

Membuat sebuah project website dan menghubungkan component menjadi beberapa halaman menggunakan Route

Pengertian:

Proyek ini bertujuan untuk membuat sebuah website dengan menggunakan React.js dan menghubungkan beberapa komponen menjadi beberapa halaman menggunakan Route. Dengan demikian, pengguna dapat berpindah antara halaman-halaman yang berbeda dengan lancar.

Pembelajaran pada modul:

Dalam modul ini, mahasiswa akan mempelajari bagaimana membuat sebuah kalkulator sederhana dengan React.js.

- Langkah pertama adalah membuat file Calculator.js di dalam folder components dan file Calculator.css untuk mengatur tampilan kalkulator.
- Kemudian, import file CSS untuk menambahkan style, serta import komponen Navbar dan Footer untuk menampilkan navbar pada halaman kalkulator.
- Selanjutnya, dalam fungsi const, mahasiswa akan mendeklarasikan fungsi-fungsi untuk melakukan operasi pada kalkulator. Untuk menampilkan fungsi operasi tersebut, mahasiswa akan mendeklarasikan view pada fungsi return, serta menambahkan Navbar dan Footer.

Selain itu, mahasiswa juga akan mempelajari bagaimana membuat halaman Home dengan file Home.js dan Home.css.

- File CSS tersebut akan digunakan untuk mengatur tampilan halaman Home dengan gaya yang menarik.
- Kemudian, import file Navbar, Footer, dan Navbar.css untuk menambahkan navbar pada halaman Home.
- Setelah itu, mahasiswa akan membuat fungsi const untuk menampilkan Navbar serta Footer, dan menambahkan tampilan pada menu navbar.

Selanjutnya, mahasiswa akan belajar menghubungkan komponen-komponen tersebut menggunakan Route pada React.js.

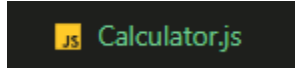
- Pada halaman App.js, akan dilakukan import React root dom dan penambahan fungsi Const dan Return untuk menghubungkan komponen-komponen. Selanjutnya, pada Login.js, mahasiswa akan menambahkan fungsi navigate untuk menghubungkan ke halaman lainnya.

Kesimpulan:

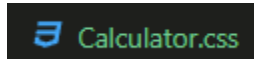
Modul ini memberikan pemahaman yang komprehensif tentang bagaimana membuat sebuah website interaktif menggunakan React.js. Dengan menggunakan komponen-komponen seperti kalkulator dan halaman Home, serta menghubungkannya dengan menggunakan Route, mahasiswa dapat membuat sebuah website yang lebih dinamis dan fungsional. Dengan demikian, mahasiswa memiliki pengetahuan dan keterampilan yang cukup untuk membuat website yang kompleks dan menarik dengan menggunakan React.js.

Langkah-langkah membuat Calculator sederhana, Membuat Halaman Home menghubungkan setiap component dengan route serta navigate:

1. Buatlah File Calculator.js di dalam folder components.



2. Selanjutnya Buatlah file Calculator.css untuk membuat tampilan dari halaman calculator menjadi lebih menarik.



3. Berikut adalah isi dari file Calculator.css

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

.keypad {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 10px;
}

.keypad button {
  padding: 20px;
  background-color: #333;
  color: white;
  border: none;
  cursor: pointer;
}

.keypad button:hover {
  background-color: #555;
}

#clear {
  grid-column: span 2;
}

#result {
  grid-column: span 2;
}
```

4. Selanjutnya pada file Calculator.js tambahkan import use state, import file .css untuk menampilkan style, import Navbar dan import Footer untuk menampilkan Navbar pada halaman calculator.

```
import React, { useState } from 'react';
import './Calculator.css'
import Navbar from './Navbar';
import Footer from './Footer';
```

5. Selanjutnya dalam fungsi const diisi dengan deklarasi dari fungsi-fungsi untuk membuat operasi dari calculator.

```
const Calculator = () => {
  const [result, setResult] = useState('');

  const handleClick = (e) => {
    setResult(result.concat(e.target.name));
  };

  const clear = () => {
    setResult('');
  };

  const backspace = () => {
    setResult(result.slice(0, -1));
  };

  const calculate = () => {
    try {
      setResult(eval(result).toString());
    } catch (err) {
      setResult('Error');
    }
  };
};
```

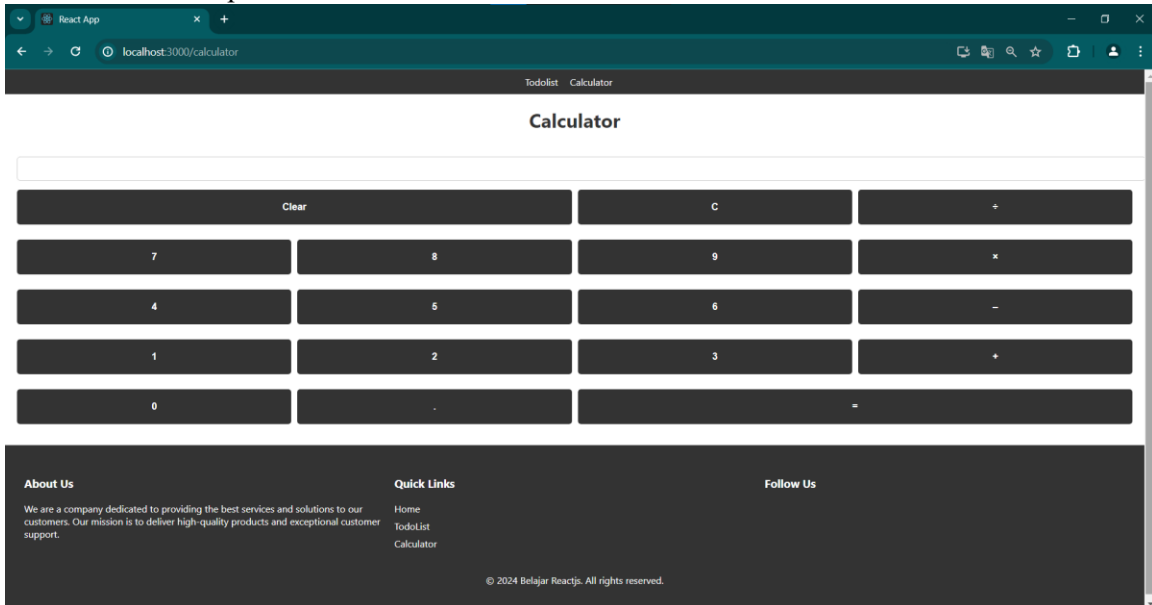
6. Untuk menampilkan fungsi operasi yang berada pada fungsi const deklarasikan view berikut ini pada fungsi return. Deklarasikan <Navbar/> dan <Footer/> untuk menampilkan Navbar.js dan Footer.js.

```
<div>
  <Navbar/>
  <h1>Calculator</h1>
  <div className="calculator-container">
    <form>
      <input type="text" value={result} readOnly />
    </form>
    <div className="keypad">
      <button onClick={clear} id="clear">Clear</button>
      <button onClick={backspace} id="backspace">C</button>
      <button name="/" onClick={handleClick}>&divide;</button>
      <button name="7" onClick={handleClick}>7</button>
      <button name="8" onClick={handleClick}>8</button>
      <button name="9" onClick={handleClick}>9</button>
      <button name="*" onClick={handleClick}>&times;</button>
      <button name="4" onClick={handleClick}>4</button>
      <button name="5" onClick={handleClick}>5</button>
      <button name="6" onClick={handleClick}>6</button>
      <button name="-" onClick={handleClick}>&ndash;</button>
      <button name="1" onClick={handleClick}>1</button>
      <button name="2" onClick={handleClick}>2</button>
      <button name="3" onClick={handleClick}>3</button>
      <button name="+" onClick={handleClick}>+</button>
      <button name="0" onClick={handleClick}>0</button>
      <button name="." onClick={handleClick}>.</button>
      <button onClick={calculate} id="result"></button>
    </div>
  </div>
  <Footer />
</div>
);
```

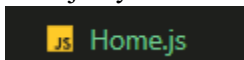
7. Selanjutnya pada file Navbar.js tambahkan source code berikut agar tampil link yang mengakses halaman calculator pada navbar.

```
<li className="navbar-item"><a href="/calculator" className="navbar-link">Calculator</a></li>
```

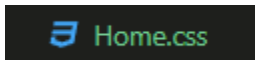
8. Berikut adalah tampilan dari halaman calculator.



9. Selanjutnya membuat halaman Home.js



10. Buatlah file Home.css



11. Berikut adalah isi dari file Home.css.

```
.home-container {  
  font-family: 'Arial', sans-serif;  
  color: #333 ;  
  text-align: center;  
  padding: 20px;  
  background-color: #f9f9f9;  
}  
  
.home-header {  
  background-color: #82caff;  
  color: white;  
  padding: 50px 0;  
}  
  
.home-header h1 {  
  margin: 0;  
  font-size: 3em;  
}
```

```
.home-header p {
  font-size: 1.5em;
  margin: 10px 0 0;
}

.features {
  display: flex;
  justify-content: space-around;
  flex-wrap: wrap;
  margin: 20px 0;
}

.feature-card {
  background-color: white;
  border: 1px solid #ddd;
  border-radius: 8px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  margin: 10px;
  padding: 20px;
  width: 300px;
  transition: transform 0.2s;
}

.feature-card h2 {
  color: #82caff;
  font-size: 1.5em;
  margin-bottom: 10px;
}

.feature-card p {
  color: #666;
  font-size: 1em;
}

.feature-card:hover {
  transform: scale(1.05);
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}

.feature-link {
  text-decoration: none;
  color: #000000;
}
```

```
.feature-link:hover {  
  text-decoration: underline;  
}
```

12. Selanjutnya import file Navbar, Footer, dan file Navbar.css.

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

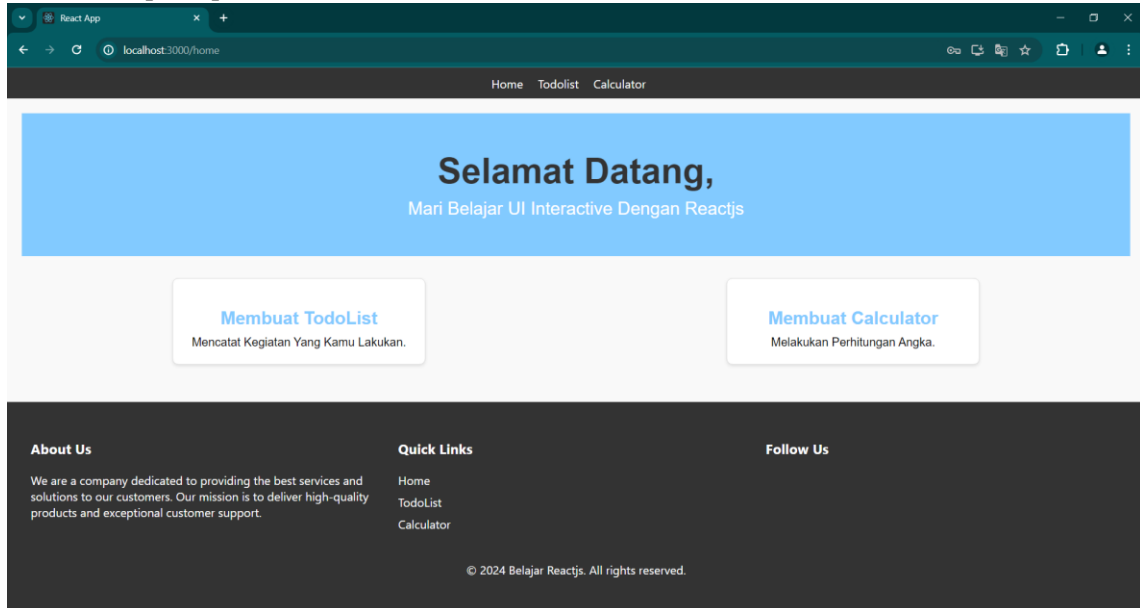
13. Selanjutnya membuat fungsi const dan menampilkan Navbar serta footer

```
const Home = ({ username }) => {  
  return (  
    <div>  
      <Navbar />  
      <div className="home-container">  
        <header className="home-header">  
          <h1>Selamat Datang, {username}</h1>  
          <p>Mari Belajar UI Interactive Dengan Reactjs</p>  
        </header>  
        <section className="features">  
          <div className="feature-card">  
            <h2>Membuat TodoList</h2>  
            <a href="/todolist" className="feature-link">Mencatat Kegiatan Yang Kamu Lakukan.</a>  
          </div>  
          <div className="feature-card">  
            <h2>Membuat Calculator</h2>  
            <a href="/calculator" className="feature-link">Melakukan Perhitungan Angka.</a>  
          </div>  
        </section>  
      </div>  
      <Footer />  
    </div>  
  );  
};
```

14. Tampilkan juga pada menu navbar.

```
<li className="navbar-item"><a href="/home" className="navbar-link">Home</a></li>
```

15. Berikut tampilan pada website.



16. Menghubungkan components. Instalasi lah route pada react.js

```
npm install react-router-dom
```

17. Kemudian pada File App.js import react root dom

```

import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Home from './components/websitereact/Home';
import Login from './components/websitereact/Login';
import Calculator from './components/websitereact/Calculator';
import Todolist from './components/websitereact/Todolist';
import Hello from './components/helloword/Hello';
import { Counter } from './components/counter/Counter';
import LoginForm from './components/form/LoginForm';
import Navbar from './components/websitereact/Navbar';

```

18. Selanjutnya pada fungsi Const dan Return seperti berikut ini.

```

const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Login />} />
        <Route path="/Home" element={<Home />} />
        <Route path="/Todolist" element={<Todolist />} />
        <Route path="/Calculator" element={<Calculator />} />
        <Route path="/Hello" element={<Hello />} />
        <Route path="/LoginForm" element={<LoginForm />} />
        <Route path="/Counter" element={<Counter />} />
        <Route path="/Navbar" element={<Navbar />} />
      </Routes>
    </Router>
  );
};

```

19. Selanjutnya pada Login.js tambahkan navigate untuk menghubungkan ke halaman yang lainnya.

```

import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import Swal from 'sweetalert2';
import './Login.css';

tabnine: test | explain | document | ask
export default function Login() {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [isLoading, setIsLoading] = useState(false);
  const [showPassword, setShowPassword] = useState(false);
  const navigate = useNavigate();

```

20. Pada fungsi return dibagian alert tambagan navigate agar bisa dihubungkan dengan halaman home setelah alert muncul.

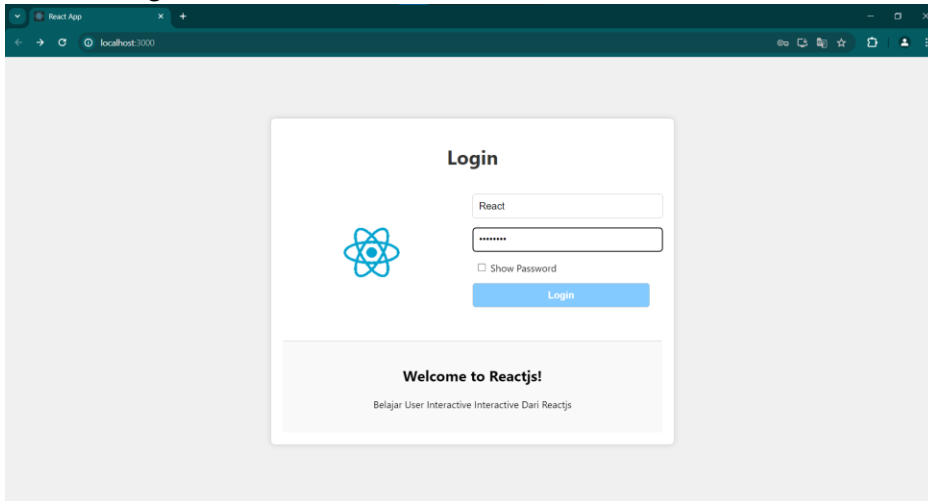
```

timer: 1000,
timerProgressBar: true,
willClose: () => {
  setIsLoading(false);
  navigate('/home');
}
});

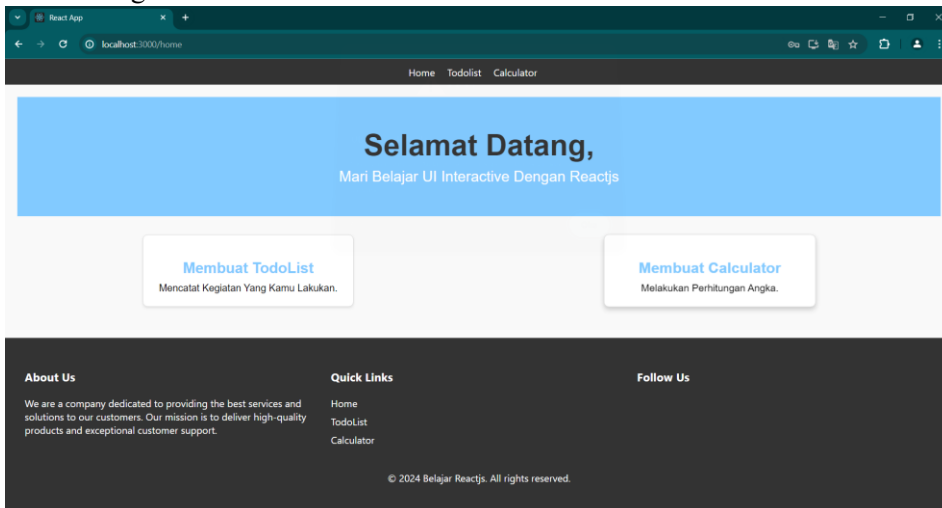
```

21. Maka setelah semua step atas dilakukan semua halaman bisa diakses setelah login seperti berikut ini.

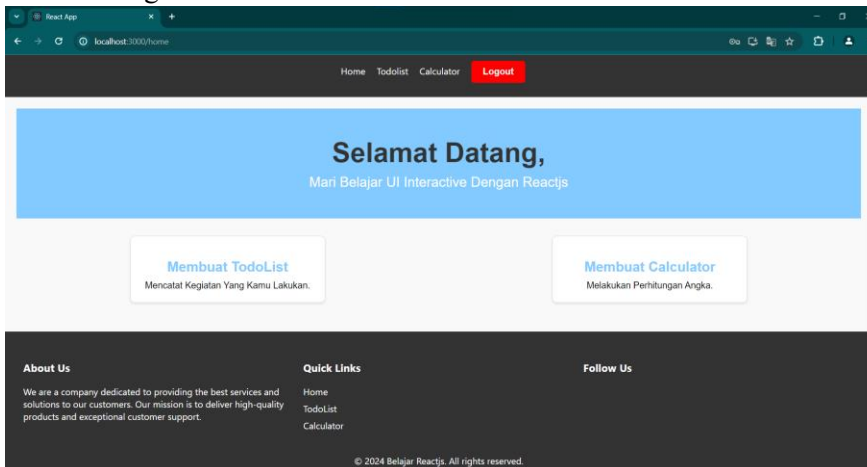
Sebelum Login.



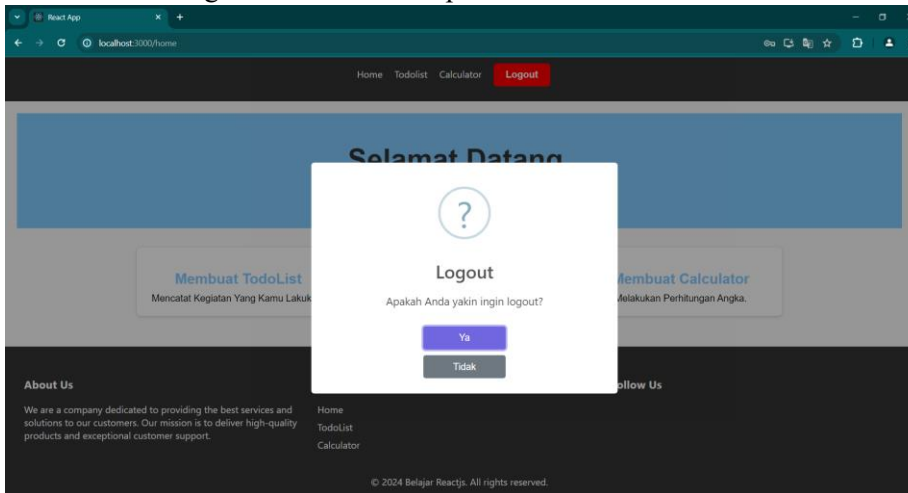
Setelah Login.



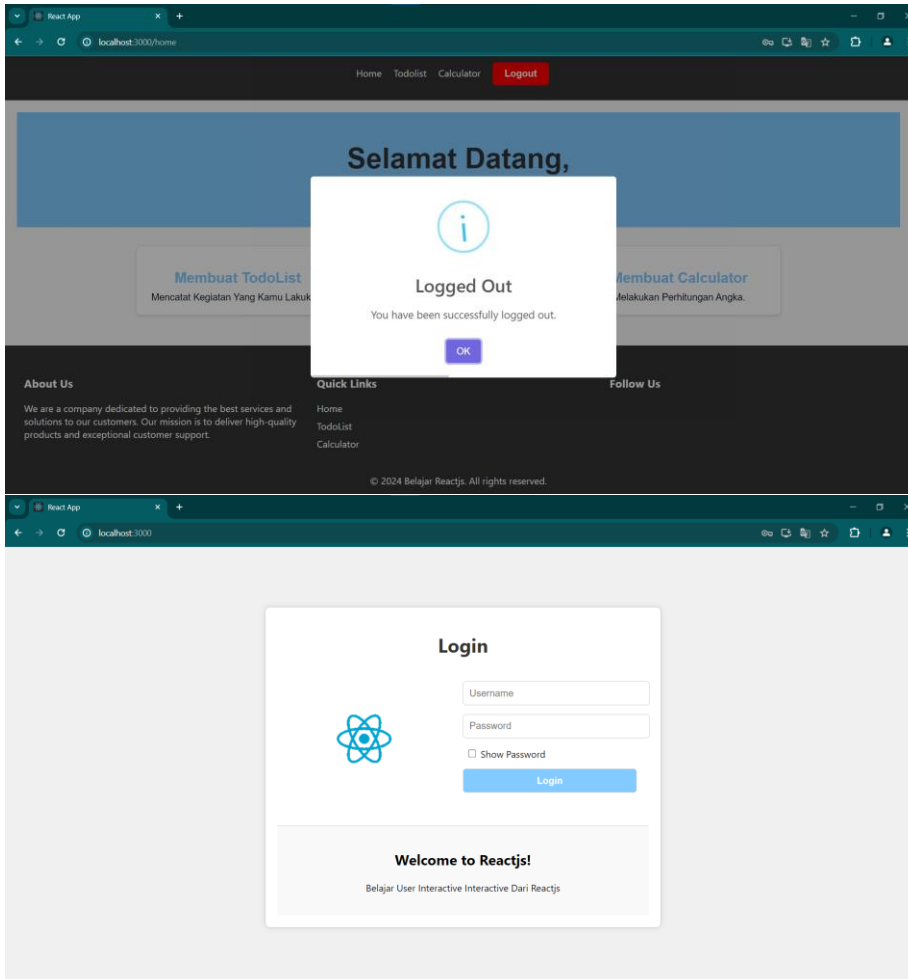
Test Case membuat button Logout pada navbar dan alert berhasil Logout kemudian Kembali ke halaman Login.



Setelah di klik logout akan muncul seperti berikut ini



Selanjutnya ketika di klik Ya maka akan muncul alert dan Kembali ke halaman login seperti berikut ini.



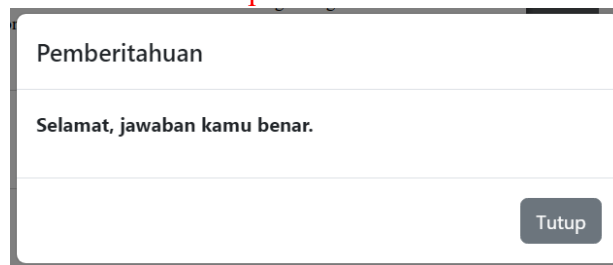
Clue

```
const handleLogout = () => {  
  Swal.fire({  
    title:   
    text:   
    icon:   
    showCancelButton:   
    confirmButtonText:   
    cancelButtonText:   
  }).then((result) => {  
    if (result.isConfirmed) {  
      Swal.fire({  
        title:   
        text:   
        icon:   
        timer: 1000,  
        timerProgressBar: true,  
          
      })  
    }  
  });  
};
```

Jika sudah upload file Navbar.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:

