

# Membuat form login dengan style dan Sweet alert

## Pengertian:

React.js adalah pustaka JavaScript yang digunakan untuk membangun antarmuka pengguna yang interaktif dan dinamis. Dengan menggunakan komponen dan state, React mempermudah pengembangan aplikasi web yang responsif. Dalam modul ini, mahasiswa akan belajar membuat form login dengan styling dan integrasi SweetAlert untuk menampilkan notifikasi.

## Pembelajaran pada modul:

Dalam modul ini, mahasiswa belajar membuat form login dengan styling dan SweetAlert.

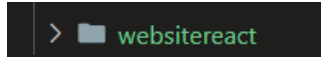
- Langkah pertama adalah membuat folder baru bernama websitereact di dalam folder components, lalu membuat file Login.js di dalam folder tersebut.
- Setelah itu, salin isi dari file Testcase2.js yang telah dibuat pada jobsheet RUII2 ke dalam Login.js.
- Selanjutnya, buat file CSS bernama Login.css di dalam folder websitereact untuk mengubah tampilan Login.js.
- Tambahkan berbagai kelas CSS untuk menata halaman login, seperti .login-page, .login-container, .container, .login-content, dan lainnya.
- Tambahkan perintah import untuk mengakses file CSS dari Login.js. Buat fungsi const untuk mengelola state dan handle login, dan definisikan fungsi return untuk menampilkan form login yang telah dideklarasikan.
- Tambahkan folder images di dalam folder public dan tambahkan gambar React.gif dari link yang diberikan. Deklarasikan Login.js di App.js, jalankan npm start, dan periksa hasilnya.
- Buat file Todolist.js dan Todolist.css di dalam folder websitereact, serta tambahkan gaya untuk tampilan to-do list.
- Buat file Navbar.js, Navbar.css, Footer.js, dan Footer.css untuk membuat tampilan website lebih interaktif. Tambahkan SweetAlert untuk menampilkan notifikasi saat tombol login diklik.
- Terakhir, buat test case untuk menampilkan SweetAlert saat login berhasil atau gagal dan fitur show password.

## Kesimpulan:

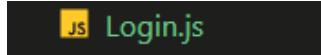
Dalam modul ini, mahasiswa belajar membuat form login yang interaktif dengan menggunakan React.js, styling dengan CSS, dan notifikasi menggunakan SweetAlert. Modul ini memberikan pemahaman tentang bagaimana membangun komponen yang dinamis dan responsif, mengelola state, serta menambahkan elemen interaktif ke dalam aplikasi web. Dengan langkah-langkah yang sistematis, mahasiswa dapat mengembangkan antarmuka pengguna yang lebih menarik dan fungsional.

## Membuat sebuah website User interface interactive sederhana menggunakan React.js:

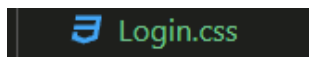
1. Masih dalam 1 project buatlah folder baru bernama websitereact di dalam folder components.



2. Kemudian buat file Login.js



3. Selanjutnya Copy isi pada file Testcase2.js yang telah anda buat pada jobsheet RUII2.
4. Selanjutnya Buatlah file css untuk mengubah tampilan Login.js agar lebih menarik di dalam folder websitereact.



5. Berikut isi dari file Login.css.

```
.login-page {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background-color: #f0f0f0;
}

.login-container {
  text-align: center;
  padding: 20px;
}

.container {
  max-width: 800px;
  margin: 0 auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 8px;
  background-color: #fff;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.login-content {
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

```
.login-image {
  flex: 1;
}

.login-images {
  max-width: 100%;
  height: auto;
}

.login-form {
  flex: 1;
  padding: 20px;
}

input[type="text"],
input[type="password"],
button {
  width: 100%;
  padding: 10px;
  margin-bottom: 15px;
  border: 1px solid #ccc;
  border-radius: 5px;
  font-size: 16px;
}

button {
  background-color: #82caff;
  color: white;
  font-weight: bold;
  cursor: pointer;
  transition: background-color 0.3s;
}

button:hover {
  background-color: #60b5ff;
}

.checkbox-container {
  display: flex;
  align-items: center;
  justify-content: flex-start;
  margin-bottom: 15px;
}

.checkbox-label {
```

```

margin-left: 5px;
}

.error {
  color: #ff0000;
  margin-bottom: 15px;
}

.loading {
  font-style: italic;
}

.additional-container {
  margin-top: 20px;
  padding: 20px;
  background-color: #f9f9f9;
  border-top: 1px solid #ccc;
}

```

6. Agar file Login.js dapat mengakses file css tambahkan perintah import pada file Login.js

```
import './Login.css';
```

7. Kemudian buatlah fungsi const seperti berikut ini.

```

export default function Login() {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [isLoading, setIsLoading] = useState(false);
  const

```

8. Selanjutnya buatlah fungsi retrun seperti berikut ini untuk menampilkan form login yang telah di deklarasikan diatas.

```

return (
  <div className="login-page">
    <div className="login-container">
      {isLoggedIn ? (
        <Home username={username} onLogout={handleLogout} />
      ) : (
        <div className="container">
          <h1>Login</h1>
          <div className="login-content">
            <div className="login-image">
              
            </div>
            <div className="login-form">
              <input
                type="text"
                placeholder="Username"
                value={username}
                onChange={(e) => setUsername(e.target.value)}
              />
            </div>
          </div>
        </div>
      )}
    </div>
  </div>
)

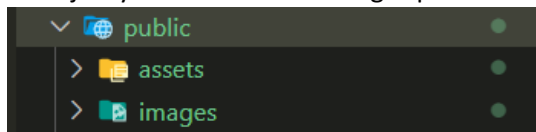
```

```

</div>
<div className="checkbox-container">
  <label className="checkbox-label">
    <input type="checkbox" />
  </label>
</div>
<div>
  <button onClick={handleLogin} disabled={isLoading}>
    {isLoading ? 'Logging in...' : 'Login'}
  </button>
</div>
</div>
<div className="additional-container">
  <h2>Welcome to Reactjs!</h2>
  <p>Belajar User Interactive Interactive Dari Reactjs</p>
</div>
</div>
</div>
);

```

9. Selanjutnya buatlah folder images pada folder public.



10. Downloadlah gambar dari link berikut ini <https://lottiefiles.com/animations/react-logo-iD74FPhCKX> setelah di download masukan gambar yang telah di download ke dalam folder images dengan nama React.gif.

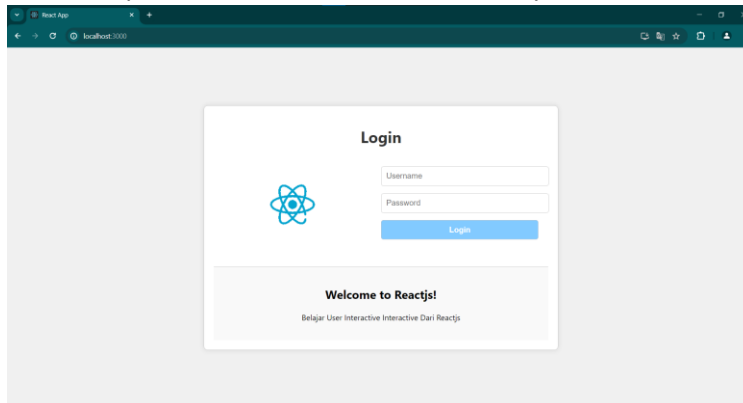
11. Setelah itu deklarasikan file login.js pada file App.js.

```

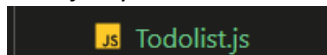
import Login from "./components/websitreact/Login";
function App() {
  return (
    <div className="App">
      <Login/>
    </div>
  );
}

```

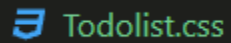
12. Lakukan npm start dan berikut adalah hasilnya.



13. Selanjutnya buatlah file baru bernama Todolist.js di dalam folder websitreact



14. Selanjutnya buatlah file css bernama Todolist.css



15. Berikut adalah isi dari file css diatas.

```
.todo-container {
  max-width: 600px;
  margin: auto;
  padding: 6rem;
  background-color: #f8f9fa;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  text-align: center;
}

.todo-container h2 {
  font-size: 2rem;
  margin-bottom: 1.5rem;
}

.todo-container input[type="text"] {
  width: calc(100% - 80px);
  padding: 0.5rem;
  margin-bottom: 1rem;
  border-radius: 4px;
  border: 1px solid #ced4da;
}

.button-add {
  padding: 0.5rem 1rem;
  margin-left: 1rem;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  background-color: #038af9;
  color: #fff;
}

.todo-container ul {
  list-style-type: none;
  padding: 0;
}

.todo-container li {
  display: flex;
  align-items: center;
}
```

```

margin-bottom: 0.5rem;
}

.delete-button {
margin-left: auto;
background-color: #dc3545;
width: 20%;
}

.delete-button:hover {
background-color: #c82333;
}

```

16. Selanjutnya isi fungsi import seperti berikut ini.

```

import React, { useState } from 'react';
import './Todolist.css';

```

17. Kemudian pada fungsi const deklarasikan fungsi fungsi apa saja yang akan digunakan seperti berikut ini.

```

const Todolist = () => {
const [todos, setTodos] = useState([]);
const [newTodo, setNewTodo] = useState('');

const addTodo = () => {
if (newTodo.trim()) {
setTodos([...todos, newTodo.trim()]);
setNewTodo('');
}
};

const deleteTodo = (indexToDelete) => {
const updatedTodos = todos.filter((_, index) => index !== indexToDelete);
setTodos(updatedTodos);
};
}

```

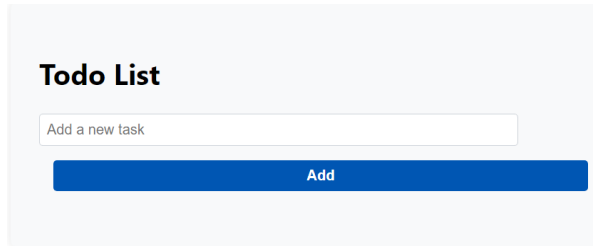
18. Kemudian setelah mendeklarasikan fungsi-fungsi yang digunakan buatlah tampilan pada website agar fungsi-fungsi diatas dapat ditampilkan pada website.

```

return (
<div>
<Navbar />
<div className="todo-container">
<h2>Todo List</h2>
<input
type="text"
value={newTodo}
onChange={(e) => setNewTodo(e.target.value)}
placeholder="Add a new task"
/>
<button onClick={addTodo}>Add</button>
<ul>
{todos.map((todo, index) => (
<li key={index}>
{todo}
<button className="delete-button" onClick={() => deleteTodo(index)}>Delete</button>
</li>
))}
</ul>
</div>
<Footer />
</div>
);

```

19. Berikut adalah tampilan pada website.



20. Selanjutnya agar tampilan dari website lebih interactive buatlah File Navbar.js dan Navbar.css  
Navbar.js:

```
import React from 'react';
import './Navbar.css';

const Navbar = () => {

  return (
    <nav className="navbar">
      <ul className="navbar-list">
        <li className="navbar-item"><a href="/todolist" className="navbar-link">Todolist</a></li>
      </ul>
    </nav>
  );
};

export default Navbar;
```

Navbar.css:

```
h1 {
  text-align: center;
  margin-bottom: 20px;
  color: #333;
}

.navbar {
  background-color: #333;
  padding: 10px 20px;
}

.navbar-list {
  list-style-type: none;
  margin: 0;
  padding: 0;
  text-align: center;
}

.navbar-item {
  display: inline-block;
  margin-right: 20px;
}
```



```

.navbar-link {
  text-decoration: none;
  color: #fff;
}

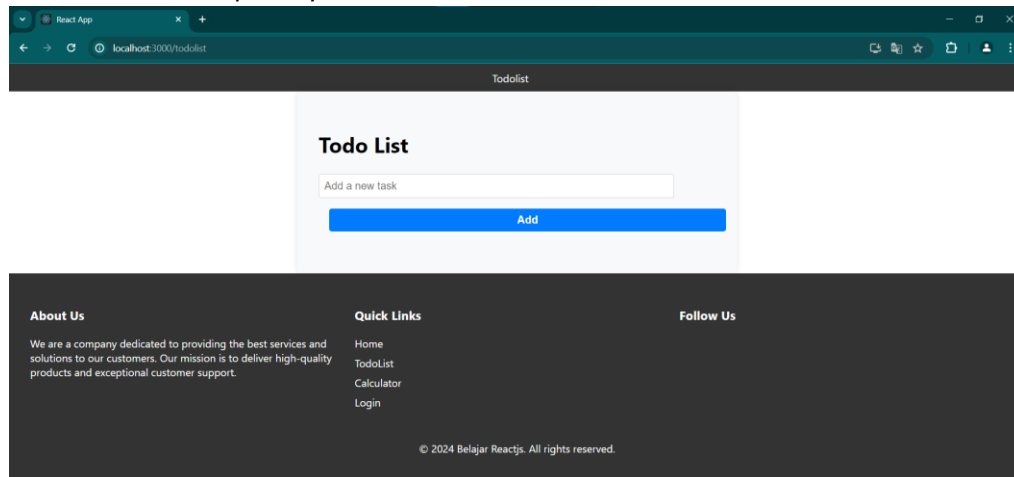
.navbar-link:hover {
  color: #82caff;
}

.logout-button {
  background-color: red;
  color: white;
  border: none;
  padding: 10px 20px;
  font-size: 16px;
  cursor: pointer;
  border-radius: 5px;
}

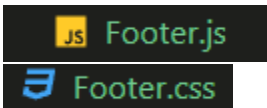
.logout-button:hover {
  background-color: darkred;
}

```

Berikut adalah tampilannya:



21. Selanjutnya kita membuat file Footer.js dan Footer.css agar tampilan dari sebuah website lebih interactive



Berikut adalah isi file css.

```

.footer {
  background-color: #333;
  color: #fff;
}

```

```
padding: 2rem;
}

.footer-content {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
grid-gap: 1rem;
}

.footer-section {
margin-bottom: 1rem;
}

.footer-heading {
color: #fff;
}

.footer-section ul {
list-style-type: none;
padding: 0;
margin: 0;
}

.footer-section ul li {
margin-bottom: 0.5rem;
}

.footer-section ul li a {
color: #fff;
text-decoration: none;
}

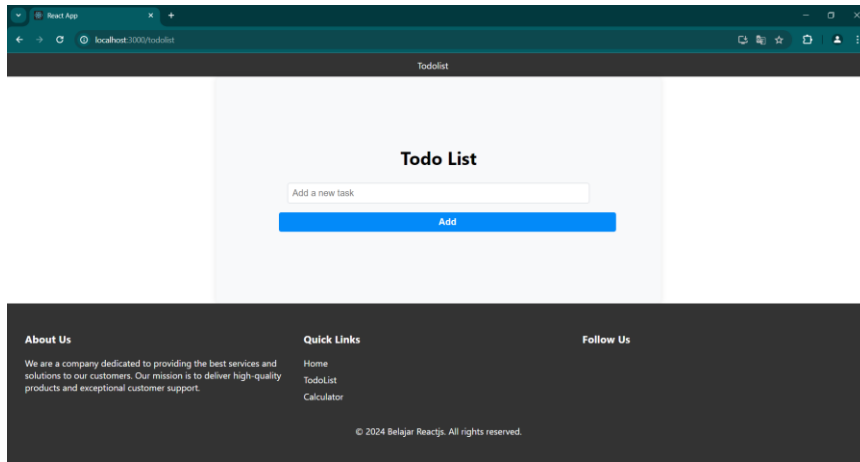
.footer-bottom {
margin-top: 1rem;
text-align: center;
}
```

Berikut adalah isi dari file Footer.js

```
import React from 'react';
import './Footer.css'

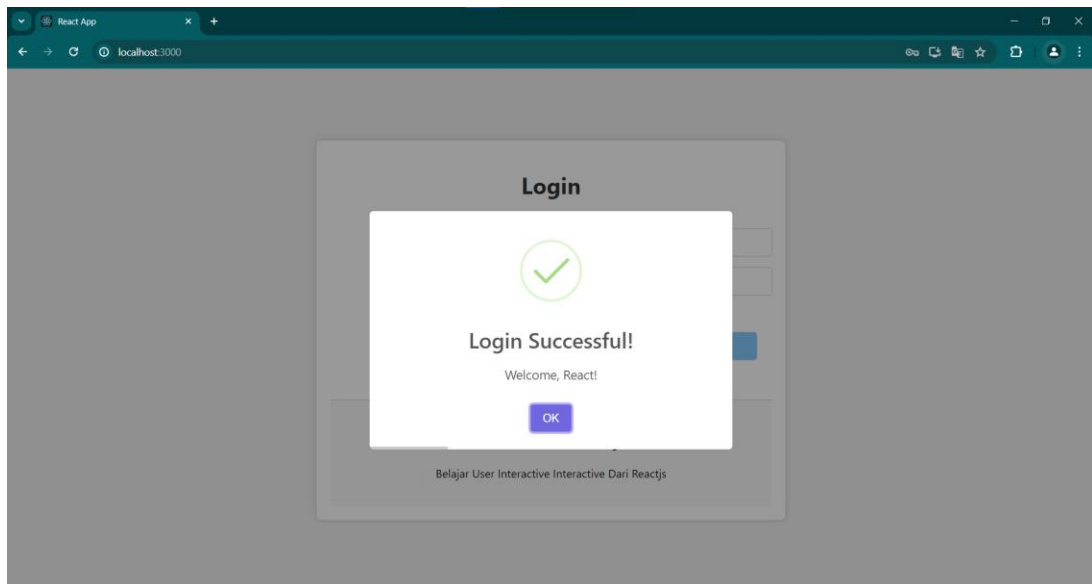
const Footer = () => {
  return (
    <footer className="footer">
      <div className="footer-content">
        <div className="footer-section about">
          <h3 className="footer-heading">About Us</h3>
          <p>We are a company dedicated to providing the best services and solutions to our customers. Our mission is to deliver high-quality products and exceptional customer support.</p>
        </div>
        <div className="footer-section links">
          <h3 className="footer-heading">Quick Links</h3>
          <ul>
            <li><a href="/todolist">Todolist</a></li>
          </ul>
        </div>
        <div className="footer-section social">
          <h3 className="footer-heading">Follow Us</h3>
        </div>
      </div>
      <div className="footer-bottom">
        <p>© 2024 Belajar Reactjs. All rights reserved.</p>
      </div>
    </footer>
  );
};
```

Berikut adalah tampilannya.

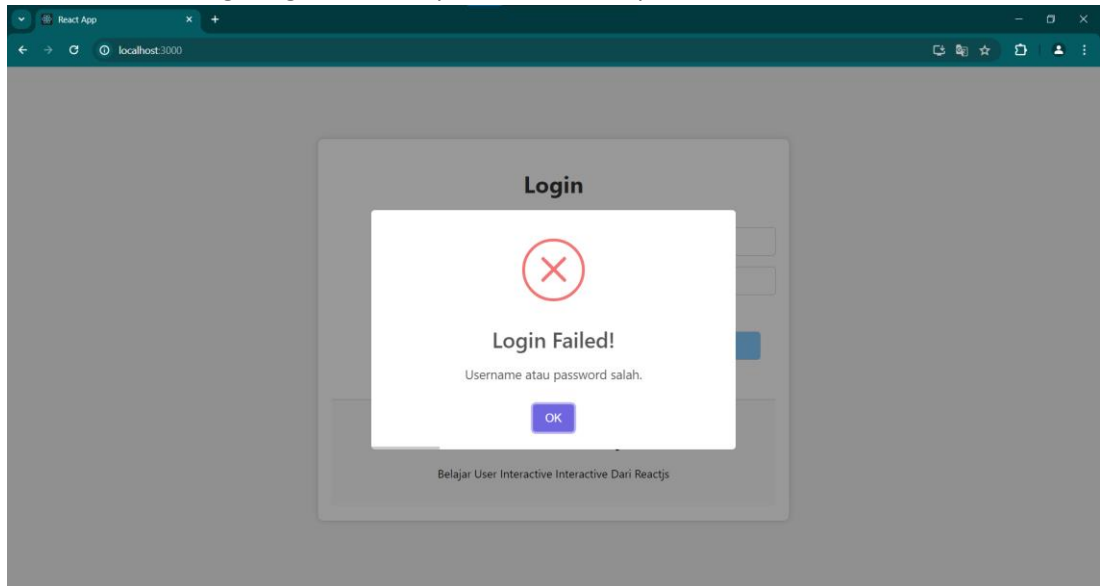


Test case 1

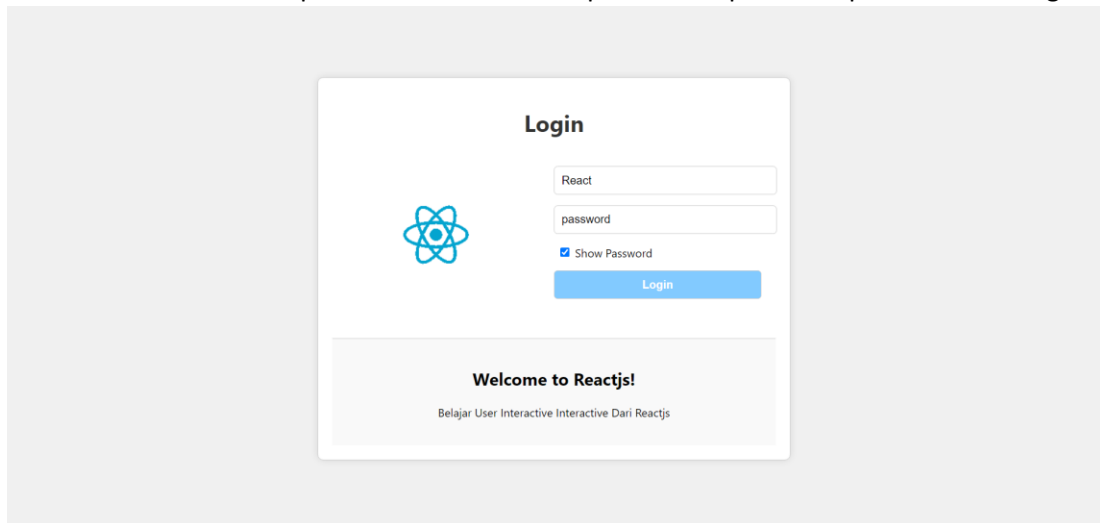
Buatlah sweet alert ketika button login di klik maka akan menampilkan sweet alert seperti berikut ini.



Kemudian saat Gagal login akan tampil sweet alert seperti berikut ini.



Kemudian buatlah show password untuk menampilkan form password pada halaman login



Clue

Instalasi sweetalert

```
npm install sweetalert2
```

## Fungsi Const

```
export default function Login() {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [isLoading, setIsLoading] = useState(false);

  const handleLogin = () => {
    setIsLoading(true);
    setTimeout(() => {
      if (username === 'React' && password === 'password') {
        Swal.fire({
          title: 'Success',
          text: 'Login successful!',
          icon: 'success',
          confirmButtonText: 'OK'
        });
      } else {
        Swal.fire({
          title: 'Error',
          text: 'Invalid credentials!',
          icon: 'error',
          confirmButtonText: 'OK'
        });
      }
      setIsLoading(false);
    }, 1000);
  };
};
```

Fungsi return

```

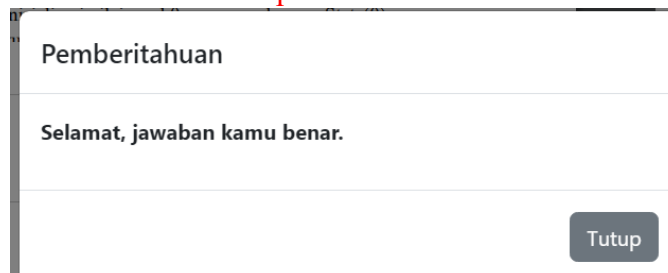
return (
  <div className="login-page">
    <div className="login-container">
      <div className="container">
        <h1>Login</h1>
        <div className="login-content">
          <div className="login-image">
            
          </div>
          <div className="login-form">
            <input
              type="text"
              placeholder="Username"
              value={username}
              onChange={(e) => setUsername(e.target.value)}
            />
            <input
              type={showPassword ? "text" : "password"}
              placeholder="Password"
              value={password}
              onChange={(e) => setPassword(e.target.value)}
            />
            <div className="checkbox-container">
              <label className="checkbox-label">
                <input
                  type="checkbox"
                />
                Show Password
              </label>
            </div>
            <button onClick={handleLogin} disabled={isLoading}>
              {isLoading ? 'Logging in...' : 'Login'}
            </button>
          </div>
        </div>
        <div className="additional-container">
          <h2>Welcome to Reactjs!</h2>
          <p>Belajar User Interactive Interaktif Dari Reactjs</p>
        </div>
      </div>
    </div>
  </div>
);

```

**Jika sudah upload file FormStyle.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:

