

Pengenalan tentang form pada React.js

Pengertian:

Form di React.js digunakan untuk menangani input dari pengguna. Dengan menggunakan state, kita dapat mengelola data input dan merespon perubahan input secara dinamis. React.js memanfaatkan `useState` untuk mengelola nilai dari form. `useState` adalah hook yang mengembalikan sepasang nilai: state saat ini dan fungsi untuk memperbarui state tersebut. Ketika `useState` dipanggil pertama kali, ia menerima argumen berupa nilai awal dari state.

Pembelajaran pada modul:

Dalam modul ini, mahasiswa diajarkan untuk membuat halaman form sederhana di React.js.

- Proses dimulai dengan membuat folder baru di dalam folder `components` dan menambahkan file `Form.js` di dalamnya.
- Langkah pertama adalah mengimport objek yang diperlukan dari React, seperti `useState`, yang digunakan untuk mengelola state.
- Setelah itu, `useState` dideklarasikan untuk menyimpan dan mengubah nilai state. Fungsi `return` diisi dengan elemen form input dan tombol `SetNama`.
- Ketika tombol diklik, nilai state akan berubah sesuai input. Selanjutnya, mahasiswa diminta membuat form login di file `LoginForm.js`, yang juga menggunakan `useState` untuk menyimpan `username`, `password`, dan `loggedInUsername`. Fungsi `handleLogin` dideklarasikan untuk menangani login, dan hasil login ditampilkan dengan pemanggilan fungsi `LoggedInUsername`. Mahasiswa juga membuat logika `if-else` untuk validasi login, memastikan hanya admin dan password yang valid.

Kesimpulan:

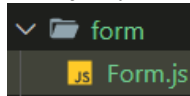
Dalam modul ini, mahasiswa belajar bagaimana membuat dan mengelola form di React.js menggunakan state. Dengan mengikuti langkah-langkah untuk membuat form dan login form, mahasiswa mendapatkan pemahaman praktis tentang bagaimana React.js menangani input dan perubahan state secara dinamis. Test case yang diberikan memperkuat pemahaman dengan implementasi logika dasar `if-else` dalam aplikasi form login.

Membuat halaman form sederhana:

1. Buatlah Folder baru di dalam folder components dengan nama form



2. Selanjutnya Buatlah file di dalam folder form dengan nama Form.js



3. Kemudian pada file Form.js baris pertama diisi dengan mengimport objek dari react.js yang diperlukan untuk menulis komponen React seperti Berikut ini.

```
1 import React, { useState } from 'react';
```

useState memiliki fungsi sebagai mengembalikan sepaasang nilai:state saat ini dan memperbarui state. Saat useState ini dipanggil pertama kali useState menerima argument yaitu nilai awal dari state tersebut.

4. Kemudian tambahkan export default agar file Form dapat diakses oleh file App.js dan ditampilkan pada halaman website.

```
export default function Form() {
```

5. Kemudian deklarasikan useState yang akan digunakan untuk menyimpan nilai state dan mengubah nilai state tersebut.

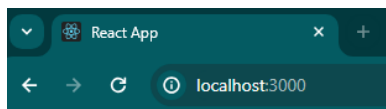
```
const [name, setName] = useState('');  
const [inputName, setInputName] = useState('');
```

6. Selanjutnya pada fungsi return isi dengan Informasi atau topik yang sudah di deklarasikan pada baris sebelumnya, seperti memanggil fungsi nama, kemudian meletakkan form input, dan button SetNama.

```
return (  
  <div>  
    <h1>{name}</h1>  
    <input  
      type="text"  
      name="inputNama"  
      value={inputName}  
      onChange={(e) => setInputName(e.target.value)}  
    />  
    <button className="set-button" onClick={() => setName(inputName)}>Set Nama</button>  
  </div>  
);
```

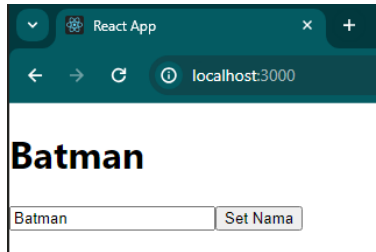
7. Setelah melakukan semua perintah di atas maka tampilan pada halaman website akan seperti Berikut ini.

Sebelum di Set Nama

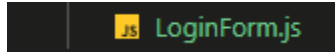


Set Nama

Sesudah di klik Set Nama



8. Selanjutnya membuat form login. Buatlah file baru bernama LoginForm.js



9. Kemudian selanjutnya pada halaman export ada perubahan yaitu terdapat fungsi useState username, password dan loggedInUsername.

```
tabnine: test | explain | document | ask
export default function LoginForm() {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [loggedInUsername, setLoggedInUsername] = useState('');
```

10. Berikutnya deklarasi dari handleLogin.

```
const handleLogin = () => {
  setLoggedInUsername(username);
};
```

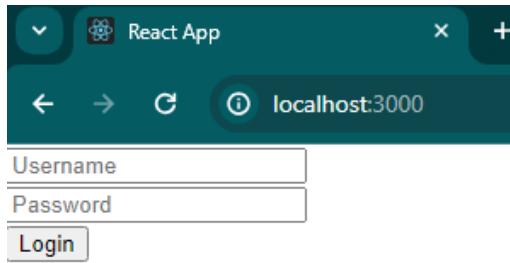
11. Selanjutnya pada fungsi return terdapat penambahan pemanggilan fungsi LoggedInUsername untuk menampilkan username pada halaman baru.

```
<div>
  {loggedInUsername ? (
    <h1>Halo, {loggedInUsername}!</h1>
  ) : (
    <div>
```

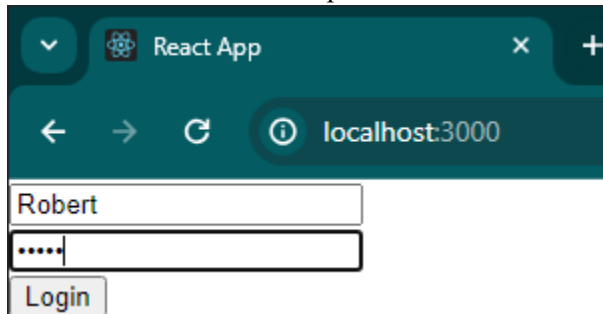
12. Kemudian berikut ini isi dari form input yaitu berupa username, password, dan button Login untuk ditampilkan pada halaman website.

```
<div>
  <input
    type="text"
    placeholder="Username"
    value={username}
    onChange={(e) => setUsername(e.target.value)}
  />
  <br />
  <input
    type="password"
    placeholder="Password"
    value={password}
    onChange={(e) => setPassword(e.target.value)}
  />
  <br />
  <button onClick={handleLogin}>Login</button>
</div>
)}
</div>
```

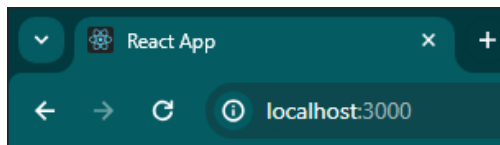
13. Tampilan dari website .



14. Memasukan username dan password.



Hasil

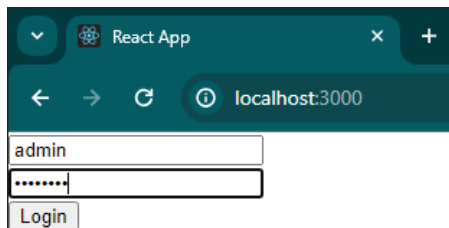


Halo, Robert!

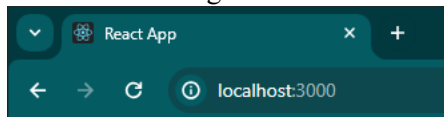
Test Case

Buatlah form login memiliki permissalan if else agar hanya username 'admin' dan password 'password' yang dapat lanjut ke tampilan halaman baru seperti Berikut ini.

Jika berhasil

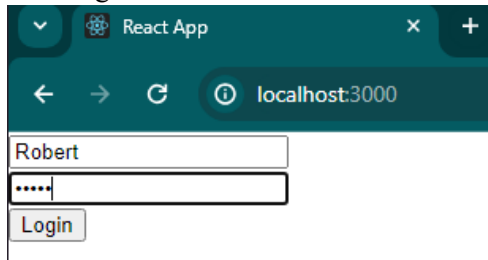


Setelah di klik login

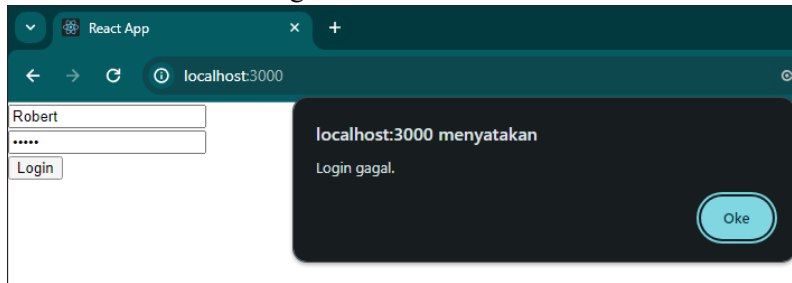


Halo, admin!

Jika Gagal



Setelah diklik tombol login



Clue

```
import React, { useState } from 'react';

tabnine: test | explain | document | ask
export default function TestCase2() {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [isLoggedIn, setIsLoggedIn] = useState(false);

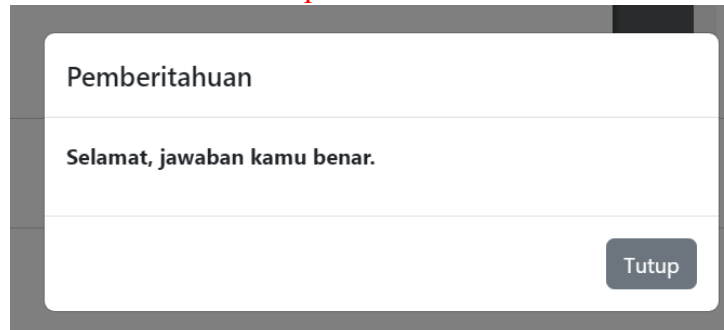
  const handleLogin = () => {
    if ( username && password ) {
      setIsLoggedIn(true);
    } else {
      alert("Login gagal.");
    }
  };

  return (
    <div>
      {isLoggedIn ? (
        <h1>Halo, {username}!</h1>
      ) : (
        <div>
          <input
            type="text"
            placeholder="Username"
            value={username}
            onChange={(e) => setUsername(e.target.value)}
          />
          <br />
          <input
            type="password"
            placeholder="Password"
            value={password}
            onChange={(e) => setPassword(e.target.value)}
          />
          <br />
          <button onClick={handleLogin}>Login</button>
        </div>
      )}
    </div>
  );
}
```

Jika sudah upload file Form.js ke dalam iCLOP:

Warning!

Jika file yang di Upload sesuai dengan perintah praktikum diatas maka akan muncul seperti berikut:



Jika file tidak sesuai dengan perintah diatas praktikum maka akan muncul seperti berikut:

