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 modu:**

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.



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



1. 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;

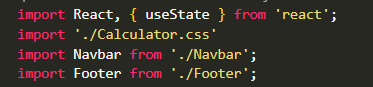
}

#result {

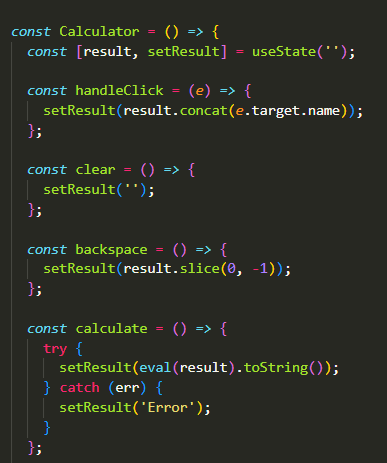
*grid-column*: span 2;

}

1. 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.



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



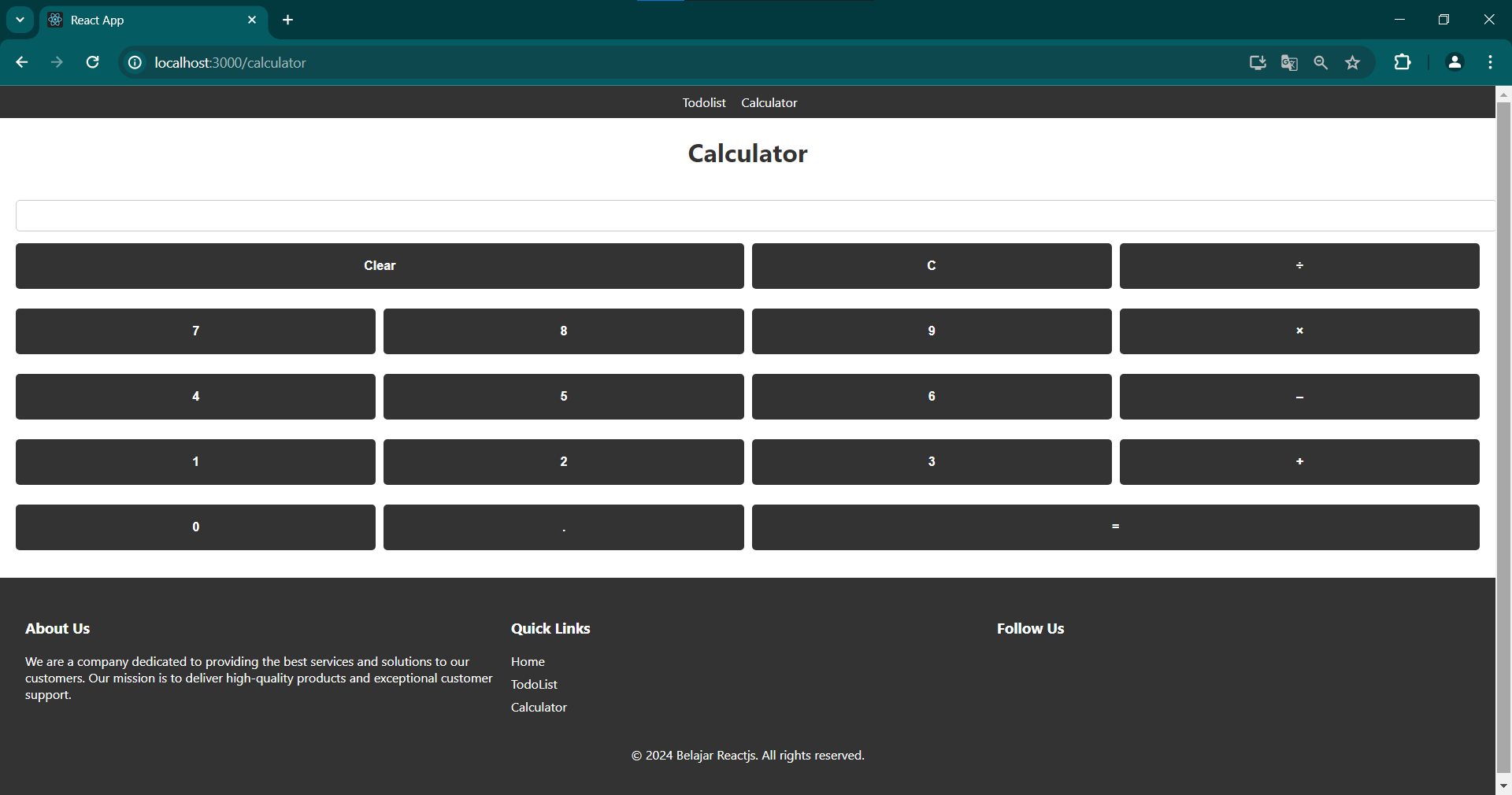
1. Untuk menampilkan fungsi operasi yang berada pada fungsi const deklarasikan view berikut ini pada fungsi return. Deklarasikan <Navbar/> dan <Footer/> untuk menampilkan Navbar.js dan Footer.js.



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



1. Berikut adalah tampilan dari halaman calculator.



1. Selanjutnya membuat halaman Home.js



1. Buatlah file Home.css



1. 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;

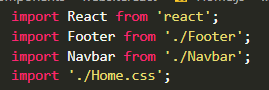
}

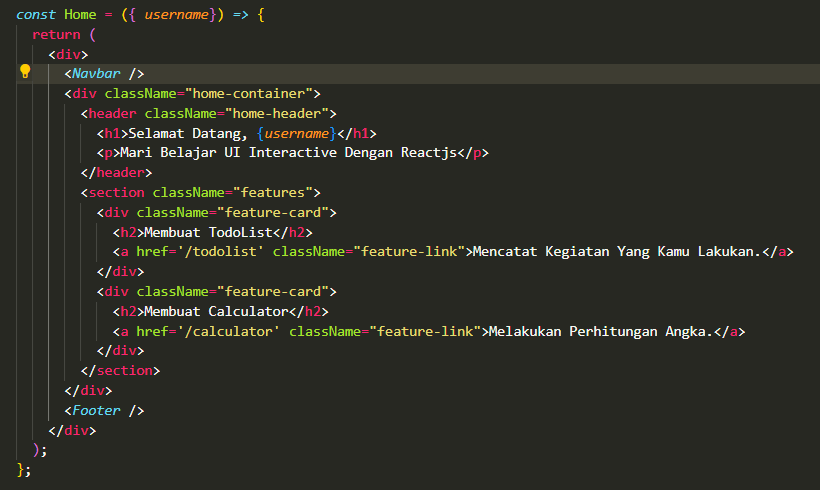
.feature-link:hover {

*text-decoration*: underline;

}

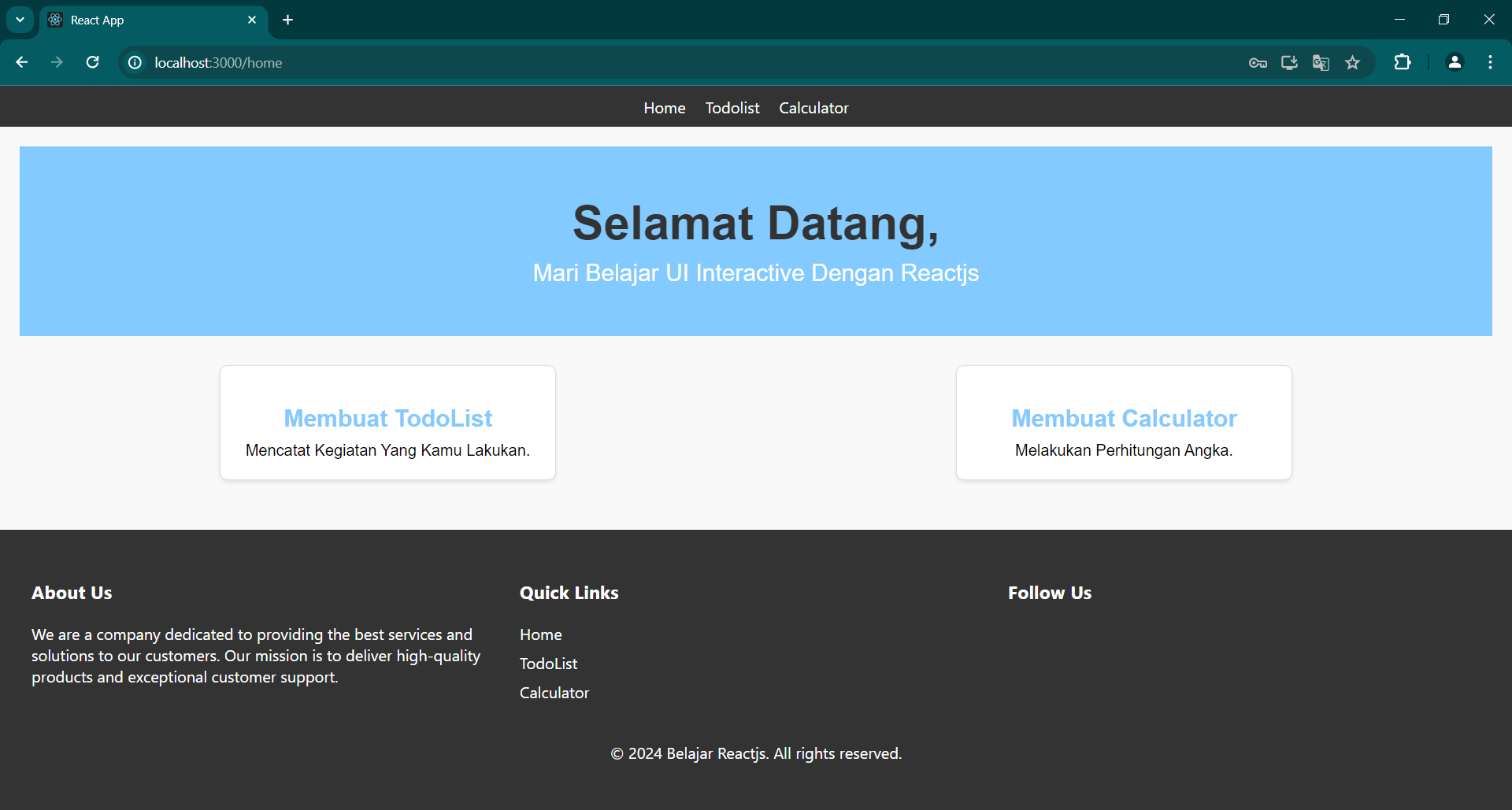
1. Selanjutnya import file Navbar, Footer, dan file Navbar.css.



1. Selanjutnya mebuat fungsi const dan menampilkan Navbar serta footer  
   
2. Tampilkan juga pada menu navbar.



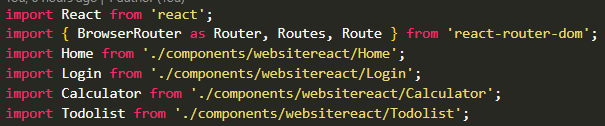
1. Berikut tampilan pada website.



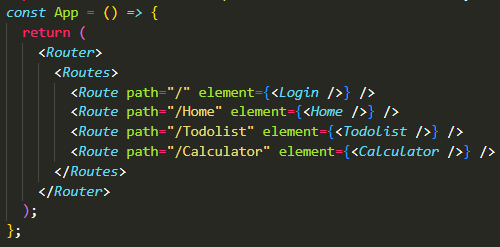
1. Menghubungkan components. Instalasi lah route pada react.js



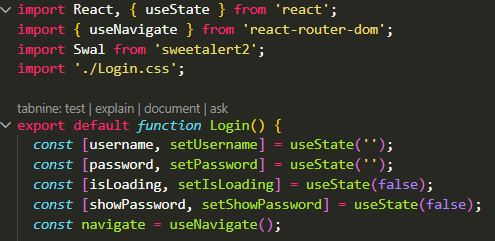
1. Kemudian pada halaman App.js import react root dom



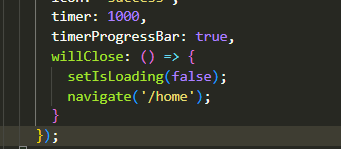
1. Selanjutnya pada fungsi Const dan Return seperti berikut ini.



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

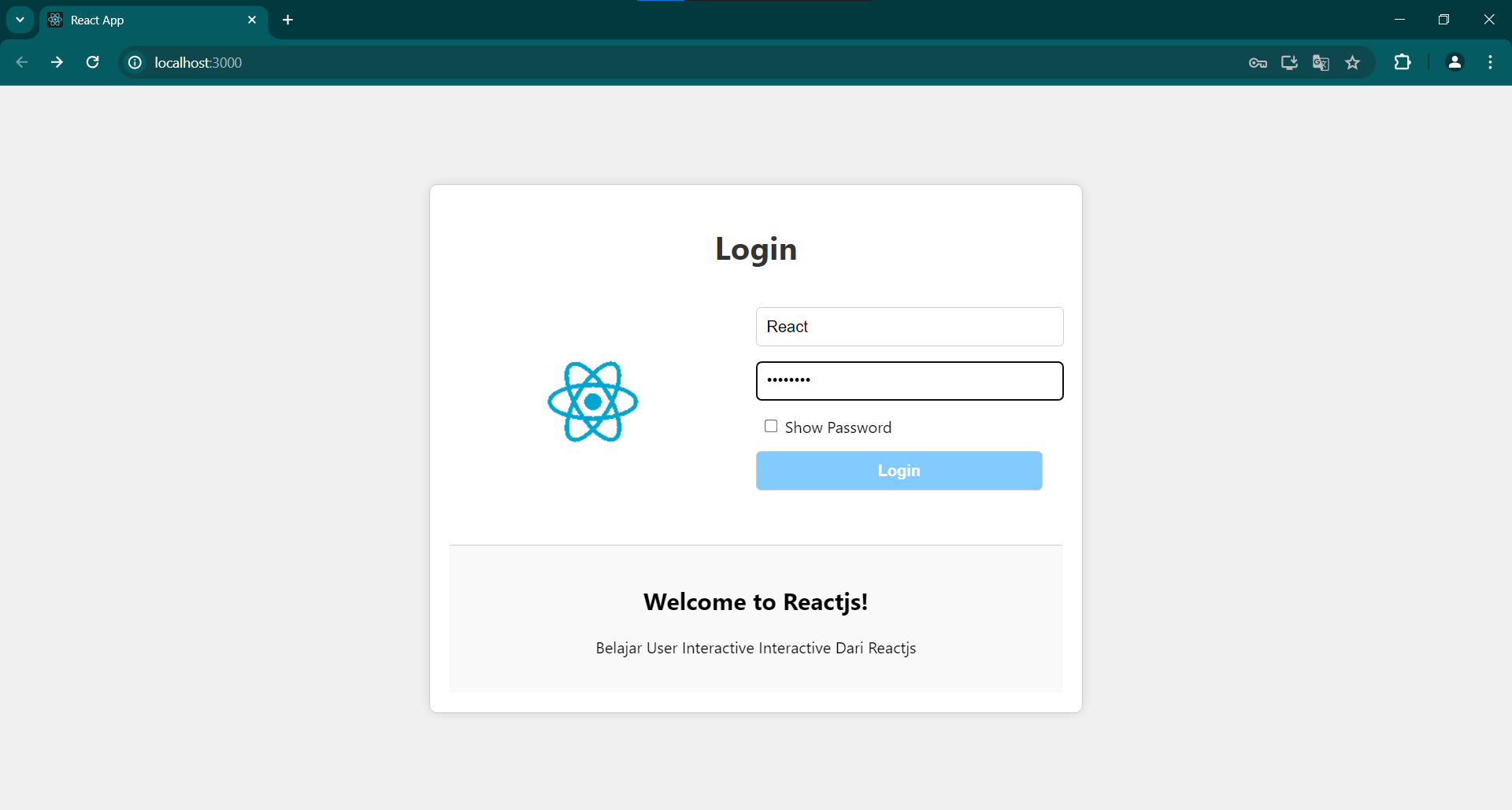


1. Pada fungsi return dibagian alert tambagan navigate agar bisa dihubungkan dengan halaman home setelah alert muncul.

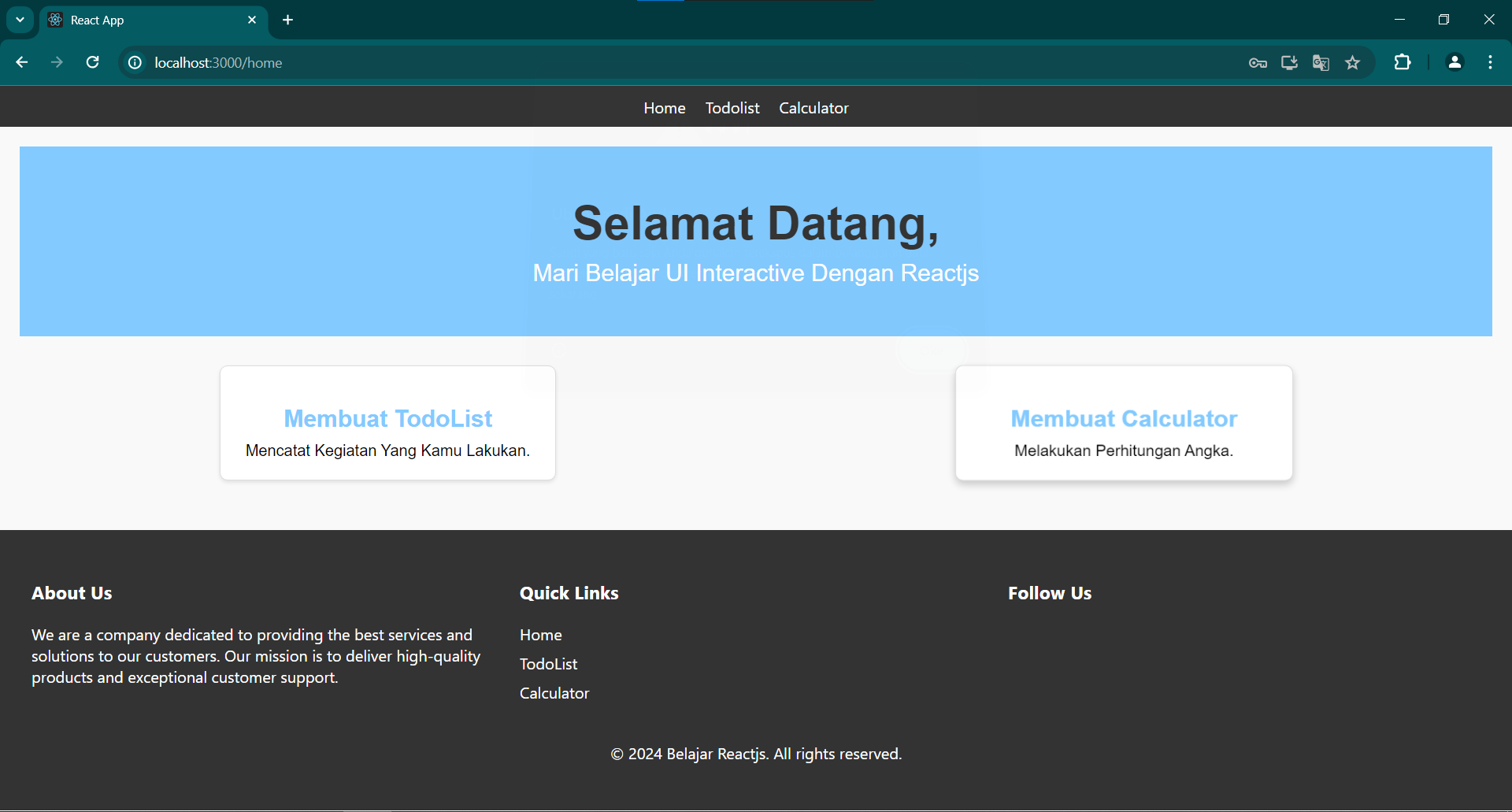


1. Maka setelah semua step atas dilakukan semua halaman bisa diakses setelah login seperti berikut ini.

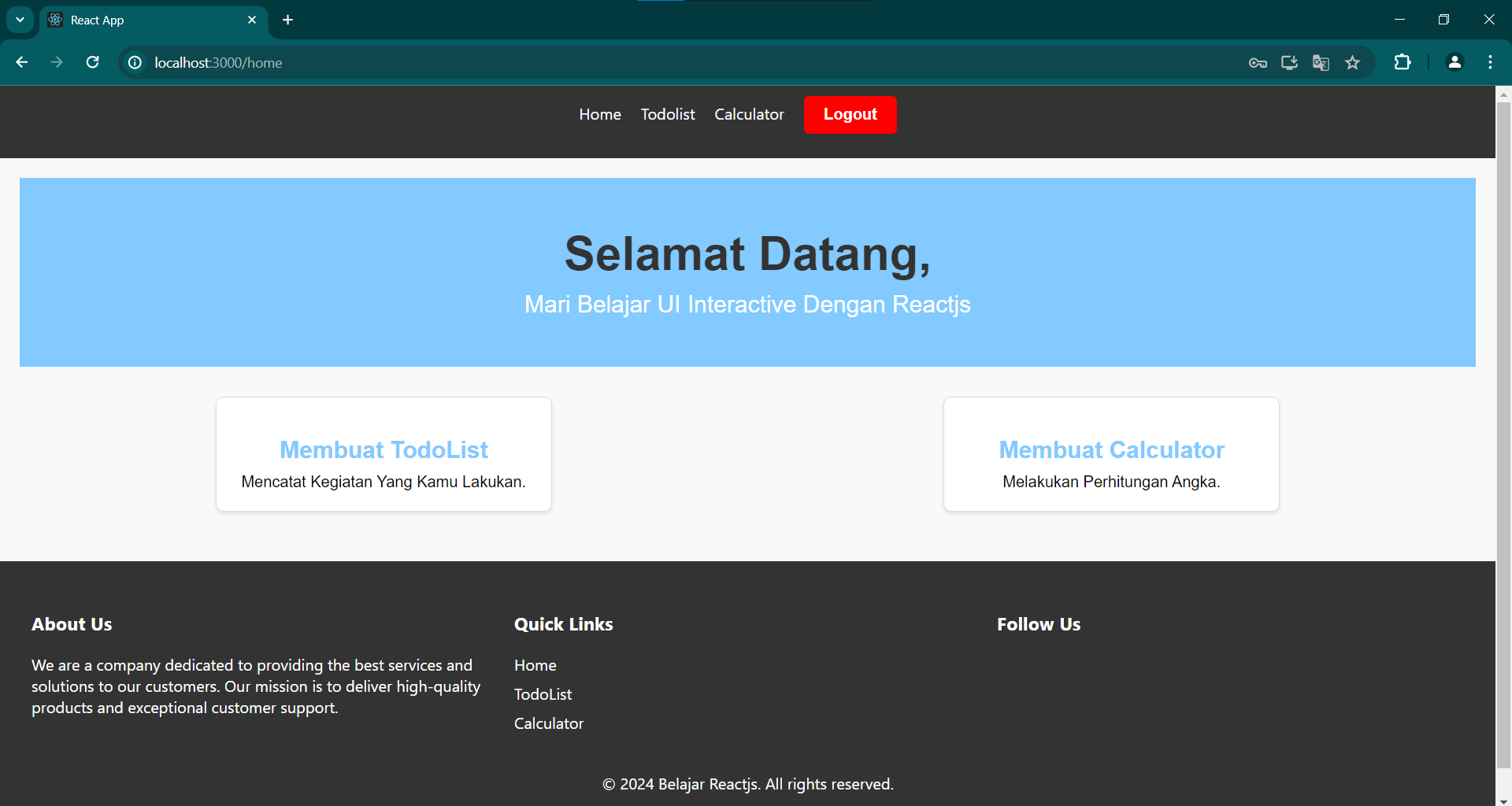
Sebelum Login.



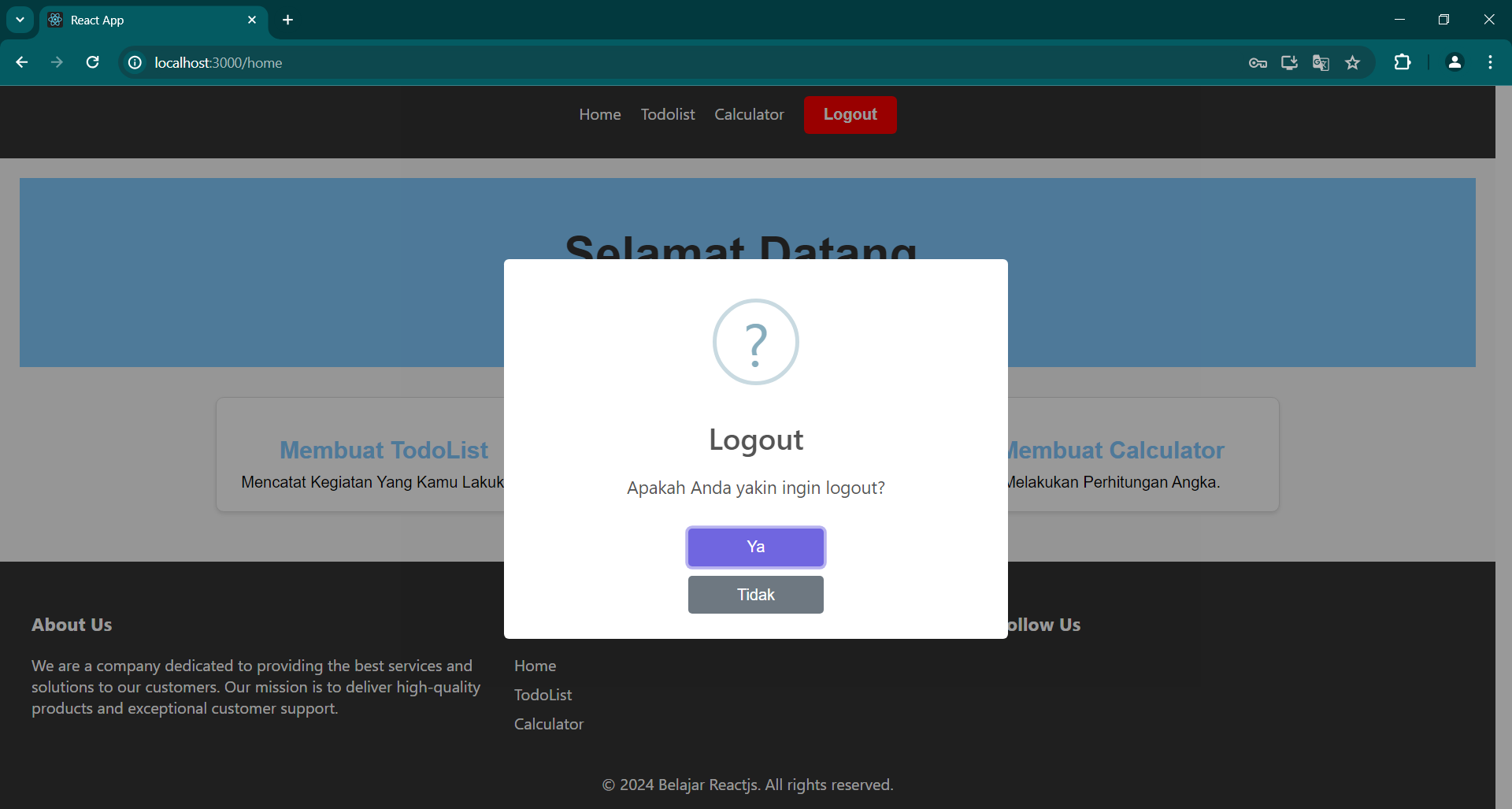
Setelah Login.



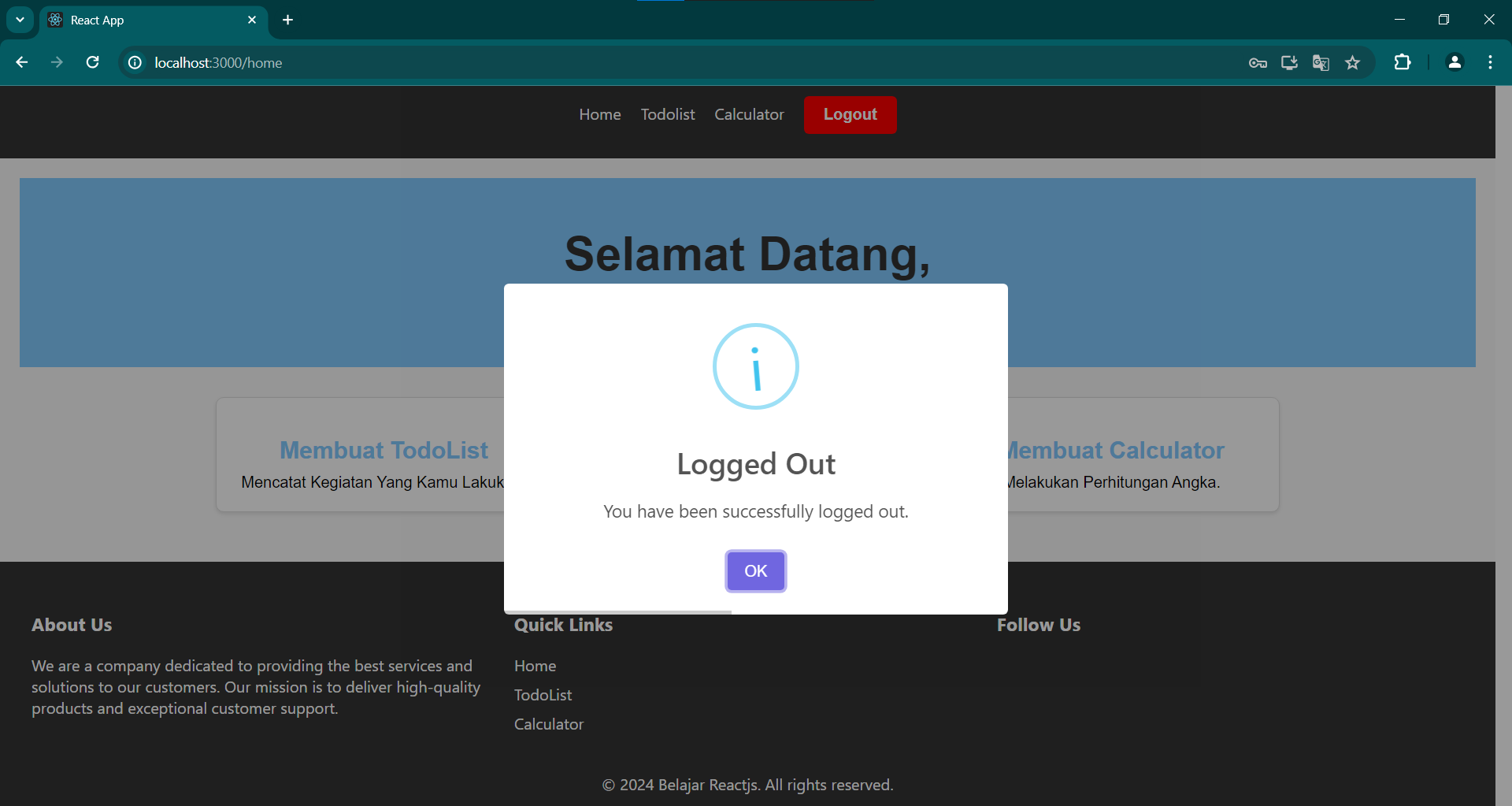
Test Case membuat button Logout pada navbar dan alert berhasil Logout kemudian Kembali ke halaman Login.

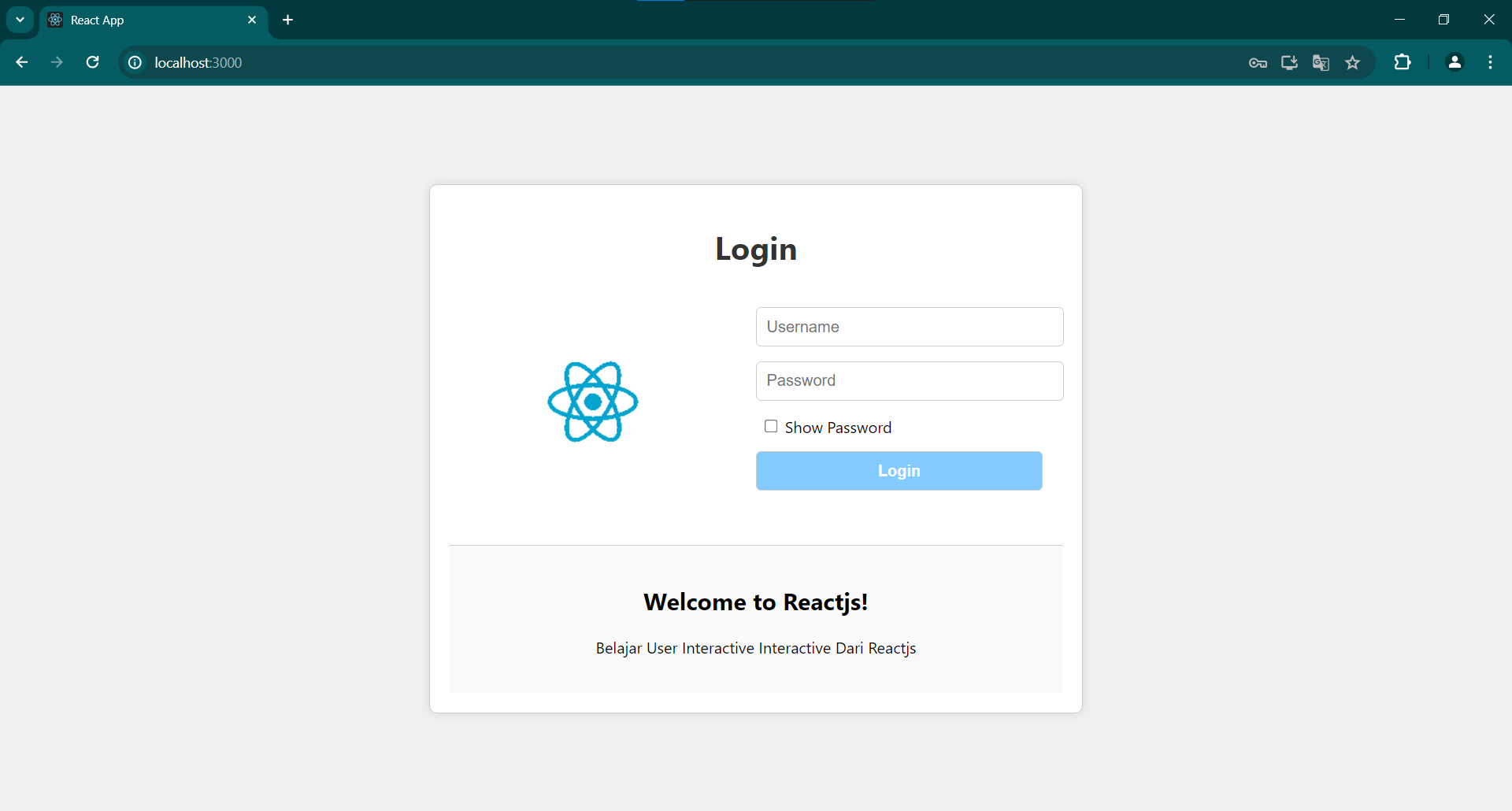


Setelah di klik logout akan muncul seperti berikut ini

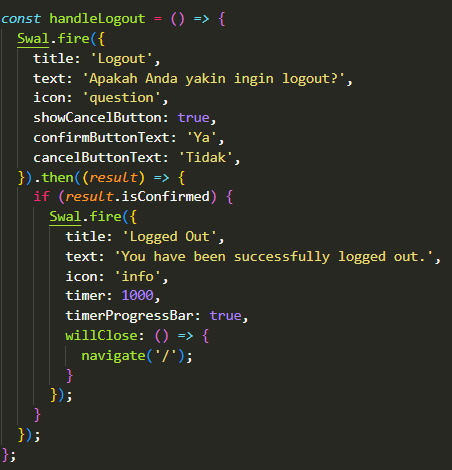


Selanjtunya ketika di klik Ya maka akan muncul alert dan Kembali ke halaman login seperti berikut ini.





Clue



Setelah selesai upload file dengan nama Navbar.js ke website iCLOP