**BAB IV**

**PEMBUATAN PROGRAM DAN UJICOBA**

1. **Implementasi**

Pada bab sebelumnya telah dilakukan rancangan alur aplikasi dan rancangan tampilan aplikasi. Maka, dilanjutkan dengan pengujicobaan aplikasi. Aplikasi ini memuat film yang sedang tayang, durasi film dan juga jadwal tayang film dalam satu aplikasi.

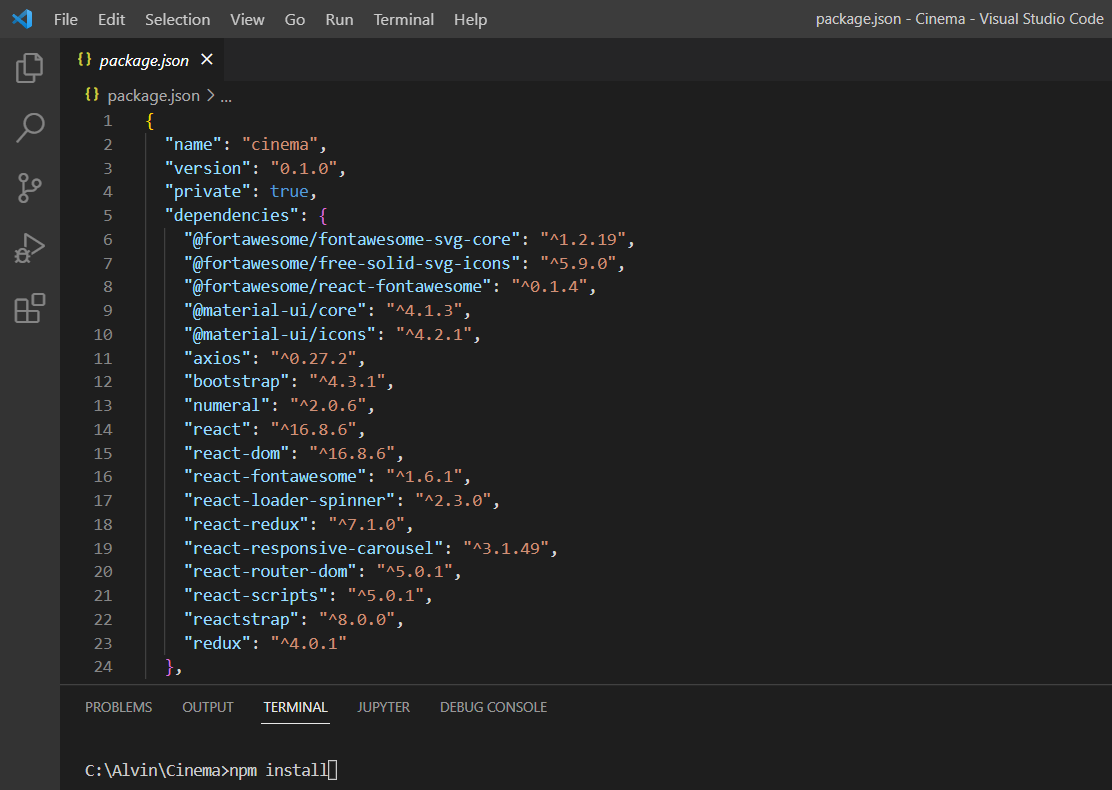
1. **Pembuatan Aplikasi**

Pada tahap ini akan dijabarkan langkah-langkah dalam pembuatan Aplikasi Cinema Seat Reservation berbasis website. Tahapan pembuatan aplikasi ini dimulai dengan memuat beberapa package dependency untuk diinstall, kemudian pembuatan halaman aplikasi seperti tampilan halaman register, halaman login, homepage admin, menu manage movie, homepage user, menu movies list, movie details, buy ticket, order seat, cart, checkout dan history purchase. Aplikasi ini dibuat dengan menggunakan library React sebagai frontend dan Node sebagai backend.

1. **Install Dependency**

Terdapat beberapa langkah yang harus dilakukan dalam melakukan install dependency, diantaranya :

1. Buka text editor Visual Studio Code.
2. Buat folder baru bernama Cinema.
3. Change directory ke dalam folder Cinema.
4. Buat file bernama package yang berektensi .json.
5. Masukkan dependecy yang dibutuhkan seperti pada **gambar 4.1** kemudian buka terminal pada Visual Studio Code lalu ketikkan command npm install untuk menginstallnya agar dapat digunakan.



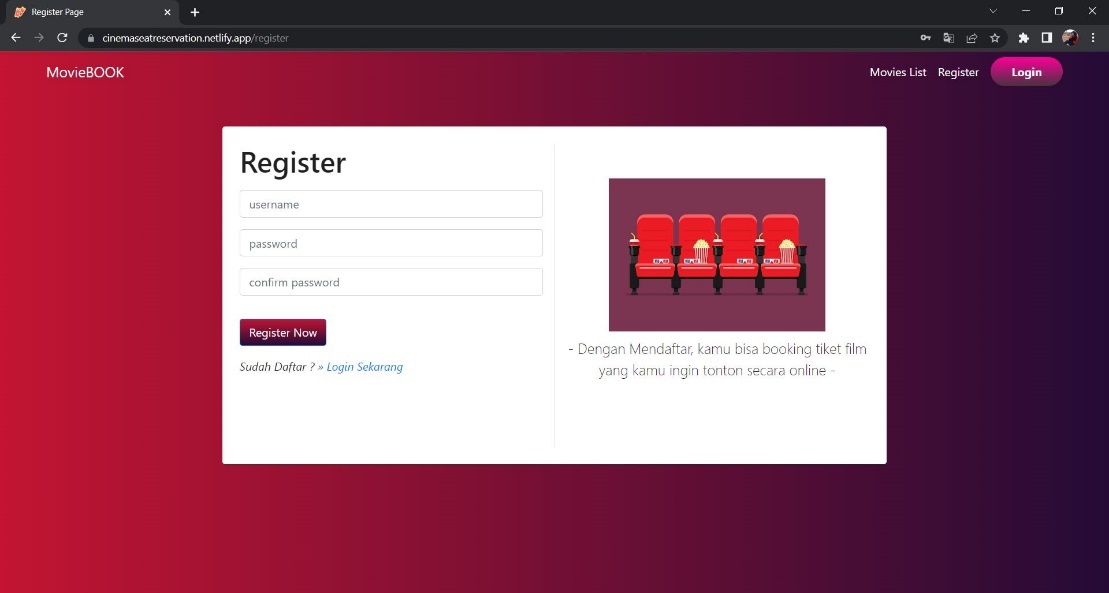
**Gambar 4.1** Dependency yang dibutuhkan

1. **Pembuatan Halaman Register**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari halaman register. Terdapat 3 variable yang didefinisikan yaitu username, password dan confirm password. Ketiga variable tersebut akan mengambil value yang diinput oleh pengguna kemudian diberi kondisi apabila username, password atau confirm password tidak diisi, maka diberi peringatan bahwa semua data harus diisi. Apabila inputan password dan confirm password tidak sama, maka diberi peringatan bahwa password dan confirm password harus sama. Jika semua sudah sesuai namun username telah terpakai, maka diberi peringatan bahwa username sudah ada. Kondisi terakhir yaitu apabila username tidak ada sebelumnya dan password sudah sama dengan confirm password, maka user telah berhasil melakukan registrasi yang kemudian langsung me-redirect ke halaman login. Berikut ini adalah potongan program untuk pembuatan halaman register.



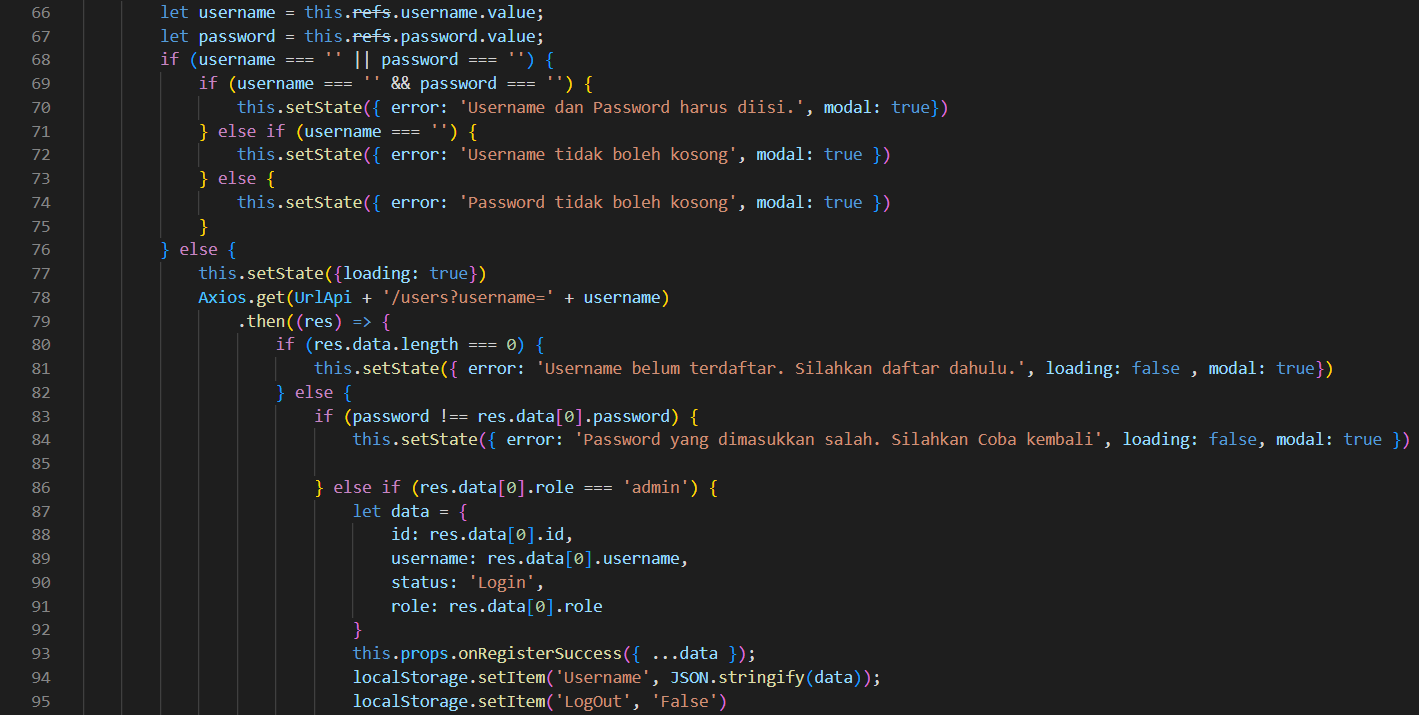
Pada **gambar 4.2** merupakan hasil implementasi pembuatan halaman register. Halaman register terdiri dari 3 input form, 1 button submit dan 1 image view.



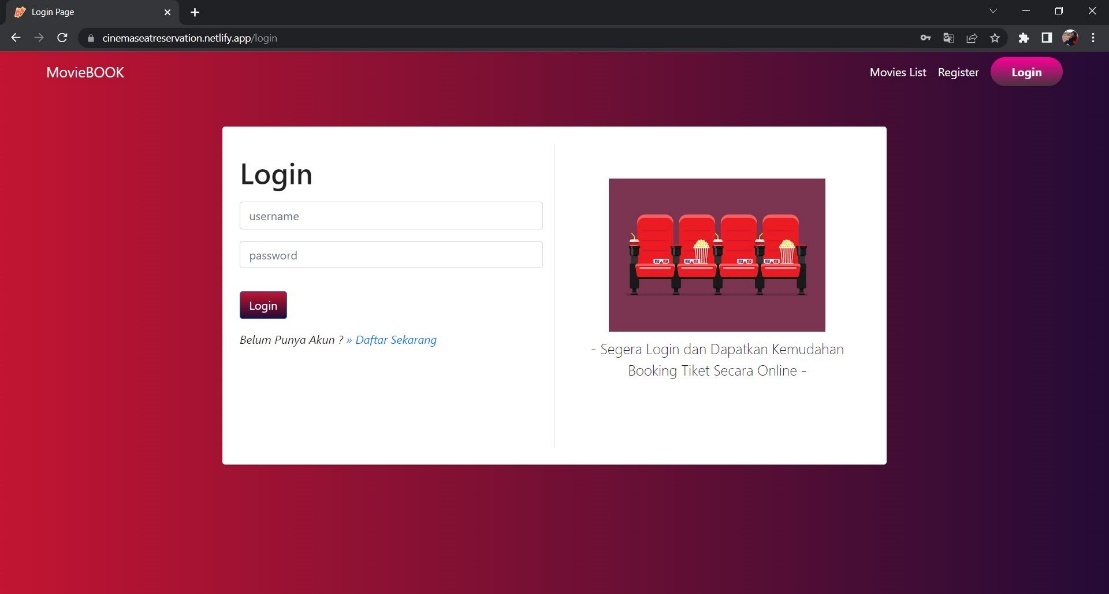
**Gambar 4.2** Tampilan Halaman Register

1. **Pembuatan Halaman Login**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari halaman login. Terdapat 2 variable yang didefinisikan yaitu username dan password. Kedua variable tersebut akan mengambil value yang diinput oleh pengguna kemudian memiliki kondisi error saat username dan password tidak diisi keduanya maupun tidak diisi salah satunya. Berikut ini adalah potongan program untuk pembuatan halaman login.



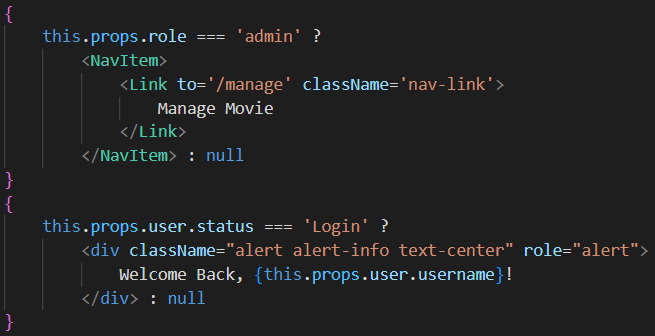
Pada **gambar 4.3** merupakan hasil implementasi pembuatan halaman register. Halaman register terdiri dari 2 input form, 1 button submit dan 1 image view.



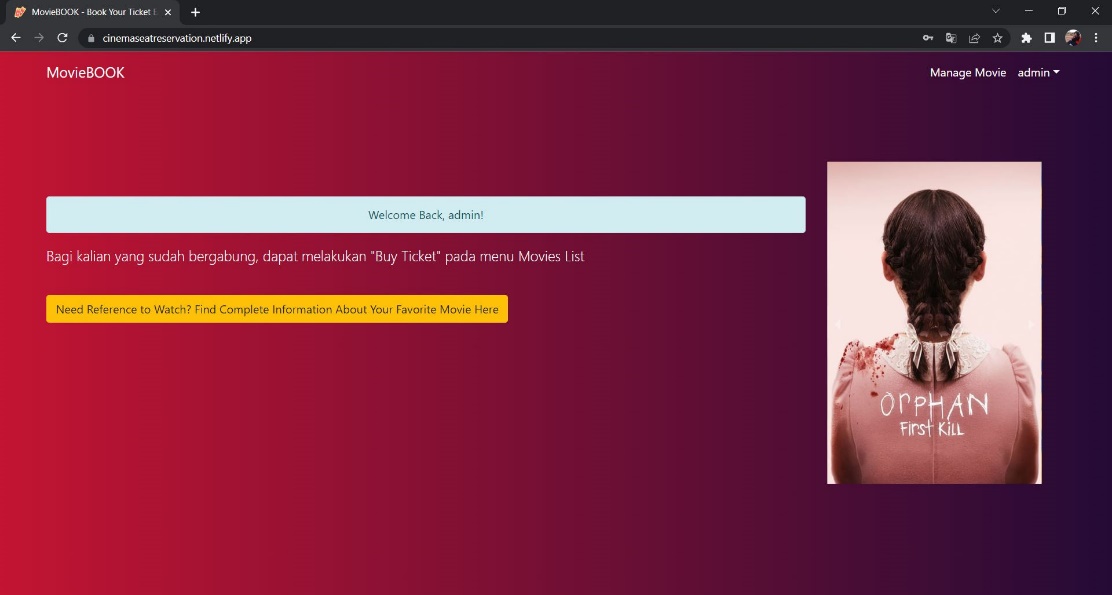
**Gambar 4.3** Tampilan Halaman Login

1. **Pembuatan Homepage Admin**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari homepage admin. Dimana jika role sama dengan admin, maka akan menampilkan menu manage movie pada navbar. Pada alert info akan me-return value dari username-nya yaitu admin. Berikut ini adalah potongan program untuk pembuatan homepage admin.



Pada **gambar 4.4** merupakan hasil implementasi pembuatan homepage admin. Homepage admin terdiri dari 2 navbar, alert info dan slider otomatis yang menampilkan beberapa poster film.



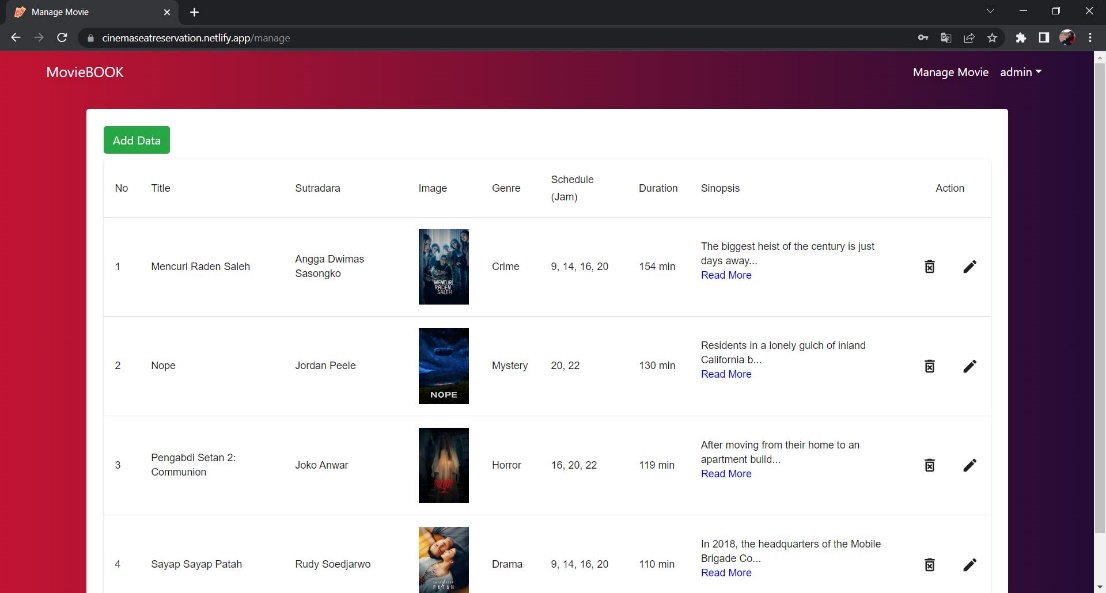
**Gambar 4.4** Tampilan Homepage Admin

1. **Pembuatan Menu Manage**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari menu manage movie. Dimana pada menu ini dapat menampilkan daftar film seperti title, director, poster, genre, playing at, runtime dan plot film yang berasal dari API. Button yang tersedia yaitu delete dan edit. Berikut ini adalah potongan program untuk pembuatan menu manage movie.



Pada **gambar 4.5** merupakan hasil implementasi pembuatan menu manage movie. Dimana admin dapat mengatur daftar film yang sedang tayang, melihat selengkapnya mengenai sinopsis film yang ada serta melakukan delete maupun update data film.



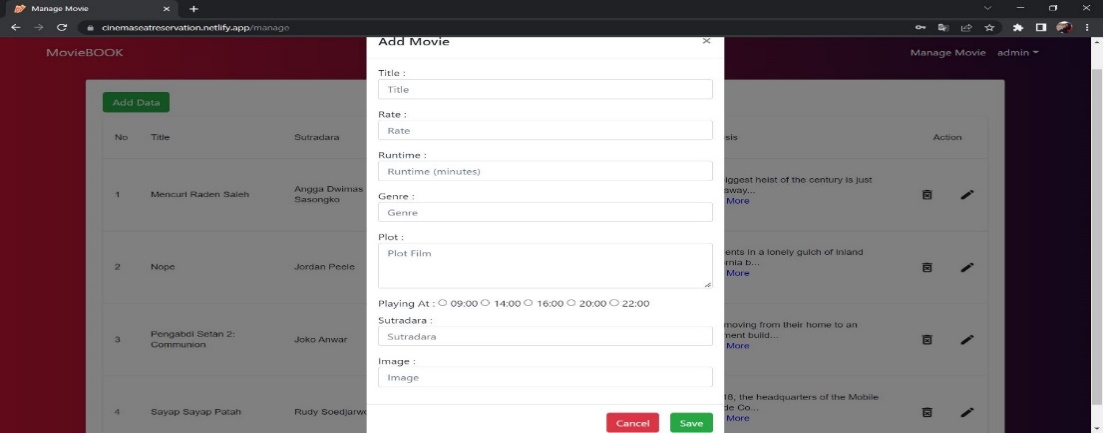
**Gambar 4.5** Tampilan Menu Manage

1. **Pembuatan Menu Add Data**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari menu add data. Dimana terdiri dari 7 variable yang didefinisikan berdasarkan inputan dari admin. Berikut ini adalah potongan program untuk pembuatan menu add data.



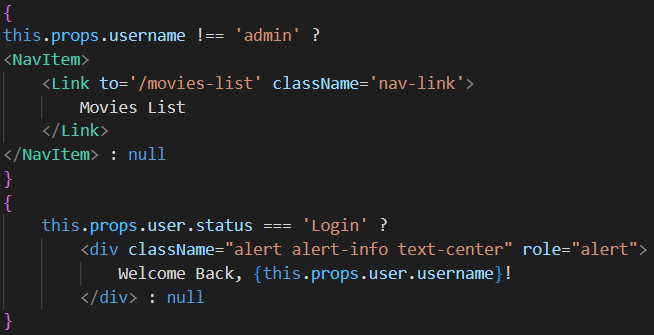
Pada **gambar 4.6** merupakan hasil implementasi pembuatan add data. Dimana admin dapat memasukkan data ke dalam form untuk di-submit yang kemudian akan ditampilkan pada menu manage movie.



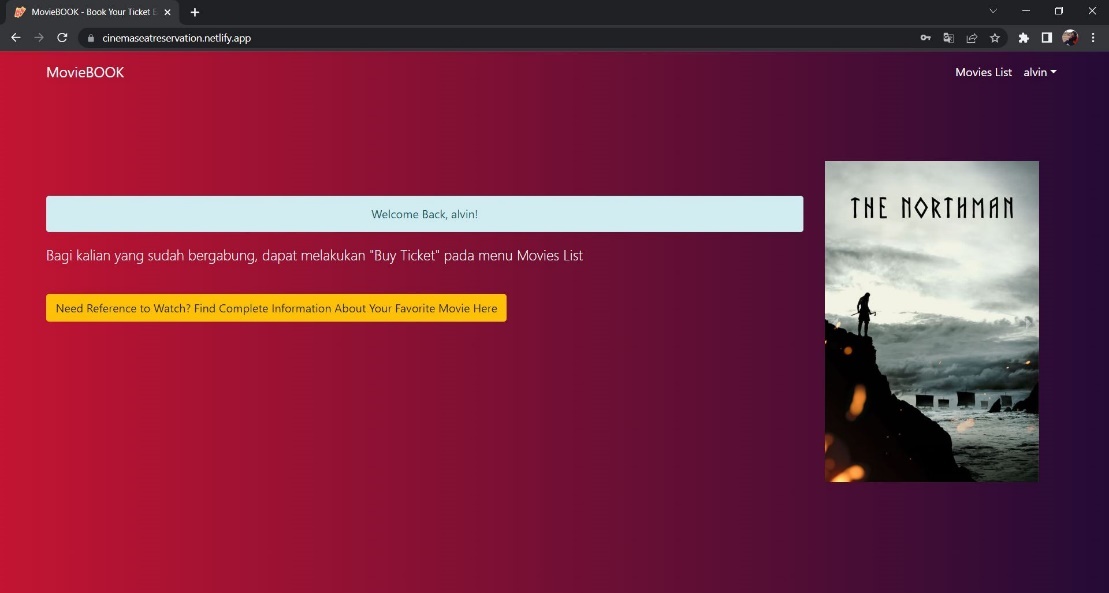
**Gambar 4.6** Tampilan Menu Add Data

1. **Pembuatan Homepage User**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari homepage user. Dimana jika role tidak sama dengan admin, maka akan menampilkan menu movies list pada navbar. Pada alert info akan me-return value dari username-nya yaitu user. Berikut ini adalah potongan program untuk pembuatan homepage user.



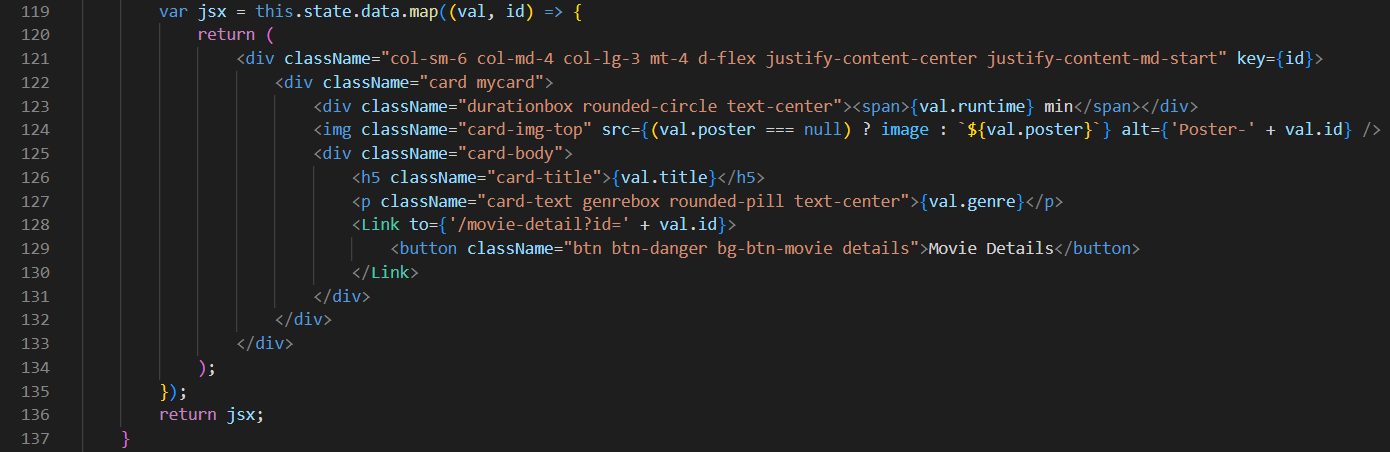
Pada **gambar 4.7** merupakan hasil implementasi pembuatan homepage user. Homepage user terdiri dari 2 navbar, alert info dan slider otomatis yang menampilkan beberapa poster film.



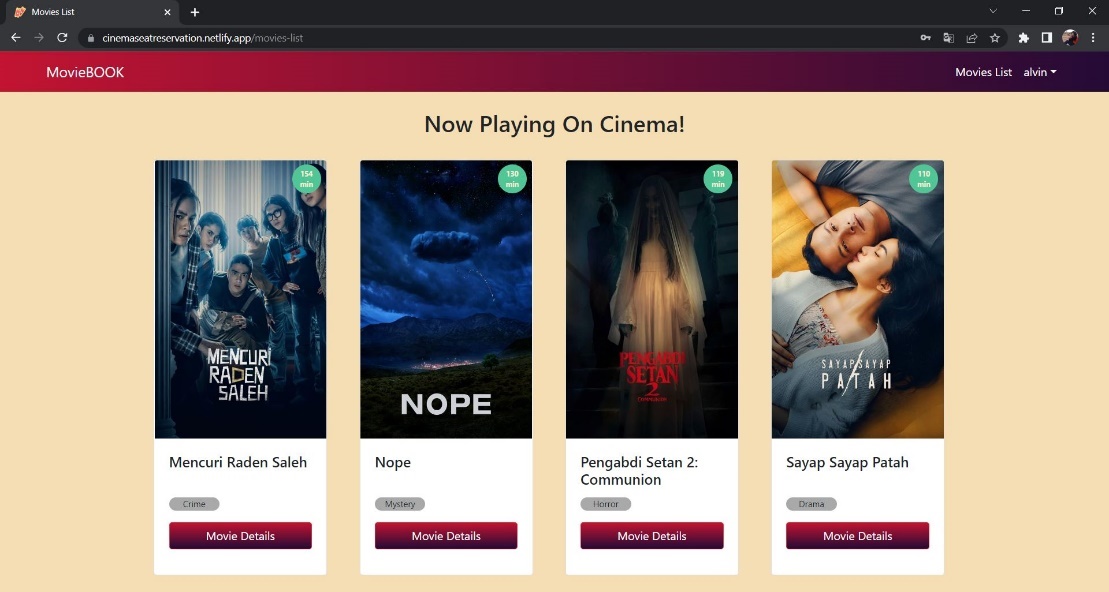
**Gambar 4.7** Tampilan Homepage User

1. **Pembuatan Menu Movies List**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari menu movies list. Dimana data pada API dilakukan mapping ke dalam class card untuk ditampilkan seperti runtime, poster, judul dan genre dari film. Serta button movie details yang mengarah ke url custom sesuai dengan id film yang akan dipilih. Berikut ini adalah potongan program untuk pembuatan menu movies list.



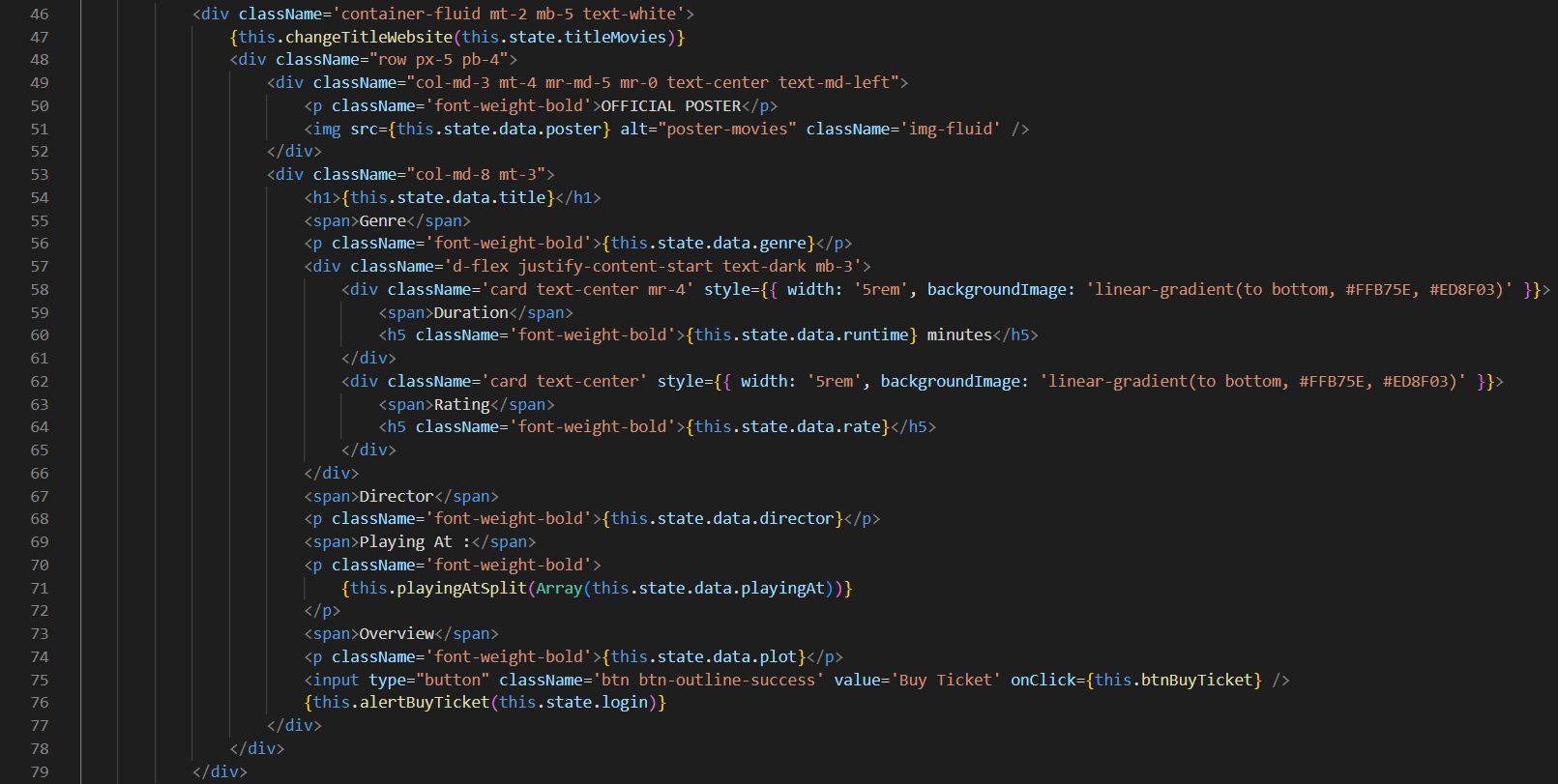
Pada **gambar 4.8** merupakan hasil implementasi pembuatan menu movies list. Menu movies list terdiri dari card component berisi film serta button movie details.



**Gambar 4.8** Tampilan Menu Movies List

1. **Pembuatan Halaman Movie Details**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari halaman movie details. Dimana pada halaman ini menampilkan informasi detail film yang dipilih sesuai pada menu sebelumnya. Terdapat opsi buy ticket untuk melakukan pembelian bagi user. Berikut ini adalah potongan program untuk pembuatan halaman movie details.



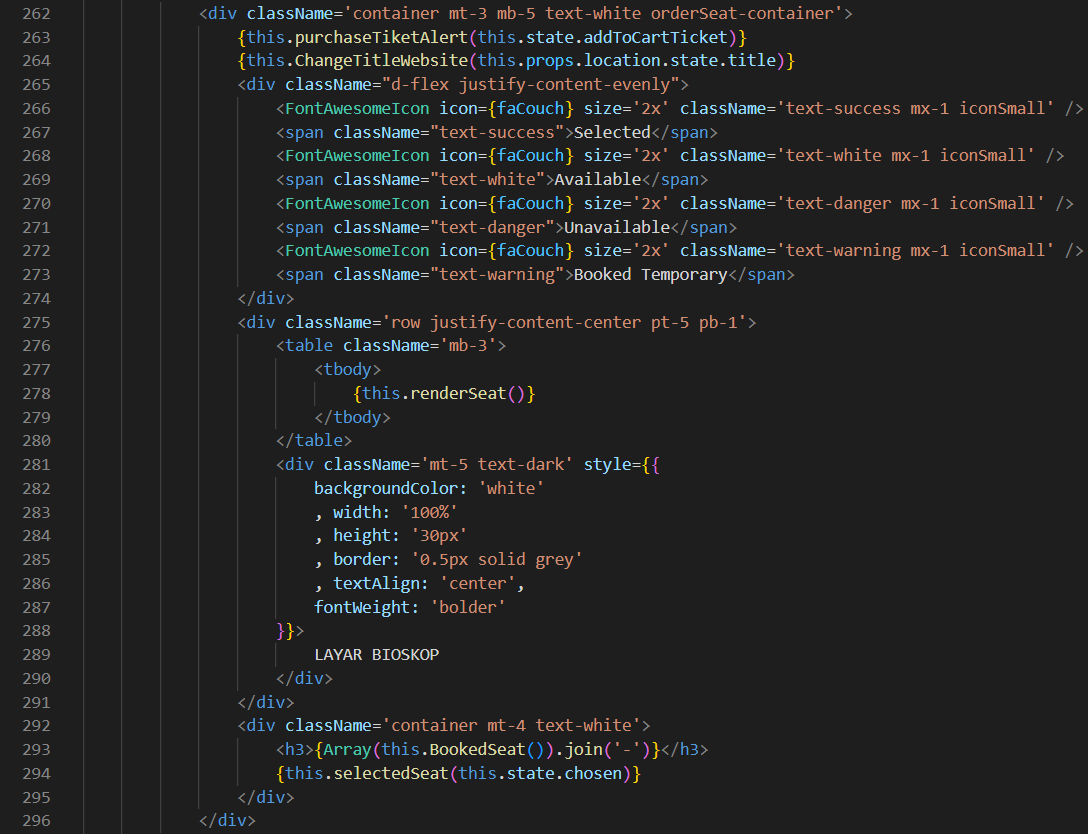
Pada **gambar 4.9** merupakan hasil implementasi pembuatan halaman movie details. Halaman movie details terdiri dari poster, judul, genre, director, playing at dan overview dari film yang dipilih.



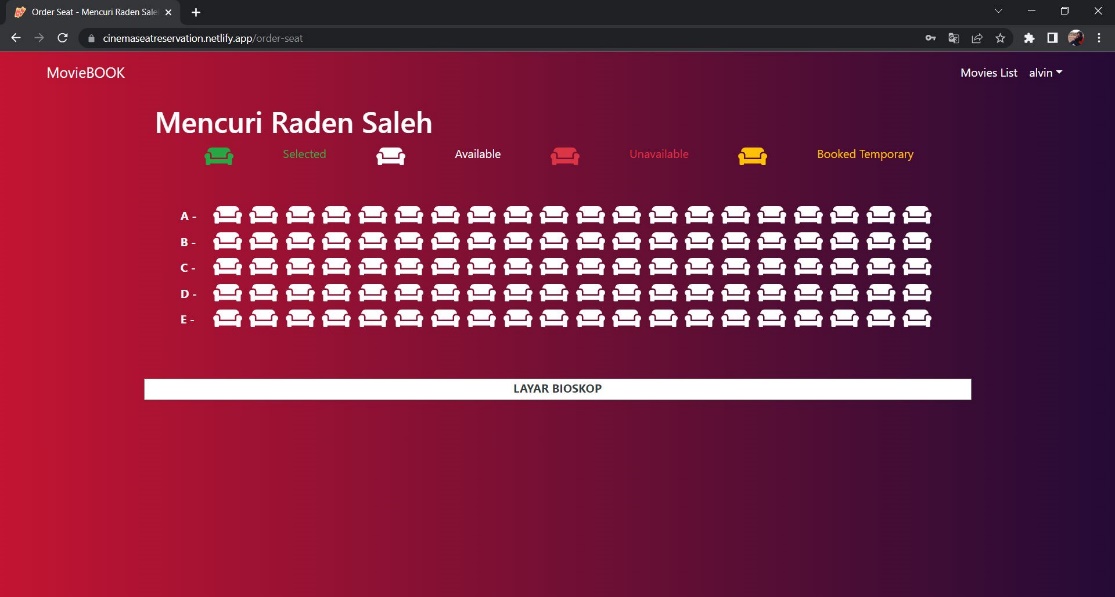
**Gambar 4.9** Tampilan Halaman Movie Details

1. **Pembuatan Halaman Order Seat**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari halaman order seat. Dimana pada halaman ini user dapat memilih kursi yang tersedia untuk dipesan. Terdapat 4 keterangan yang membantu apakah kursi sudah ditempati atau belum, yaitu selected, available, unavailable dan booked temporary. Hal lainnya memiliki penambahan rectangle untuk mengetahui layar bioskop menghadap kearah mana. Berikut ini adalah potongan program untuk pembuatan halaman order seat.

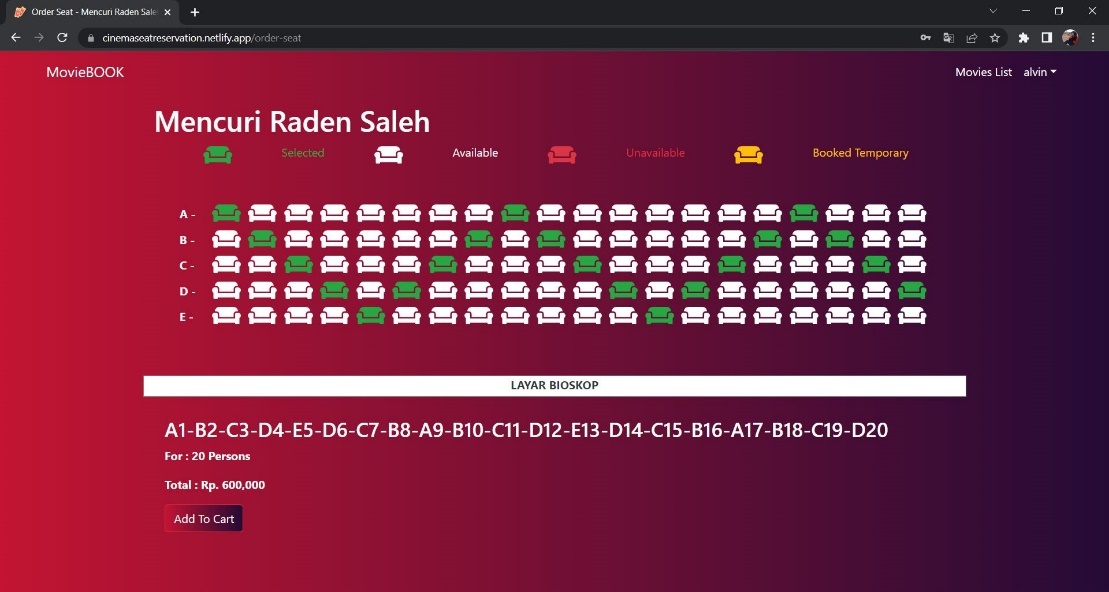


Pada **gambar 4.10** merupakan hasil implementasi pembuatan halaman order seat saat belum terpilih. Halaman order seat saat belum terpilih memiliki warna putih dan tidak ada pilihan yang dapat diproses sehingga tidak menampilkan button untuk menambahkan ke dalam cart.



**Gambar 4.10** Tampilan Halaman Order Seat Available

Pada **gambar 4.11** merupakan hasil implementasi pembuatan halaman order seat saat terpilih. Halaman order seat saat terpilih memiliki warna hijau dan memunculkan pilihan nomor kursi, jumlah yang dipilih, total harga dan button add to cart.



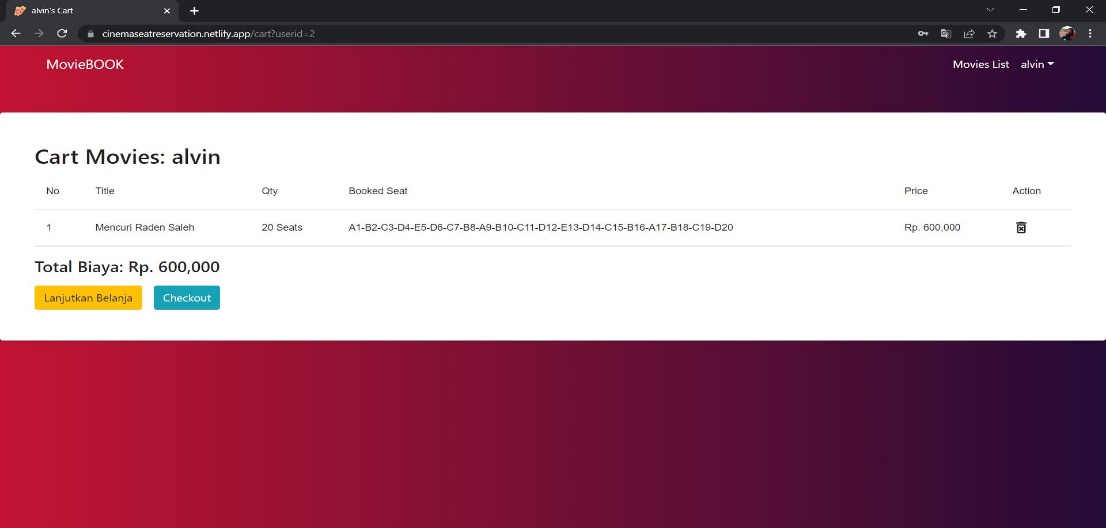
**Gambar 4.11** Tampilan Halaman Order Seat Selected

1. **Pembuatan Halaman User’s Cart**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari halaman user’s cart. Saat user memilih add to cart pada halaman sebelumnya, maka akan beralih ke halaman ini untuk melihat pesanan sebelum melakukan checkout. User pun dapat melanjutkan belanja kembali. Berikut ini adalah potongan program untuk pembuatan halaman user’s cart.



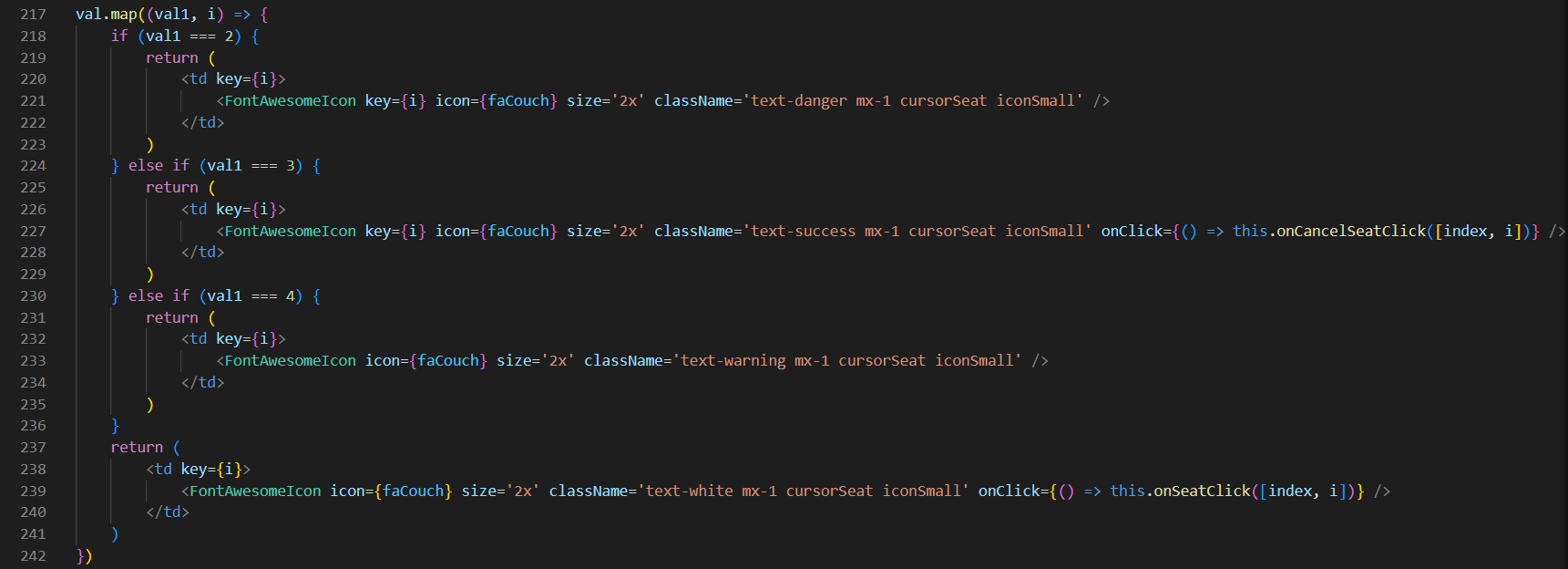
Pada **gambar 4.12** merupakan hasil implementasi pembuatan halaman user’s cart. Halaman user’s cart terdiri dari sekumpulan pesanan yang belum dilakukan pembayaran oleh user.



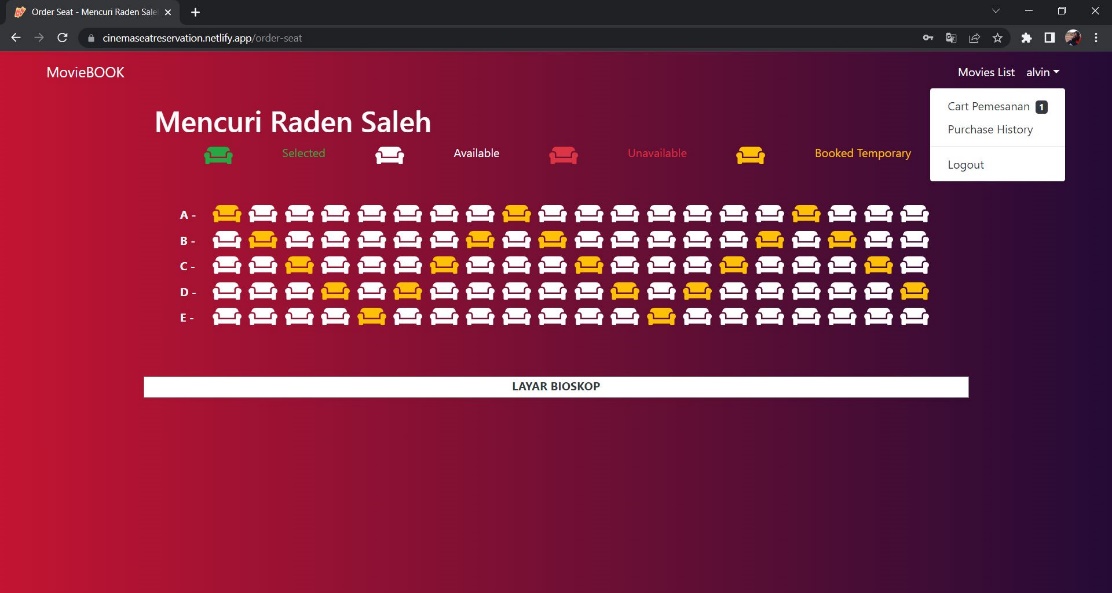
**Gambar 4.12** Tampilan Halaman User’s Cart

1. **Pembuatan Halaman Order Seat Booked Temporary**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari halaman order seat booked temporary. Saat user melanjutkan belanja dan memilih film yang sama seperti sebelumnya, maka akan terdapat tanda kursi booked temporary. Berikut ini adalah potongan program untuk pembuatan halaman order seat temporary.



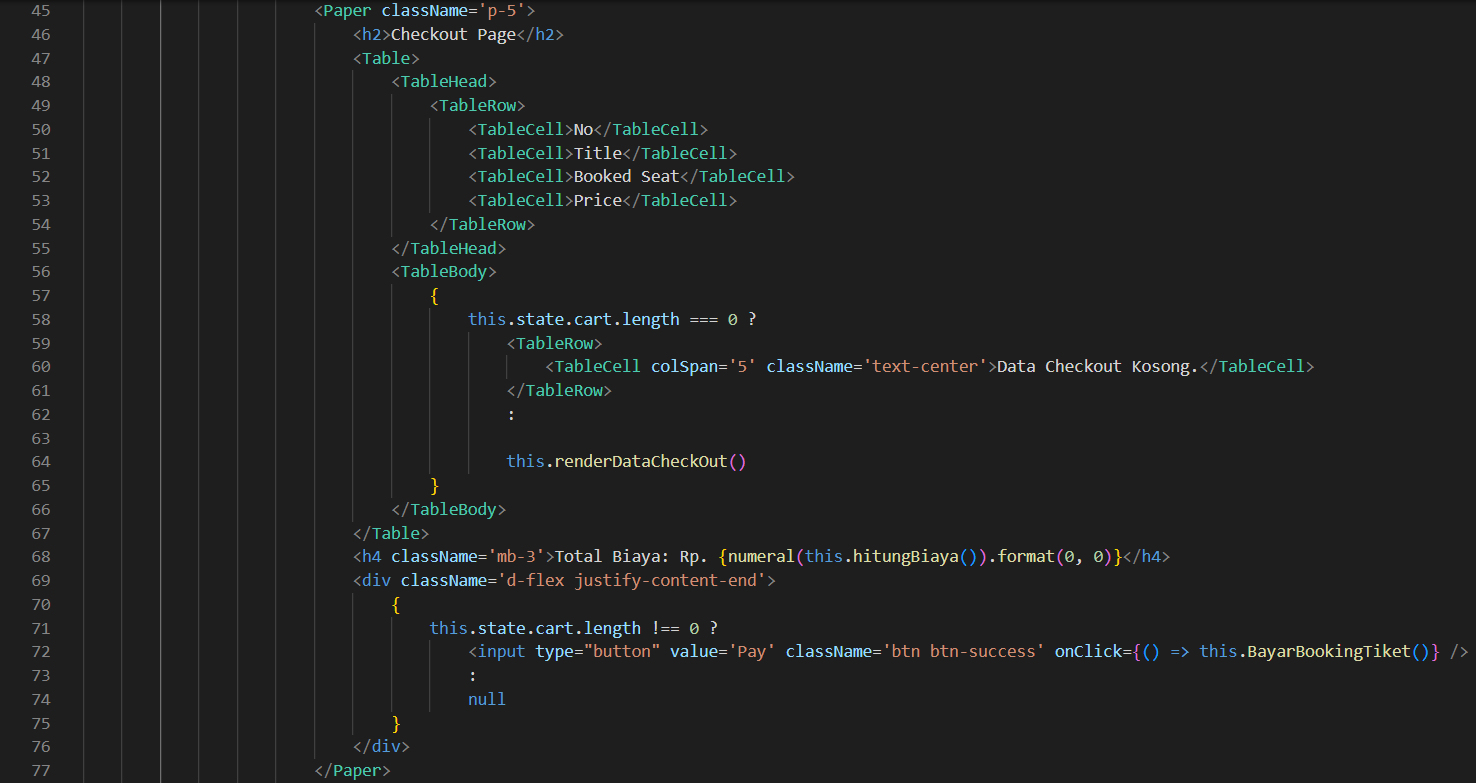
Pada **gambar 4.12** merupakan hasil implementasi pembuatan halaman order seat booked temporary. Halaman order seat booked temporary memiliki tanda warna kuning yang berarti pesanan sudah berada pada cart user namun pesanan tersebut belum dilakukan checkout.



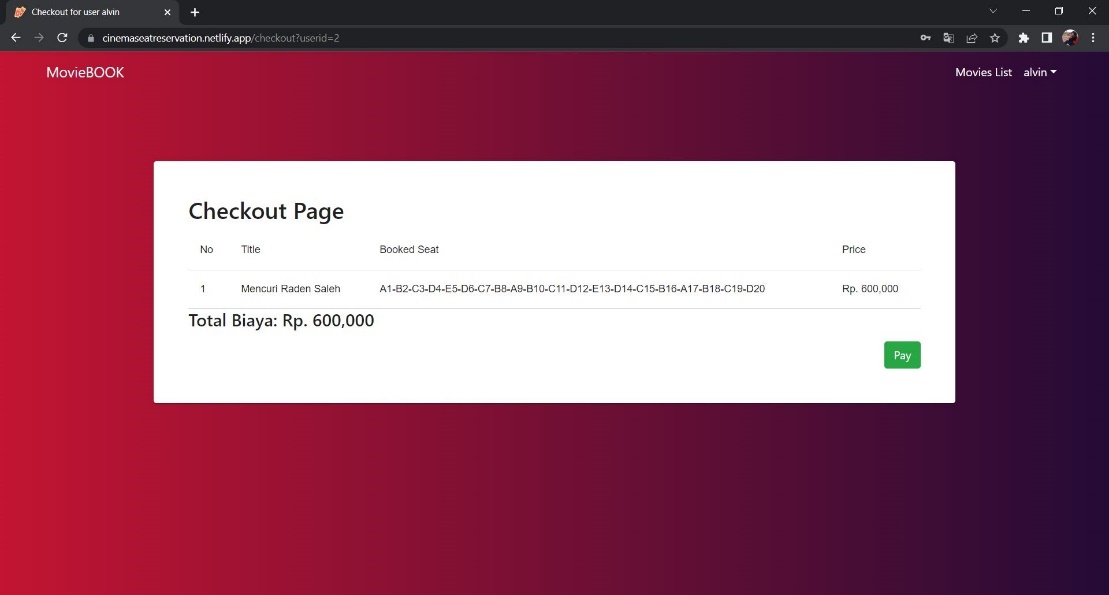
**Gambar 4.12** Tampilan Halaman Order Seat Booked Temporary

1. **Pembuatan Halaman Checkout**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari halaman checkout. Data yang ditampilkan berdasarkan dari pesanan user’s cart untuk dilakukan pembayaran. Berikut ini adalah potongan program untuk pembuatan halaman checkout.



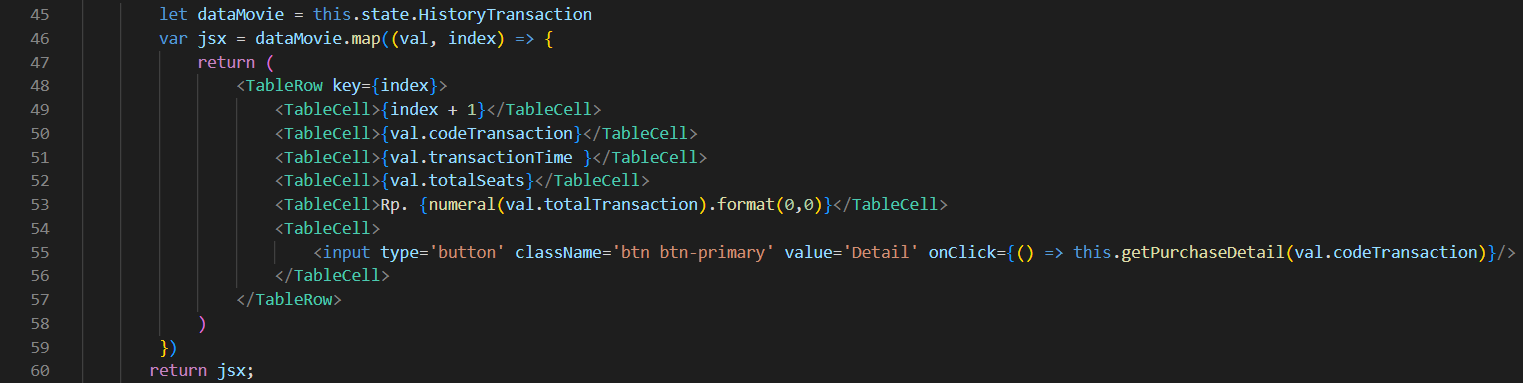
Pada **gambar 4.13** merupakan hasil implementasi pembuatan halaman checkout. Halaman ini memiliki button pay yang digunakan untuk melakukan pembayaran.



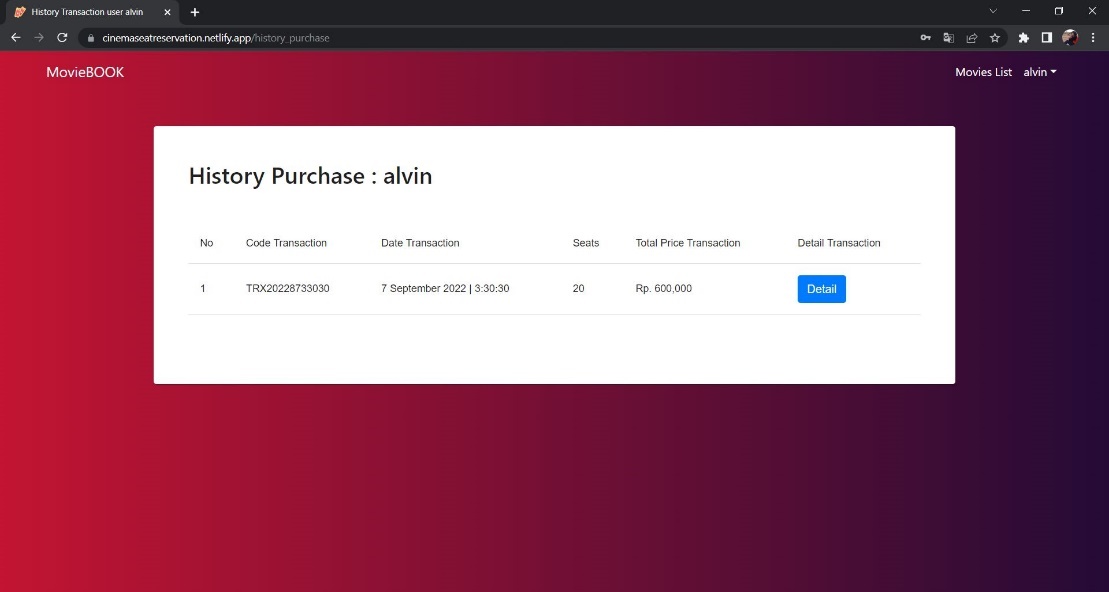
**Gambar 4.13** Tampilan Halaman Checkout

1. **Pembuatan Halaman History Purchase**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari halaman History Purchase. Data yang ditampilkan akan ter-generate pada saat user melakukan pembayaran terutama untuk kolom code transaction dan date transaction serta total seat dan total price. Berikut ini adalah potongan program untuk pembuatan halaman history purchase.



Pada **gambar 4.14** merupakan hasil implementasi pembuatan halaman purchase. Halaman ini menampilkan code transaction, date transaction, seat dan total price. Detail transaction perlu menekan button.



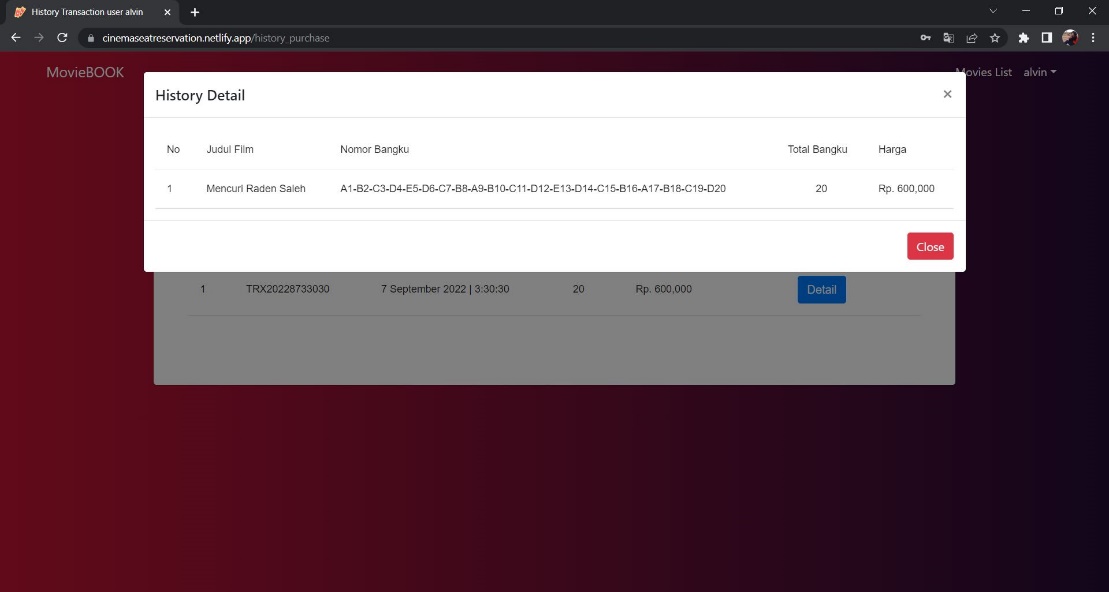
**Gambar 4.14** Tampilan Halaman Checkout

1. **Pembuatan Menu History Detail**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari menu history detail. Pada menu ini menampilkan pesanan yang sudah dibayarkan oleh user. Berikut ini adalah potongan program untuk pembuatan menu history detail.



Pada **gambar 4.15** merupakan hasil implementasi pembuatan menu history detail. Halaman ini menampilkan judul film dan nomor bangku.



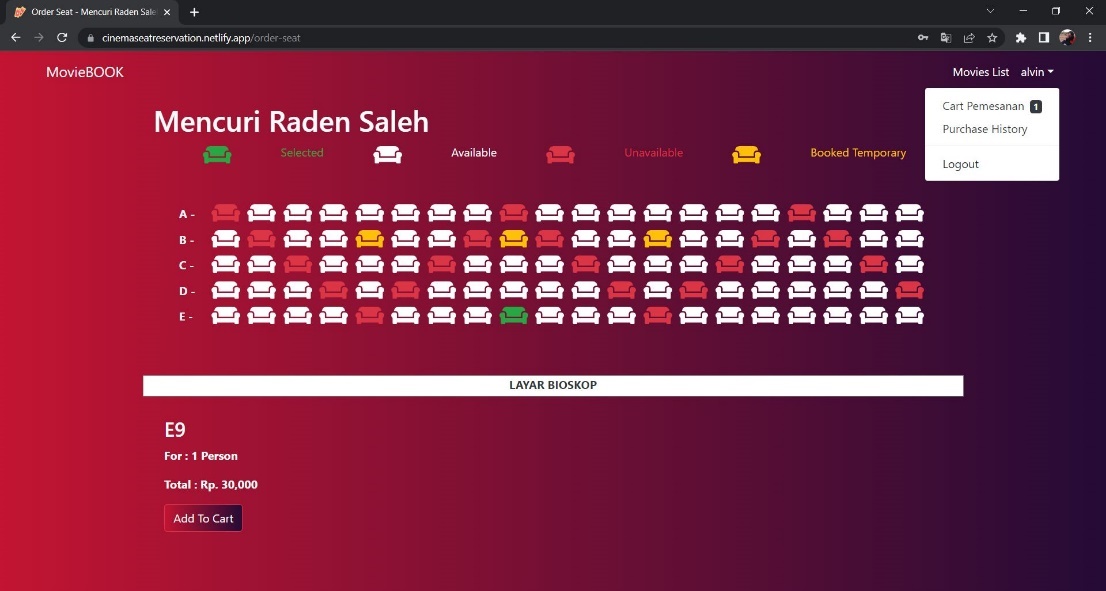
**Gambar 4.15** Tampilan Menu History Detail

1. **Pembuatan Halaman Order Seat Mixed**

Pada tahap ini dilakukan pembuatan logika program dan tampilan dari halaman order seat mixed. Terdapat 3 state yang dilakukan perulangan diantaranya temporarySeat dengan value 4, bookedSeat dengan value 2 dan chosen dengan value 3. Dimana kondisi terakhir akan me-return nilai default yaitu available seat dengan class text-white. Berikut ini adalah potongan program untuk pembuatan halaman order seat mixed.



Pada **gambar 4.16** merupakan hasil implementasi pembuatan halaman seat order mixed. Kursi yang berwarna merah sudah dibayar oleh user, kursi yang berwarna kuning sudah masuk ke dalam cart, kursi yang berwarna putih masih tersedia dan kursi yang berwarna hija sedang dipilih.



**Gambar 4.16** Tampilan Halaman Order Seat Mixed

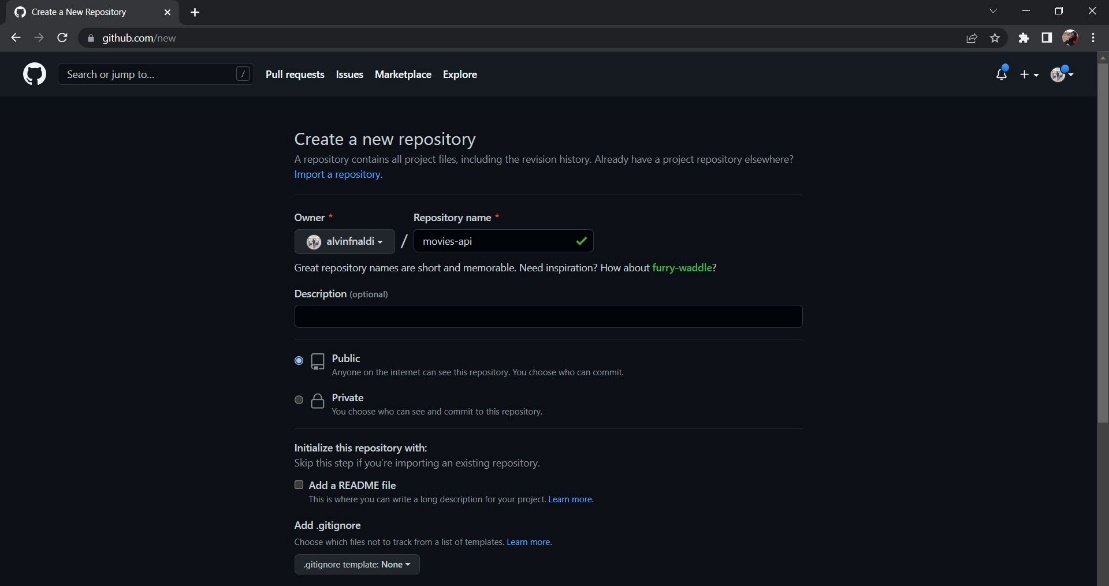
1. **Deploy Backend**

Setelah berhasil melakukan testing terhadap API secara offline, diperlukan untuk deploy ke hosting agar data dapat diakses secara online tanpa perlu running server pada terminal text editor dari file project. Masukkan file db.json ke dalam folder MovieAPI. Tampilan konfigurasi file server disajikan pada **Gambar 4.17** berikut.



**Gambar 4.17** Konfigurasi file server.js

Setelah berhasil konfigurasi file server, buat repository baru pada github dengan nama movies-api. Tampilan repository baru github disajikan pada **Gambar 4.18** berikut.



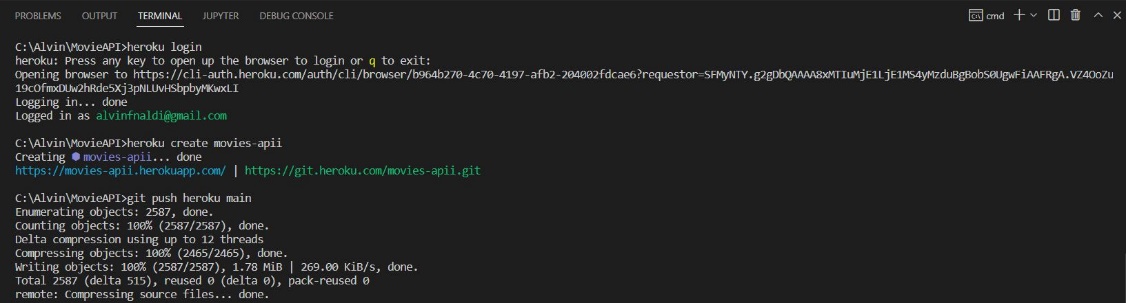
**Gambar 4.18** Tampilan Repository Baru Github

Pada terminal Visual Studio Code, push kodingan ke repository github yang sudah dibuat menggunakan command line berikut. Tampilan push code to github repo disajikan pada **Gambar 4.19** berikut.



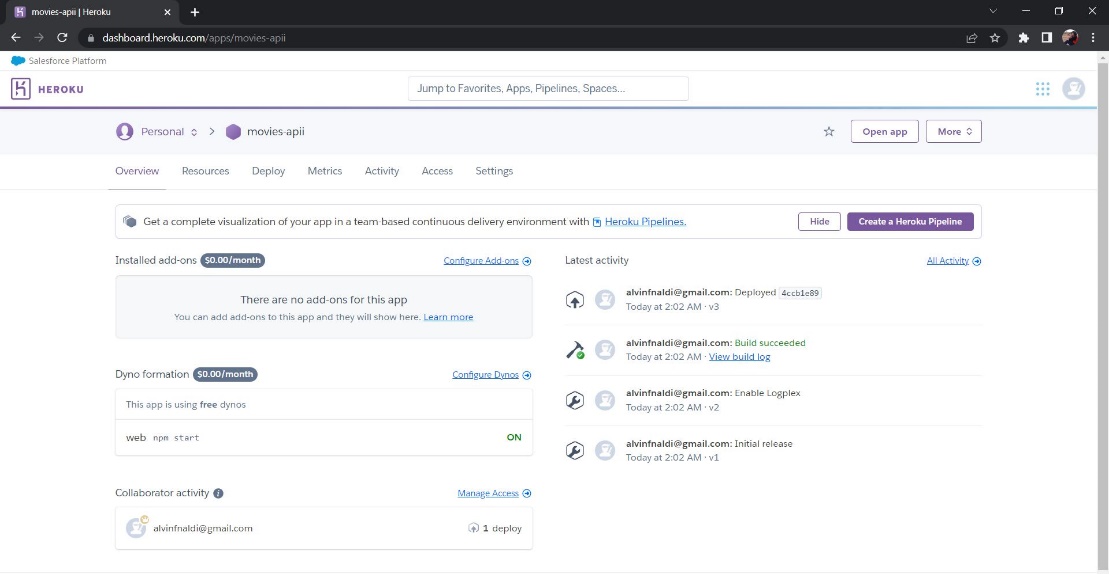
**Gambar 4.19** Push Code to Github Repo

Selanjutnya, cloud platform yang mendukung NodeJS diperlukan pada deployment server. Kali ini digunakan cloud platform yang bernama heroku. Lakukan login, create repository dan push ke hosting heroku menggunakan command line berikut. Tampilan deploy backend heroku disajikan pada **Gambar 4.20** berikut.



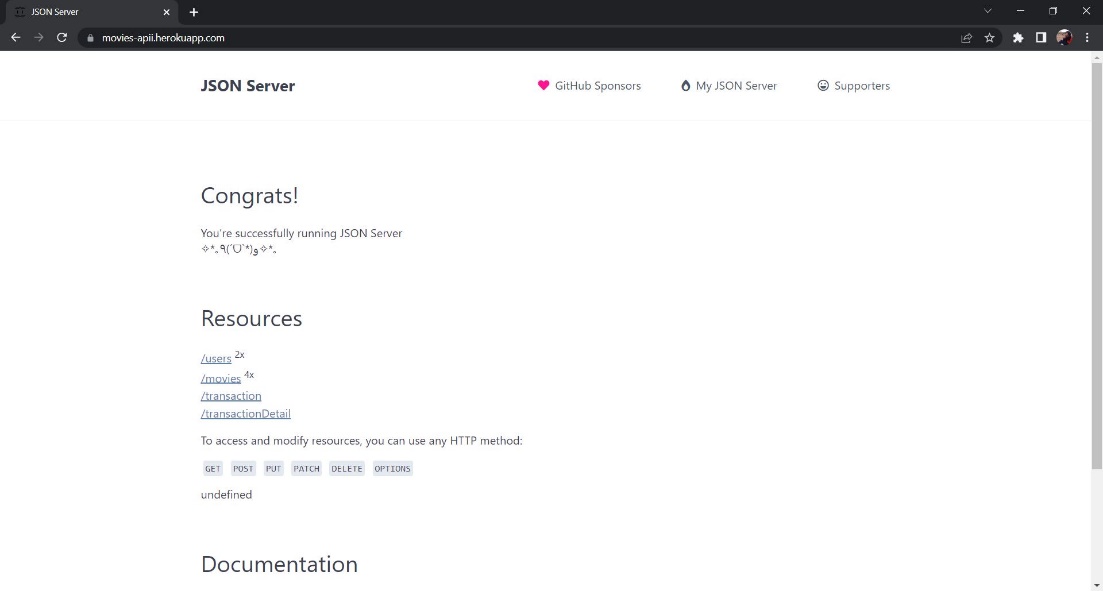
**Gambar 4.20** Deploy Backend Heroku

Pastikan kembali pada dashboard aplikasi telah build succeeded. Tampilan build succeeded disajikan pada **Gambar 4.21** berikut.



**Gambar 4.21** Build succeeded

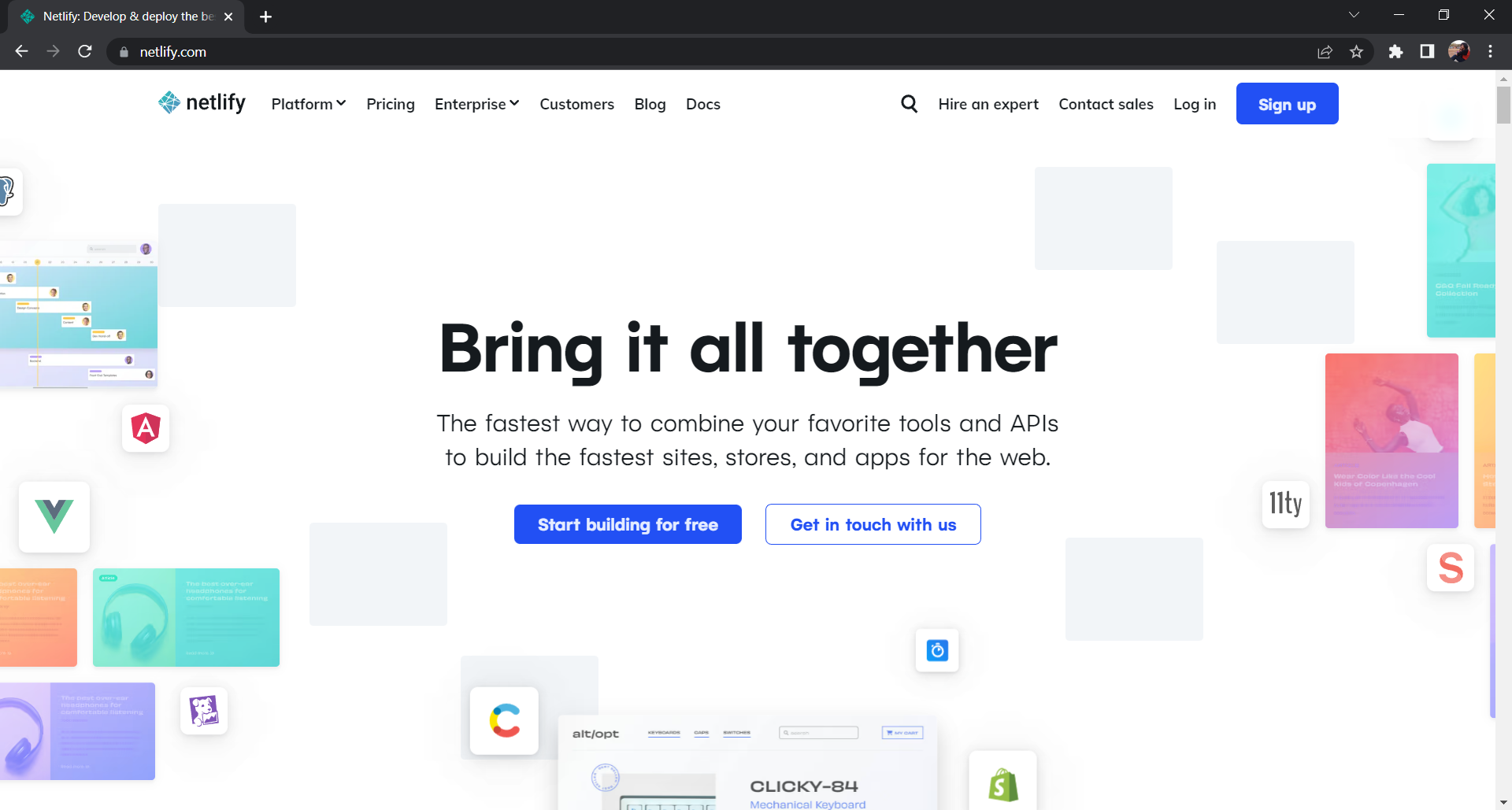
Akses aplikasi backend dengan mengunjungi url <https://movies-apii.herokuapp.com/>. Tampilan aplikasi backend dengan 4 endpoint disajikan pada **Gambar 4.22** berikut.



**Gambar 4.22** Server online

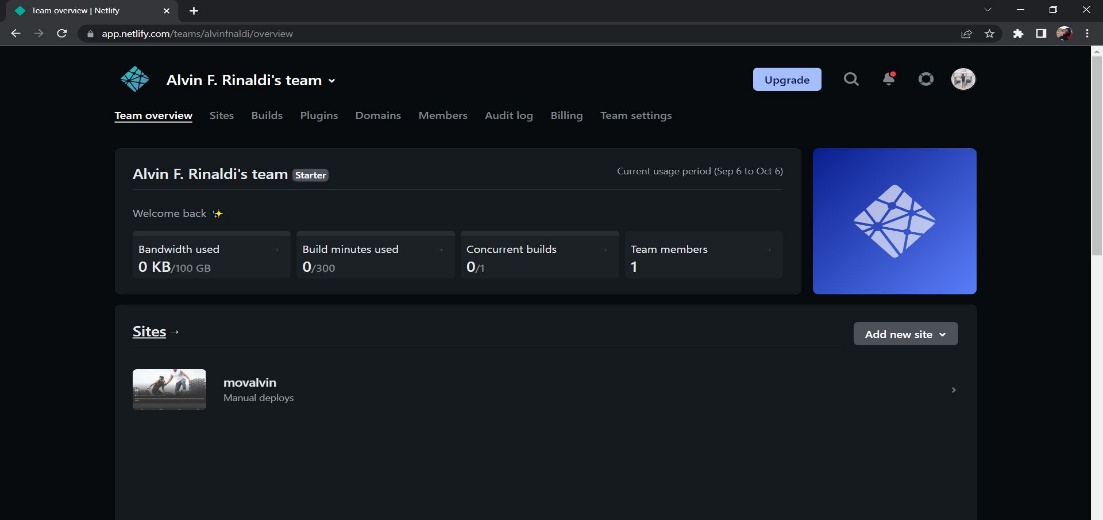
1. **Deploy Frontend**

Setelah aplikasi backend terdeploy, aplikasi frontend perlu juga di-hosting agar user interface website dapat diakses oleh semua orang yang ingin melakukan registrasi ke dalam web. 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 4.23** berikut.



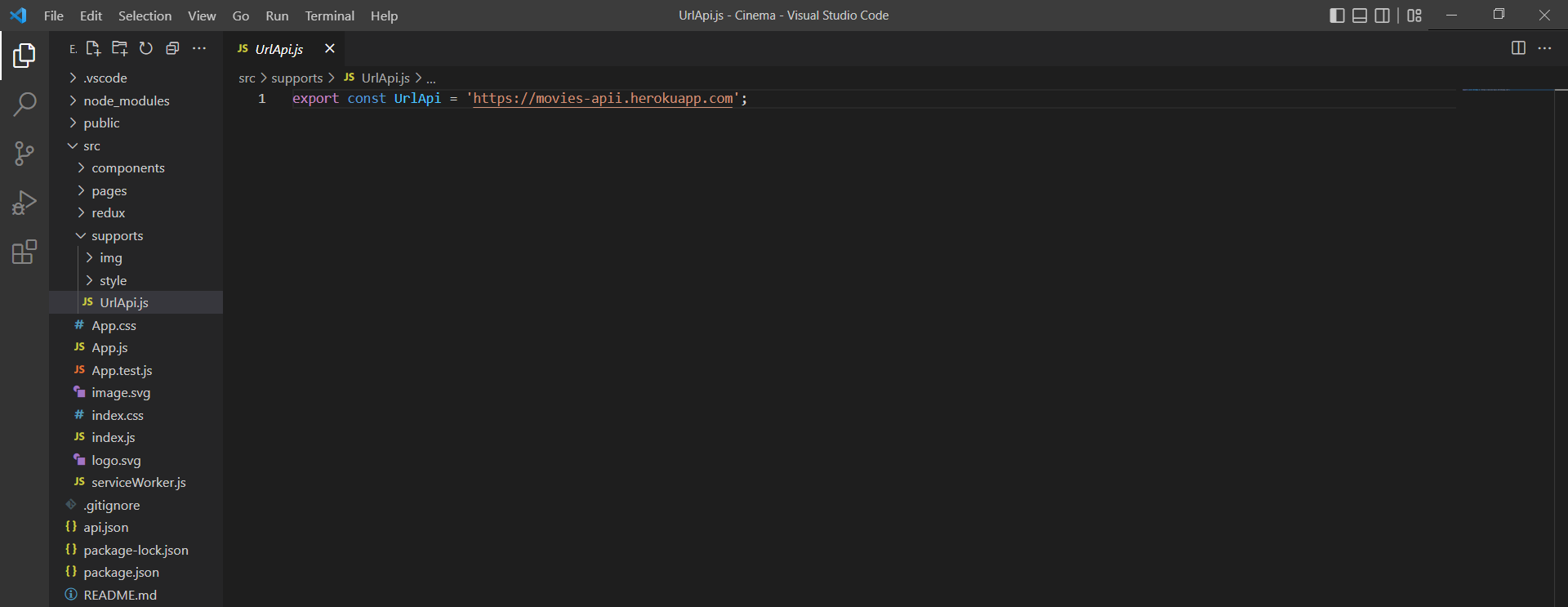
**Gambar 4.23**

Setelah berhasil login, maka akan menampilkan menu add new site untuk dilakukan manual deploy. Tampilan halaman Netlify sesudah login disajikan pada **Gambar 4.24** berikut.

****

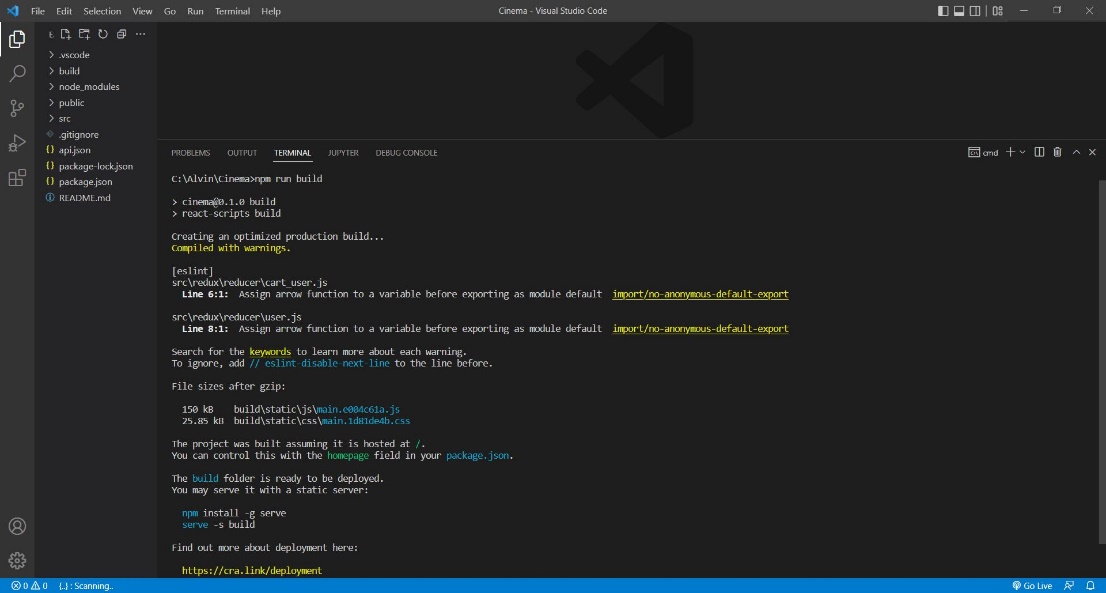
**Gambar 4.24** Tampilan Halaman Netlify setelah login

Definisikan terlebih dahulu sebuah variable UrlApi yang bernilai url aplikasi backend yang sudah di-hosting sebelumnya yaitu [https://movies-apii.herokuapp.com](https://movies-apii.herokuapp.com/). Tampilan setting url backend disajikan pada **Gambar 4.25** berikut.



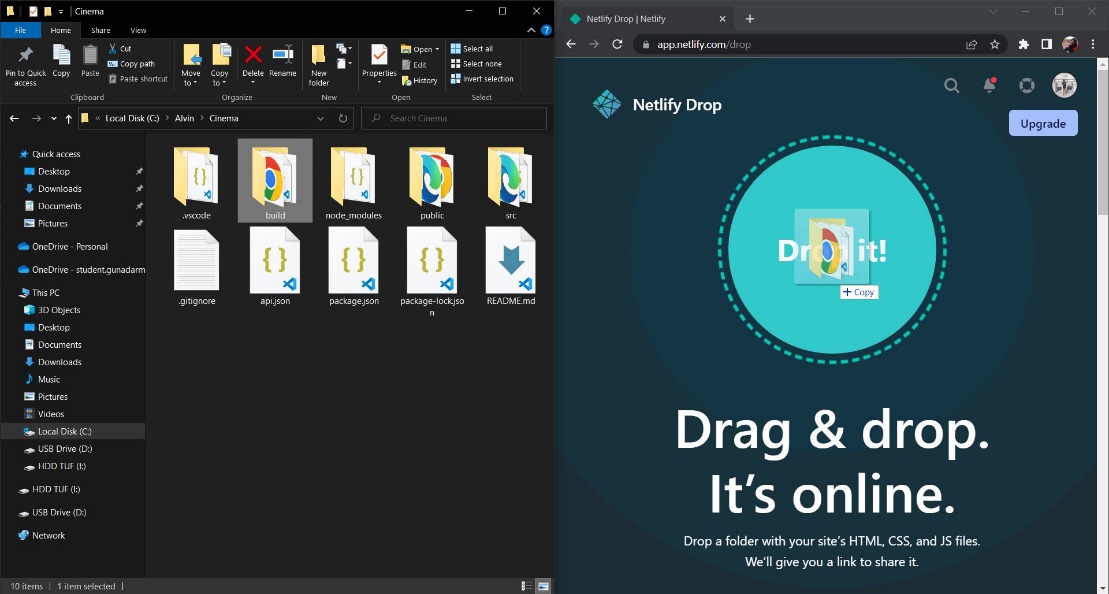
**Gambar 4.25** Setting url backend

Build program dengan perintah npm run build pada terminal kemudian tunggu beberapa saat sampai proses selesai dan folder build berhasil terbuat. Tampilan folder build disajikan pada **Gambar 4.26** berikut.

****

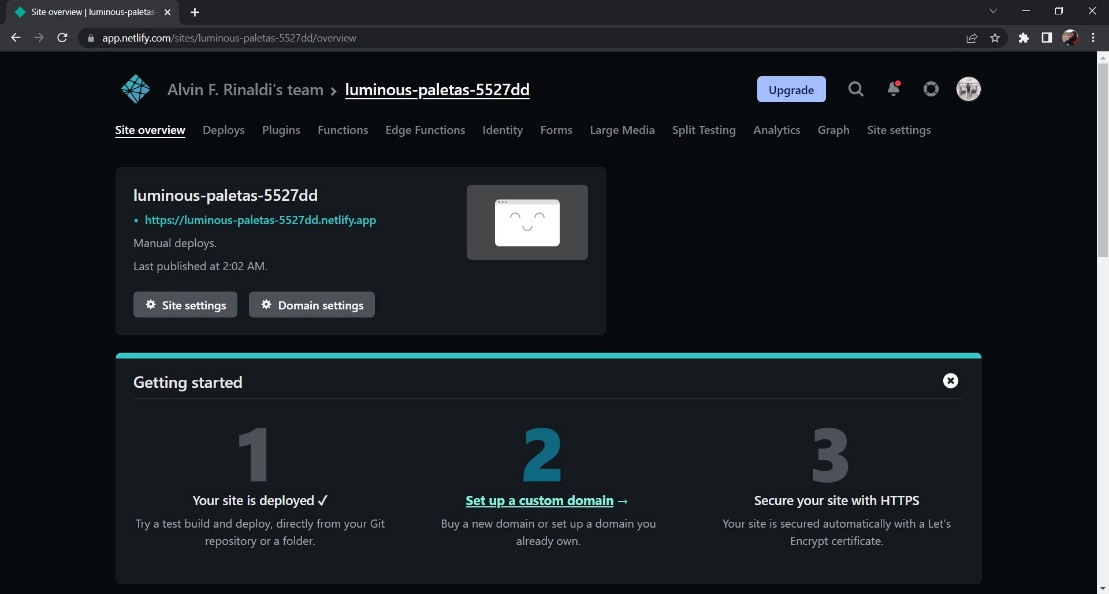
**Gambar 4.26** Pembuatan folder build

Selanjutnya, drag and drop folder build ke situs Netlify. Kemudian tunggu proses upload tersebut. Tampilan pemindahan folder build disajikan pada **Gambar 4.27** berikut.

****

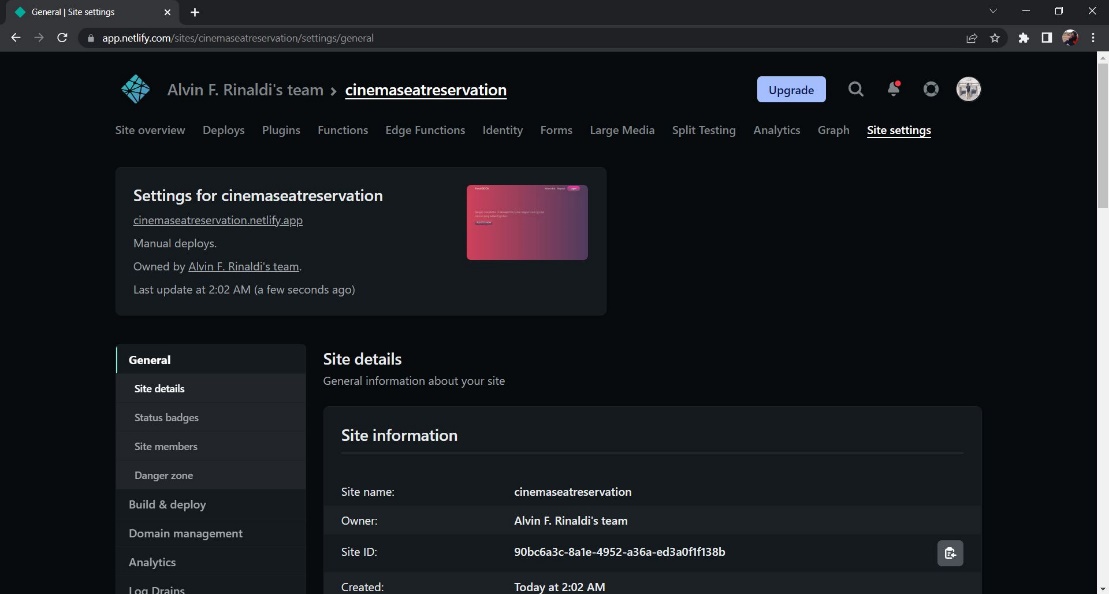
**Gambar 4.27** Drag and drop folder build

Selesai melakukan drag and drop folder build dari project yang dibuat, halaman 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 4.28** berikut.

****

**Gambar 4.28** Client online

Nama situs yang sudah terbuat dapat diubah melalui site settings. Agar nama situs relevan dengan aplikasi, ubah nama situs menjadi <https://cinemaseatreservation.netlify.app/>. Tampilan perubahan nama situs disajikan pada **Gambar 4.29** berikut.

****

**Gambar 4.29** Kustomisasi nama situs

1. **Uji Coba Aplikasi**

Setelah aplikasi diimplementasikan, uji coba aplikasi dilakukan terlebih dahulu sebelum digunakan untuk menemukan kesalahan yang mungkin terjadi saat sistem sedang berjalan pada browser Google Chrome dan Mozilla Firefox. **Tabel 3.1** menampilkan hasil pengujian Black Box.

**Tabel 3.1** Hasil Uji Coba Black Box

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Pengujian | Test Case | Hasil Yang Diharapkan | Hasil Pengujian | Google Chrome | Mozilla Firefox |
| 1 | Homepage Admin | Login dengan akun admin | Menampilkan pesan “Welcome Back, admin!” | Sesuai harapan | Berhasil | Berhasil |
| 2 | Manage Movie | Klik menu Manage Movie | Menampilkan Data Movie default dari API | Sesuai harapan | Berhasil | Berhasil |
| 3 | Add Movie | Klik Add Data | Menampilkan modal berisi form untuk input data movie | Sesuai harapan | Berhasil | Berhasil |
| 4 | Delete Movie | Klik icon delete pada salah satu data movie | Menghapus data movie yang dipilih | Sesuai harapan | Berhasil | Berhasil |
| 5 | Update Movie | Klik icon update pada salah satu data movie | Menampilkan modal berisi form input berikut dengan value-nya | Sesuai harapan | Berhasil | Berhasil |
| 6 | Homepage User | Login dengan akun user | Menampilkan pesan “Welcome Back, user!” | Sesuai harapan | Berhasil | Berhasil |
| 7 | Movie List | Klik menu Movie List | Menampilkan daftar film yang sedang tayang | Sesuai harapan | Berhasil | Berhasil |
| 8 | Movie Details | Klik menu movie details pada salah satu film | Menampilkan info berupa poster, judul, genre, durasi, rating, sutradara, playing at dan plot dari film tersebut | Sesuai harapan | Berhasi | Berhasil |
| 9 | Buy Ticket | Klik button buy ticket | Redirect ke halaman order seat | Sesuai harapan | Berhasil | Berhasil |
| 10 | Order Seat | Memilih tempat duduk yang diinginkan | Tempat yang sudah terpilih tidak bisa di-klik | Sesuai harapan | Berhasil | Berhasil |
| 11 | Add to Cart | Klik button add to cart setelah memilih kursi | Status booking bersifat sementara | Sesuai harapan | Berhasil | Berhasil |
| 12 | Cart | Klik button checkout | Redirect ke halaman checkout | Sesuai harapan | Berhasil | Berhasil |
| 13 | Checkout | Klik button pay | Pesanan terbayarkan, cart menjadi kosong | Sesuai harapan | Berhasil | Berhasil |
| 14 | History purchase | Klik dropdown akun, pilih purchase history | Menampilkan history transaksi pemesanan | Sesuai harapan | Berhasil | Berhasil |

Berdasarkan hasil pengujian yang telah dilakukan dapat dihasilkan bahwa Aplikasi Cinema Seat Reservation Berbasis Website Menggunakan ReactJS dan NodeJs berjalan dengan baik sesuai fungsi-fungsi yang diinginkan.