

## BAB III METODOLOGI

### A. Waktu dan Tempat Penelitian

#### 1. Waktu Penelitian

Penelitian ini dilakukan pada semester 7 tahun ajaran 2022/2023 sebagai upaya peneliti untuk mengeksplorasi metode rendering web, khususnya implementasi *Next.js*. Pada tahap ini, peneliti telah memiliki dasar pengetahuan yang memadai dalam bidang pengembangan web dan perancangan interface. Penelitian ini mencerminkan kemampuan peneliti untuk mengintegrasikan pengetahuan teoritis dengan praktik implementasi pada studi kasus yang konkret, yaitu *Website "Filmku"*. Kesempatan ini memberikan peneliti peluang untuk mendalami pemahaman kompleksitas pengembangan web di era yang terus berkembang, serta menerapkan konsep-konsep yang telah dipelajari sepanjang perjalanan studi.

#### 2. Tempat Penelitian

Dalam penelitian ini, penulis memfokuskan studi pada *Website Filmku*, *Filmku* adalah sebuah aplikasi web yang menyediakan beragam informasi tentang film, termasuk detail film, ulasan, peringkat, dan trailer. Aplikasi ini memungkinkan *user* untuk menelusuri katalog film yang luas, mencari film berdasarkan judul, genre, tahun rilis, dan aktor, serta membaca ulasan dari *user* lain. *Filmku* dibangun menggunakan *Next.js*, sebuah *framework JavaScript* yang sangat populer untuk pengembangan aplikasi web berbasis *Next.js*. Aplikasi *Filmku* bertujuan untuk memberikan pengalaman yang menyenangkan bagi para *user* yang gemar menonton film.

Beberapa fitur yang terdapat di aplikasi ini adalah sebagai berikut:

##### a. Pencarian Film

*User* dapat dengan mudah mencari film berdasarkan berbagai kriteria seperti judul, genre, tahun rilis, atau aktor.

##### b. Detail film

Aplikasi ini menyediakan informasi lengkap tentang setiap film, termasuk sinopsis, aktor, sutradara, rating, dan trailer jika tersedia.

##### c. Peringkat *User*

*User* dapat memberikan ulasan dan penilaian pada film yang ditontonnya sehingga *user* lain dapat membaca pendapatnya.

##### d. Peringkat Film

Aplikasi ini menampilkan peringkat film berdasarkan ulasan *user* serta peringkat kritikus jika tersedia.

##### e. Daftar favorit

*User* dapat menyimpan film ke daftar favorit untuk ditonton nanti.

##### f. Tampilan Responsif

Aplikasi *Filmku* dirancang *responsif*, sehingga dapat diakses dengan baik baik dari komputer desktop maupun perangkat seluler.

## B. Alat dan Bahan

Pengukuran kinerja *Website* Filmku dilakukan dengan menggunakan alat-alat berikut:

Tabel 8 Kebutuhan Hardware

No	Kebutuhan <i>Hardware</i>	Spesifikasi	Keterangan
1	Laptop Asus Vivobook 14 K413EA	RAM 16 GB NVMe, ROM 512 GB SSD, Windows 11	Digunakan untuk mencari dokumen referensi, membuat diagram, dan membuat program aplikasi.

Tabel 9 Kebutuhan *Software*

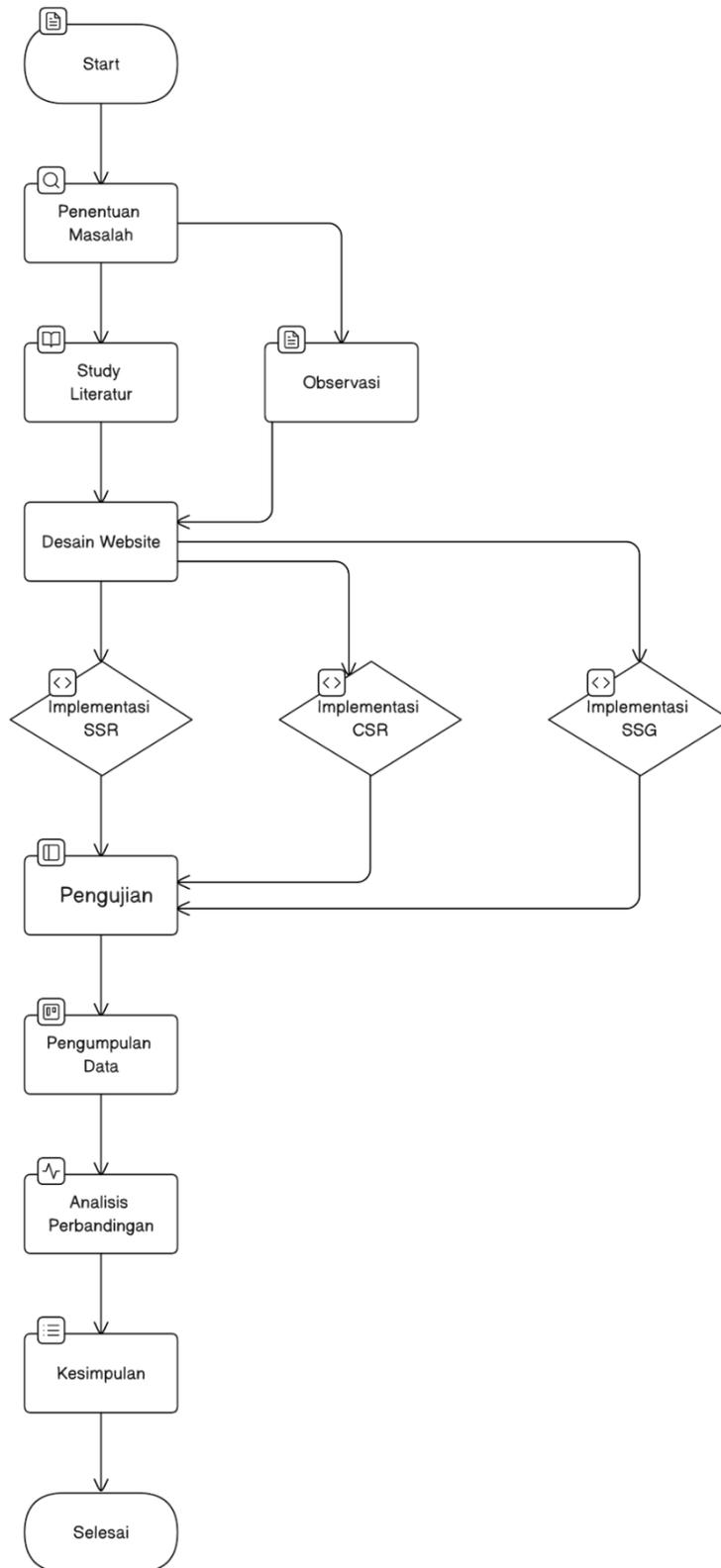
No	Kebutuhan <i>Software</i>	Keterangan
1	Windows 11 32 bit	Windows 11 adalah sistem operasi yang digunakan sebagai platform untuk menjalankan dan menguji aplikasi Filmku. Ini menyediakan lingkungan yang stabil dan kompatibel dengan aplikasi berbasis web, memungkinkan peneliti untuk mengakses dan menganalisis aplikasi Filmku dengan baik.
2	Visual Studio Code	VS Code adalah sebuah kode editor yang digunakan untuk mengedit dan menganalisis kode sumber aplikasi Filmku. Dengan fitur-fitur seperti penyorotan sintaksis, <i>debugging</i> , dan integrasi dengan Git, peneliti dapat mengakses dan memeriksa kode sumber aplikasi Filmku secara efisien.
3	Ubuntu server v22.04	Untuk server local website filmku saat pengujian
4	Vmware Pro v17	Untuk membantu menjalankan ubuntu 22.04 dalam sistem windows
5	Putty	Untuk meremote SSH ubuntu 22.04 supaya menjalankan website secara local lebih mudah.

---

No	Kebutuhan <i>Software</i>	Keterangan
5	<i>Development Tools</i>	<i>Development Tools</i> adalah proses memonitor dan menganalisis lalu lintas jaringan komputer untuk memahami, mendiagnosis, atau mengatasi masalah yang terkait dengan koneksi jaringan atau permintaan web. Ini adalah fungsi yang sering digunakan dalam pengembangan web, pemecahan masalah jaringan, dan optimisasi performa situs web.

---

### C. Prosedur Penelitian



Gambar 9 Prosedur Penelitian

Detail tahapan prosedur penelitian pada Gambar 9 adalah sebagai berikut :

1. Penentuan Masalah

Langkah pertama dalam penelitian adalah mengidentifikasi masalah atau tujuan penelitian. Pada tahap ini Penulis akan mengajukan pertanyaan penelitian dan mengidentifikasi masalah yang perlu diselesaikan melalui analisis aplikasi Filmku.

2. Studi Literatur dan Observasi

Setelah mengidentifikasi masalah, langkah selanjutnya adalah melakukan penelitian dan observasi dokumen. Penulis akan melakukan pencarian literatur yang relevan dengan topik penelitian dan meninjau aplikasi Filmku serta penelitian yang ada. Ini membantu Peneliti memahami lanskap dan tren terkini di lapangan.

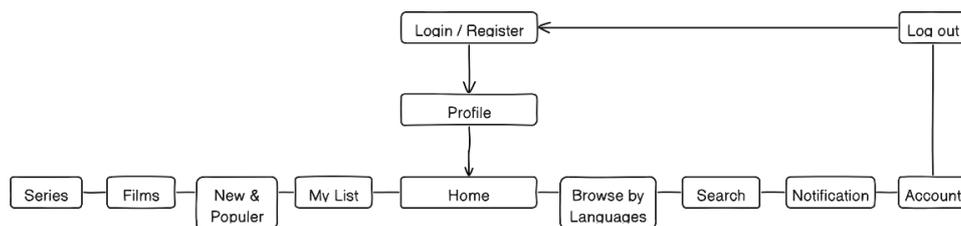
3. Desain Website

Disini peneliti akan merencanakan dan merancang Website yang akan digunakan dalam penelitian. Peneliti akan menentukan struktur, tampilan, dan fungsionalitas yang akan diterapkan pada Website yang akan dianalisis.

a. Struktur Website

Website filmku menggunakan Struktur navigasi campuran atau disebut struktur navigasi bebas. Dimana didalamnya terdapat beberapa gabungan dari struktur navigasi lainnya. Struktur navigasi campuran ini banyak digunakan dalam membuat website, karena struktur ini dapat memberikan tingkat interaksi yang lebih tinggi. Dan keterikatan dalam halaman website dapat dibuat lebih efisien dan menarik [20]. Pada masing masing halaman memuat data-data yang berbeda, tentunya dengan besar resource yang berbeda. Struktur website filmku dapat dilihat pada gambar dibawah ini.

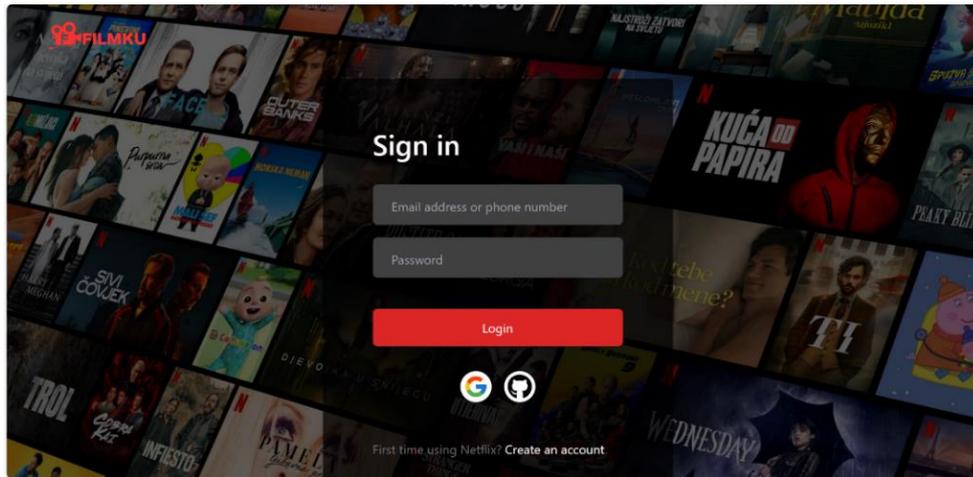
### STRUKTUR WEBSITE FILMKU



Gambar 10 Struktur Website Filmku

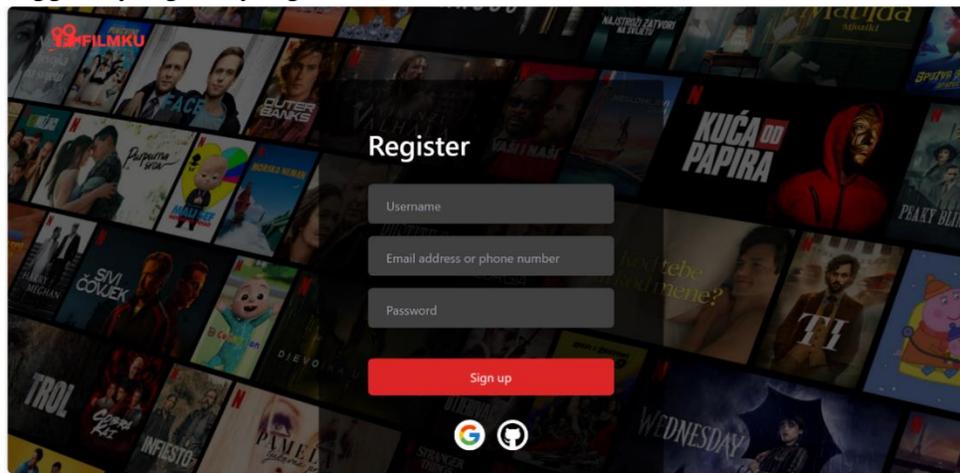
b. Tampilan dan Fungsionalitas Website

Berikut ini adalah gambar tampilan Login, Register, Profile dan halaman utama website Filmku.



Gambar 11 Tampilan Login

Halaman login adalah tempat di mana pengguna memasukkan kredensial mereka untuk mengakses akun yang sudah terdaftar. Fungsi utama halaman login adalah untuk memverifikasi identitas pengguna dengan membandingkan informasi yang dimasukkan (biasanya username/email dan password) dengan data yang telah disimpan dalam database. Proses ini memastikan bahwa hanya pengguna yang sah yang memiliki akses ke konten atau fitur tertentu.



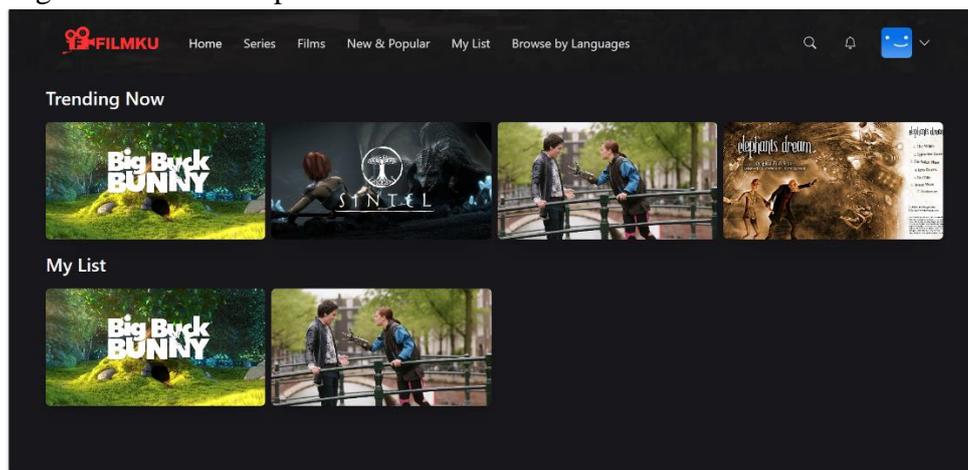
Gambar 12 Tampilan halaman Register

Halaman registrasi memungkinkan pengguna baru untuk membuat akun baru di dalam sistem. Fungsi utamanya adalah mengumpulkan informasi yang diperlukan dari pengguna dan menyimpannya ke dalam database untuk penggunaan selanjutnya. Proses ini mencakup validasi data yang dimasukkan untuk memastikan integritas dan keamanan sistem.



Gambar 13 Tampilan Halaman Utama

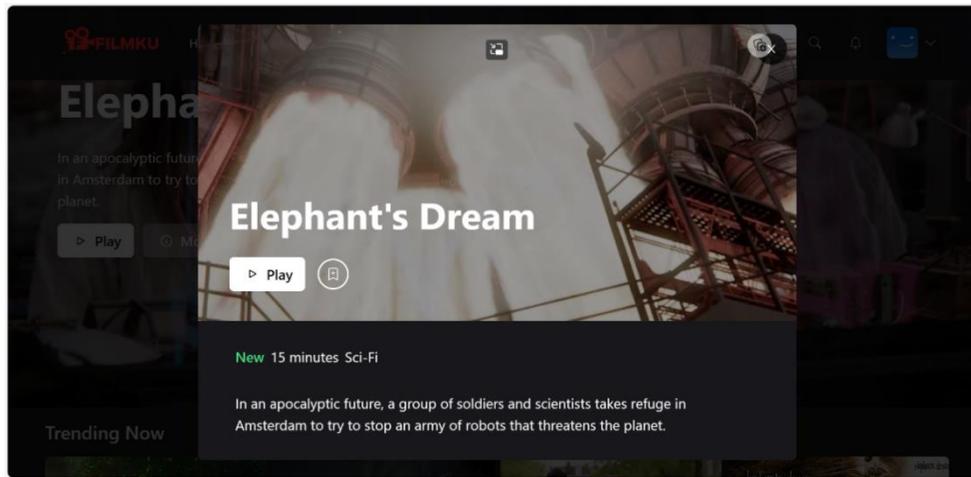
Halaman utama Filmku dirancang untuk membuat pengguna merasa tertarik, mudah menavigasi, dan menemukan konten yang sesuai dengan preferensi mereka. Fungsionalitas ini memberikan pengalaman pengguna yang lebih personal dan menyenangkan dalam mengeksplorasi berbagai film dan serial yang ditawarkan oleh platform tersebut.



Gambar 14 Tampilan halaman *Trending Now* dan *MyList*

Halaman "Trending Now" menampilkan daftar konten yang saat ini sedang populer di kalangan pengguna Netflix. Konten yang muncul di sini dipilih berdasarkan sejumlah faktor, termasuk popularitas umum, tren penonton, dan preferensi geografis. Halaman "My List" adalah tempat di mana pengguna dapat menyimpan film, acara TV, atau episode yang ingin mereka tonton nanti. Ini memungkinkan pengguna untuk membuat daftar pribadi konten yang menarik perhatian mereka tanpa harus mencarinya setiap kali mereka ingin menonton. Kedua halaman tersebut dirancang untuk meningkatkan pengalaman menonton dengan menyediakan rekomendasi yang lebih baik dan memudahkan pengguna dalam mengelola dan mengakses konten yang mereka

sukai. Dengan algoritma rekomendasi yang cerdas, Filmku berusaha untuk menyajikan konten yang sesuai dengan preferensi setiap pengguna, menjadikan pengalaman menonton lebih personal dan menyenangkan.



Gambar 15 Tampilan Detail Film

Halaman detail film di situs web Filmku merupakan halaman yang menyajikan informasi rinci tentang suatu film yang tersedia di platform tersebut. Halaman ini dirancang untuk memberikan penonton potongan informasi yang cukup untuk membuat keputusan apakah film tersebut sesuai dengan preferensi mereka. Berikut adalah beberapa elemen yang biasanya terdapat pada halaman detail film di Filmku:

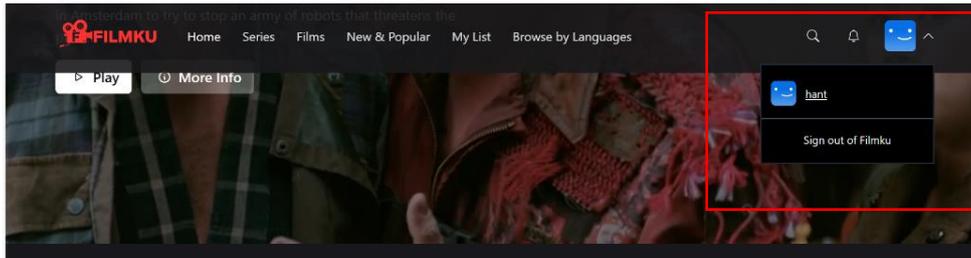
- a. Gambar dan judul film
- b. Durasi film
- c. Kategori film
- d. Deskripsi singkat film



Gambar 16 Tampilan Pemutaran Film

Halaman pemutaran film di situs web Filmku dirancang untuk memberikan pengalaman yang mudah dan nyaman bagi pengguna saat menonton film atau acara televisi. Berikut adalah beberapa elemen umum yang mungkin Anda temui di halaman pemutaran film filmku:

- a. Progres bar
- b. Tombol Paus/Play
- c. Tombol Volume
- d. Tombol Full layar
- e. Opsi lain (Unduh, Kecepatan pemutaran, Gambar dalam gambar)



Gambar 17 Tampilan detail Profil

Halaman profil adalah halaman yang menampilkan informasi pribadi tentang seorang pengguna. Tujuannya adalah memberikan gambaran umum tentang identitas, minat, dan aktivitas pengguna tersebut.

#### 4. Implementasi SSR CSR dan SSG

Setelah perancangan *Website*, peneliti mengimplementasikan SSR, CSR, dan SSG pada *Website* Filmku sesuai dengan desain yang telah dibuat. Implementasi ini melibatkan perubahan dalam cara data dinamis diambil dan ditampilkan. Untuk menguji perbandingan performa dari teknik *rendering* di *Next.js* peneliti memasukan tehnik ini ke 3 halaman yang berbeda, selanjutnya menguji satu persatu tehnik yang digunakan. Hal ini dilakukan agar mendapat nilai dan data yang akurat, dan memungkinkan untuk menemukan keunggulan serta kelemahan masing-masing tehnik *rendering* saat digunakan dalam beberapa kasus halaman yang berbeda-beda.

##### a. SSR

Berikut ini adalah implementasi SSR beserta keteranganya pada Halaman Login, Register, Halaman Utama, dan Halaman Profil

```
.....  
export async function getServerSideProps(context:  
Next.jsPageContext) {  
  const session = await getSession(context);  
  
  if (session) {  
    return {  
      redirect: {  
        destination: '/',  
        permanent: false,  
      }  
    }  
  }  
}
```

```

    }
  }

  return {
    props: {}
  }
}

const Auth = () => {
  .....
}

export default Auth;

```

Tabel 10 Detail Coding Implementasi SSR pada Halaman Login dan Register

No	Code	Keterangan
1	Fungsi getServerSideProps	Ini adalah fungsi yang digunakan untuk <i>Server-side Rendering</i> (SSR). Fungsi ini akan dipanggil saat halaman diakses, dan server akan menggunakan ini untuk memeriksa status sesi <i>user</i> . Fungsi <code>`getSession(context)`</code> digunakan untuk mencoba mendapatkan sesi <i>user</i> saat ini dengan menggunakan informasi <code>`context`</code> . Jika sesi <i>user</i> sudah ada ( <i>user</i> sudah login), <i>user</i> akan diarahkan kembali ke halaman utama dengan opsi permanen <code>`false`</code> . Ini membantu menghindari akses yang tidak sah atau ganda ke halaman autentikasi.

Selanjutnya penerapan SSR pada halaman Profile dapat dilihat pada kode di bawah ini:

```

.....
export async function getServerSideProps(context:
Next.jsPageContext) {
  const session = await getSession(context);

  if (!session) {
    return {
      redirect: {
        destination: '/auth',
        permanent: false,
      }
    }
  }

  return {
    props: {}
  }
}

const App = () => {
  .....
export default App;

```

Tabel 11 Detail Coding Implementasi SSR pada Halaman Profile

No	Code	Keterangan
1	Fungsi getServerSideProps	Fungsi ini digunakan untuk menggunakan metode SSR dalam Halaman Profile serta memeriksa apakah ada sesi <i>user</i> yang terautentikasi dengan menggunakan <code>getSession</code> dari <i>Next.js-auth/Next.js</i> . Jika <i>user</i> sudah terautentikasi (sesi ada), akan mengarahkan mereka kembali ke halaman utama dengan mengembalikan objek <code>redirect</code> . Jika <i>user</i> belum terautentikasi, halaman ini akan di-render dengan properti kosong.
3	Komponen App	Halaman utama aplikasi yang menampilkan daftar profil <i>user</i> yang tersedia. Menggunakan hook <code>useCurrentUser</code> untuk mengambil data profil <i>user</i> saat ini. Menggunakan hook

No	Code	Keterangan
		useRouter untuk mengelola perubahan rute dan fungsi selectProfile untuk mengarahkan <i>user</i> kembali ke halaman utama saat mereka memilih profil.

Selanjutnya penerapan SSR pada halaman Utama dapat dilihat pada kode di bawah ini:

```

.....
export async function getServerSideProps(context:
Next.jsPageContext) {
  const session = await getSession(context);

  if (!session) {
    return {
      redirect: {
        destination: '/auth',
        permanent: false,
      }
    }
  }

  return {
    props: {}
  }
}

const Home = () => {
.....
export default Home;

```

Tabel 12 Detail Coding Implementasi SSR pada Halaman Utama

No	Code	Keterangan
1	Fungsi getServerSideProps	Fungsi ini digunakan untuk menggunakan metode SSR pada halaman Utama serta memeriksa apakah ada sesi <i>user</i> yang terautentikasi dengan menggunakan <i>getSession</i> dari <i>Next.js-auth/Next.js</i> . Jika <i>user</i> sudah terautentikasi (sesi ada), akan mengarahkan mereka kembali ke

No	Code	Keterangan
		halaman utama dengan mengembalikan objek <code>redirect</code> . Jika <i>user</i> belum terautentikasi, halaman ini akan di-render dengan properti kosong.
2	Komponen Home	Ini adalah halaman utama aplikasi yang menampilkan berbagai komponen seperti <code>Navbar</code> , <code>Billboard</code> , dan daftar film dari komponen <code>MovieList</code> . Menggunakan beberapa hook kustom seperti <code>useMovieList</code> , <code>useFavorites</code> , dan <code>useInfoModalStore</code> untuk mengambil data film, daftar favorit, dan mengelola modal informasi. Memuat komponen <code>InfoModal</code> untuk menampilkan modal informasi jika <code>isOpen</code> bernilai <code>true</code> .

b. CSR

Berikut ini adalah implementasi CSR beserta Keterangannya pada Halaman Login dan Register.

```

.....
const Home = () => {
  const [movies, setMovies] = useState([]);
  const [favorites, setFavorites] = useState([]);
  const { isOpen, closeModal } = useInfoModalStore();

  useEffect(() => {

    return (
      .....
    );
  });

  export default Home;

```

Tabel 13 Detail Coding Implementasi CSR pada Halaman Utama

No	Code	Keterangan
1	UseEffect	Menggunakan <code>useEffect</code> untuk menggunakan metode CSR pada halaman Utama serta menjalankan kode pengambilan data setelah komponen di-render. Kode ini melakukan dua permintaan ke API untuk mengambil data film dan daftar favorit menggunakan fungsi <code>fetch</code> . Data yang diambil akan mengisi state <code>movies</code> dan <code>favorites</code> saat respons diterima

c. SSG

Berikut ini adalah implementasi SSR beserta Keterangannya pada Halaman Login dan Register.

```

.....
export async function getStaticProps(context:
Next.jsPageContext) {
  const session = await getSession(context);

  if (session) {
    return {
      redirect: {
        destination: '/',
        permanent: false,
      }
    }
  }

  return {
    props: {}
  }
}

const Auth = () => {
  .....
export default Auth;

```

Tabel 14 Detail Coding Implementasi SSG Halaman Login dan Register

No	Code	Keterangan
1	Fungsi getStaticProps	Menggunakan <code>getStaticProps</code> yang berfungsi untuk <i>Static Site Generation</i> (SSG). Ini digunakan untuk menghasilkan halaman statis pada waktu kompilasi. Dalam kasus ini, kode hanya mengecek apakah ada sesi <i>user</i> yang terautentikasi dengan menggunakan <code>getSession</code> dari <i>Next.js-auth/Next.js</i> .

Selanjutnya penerapan SSG pada halaman Profile dapat dilihat pada kode di bawah ini:

```

.....
export async function getStaticProps(context:
Next.jsPageContext) {
  const session = await getSession(context);

  if (session) {
    return {
      redirect: {
        destination: '/',
        permanent: false,
      }
    }
  }

  return {
    props: {}
  }
}
return (
.....

const App = () => {
.....
export default App;

```

Tabel 15 Detail Coding Implementasi SSG pada Halaman Profile

No	Code	Keterangan
1	Fungsi getStaticProps	Fungsi ini digunakan untuk menggunakan metode SSG pada halaman profile dan memeriksa apakah ada sesi <i>user</i> yang terautentikasi dengan menggunakan getSession dari <i>Next.js-auth/Next.js</i> . Jika <i>user</i> sudah terautentikasi (sesi ada), akan mengarahkan mereka kembali ke halaman utama dengan mengembalikan objek <i>redirect</i> . Jika <i>user</i> belum terautentikasi, halaman ini akan di-render dengan properti kosong.
3	Komponen App	Halaman utama aplikasi yang menampilkan daftar profil <i>user</i> yang tersedia. Menggunakan hook <i>useCurrentUser</i> untuk mengambil data profil <i>user</i> saat ini.

Selanjutnya penerapan SSG pada halaman Utama dapat dilihat pada kode di bawah ini:

```

.....
export async function getStaticProps(context:
Next.jsPageContext) {
  const session = await getSession(context);
  if (!session) {
    return {
      redirect: {
        destination: '/auth',
        permanent: false,
      }
    }
  }

  return {
    props: {}
  }
}
const Home = () => {
.....
export default Home;

```

Tabel 16 Detail Coding Implementasi SSG pada Halaman Utama

No	Code	Keterangan
1	Fungsi <code>getStaticProps</code>	Fungsi ini digunakan untuk menggunakan Metode SSG pada halaman ini kemudian memeriksa apakah ada sesi <i>user</i> yang terautentikasi dengan menggunakan <code>getSession</code> dari <i>Next.js-auth/Next.js</i> . Jika <i>user</i> sudah terautentikasi (sesi ada), akan mengarahkan mereka kembali ke halaman utama dengan mengembalikan objek <code>redirect</code> . Jika <i>user</i> belum terautentikasi, halaman ini akan di-render dengan properti kosong.
2	Komponen Home	Ini adalah halaman utama aplikasi yang menampilkan berbagai komponen seperti Navbar, Billboard, dan daftar film dari komponen <code>MovieList</code> . Menggunakan beberapa hook kustom seperti <code>useMovieList</code> , <code>useFavorites</code> , dan <code>useInfoModalStore</code> untuk mengambil data film, daftar favorit, dan mengelola modal informasi. Memuat komponen <code>InfoModal</code> untuk menampilkan modal informasi jika <code>isOpen</code> bernilai <code>true</code> .

5. Pengambilan data Implementasi SSR, CSR, dan SSG  
 Pada tahap ini, data kinerja *Website* Filmku saat implementasi *Server-side Rendering* (SSR), *Client-side Rendering* (CSR), dan *Static-Side Generator* (SSG) dikumpulkan. Pengambilan data ini dilakukan pada 3 halaman yang berbeda kasus di *Website*. Data ini merupakan yang akan menjadi perbandingan kinerja masing-masing tehnik *rendering* pada kasus-kasus tertentu. Pada browser halaman akan di reload beberapa kali sampai data menunjukkan grafik yang stabil dan kemudian diambil rata-rata nya, hal ini dilakukan untuk mendapatkan data yang valid.
6. Perbandingan Performa

Pada tahap ini peneliti akan membandingkan data yang telah di dapat kemudian di bandingkan dengan metode-metode *rendering* lainnya untuk mendapatkan hasil kesimpulan pada penelitian yang dilakukan.

#### 7. Kesimpulan

Dalam tahap ini, peneliti menganalisis data yang telah dikumpulkan pada pengujian SSR, CSR, dan SSG. Kesimpulan ditarik berdasarkan perbandingan kinerja *Website* Filmku ketika implementasi SSR, CSR, dan SSG pada berbagai kasus, serta dampak positif atau negatif yang mungkin muncul. Hasil penelitian digunakan untuk memberikan wawasan tentang efektivitas perubahan yang dilakukan pada *Website* Filmku.

#### D. Analisis Data

Penelitian ini menggunakan metode pengujian Loading Time halaman, Loading Time adalah waktu yang diperlukan browser untuk menampilkan sepenuhnya informasi situs web yang diminta user [16]. Peneliti memilih metode ini karena mengacu pada kualitas kepuasan user secara experience, Kualitas suatu web dikatakan baik jika mampu dengan cepat merespon setiap permintaan URL (Uniform *Resource* Locator) user dan mampu meminimalisir terjadinya kesalahan. Faktor kestabilan web sangat penting untuk mempercepat proses loading web. Performa dalam membuka sebuah website menjadi salah satu faktor yang akan membuat pengguna nyaman dalam menggunakan layanan [21].

Pengujian ini dilakukan pada *Browser* Chrome versi 118.0.5993.90. Halaman yang digunakan pada penelitian ini adalah halaman Auth, Profile, dan Halaman Utama. Pada browser halaman akan di reload beberapa kali sampai data menunjukkan grafik yang stabil, Hal ini dikarenakan Untuk mendapatkan hasil yang valid. Walaupun dilakukan pada metode yang berbeda, namun dalam pengujiannya tetap menggunakan simulasi Ubuntu Server versi 22.04 dengan Dual Processor dan Dual core. Kemudian menggunakan Putty untuk remote SSH hal ini untuk memudahkan peneliti dalam menjalankan website. Website yang diuji dalam Localhost, hal ini di pilih agar jaringan yang sering berubah sewaktu-waktu tidak menjadi penghambat dalam pengujian, serta untuk memaksimalkan dalam menguji metode rendering. Untuk skenario pengujian dapat dilihat pada tabel dibawah ini.

Tabel 17 Skenario Pengujian

No	Metode Rendering	Halaman	Matrik Pengukuran	Jumlah reload
1	Server-Side Rendering	Auth	Request, Resource,	30x
		Profile	DOM Content	30x
		Utama	Loaded, Load, Finish	20x
2	Client-Side Rendering	Auth	Request, Resource,	30x
		Profile	DOM Content	30x
		Utama	Loaded, Load, Finish	20x

No	Metode Rendering	Halaman	Matrik Pengukuran	Jumlah reload
3	Static Site Generator	Auth	Request, Resource,	30x
		Profile	DOM Content	30x
		Utama	Loaded, Load, Finish	20x

Kecepatan waktu Loading dari browser diukur dalam satuan detik, kemudian di ambil rata-rata dari setiap metode yang digunakan. Perbandingan waktu setiap metode kemudian di simpulkan dalam bentuk persen (%), hal ini di gunakan untuk memudahkan pembaca dalam mengetahui perbandingan masing-masing metode. Adapun untuk mendapatkan perbandingan dalam bentuk persen (%) menggunakan rumus berikut [21].

$$\frac{\text{Nilai A} - \text{Nilai B}}{\text{Nilai C}} * 100\% \quad (1)$$

Ket:

Nilai A = Nilai yang dibandingkan

Nilai B = Nilai Pembanding

Nilai C = Nilai Terbesar dari Nilai A & Nilai B

Pengujian kecepatan waktu loading halaman pada halaman web memanfaatkan fitur panel jaringan di *DevTools browser Chrome*. Pengujian dilakukan pada setiap halaman website, kemudian kecepatan waktu loading antara program versi 1, kode program versi 2, dan kode versi 3 dibandingkan. Untuk Matrik pengukuran dapat dilihat pada Tabel 18.

Tabel 18 Matrik Pengukuran

No	Matrik Pengukuran	Keterangan
1	Request	Jumlah Request yang ada pada halaman yang di render.
2	<i>Resource</i>	Jumlah data yang ada pada halaman yang di render
3	DOM Content Loaded	DOM Ready atau DOM Content Loaded adalah event yang akan terpicu ketika telah berhasil membaca keseluruhan elemen DOM, yaitu dari bagian awal sampai akhir halaman web, atau lebih sering terlihat pada

No	Matrik Pengukuran	Keterangan
		animasi loading dinavbar bagian atas browser.
4	Load	Waktu yang dibutuhkan halaman untuk Memuat halaman saja
5	Finish	Waktu yang dibutuhkan untuk halaman menyelesaikan semua rendering, mulai dari halaman, conten, video, dan berbagai informasi lainnya.

Matrik ini mengacu pada respon dan data dari semua traksaksi yang dilakukan sesuai dengan yang terdapat pada Dev Tools pada *browser* Chrome, Matrik ini digunakan untuk melihat berapa efektif dan seberapa cepat website tersebut berinteraksi, dan menampilkan data secara keseluruhan kepada user. Matrik yang terpenting adalah finish, ini dikarenakan finish merupakan waktu untuk metode rendering menyelesaikan proses rendering halamannya. Setelah data diambil per halaman akan di beri rangking untuk setiap metode rendering, hal ini digunakan untuk penilaian akhir supaya lebih mudah untuk membandingkan masing masing metode rendering. Penilaian dari setiap rangking dapat dilihat pada tabel dibawah ini.

Tabel 19 Nilai Rangking setiap halaman

Rangking	Nilai	Persentase Halaman
1	3	10%
2	2	8%
3	1	82%

Persentase di setiap halaman didapatkan dari jumlah *resource* yang ada dalam halaman tersebut, semakin besar *resource* yang dimiliki halaman maka semakin besar presentase serta Nilai yang terdapat pada halaman tersebut. Rumus yang digunakan dalam mengambil persentase Halaman adalah sebagai berikut.

$$\text{Nilai Persentase halaman} = \frac{\text{Nilai } Y}{\text{Nilai } Z} * 100\% \quad (2)$$

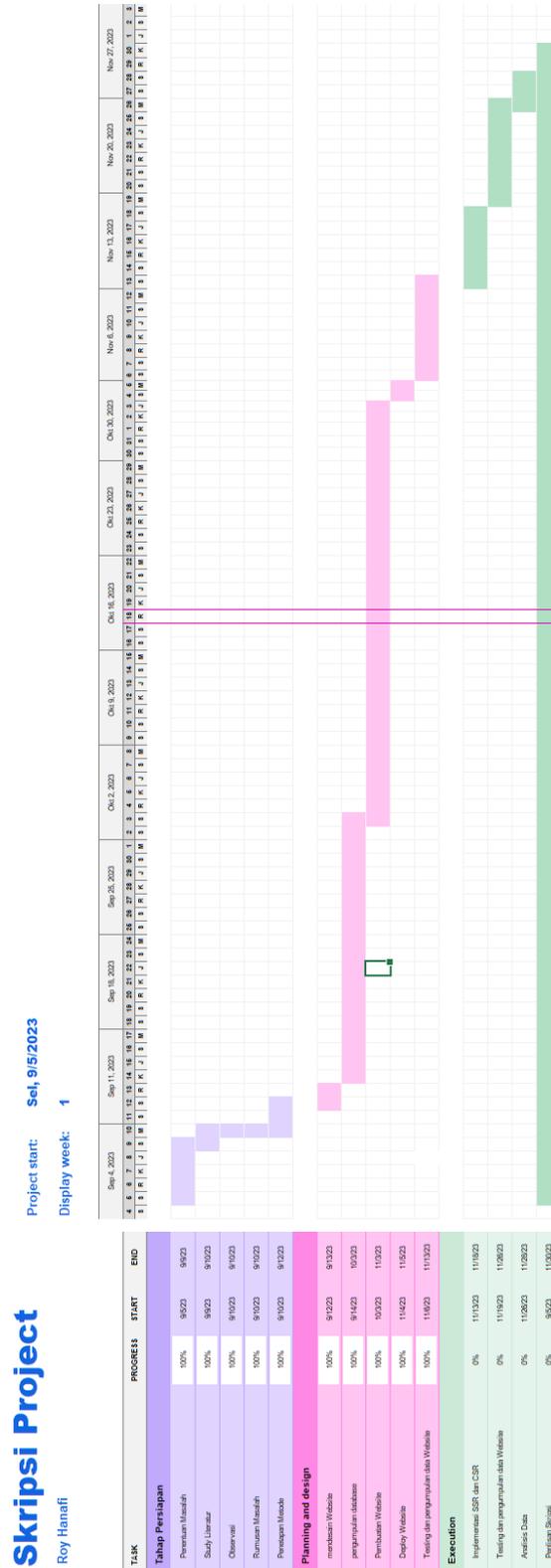
Ket:

Nilai Y = Jumlah resource halaman

Nilai Z = Jumlah resource semua halaman

## E. Jadwal Penelitian

Jadwal penelitian secara rinci dapat dilihat pada tabel microsoft project berikut:



Gambar 18 Jadwal Penelitian