

MA

Buatkan script reactjs untuk rekomendasi 10 film setiap hari dengan api themoviedb dan tampilan dari tailwind juga untuk memper indah dan buat per file nya juga secara detail dan pengistalan tailwindnya juga Jelaskan langkah demi langkah



Untuk membuat aplikasi rekomendasi 10 film setiap hari menggunakan ReactJS, Tailwind CSS, dan API themoviedb.org, ikuti langkah-langkah berikut:

1. Instalasi dan konfigurasi awal:

- Instal Node.js jika belum diinstal di komputer Anda (<https://nodejs.org/en/>)
- Buka terminal atau command prompt dan jalankan perintah berikut untuk membuat proyek React baru:

```
lua
Copy code
npx create-react-app movie-recommendation
```

- Masuk ke direktori proyek:

```
bash
Copy code
cd movie-recommendation
```

- Instal dependensi Tailwind CSS:

```
java
Copy code
npm install tailwindcss@latest postcss@latest autoprefixer@latest
```

- Inisialisasi Tailwind CSS:

```
csharp
Copy code
npx tailwindcss init -p
```

- Buka file `tailwind.config.js` dan tambahkan konfigurasi berikut:

```
javascript
Copy code
module.exports = {
  purge: ['./src/**/*.{js,jsx,ts,tsx}', './public/index.html'],
  darkMode: false,
  theme: {
    extend: {},
  },
  variants: {
    extend: {},
  },
  plugins: [],
}
```

- Buka file `src/index.css` dan tambahkan:

```
css
Copy code
@tailwind base;
@tailwind components;
@tailwind utilities;
```

- Import `index.css` ke `src/index.js`:

```
javascript
Copy code
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);

reportWebVitals();
```

2. Daftarkan akun di themoviedb.org (<https://www.themoviedb.org/signup>) dan dapatkan API key dari dashboard.

3. Membuat komponen utama:

- Buat folder `src/components` untuk menyimpan komponen.
- Buat file `src/components/MovieCard.js` untuk menampilkan informasi film:

```
javascript
Copy code
import React from 'react';

const MovieCard = ({ movie }) => {
  const imageUrl = `https://image.tmdb.org/t/p/w200${movie.poster_path}`;

  return (
    <div className="w-64 p-4 bg-white rounded-lg shadow-md">
      <img src={imageUrl} alt={movie.title} className="w-full h-96 rounded-lg" />
      <h2 className="mt-2 text-xl font-bold">{movie.title}</h2>
      <p className="mt-1 text-gray-600">{movie.release_date}</p>
      <p className="mt-1 text-gray-800">{movie.overview}</p>
    </div>
  );
};

export default MovieCard;
```

- Buat file `src/components/MovieList.js` untuk menampilkan daftar film:

```
javascript
Copy code
import React, { useEffect, useState } from 'react';
import MovieCard from './MovieCard';

const API_KEY = 'API_KEY_ANDA';
const API_URL = `https://api.themoviedb.org/3/discover/movie?api_key=${API_KEY}&sort_by=popularity.desc`;

const MovieList = () => {
  const [movies, setMovies] = useState([]);

  useEffect(() => {
    fetchMovies();
  }, []);

  const fetchMovies = async () => {
    try {
      const response = await fetch(API_URL);
      const data = await response.json();
      setMovies(data.results.slice(0, 10));
    } catch (error) {
      console.error(error);
    }
  };

  return (
    <div className="container mx-auto">
      <h1 className="text-4xl font-bold mb-8">Top 10 Movies Today</h1>
      <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4">
        {movies.map((movie) => (
          <MovieCard key={movie.id} movie={movie} />
        ))}
      </div>
    </div>
  );
};

export default MovieList;
```

MA

lanjutkan



```
10));
} catch (error) {
  console.error(error);
}
};

return
```

MA

Buat file src/components/MovieList.js untuk menampilkan daftar film:



Buat file `src/components/MovieList.js` untuk menampilkan daftar film:

```
javascript
Copy code
import React, { useEffect, useState } from 'react';
import MovieCard from './MovieCard';

const API_KEY = 'API_KEY_ANDA';
const API_URL = `https://api.themoviedb.org/3/discover/movie?api_key=${API_KEY}&sort_by=popularity.desc`;

const MovieList = () => {
  const [movies, setMovies] = useState([]);

  useEffect(() => {
    fetchMovies();
  }, []);

  const fetchMovies = async () => {
    try {
      const response = await fetch(API_URL);
      const data = await response.json();
      setMovies(data.results.slice(0, 10));
    } catch (error) {
      console.error(error);
    }
  };

  return (
    <div className="container mx-auto">
      <h1 className="text-4xl font-bold mb-8">Top 10 Movies Today</h1>
      <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4">
        {movies.map((movie) => (
          <MovieCard key={movie.id} movie={movie} />
        ))}
      </div>
    </div>
  );
};

export default MovieList;
```

Dalam kode di atas:

- Kami mengimpor komponen `MovieCard` yang telah dibuat sebelumnya.
- Kami mendefinisikan API key dan URL untuk memanggil API themoviedb.org.
- Kami menggunakan hook `useState` untuk menyimpan daftar film.
- Kami menggunakan hook `useEffect` untuk mengambil film saat komponen dimuat.
- Kami mendefinisikan fungsi `fetchMovies` untuk mengambil film dari API dan menyimpannya dalam state.
- Kami menampilkan daftar film menggunakan grid responsif dengan Tailwind CSS.

Setelah membuat komponen `MovieList`, import dan gunakan komponen ini di dalam file `src/App.js`:

```
javascript
Copy code
import React from 'react';
import MovieList from './components/MovieList';

function App() {
  return (
    <div className="bg-gray-100 min-h-screen">
      <MovieList />
    </div>
  );
}

export default App;
```

Sekarang Anda dapat menjalankan aplikasi dengan perintah berikut:

```
sql
Copy code
npm start
```

Aplikasi akan membuka di browser, menampilkan 10 film teratas setiap hari menggunakan API themoviedb.org dan tampilan dari Tailwind CSS.