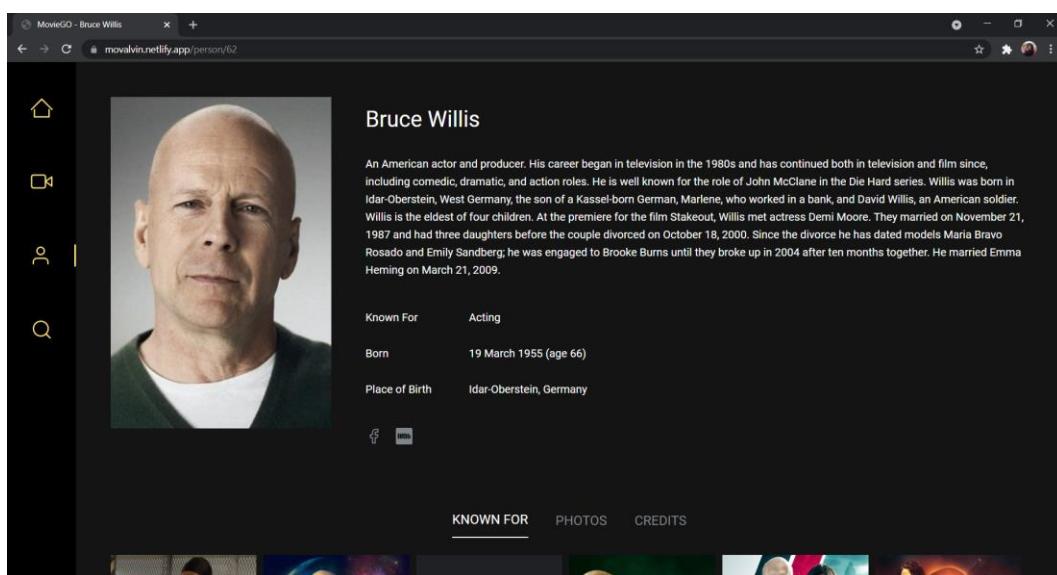
- Program *Detail Person*

Pada program ini dapat mengambil data diri dari aktor yang dipilih dari menu *person* atau menu *search*.

```
export default class Person extends Component {
  constructor(props) {
    super(props); photo: null, name: "", biography: "", knownFor: "", birthday: "",
    placeBirth: "", social: [],
  }
}

componentDidMount() {
  this.movieService.getPerson(this.personId, (person) => {
    this.setState({
      photo: person.photo,
      name: person.name,
      biography: person.biography,
      knownFor: person.knownFor,
      birthday: person.birthday,
      placeBirth: person.placeBirth,
      social: person.social,
    });
  });
}
```

Contoh tampilan *Detail Person* disajikan pada **Gambar 3.26** berikut.



Gambar 3.26 Tampilan *Detail Person*

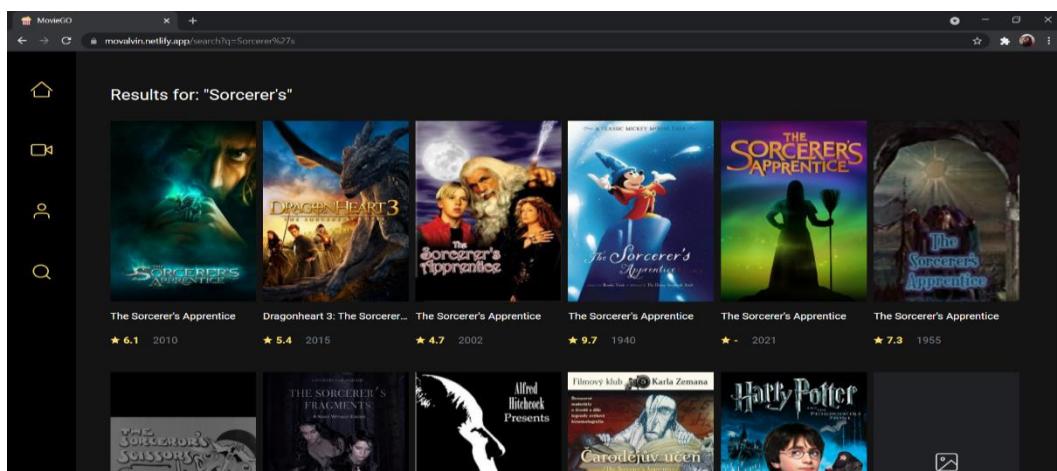
■ Program *Detail Search*

Pada program ini kurang lebih sama dengan menu *search*, hanya saja disini tanpa menggunakan pengkondisian *if else*, yaitu menggunakan pengkondisian *ternary* dalam membandingkan kedua *statement*. Input yang diketikkan oleh pengguna misalnya berupa angka tahun rilis, maka halaman akan mencari inputan tersebut lalu menampilkan tulisan “*Movies release in year: ...*” serta daftar film apa saja yang rilis di tahun tersebut. Namun jika pengguna menggunakan inputan huruf, maka tulisan yang muncul adalah “*Result for: ...*” dan tetap menampilkan daftar film atau *person* tergantung inputan dari pengguna pada menu *search*.

```
render() {
  const desc = (this.searchByYear) ?
    `Movies release in year: ${this.state.keyword}` : `Results for:
    "${this.state.keyword}"`;

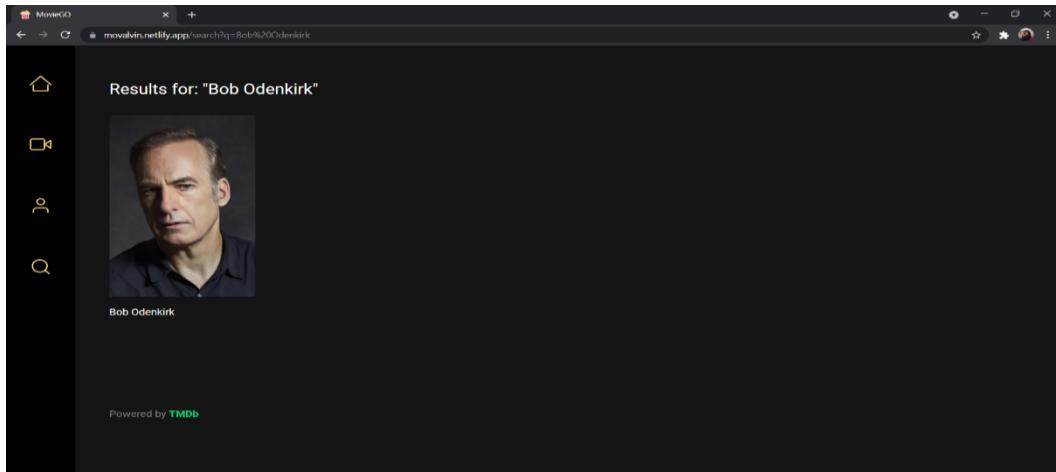
  return (
    <div>
      <div id="content-search" class="content">
        <div class="title-section"><h3> {desc} </h3> </div>
        {this.renderResults(this.state.results)} </div>
      </div> )
}
```

Contoh tampilan *Detail Search Movie* disajikan pada **Gambar 3.27** berikut.



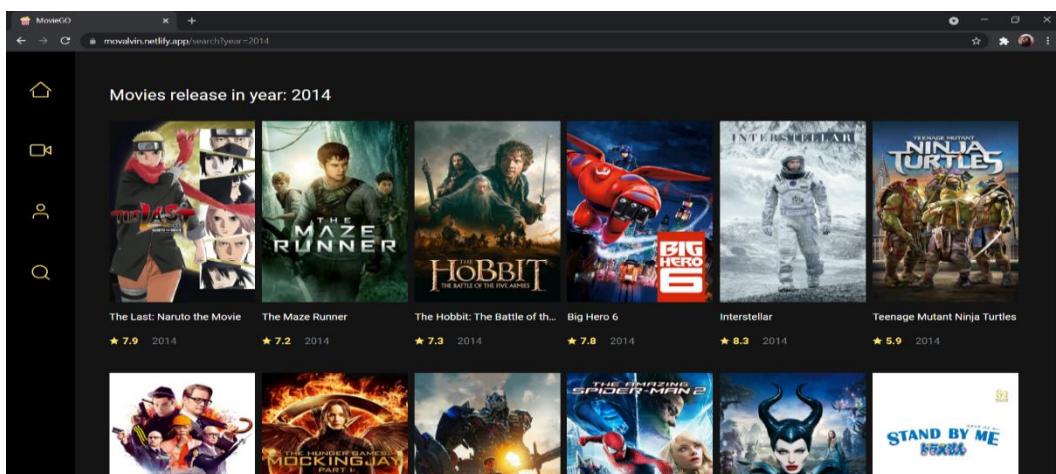
Gambar 3.27 Tampilan *Detail Search Movie*

Contoh tampilan *Detail Search Person* disajikan pada **Gambar 3.28** berikut.



Gambar 3.28 Tampilan *Detail Search Person*

Contoh tampilan *Detail Search Year* disajikan pada **Gambar 3.29** berikut.



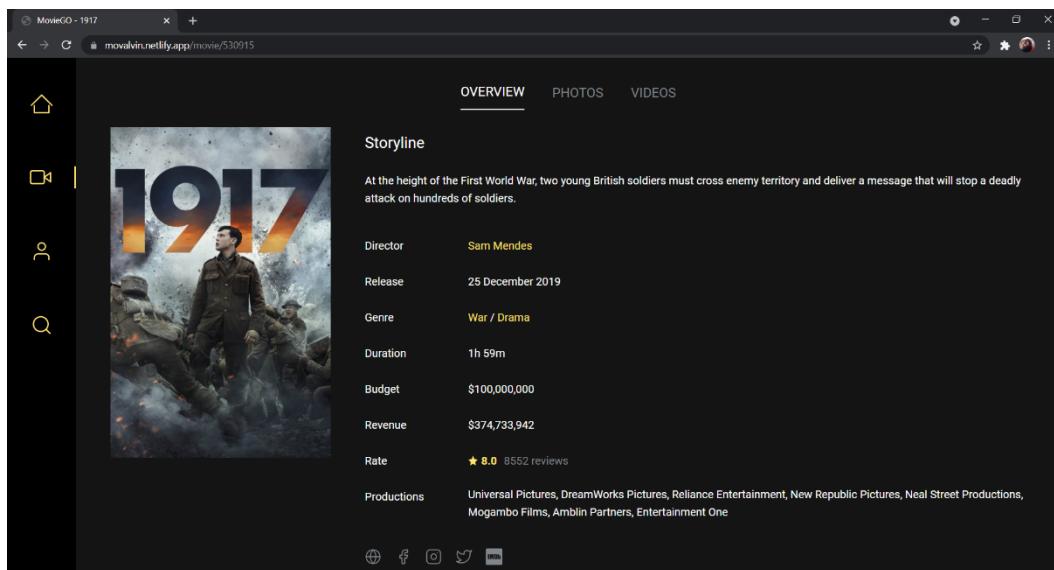
Gambar 3.29 Tampilan *Detail Search Year*

- Program *Detail Film*

Program ini akan ditampilkan apabila pengguna telah memilih film yang diinginkan melalui menu *movie* atau menu *search*.

```
export default class Movie extends Component {
  constructor(props) {
    super(props); movie: null, : null, title: "", release: "", rate: 0, vote: "", genres: [],
    overview: "", director: "", duration: "", budget: "", revenue: "", productions: "",
    social: [],
  }
  componentDidMount() {
    this.movieService.getMovie(this.movieId, (movie) => {
      this.setState({
        movie, poster: movie.poster, title: movie.title, release: movie.release,
        rate: movie.rate, vote: movie.vote, genres: movie.genres, overview: movie.overview,
        director: movie.director, duration: movie.duration,
        budget: movie.budget, revenue: movie.revenue,
        productions: movie.productions, social: movie.social,
      });
    });
  }
}
```

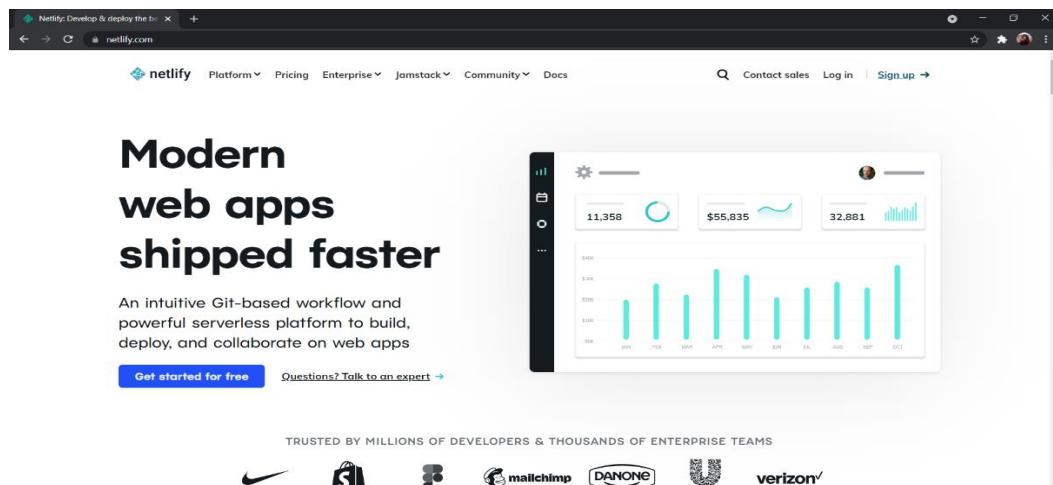
Contoh tampilan *Detail Film* disajikan pada **Gambar 3.30** berikut.



Gambar 3.30 Tampilan *Detail Film*

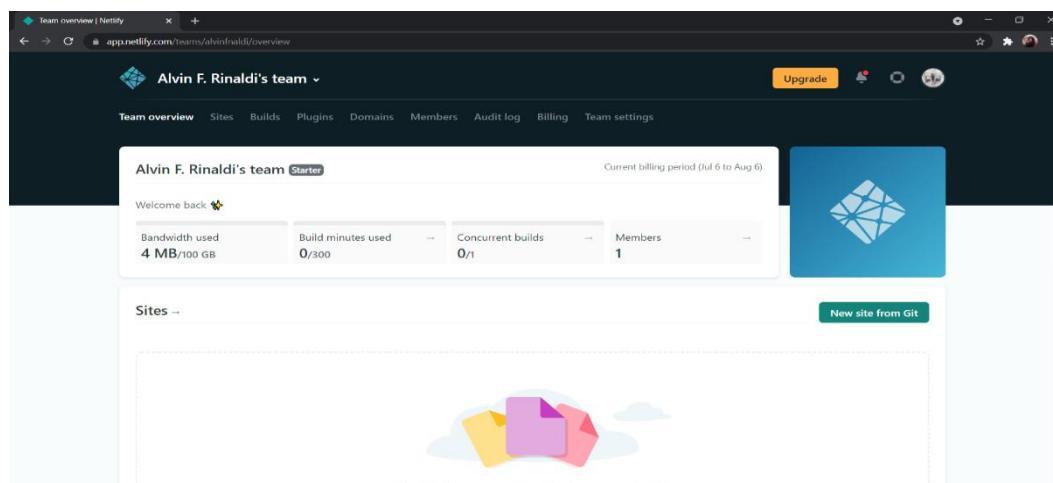
3.3.3 Hosting Website

Agar aplikasi dapat digunakan oleh semua orang, penulis akan mengonlinekan melalui aplikasi Netlify. Hal yang pertama dilakukan ialah mengunjungi situs <https://www.netlify.com/>. Di halaman ini ditampilkan menu utama dari aplikasi Netlify. Jika sudah memiliki akun, dapat langsung melakukan akses *login*. Akses *sign up* apabila belum mendaftarkan akun. Tampilan halaman awal Netlify disajikan pada **Gambar 3.31** berikut.



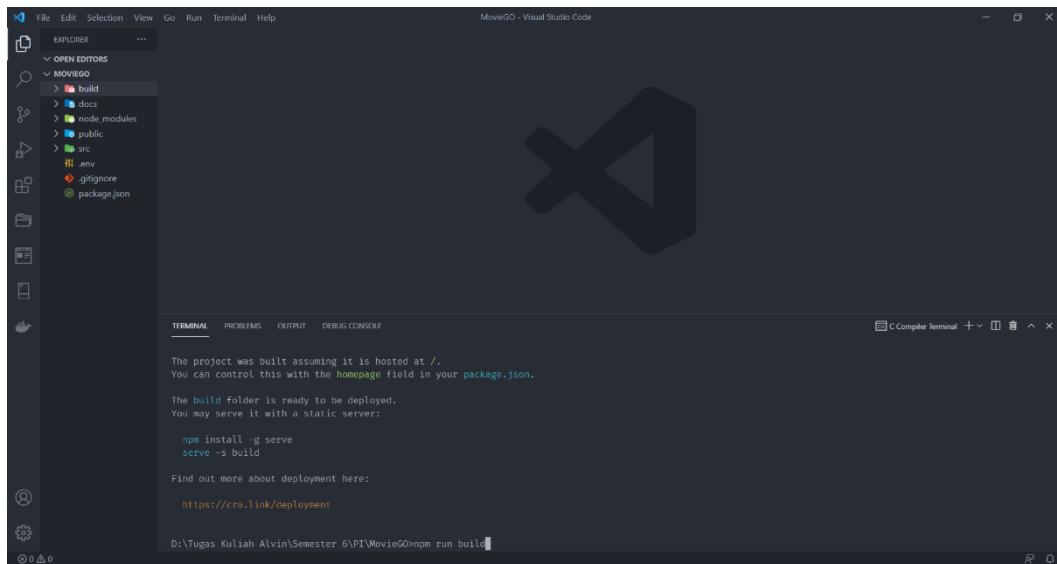
Gambar 3.31 Tampilan Halaman Netlify

Setelah berhasil *login*, maka akan menampilkan halaman kosong tanpa adanya file project karena belum menambahkan project. Tampilan halaman Netlify sesudah *login* disajikan pada **Gambar 3.32** berikut.



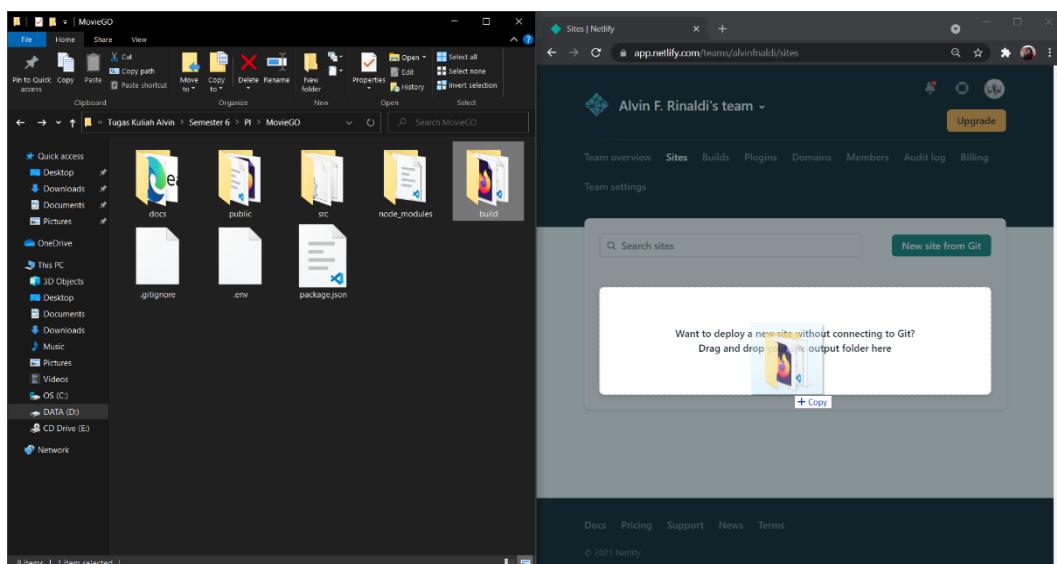
Gambar 3.32 Tampilan Halaman Netlify setelah *login*

Sebelum meng-*onlinekan*, pastikan aplikasi sudah di-*build* dengan cara membuka *command line* pada visual studio code, kemudian ketikkan perintah ***npm run build***. Tunggu prosesnya beberapa saat sampai folder *build* muncul pada sidebar menu. Tampilan folder *build* disajikan pada **Gambar 3.33** berikut.



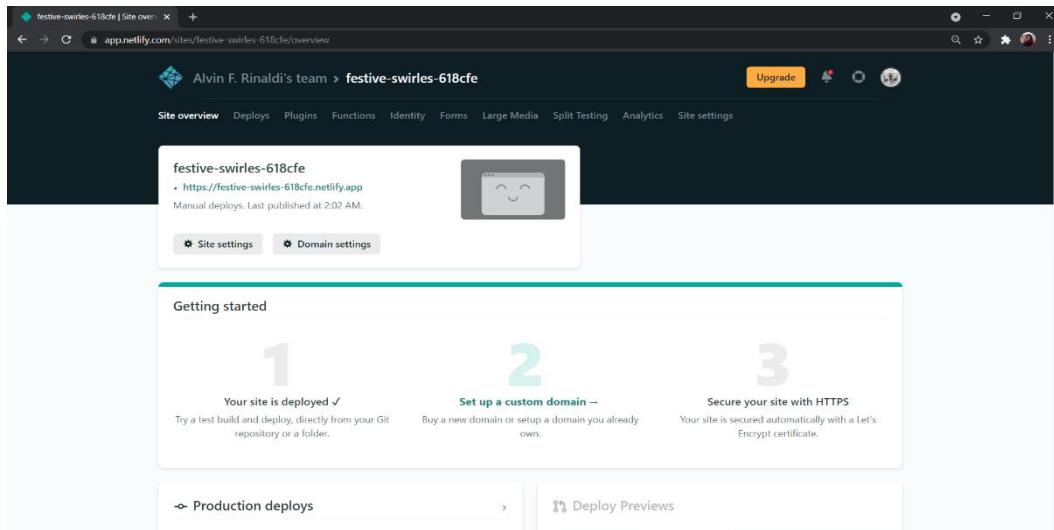
Gambar 3.33 Pembuatan folder *build*

Selanjutnya, *drag and drop* folder *build* ke situs Netlify. Kemudian tunggu proses *upload* folder tersebut. Tampilan pemindahan folder *build* disajikan pada **Gambar 3.34** berikut.



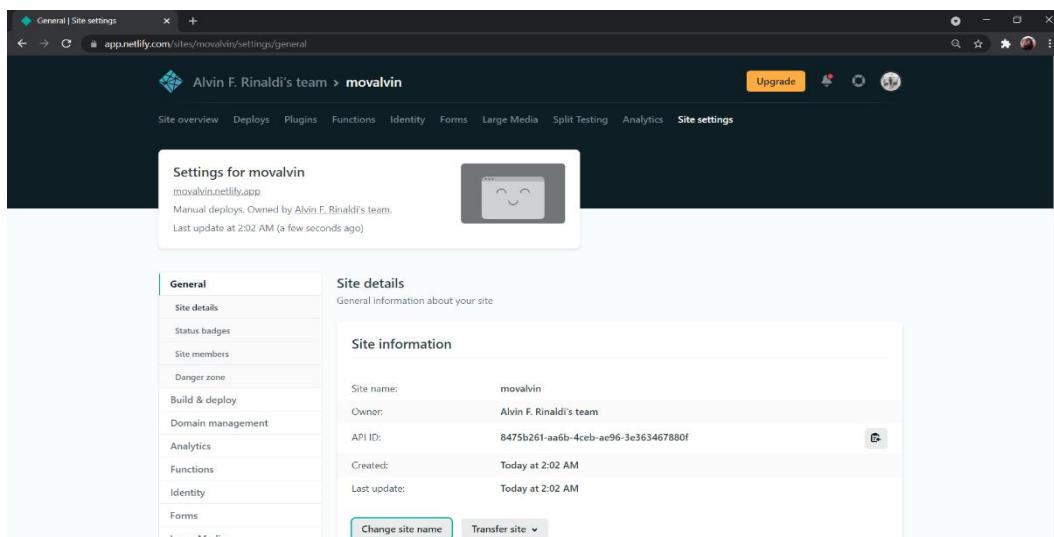
Gambar 3.34 *Drag and drop* folder *build*

Selesai melakukan *drag and drop* folder *build* dari project yang dibuat, halaman yang sebelumnya kosong akan memunculkan pemberitahuan bahwa situs telah siap, berhasil *online* dan dapat diakses secara umum melalui *link URL* yang sudah disediakan oleh Netlify. Tampilan setelah folder *build* terupload disajikan pada **Gambar 3.35** berikut.



Gambar 3.35 Website sudah *online*

Nama situs yang sudah terbuat dapat diubah melalui *site settings*. Untuk lebih memudahkan, penulis mengganti nama situs menjadi <https://movalvin.netlify.app/>. Tampilan perubahan nama situs disajikan pada **Gambar 3.36** berikut.



Gambar 3.36 Kustomisasi nama situs

3.4 Uji Coba Aplikasi

Setelah aplikasi diimplementasikan, penulis mencoba ujikan aplikasi sebelum digunakan untuk menemukan kesalahan yang mungkin terjadi saat sistem sedang berjalan. Pada Penulisan Ilmiah ini, dilakukan dua jenis pengujian, yaitu pengujian Black Box untuk menguji fungsi-fungsi pada aplikasi di platform windows dan pengujian berdasarkan spesifikasi dari platform android yang digunakan untuk menguji rancangan tampilan. Pengujian dilakukan dengan mengakses url <https://movalvin.netlify.app/> pada browser. **Tabel 3.1** menampilkan hasil pengujian Black Box, sedangkan **Tabel 3.2** menampilkan hasil pengujian berdasarkan spesifikasi smartphone android yang digunakan.

Tabel 3.1 Hasil Uji Coba Black Box

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Google Chrome	Mozilla Firefox
1	Halaman Utama	Klik menu Home	Menampilkan rekomendasi film, trending movies dan trending person	Sesuai harapan	Berhasil	Berhasil
2	Movies	Klik menu Movies	Menampilkan daftar genre film, now playing, indonesian movies, most popular dan coming soon	Sesuai harapan	Berhasil	Berhasil
3	Person	Klik menu Person	Menampilkan popular person	Sesuai harapan	Berhasil	Berhasil

			dengan infinite scrolling			
4	Search	Klik menu Search	Memunculkan dropdown input field	Sesuai harapan	Berhasil	Berhasil
5	Detail Person	Klik poster atau nama aktor yang ada di menu person	Menampilkan biografi, peran, tempat dan tanggal lahir aktor	Sesuai harapan	Berhasil	Berhasil
6	Detail Search	Ketikkan secara spesifik judul film, aktor atau tahun rilis film	Menampilkan daftar dari film, aktor atau tahun rilis yang dicari.	Sesuai harapan	Berhasil	Berhasil
7	Detail Film	Klik poster atau judul film yang berada pada halaman utama, menu movies atau berdasarkan pencarian spesifik.	Menampilkan sinopsis, sutradara, tanggal rilis, genre, durasi, anggara, pendapatan, rating, produksi dan social media dari film.	Sesuai harapan	Berhasil	Berhasil

Berdasarkan hasil uji coba menggunakan blackbox testing terhadap fungsi-fungsi aplikasi yang hasilnya ditampilkan pada **Tabel 3.1** menyatakan aplikasi berjalan baik sesuai fungsi-fungsi yang diinginkan. Selanjutnya aplikasi yang telah diuji tersebut akan diujikan juga pada beberapa smartphone. Hasil pengujian dapat dilihat pada **Tabel 3.2**.

Table 3.2 Hasil Uji Coba Berdasarkan Spesifikasi Smartphone

Tipe Smartphone	Spesifikasi				Keterangan
	OS	Resolusi	Layar	Memory	
Xiaomi Redmi Note 3 Pro	Android 6.0 (Marshmallow)	1080 x 1920 pixels	5,5 inch	3 GB RAM, 32 GB ROM	Tampilan berhasil tampil, sidebar menu pindah ke footer
Oppo F11 Pro	Android 9.0 (Pie)	1080 x 2340 pixels	6,53 inch	6 GB RAM, 64 GB ROM	Tampilan berhasil tampil, sidebar menu pindah ke footer
Realme C3	Android 10	720 x 1600 pixels	6,5 inch	3 GB RAM, 32 GB ROM	Tampilan berhasil tampil, sidebar menu pindah ke footer
Samsung Galaxy M62	Android 11	1080 x 2400 pixels	6,7 inch	8 GB RAM, 256 GB ROM	Tampilan berhasil tampil,

					sidebar menu pindah ke footer
--	--	--	--	--	-------------------------------------

Berdasarkan hasil uji coba pada beberapa smartphone seperti pada **Tabel 3.2**, aplikasi berjalan baik dengan sedikit perubahan pada sidebar menu yang berpindah ke bagian footer.

BAB IV

PENUTUP

4.1 Kesimpulan

Kesimpulan yang dapat diambil dari penulisan ilmiah yang berjudul “Aplikasi Pencarian Informasi Film Berbasis Website Menggunakan ReactJS dan NodeJS” yaitu telah berhasil dibuat dengan berdasarkan hasil uji coba yang telah dilakukan. Aplikasi ini dilakukan dengan pengujian menggunakan 2 browser pada platform windows dan 4 perangkat smartphone android yang berbeda dengan minimal sistem operasi Android Marshmallow, yaitu Xiaomi Redmi Note 3 Pro, Oppo F11 Pro, Realme C3 dan Samsung Galaxy M62. Berdasarkan hasil uji coba, fungsi-fungsi yang ada pada aplikasi berjalan dengan baik sesuai dengan rancangan. Semua informasi dalam bentuk gambar, teks dan video dapat ditampilkan sesuai rancangan pada platform windows. Sementara pada perangkat smartphone juga berjalan dengan baik, namun dibuat tampilan yang berbeda, yaitu sidebar menu dipindahkan menjadi footer menu untuk memudahkan pengguna dalam menjangkau menu tanpa harus terganggu akan tampilan yang menutupi sisi layar saat menggunakan perangkat smartphone.

4.2 Saran

Dalam implementasi aplikasi ini masih belum memiliki database yang membuat pengguna tidak mendapatkan akses untuk melakukan login dan memakai fitur lainnya seperti memberi rating atau review film serta menambahkan film favorit ke akun pengguna untuk dinonton nanti. Kepada pengembang selanjutnya, diharapkan dapat menambahkan fitur tersebut dan fasilitas lain yang sekiranya dibutuhkan oleh banyak orang sehingga aplikasi tersebut tetap dapat diandalkan dan bermanfaat.

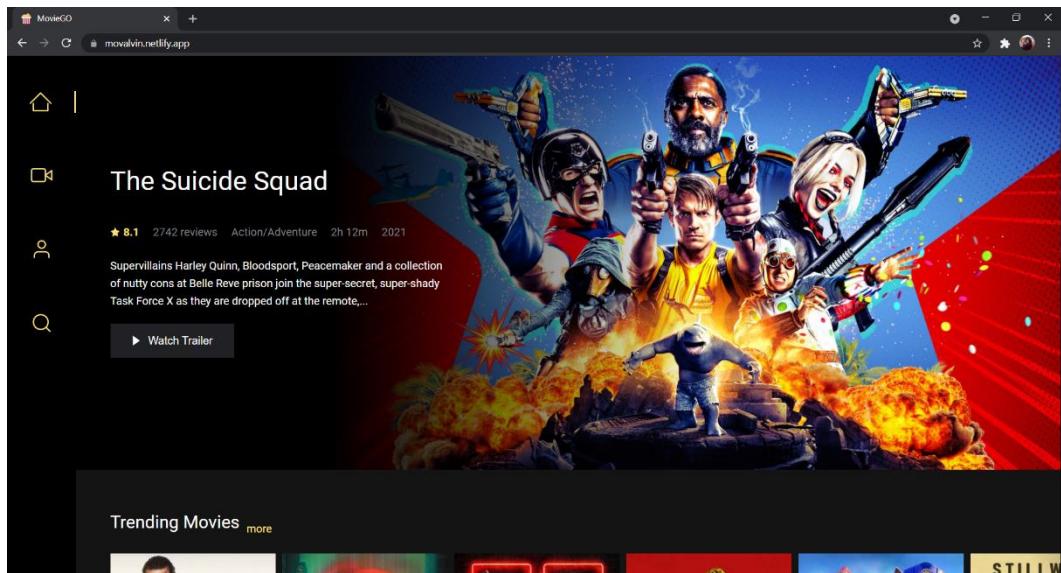
DAFTAR PUSTAKA

- [1] Adani, Muhammad Robith. 2021. "Mengenal Apa itu Use Case dan Teknik Pembuatannya", <https://www.sekawanmedia.co.id/use-case-diagram/>. Diakses pada 26 Agustus 2021.
- [2] Ansori, Ahmad. 2020. "Pengertian Activity Diagram", [https://www.ansoriweb.com/2020/03/pengertian-activity-diagram.html/](https://www.ansoriweb.com/2020/03/pengertian-activity-diagram.html). Diakses pada 26 Agustus 2021.
- [3] Bae, Sammie. 2019. JavaScript Data Structures and Algorithms: An Introduction to Understanding and Implementing Core Data Structure and Algorithm Fundamentals. Apress.
- [4] Bertoli, Michele. 2017. React Design Patterns and Best Practices: Build easy to scale modular applications using the most powerful components and design patterns. Packt Publishing.
- [5] Frelis, Jonathan. 2020. "UML (Unified Modelling Language)", <https://cloudweeb.com/blog/uml-unified-modelling-language/>. Diakses pada 6 Juli 2021.
- [6] R.H, Sianipar. 2015. Pemrograman Javascript: Teori Dan Implementasi. Bandung: Informatika.
- [7] Roziq, Mambaur. 2020. "Mengenal Apa Itu JSON (Javascript Object Notation)", <https://caraguna.com/mengenal-apa-itu-json-javascript-object-notation/>. Diakses pada 6 Juli 2021.
- [8] Setiawati, Popong. 2018. ANALISA DAN PERANCANGAN SISTEM INFORMASI PENYEDIA LOWONGAN PEKERJAAN YANG DIREKOMENDASI BERDASARKAN STANDAR KOMPETENSI KERJA NASIONAL INDONESIA (SKKNI). JIK: Jurnal Ilmu Komputer. 3(2), 138 – 139.
- [9] Sidelnikov, Greg. 2019. JavaScript Grammar, Node.js Server Setup. Learning Curve Books.
- [10] Widharma, I Wayan. 2018. "Apa itu Film?", <http://csinema.com/apa-itu-film/>. Diakses pada 22 Agustus 2021.

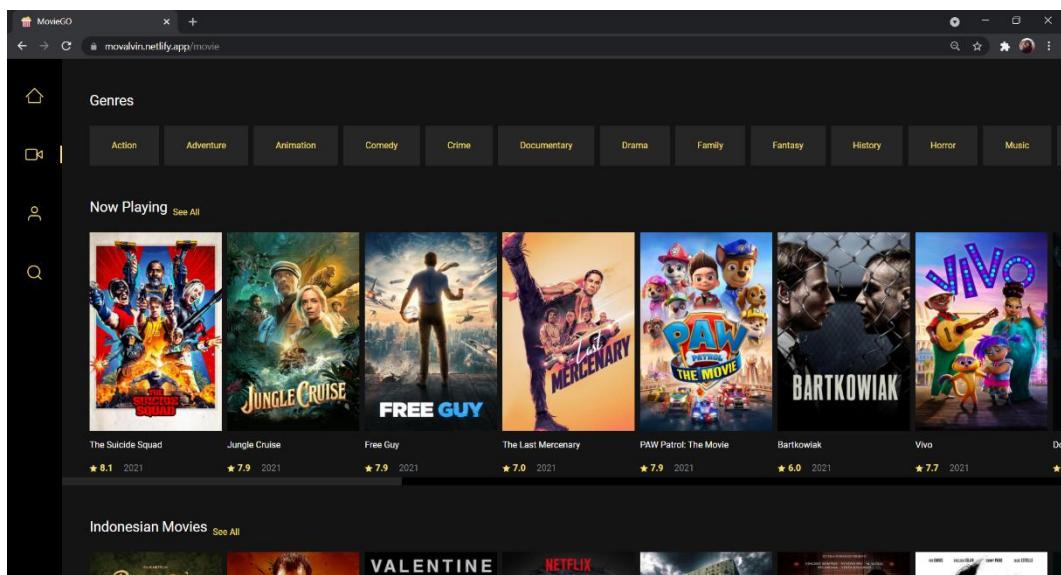
Lampiran Output Program

Versi Desktop

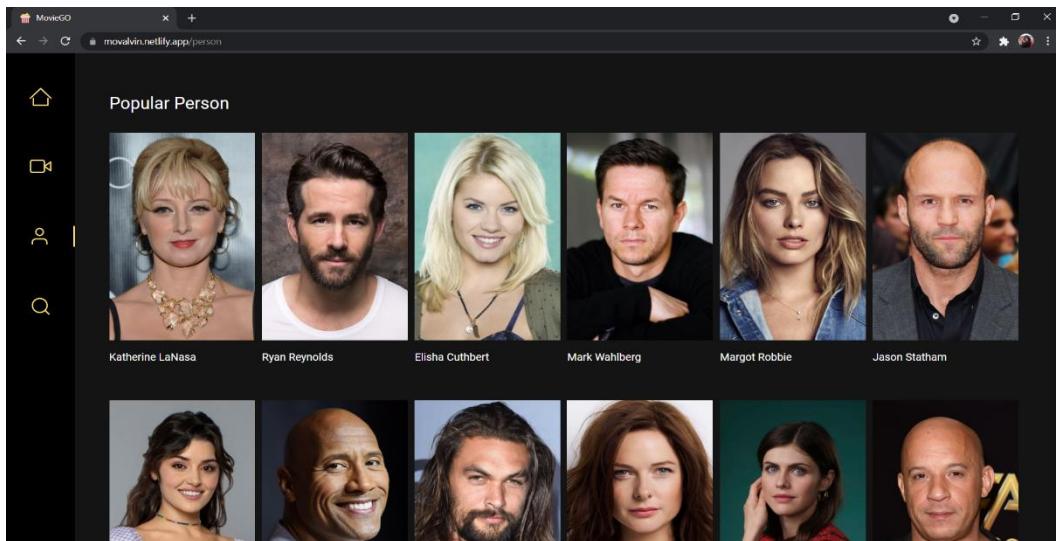
1. Halaman Utama



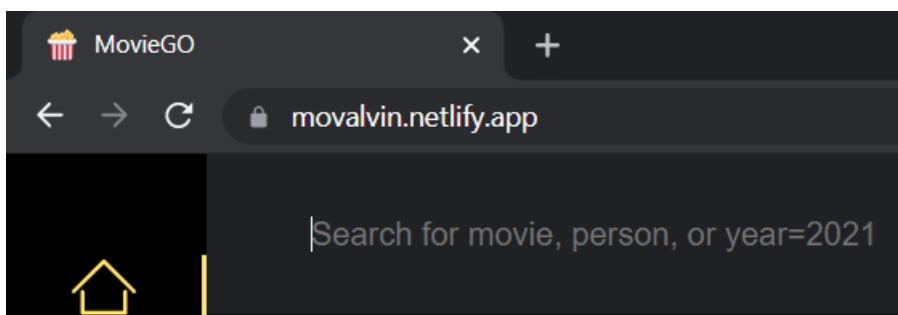
2. Menu Movie



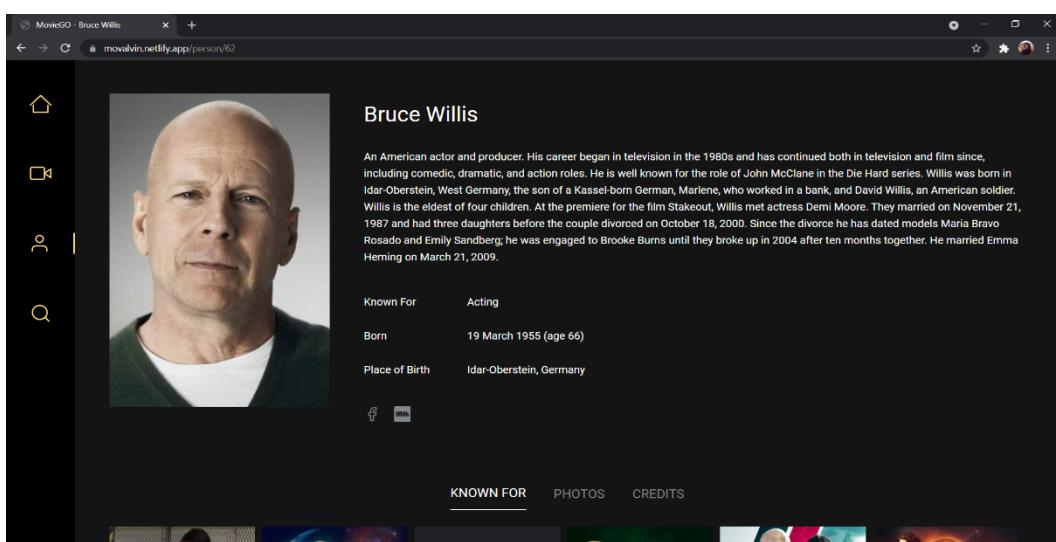
3. Menu Person



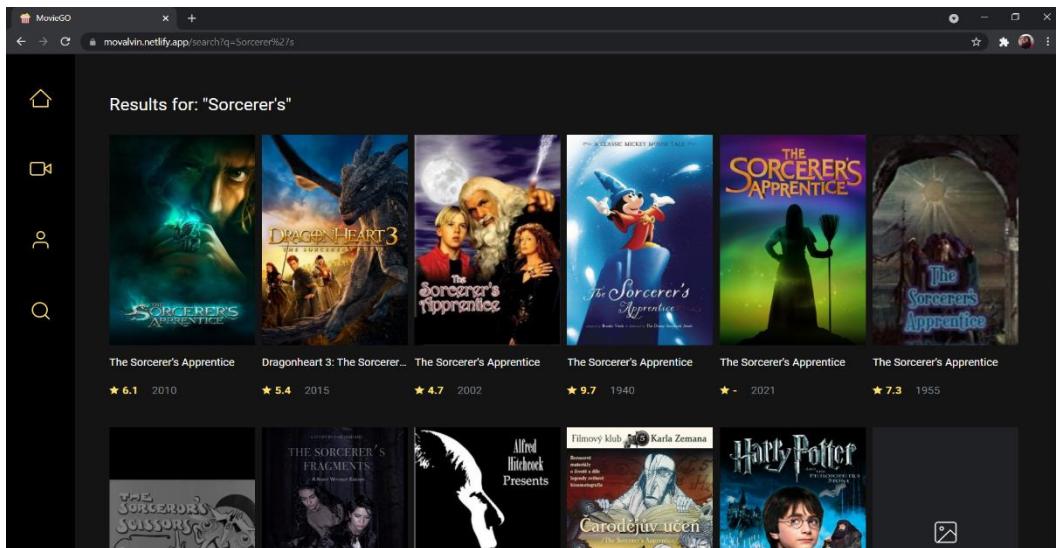
4. Menu Search



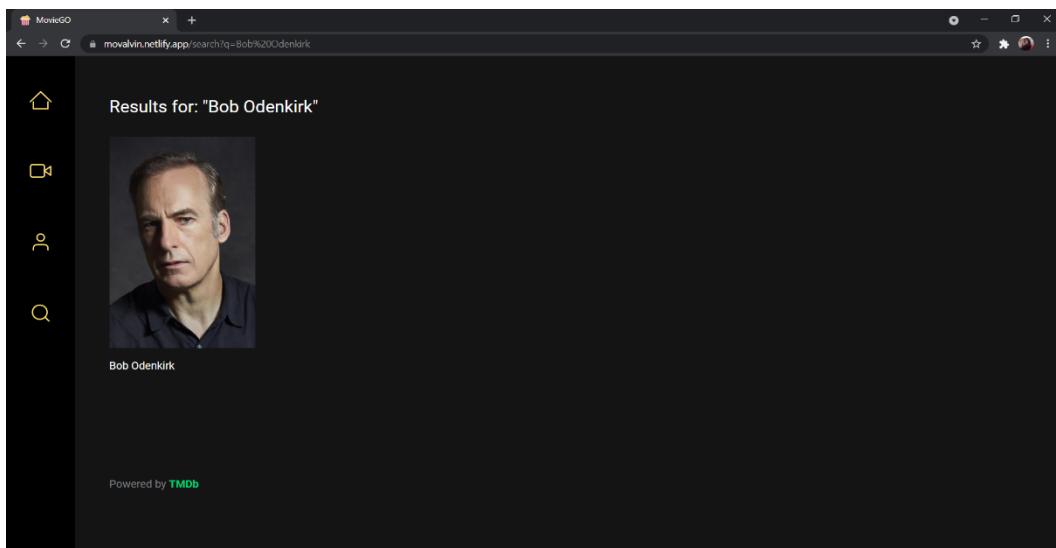
5. Detail Person



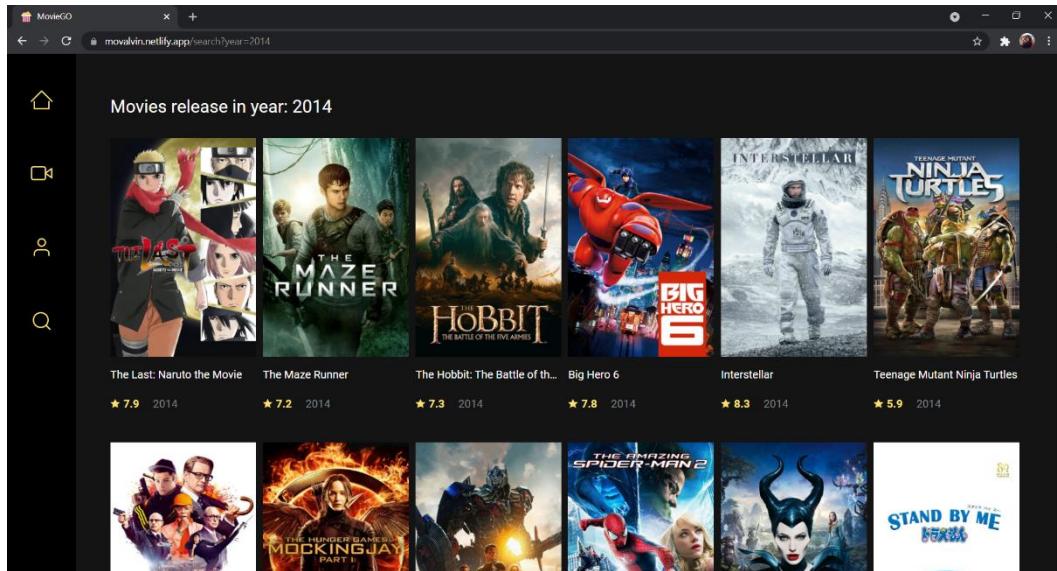
6. Detail Search Movie



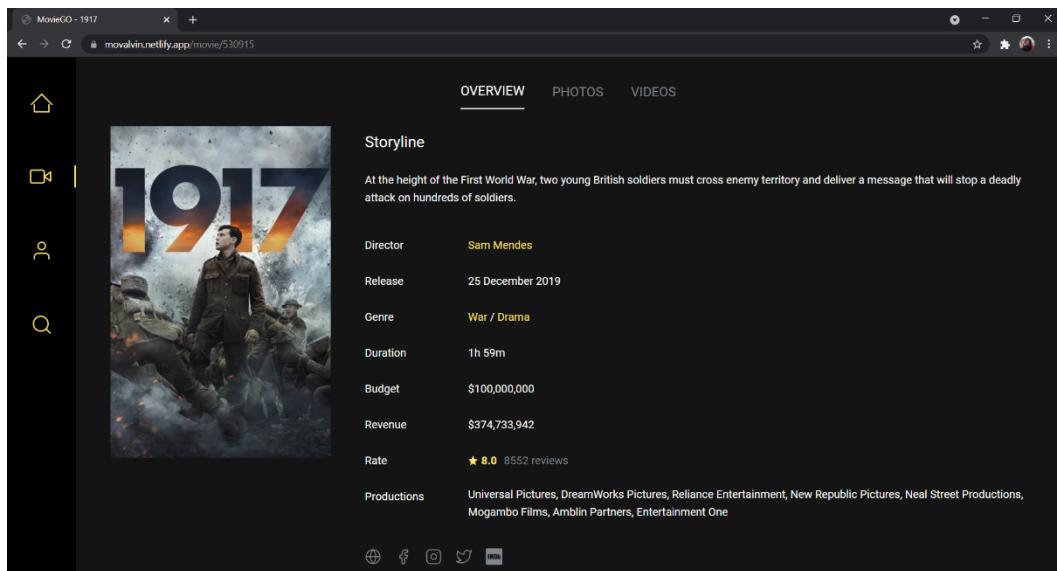
7. Detail Search Person



8. Detail Search Year

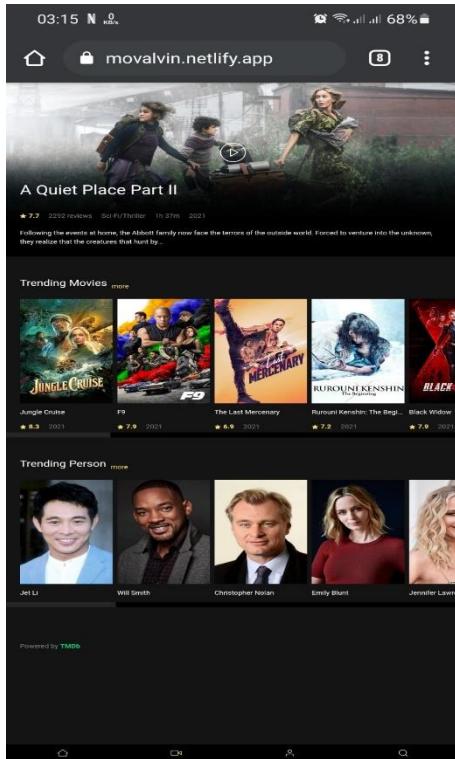


9. Detail Film

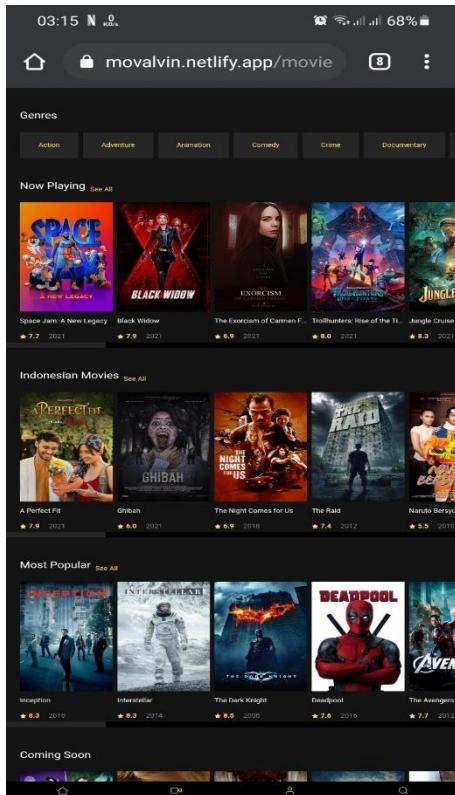


Versi Mobile

1. Halaman Utama



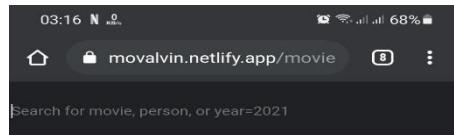
2. Menu Movie



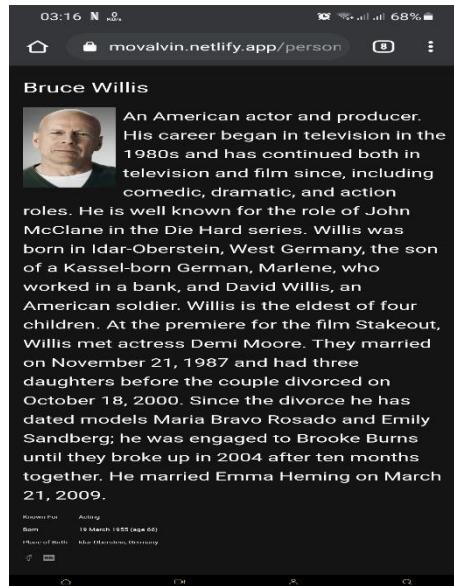
3. Menu Person



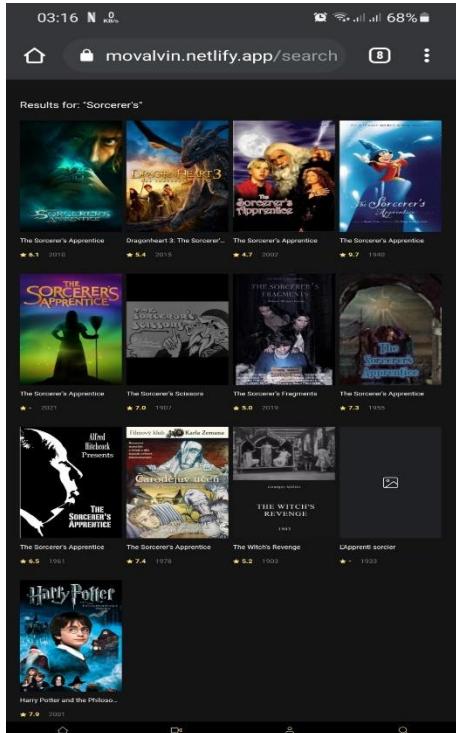
4. Menu Search



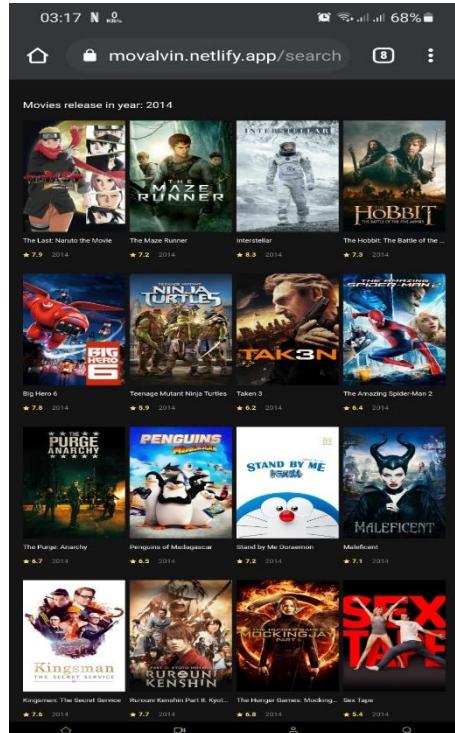
5. Detail Person



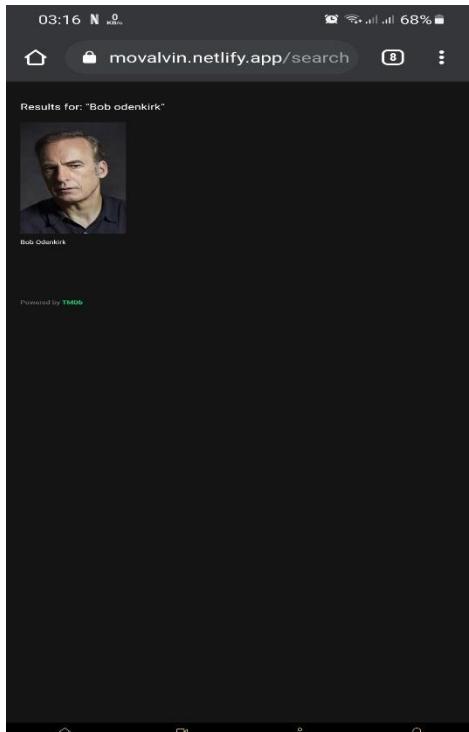
6. Detail Search Movie



8. Detail Search Year



7. Detail Search Person



9. Detail Film

