

React

Form & Lifting State

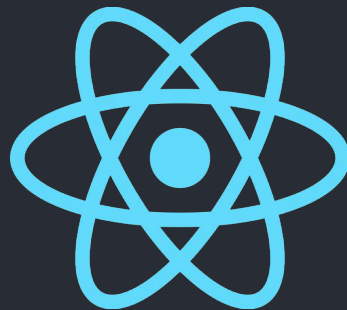


Table of Contents

- Forms
- Lifting State
- Conditional Rendering
- Composition

Forms



Forms

- React form is different than HTML form (handling).
- Controlled component: Component yang dikontrol oleh React.
- Controlled component: input, select, textarea
- Using `state` to control form

Referensi: [React- Forms](#).

Form Event

- onSubmit: Ketika component di-submit (form).
- onChange: Ketika nilai component berubah (input, select, textarea).

Referensi: [React-Forms](#).

Handle Title

- Menghandle ketika nilai input (title) berubah.
- Menggunakan event `onChange`.
- Menggunakan `state` untuk menyimpan value.

Referensi: [React- Forms](#).



AddMovieForm.js

```
function AddMovieForm() {  
  /**  
   * Ini hanya snippet(potongan) code.  
   * Kode yang lainnya tetap sama.  
   */  
  
  // Membuat state title  
  const [title, setTitle] = useState("");  
  
  return (  
    <input  
      id="title"  
      type="text"  
      // Memberikan value input: title  
      value={title}  
    />  
  );  
}
```



AddMovieForm.js

```
function AddMovieForm() {  
  /**  
   * Ini hanya snippet(potongan) code.  
   * Kode yang lainnya tetap sama.  
   */  
  
  // Membuat state title  
  const [title, setTitle] = useState("");  
  
  /**  
   * Membuat fungsi handleTitle  
   * Dijalankan ketika nilai input berubah  
   */  
  function handleTitle(e) {  
    /**  
     * Jalankan fungsi setTitle.  
     * Set title nilai baru: input saat ini.  
     */  
    setTitle(e.target.value);  
  }  
  
  return (  
    <input  
      id="title"  
      type="text"  
      // Memberikan value input: title  
      value={title}  
      // Memberikan event onChange  
      onChange={handleTitle}  
    />  
  );  
}
```

Handle Date

- Menghandle ketika nilai input (date) berubah.
- Menggunakan event `onChange`.
- Menggunakan `state` untuk menyimpan value.

Referensi: [React- Forms](#).



AddMovieForm.js

```
function AddMovieForm() {  
  /**  
   * Ini hanya snippet(potongan) code.  
   * Kode yang lainnya tetap sama.  
   */  
  
  // Membuat state date  
  const [date, setDate] = useState("");  
  
  return (  
    <input  
      id="date"  
      type="text"  
      // Memberikan value input: date  
      value={date}  
    />  
  );  
}
```



AddMovieForm.js

```
function AddMovieForm() {  
  /**  
   * Ini hanya snippet(potongan) code.  
   * Kode yang lainnya tetap sama.  
   */  
  
  // Membuat state date  
  const [date, setDate] = useState("");  
  
  /**  
   * Membuat fungsi handleDate  
   * Dijalankan ketika nilai input berubah  
   */  
  function handleDate(e) {  
    /**  
     * Jalankan fungsi setDate.  
     * Set date nilai baru: input saat ini.  
     */  
    setDate(e.target.value);  
  }  
  
  return (  
    <input  
      id="date"  
      type="text"  
      // Memberikan value input: date  
      value={date}  
      // Memberikan event onChange  
      onChange={handleDate}  
    />  
  );  
}
```

Handle Submit

- Menghandle ketika form di-submit.
- Menggunakan event `onSubmit`.

Referensi: [React- Forms](#).



AddMovieForm.js

```
function AddMovieForm() {  
  /**  
   * Ini hanya snippet(potongan) code.  
   * Kode yang lainnya tetap sama.  
   */  
  
  function handleSubmit(e) {  
    /**  
     * Mencegah perilaku default form.  
     * Mencegah form direfresh ketika disubmit.  
     */  
    e.preventDefault();  
  
    const movie = {  
      id: nanoid(), title: title,  
      year: date, type: "Movie",  
      poster: "https://picsum.photos/300/400",  
    };  
  
    // HELP: HOW TO ADD MOVIE TO MOVIES :(  
  }  
  
  return (  
    <form onSubmit={handleSubmit}>  
  
    </form>  
  );  
}
```

- HELP :(
- How to add `movie` to state `movies`?
- What is solution?

How to add movie to state movies?

movie

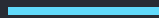
<AddMovieForm />

movies

<Movies />

- Component
- State
- Data Movie

Lifting State

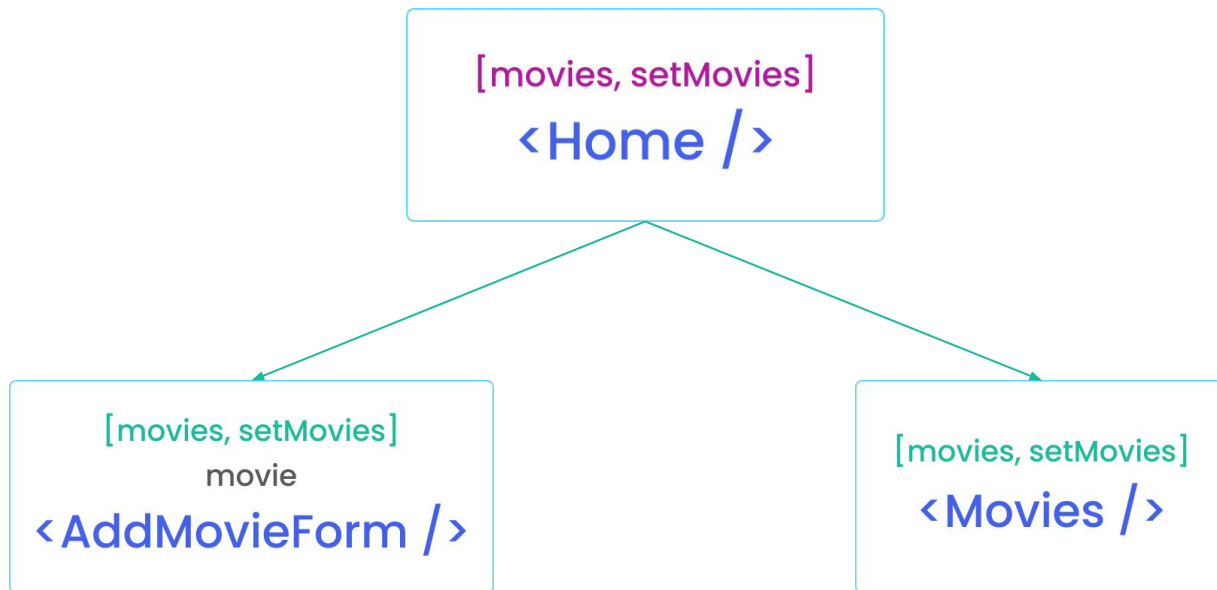


Lifting State

- Several components need the same state.
- Lifting state to their closest parent component.

Referensi: [React- Lifting State](#).

Lifting State



- **Component**
- **State**
- **Props**
- **Data Movie**



Home.js

```
function Main() {  
  /**  
   * Ini hanya snippet(potongan) code.  
   * Kode yang lainnya tetap sama.  
   */  
  
  /**  
   * Mengangkat stave movies: lifting state.  
   * Dari Component Movies ke Component Home.  
   * Agar bisa digunakan oleh Component yang lain.  
   */  
  const [movies, setMovies] = useState(data);  
  
  return (  
    <main>  
      <Hero />  
      { /* Mengirim props: state movies */}  
      <Movies movies={movies} setMovies={setMovies} />  
      <AddMovieForm movies={movies} setMovies={setMovies} />  
    </main>  
  );  
}
```




Movies.js

```
// Menangkap props
function Movies(props) {
  /**
   * Ini hanya snippet(potongan) code.
   * Kode yang lainnya tetap sama.
   */

  // Destructing props: state movies
  const { movies, setMovies } = props;
}
```



Movies.js

```
// Menangkap props
function AddMovieForm(props) {
  /**
   * Ini hanya snippet(potongan) code.
   * Kode yang lainnya tetap sama.
   */

  // Destructing props: state movies
  const { movies, setMovies } = props;

  function handleSubmit(e) {
    e.preventDefault();

    const movie = {
      id: nanoid(), title: title,
      year: date, type: "Movie",
      poster: "https://picsum.photos/300/400",
    };

    // SOLVED: HOW TO ADD MOVIE TO MOVIES :)
    setMovies([...movies, movie]);
  }
}
```

Conditional Rendering

Conditional Rendering

- Render Component based on condition (state).
- Conditional in React is similar to JavaScript.

Type:

- Conditional: If, Else If, Else
- Inline If: Operator `&&`.
- Inline If-Else: Conditional Operator.

Referensi: [React- Conditional Rendering](#).



AddMovieForm.js

```
/**
 * Ini hanya snippet code.
 * Kode lainnya tetap sama
 */


Title Wajib Diisi


```



AddMovieForm.js

```
/**
 * Ini hanya snippet code.
 * Kode lainnya tetap sama
 */


Date Wajib Diisi


```

How to show error if form is empty?



AddMovieForm.js

```
function AddMovieForm(props) {  
  /**  
   * Ini hanya snippet code.  
   * Kode lainnya tetap sama.  
   */  
  
  // Membuat state: isTitleError, isDateError  
  const [isTitleError, setIsTitleError] = useState(false);  
  const [isDateError, setIsDateError] = useState(false);  
}
```



AddMovieForm.js

```
<input
  id="title"
  type="text"
  value={title}
  onChange={handleTitle}
/>

/**
 * Menambahkan Conditional Operator
 * Jika isTitleError true maka render error
 * Jika isTitleError false maka render empty string
 */
{isTitleError ? <p>Title Wajib Diisi</p> : ""}
```



AddMovieForm.js

```
<input
  id="date"
  type="text"
  value={date}
  onChange={handleDate}
/>

/**
 * Menambahkan Conditional Operator
 * Jika isDateError true maka render error
 * Jika isDateError false maka render empty string
 */
{isDateError ? <p>Date Wajib Diisi</p> : ""}
```



AddMovieForm.js

```
function AddMovieForm(props) {  
  function handleSubmit(e) {  
    // Jika title kosong, set isTitleError menjadi true  
    if (title === "") {  
      setIsTitleError(true);  
    }  
    // Jika date kosong, set isDateError menjadi true  
    else if (date === "") {  
      setIsDateError(true);  
    }  
    // Jika tidak, push movie dan set error menjadi false  
    else {  
      const movie = {  
        id: nanoid(), title: title,  
        year: date, type: "Movie",  
        poster: "https://picsum.photos/300/400",  
      };  
      setMovies([...movies, movie]);  
  
      // set error menjadi false  
      setIsTitleError(false);  
      setIsDateError(false);  
    }  
  }  
}
```




AddMovieForm.js

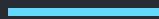
```
<input
  id="title"
  type="text"
  value={title}
  onChange={handleTitle}
/>
/**
 * Membuat inline if: operator &&
 * Jika isTitleError true maka render error
 */
{
  isTitleError && <p>Title Wajib Diisi</p>
}
```



AddMovieForm.js

```
<input
  id="date"
  type="text"
  value={date}
  onChange={handleDate}
/>
/**
 * Membuat inline if: operator &&
 * Jika isTitleError true maka render error
 */
{
  isTitleError && <p>Date Wajib Diisi</p>
}
```

Composition



Composition

- Reuse code between components
- React has a powerful composition model

Type:

- Containment: Send children (prop) to component.
- Specialization: Component be “special cases” of other components.

Referensi: [React- Composition](#).

Containment

- Some components don't know their children.
- Use special `children` prop to pass children into component.

Referensi: [React- Composition](#).



AddMovieForm.js

```
<input
  id="title"
  type="text"
  value={title}
  onChange={handleTitle}
/>
/**
 * Membuat inline if: operator &&
 * Jika isTitleError true maka render error
 */
{
  isTitleError && <p>Title Wajib Diisi</p>
}
```



AddMovieForm.js

```
<input
  id="date"
  type="text"
  value={date}
  onChange={handleDate}
/>
/**
 * Membuat inline if: operator &&
 * Jika isTitleError true maka render error
 */
{
  isTitleError && <p>Date Wajib Diisi</p>
}
```

Why don't create Error Component?
and send children as props?



AddMovieForm.js

```
// Import Component Alert
import Alert from "../Alert/Alert";

function AddMovieForm(props) {
  /**
   * Ini hanya snippet code.
   * Kode lainnya tetap sama.
   */
  return(
    <input
      id="title"
      type="text"
      value={title}
      onChange={handleTitle}
    />
    {isTitleError && <Alert>Title Wajib Diisi</Alert>}
  )
}
```



AddMovieForm.js

```
// Import Component Alert
import Alert from "../Alert/Alert";

function AddMovieForm(props) {
  /**
   * Ini hanya snippet code.
   * Kode lainnya tetap sama.
   */
  return(
    <input
      id="date"
      type="text"
      value={date}
      onChange={handleDate}
    />
    {isDateError && <Alert>Date Wajib Diisi</Alert>}
  )
}
```



AddMovieForm.js

```
function Alert(props) {  
  /**  
   * Children adalah props khusus.  
   * Berisi konten/children yang ada di dalam component.  
   */  
  const { children } = props;  
  
  return (  
    <div>  
      <span>{children}</span>  
    </div>  
  );  
}  
  
export default Alert;
```

- Composition
- Send children as props (Containment)

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.
- Forms: Handle form
- Lifting State: Mengangkat state (shareable).
- Conditional Rendering: Render component berdasarkan kondisi (state).
- Composition: Menyusun component agar dapat digunakan kembali.

Referensi: [Quick Start - React](#).

Refactor



Refactor

- Forms: Handle multiple input.
- Forms: Handle multiple input and validation.
- Navbar: Toggle Navbar.

QnA

Take Away

TODO

- Buat tag `input` untuk memasukkan link gambar.
- Buat tag `select` dan `option` untuk memilih type movie: Action, Drama, Horror, Comedy, dll.
- Handle tag input dan select menggunakan `state`.
- Pastikan movie baru muncul sesuai dengan data yang diinput di form.

Hint:

- Handle select: [React - Form](#).

Assignment

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

Attendance

—

Thanks

