React

List, Event, State

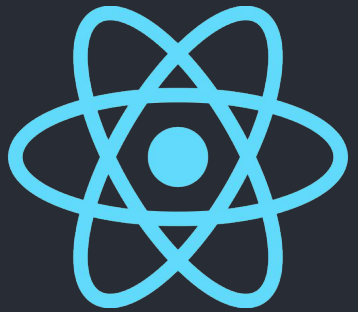


Table of Contents

React Developer Tools

React Developer Tools

- Debugging Tools for React.
- Inspect React Component Hierarchies.
- Inspect and edit: Component, Element, Props, State
- Install: Link

Dimensions: Responsive ▾ 1440 × 1295 75% ▾ No throttling ▾

Elements Components 2

Search (text or /regex/)

```
App
  Home
    Navbar
    Main
      Hero
      Movies
        Movie key="tt10872600"
        Movie key="tt2953050"
        Movie key="tt7991608"
        Movie key="tt12689248"
        Movie key="tt14060094"
        Movie key="tt8097030"
        Movie key="tt14128670"
        Movie key="tt13634480"
        Movie key="tt9848626"
        Movie key="tt9032400"
        Movie key="tt7740496"
        Movie key="tt11755740"
      Footer
```

Navbar

props
  new entry: ""

rendered by
  Home
  App
  createLegacyRoot()
  react-dom@17.0.2

source
  Home.js:28

Movie App

Home Add Movie Popular Now Playing Top Rated

Spiderman

Genre: Thriller, Drama, Romance

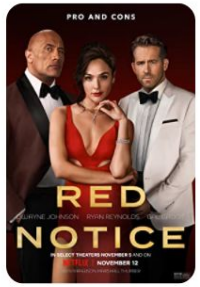
Lorem ipsum dolor, sit amet consectetur adipisicing elit. Aliquam cum accusamus quisquam earum velit ea nobis maiores exercitationem nam temporibus.

Watch

Latest Movies

Spider-Man
2021

Encanto
2021



Red Notice
2021

The Commando
2022

Refactor

Movie.js

```
function Movies() {
  return (
    <div className={styles.container}>
      <section className={styles.movies}>
        <h2 className={styles.movies__title}>Latest Movies</h2>
        <div className={styles.movie__container}>
          <div className={styles.movie}>
            <img
              className={styles.movie__image}
              src="https://picsum.photos/300/400"
              alt=""
            />
            <h3 className={styles.movie__title}>Movie Title</h3>
            <p className={styles.movie__date}>Date Title</p>
          </div>
          <div className={styles.movie}>
            <img
              className={styles.movie__image}
              src="https://picsum.photos/300/400"
              alt=""
            />
            <h3 className={styles.movie__title}>Movie Title</h3>
            <p className={styles.movie__date}>Date Title</p>
          </div>
        </div>
      </section>
    </div>
  );
}
```

- Remember Thinking in React
- Can we break into smaller component?
- Movie Component.

Movie Component

- Membuat Movie Component (single movie).
- Simpan di folder src/components/Movie/.
- Component: Movie.js.
- CSS: Movie.module.css.

Referensi: React- Thinking in React.

Movie.js

```
import styles from "./Movie.module.css";

function Movie() {
  return (
    <div className={styles.movie}>
      <img
        className={styles.movie__image}
        src="https://picsum.photos/300/400"
        alt=""
      />
      <h3 className={styles.movie__title}>Movie Title</h3>
      <p className={styles.movie__date}>Date Title</p>
    </div>
  );
}

export default Movie;
```

Movie Component

Movie.module.css

```css
/* Small Screen */
.movie {
  margin-bottom: 1rem;
}

.movie__image {
  border-radius: 25px;
  max-width: 100%;
  height: auto;
  margin-bottom: 1rem;
}

.movie__title {
  color: #4361ee;
  font-size: 1.95rem;
  margin-bottom: 0.5rem;
}

.movie__date {
  color: #64748b;
}
```

Small Screen

Movie.module.css

```css
/* Medium Screen */
@media (min-width: 768px) {
  .movie {
    flex-basis: 50%;
  }
}
```

Medium Screen

Movie.module.css

```css
/* Large Screen */
@media (min-width: 992px) {
  .movie {
    flex-basis: 25%;
    padding: 1rem;
  }
}
```

Large Screen

Movies.js

```jsx
import Movie from "../Movie/Movie";
import styles from "./Movies.module.css";

function Movies() {
  /**
   * Memanggil Component Movie di dalam Component Movies.
   */
  return (
    <div>
      <div className={styles.container}>
        <section className={styles.movies}>
          <h2 className={styles.movies__title}>Latest Movies</h2>
          <div className={styles.movie__container}>
            <Movie />
            <Movie />
            <Movie />
            <Movie />
            <Movie />
          </div>
        </section>
      </div>
    </div>
  );
}

export default Movies;
```

- Movies component looks cleaner, right?
- Break Large into Small Component.
- Movie Component (Single Component).

List and Keys

Movies.js

```
function Movies() {
  return (
    <div>
      <div className={styles.container}>
        <section className={styles.movies}>
          <h2 className={styles.movies__title}>Latest Movies</h2>
          <div className={styles.movie__container}>
            <Movie title="Superman No Home" year="2022" />
            <Movie title="Dark" year="2021" />
            <Movie title="Turning Red" year="2022" />
            <Movie title="The Batman" year="2022" />
            <Movie title="Encanto" year="2021" />
          </div>
        </section>
      </div>
    </div>
  );
}
```

- Why don’t create data (movies)?
- Looping data and render component.
- Send props to component.

Data Movie

- Menyiapkan data movies.
- Simpan di folder src/utils/constants/.
- File: data.js.
- Grab data movies: Link.

Lists

- Render multiple components/elements (list).
- Menggunakan array method: map.
- Method map mengembalikan array baru (return).

Referensi: React- List and Keys.

Movies.js

```
// Import Data Movies
import data from "../../utils/constants/data";

function Movies() {
  // Membuat Variable movies
  const movies = data;

  return (
    <div>
      <div className={styles.container}>
        <section className={styles.movies}>
          <h2 className={styles.movies__title}>Latest Movies</h2>
          <div className={styles.movie__container}>
            {/*
             * Looping movies (array).
             * Render Component Movie dan kirim props movie
             */}
            {movies.map((movie) => {
              return <Movie key={movie.id} movie={movie} />;
            })}
          </div>
        </section>
      </div>
    </div>
  );
}
```

Movie.js

```
// Component Movie menerima props
function Movie(props) {
  // Melakukan destructing props
  const { movie } = props;

  return (
    <div className={styles.movie}>
      <img
        className={styles.movie__image}
        src={movie.poster}
        alt={movie.title}
      />
      <h3 className={styles.movie__title}>{movie.title}</h3>
      <p className={styles.movie__date}>{movie.year}</p>
    </div>
  );
}
```

- More abstraction.
- One component one responsibility.
- Looks cleaner

Keys

- Mengidentifikasi component (changeable).
- Bersifat unik: id.
- Diberikan pada Component yang di-looping.

Referensi: React- List and Keys.

Movies.js

```
function Movies() {
  const movies = data;

  return (
    <div>
      <div className={styles.container}>
        <section className={styles.movies}>
          <h2 className={styles.movies__title}>Latest Movies</h2>
          <div className={styles.movie__container}>
            {movies.map((movie) => {
              return <Movie key={movie.id} movie={movie} />;
            })}
          </div>
        </section>
      </div>
    </div>
  );
}
```

- Menambahkan key
- Menggunakan ID movie sebagai nilai key

Handling Event

Event

- Action (kejadian) yang terjadi pada Component.
- Component bereaksi terhadap kejadian tersebut.
- Ex: Jika tombol diklik, maka tampilkan navbar.
- Demo: Link.

Referensi: React- Event.

Event

Common Event (React):

- onClick: Component diklik.
- onSubmit: Component di-submit (form).
- onChange: Component berubah (input).

Referensi: React- Event.

Handling Event

- Handling event in React and JavaScript is bit different.
- JavaScript: external event (addEventListener)
- React: inline event (onClick)

Difference:

- React event is camelCase.
- Pass a function in event handler (not string).
- Use preventDefault to prevent default behaviour.

Referensi: React- Handling Event.

Movies.js

```
function Movies() {
  const movies = data;

  /**
   * Fungsi untuk handle event click.
   * Dijalankan ketika button diklik.
   */
  function handleClick() {
    const movie = {
      id: "xyz", title: "Jigsaw",
      year: 2021, type: "Movie",
      poster: "https://picsum.photos/300/400",
    };

    movies.push(movie);
  }

  return (
    <div>
      <div className={styles.container}>
        <section className={styles.movies}>
          <h2 className={styles.movies__title}>Latest Movies</h2>
          <div className={styles.movie__container}>
            {movies.map((movie) => {
              return <Movie key={movie.id} movie={movie} />;
            })}
          </div>
          {/* Element button diberikan event click: onClick */}
          <button onClick={handleClick}>Add Movie</button>
        </section>
      </div>
    </div>
  );
}
```

- Apakah datanya muncul di Component? Why?
- Try log the movies. Apakah muncul?

State

State

- State similar to props.
- Make component alive.
- Component re-render if state changes
- Private and controlled by Component.
- Demo: Link.

Referensi: React- State and Lifecycle.

Handling State

- Class Component
- Function Component: Hooks State

Why not Class Component:

- Complex, hard to understand
- Hard to reuse logic
- Confuse both people and machines

Referensi: React- State and Lifecycle.

State Hooks

- Hooks: Fungsi spesial untuk menggunakan fitur React (Class Component).
- State Hooks: Menambahkan State di Function Component.
- New Feature in React 16.8.
- Use state without Class.
- Only works inside Function.

Referensi: React- State Hooks.

Counter.js

```
function Counter() {
  // Membuat variable result
  let result = 0;

  /**
   * Membuat fungsi handleClick.
   * Dijalankan ketika button dikli.
   */
  function handleClick() {
    result = result + 1;
  }

  // Menambahkan event click pada button
  return (
    <div>
      <p>Result: {result} </p>
      <button onClick={handleClick}>Add</button>
    </div>
  );
}

export default Counter;
```

- Apakah datanya berubah di component? Why?
- Try log the result.
- Solution: State.

Counter.js

```jsx
// Import useState dari React (desctructing).
import { useState } from "react";

function Counter() {
  /**
   * Membuat state menggunakan useState dan set nilai awal 0
   * useState mengembalikan 2 nilai:
   * - Berisi current value: result.
   * - Berisi fungsi untuk mengupdate result.
   * Melakukan destructing array dari hasil useState
   */
  const [result, setResult] = useState(0);

  function handleClick() {
    /**
     * Update state result menggunakan fungsi setResult
     */
    setResult(result + 1);
  }

  return (
    <div>
      <p>Result: {result} </p>
      <button onClick={handleClick}>Add</button>
    </div>
  );
}

export default Counter;
```

- Apakah datanya berubah di Component? Why?
- Component re-rendered if state changes
- Make component alive

State - Movie App

- Implement state in Movie App
- Use useState to create state movies.
- Update state when button click.

Referensi: React- State Hooks.

```
function Movies() {
  // Membuat state movies
  const [movies, setMovies] = useState(data);

  function handleClick() {
    const movie = {
      id: "xyz", title: "Jigsaw Spiral",
      year: 2021, type: "Movie",
      poster: "https://picsum.photos/300/400",
    };

    /**
     * Update state movies: setMovies
     * Melakukan teknik spread untuk copy dan merge array
     */
    setMovies([...movies, movie]);
  }

  return (
    <div>
      <div className={styles.container}>
        <section className={styles.movies}>
          <h2 className={styles.movies__title}>Latest Movies</h2>
          <div className={styles.movie__container}>
            {movies.map((movie) => {
              return <Movie key={movie.id} movie={movie} />;
            })}
          </div>
          <button onClick={handleClick}>Add Movie</button>
        </section>
      </div>
    </div>
  );
}
```

- Masih ingat spread operator?
- Menggunakan operator three dots: ...
- Digunakan untuk copy dan merge array

Generate ID

Generate ID

- Generate ID unik.
- Menggunakan package nanoid.
- Install: npm i nanoid.

Referensi: React- State Hooks.

Movies.js

```
// Import nanoid
import { nanoid } from "nanoid";

function Movies() {
  // kode lain sebelumnya

  function handleClick() {
    // Menggunakan nanoid
    const movie = {
      id: nanoid(),
      title: "Jigsaw Spiral",
      year: 2021,
      type: "Movie",
      poster: "https://picsum.photos/300/400",
    };
  }

  // kode lain setelahnya
}
```

Recap

Recap

- JSX: Menulis UI (HTML) di React.
- Component: Reusable UI, tersusun oleh element(s).
- Props: Input pada Component.
- State: Private Data pada Component.
- List and Keys: Render multiple Components.
- Event: Action yang terjadi pada Component.

QnA

Take Away

Task

Description:

- Push your work to Github.

Assignment:

- Push code to Repository Github (use branch for task management).
- No zip and don't push node_nodules folder.
- Submit link repository to elena: Link.

Attendance

Thanks