

LAPORAN PRAKTIKUM  
PEMROGRAMAN BERBASIS FRAMEWORK

Pertemuan Minggu-5

Modul 5 – React Router

*(“Install react router-Basic router, URL Parameters, Nesting, Redirect(auth)”)*



NAMA : DINA RISKY ALIN SAPUTRI  
NIM : 1841720016  
KELAS : 3F

PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG

2021

## Praktikum

Dalam membangun web SPA (*Single Page Application*) kita membutuhkan library React Router. Untuk itu langkah pertama yang kita lakukan adalah menginstall library React Router.

### Langkah Praktikum

#### a. Basic Router

Langkah	Keterangan
1	<p>Setelah kita sukses melakukan instalasi react (<i>npm install</i>), kita perlu menambahkan library react router dengan menambahkan <i>npm install react-router-dom</i>.</p> <pre>PS C:\Users\ASUS\Pemrograman-Framework\minggu5&gt; npx create-react-app react-router  Creating a new React app in C:\Users\ASUS\Pemrograman-Framework\minggu5\react-router.  Installing packages. This might take a couple of minutes. Installing react, react-dom, and react-scripts with cra-template...  &gt; core-js@2.6.12 postinstall C:\Users\ASUS\Pemrograman-Framework\minggu5\react-router\n &gt; node -e "try{require('./postinstall')}catch(e){}"  &gt; core-js@3.9.1 postinstall C:\Users\ASUS\Pemrograman-Framework\minggu5\react-router\n &gt; node -e "try{require('./postinstall')}catch(e){}"  &gt; core-js-pure@3.9.1 postinstall C:\Users\ASUS\Pemrograman-Framework\minggu5\react-rout &gt; node -e "try{require('./postinstall')}catch(e){}"  &gt; ejs@2.7.4 postinstall C:\Users\ASUS\Pemrograman-Framework\minggu5\react-router\node_m &gt; node ./postinstall.js  + cra-template@1.1.2 + react-dom@17.0.1 + react@17.0.1 + react-scripts@4.0.3 added 1918 packages from 731 contributors and audited 1921 packages in 311.975s  131 packages are looking for funding   run `npm fund` for details  found 0 vulnerabilities</pre> <pre>PS C:\Users\ASUS\Pemrograman-Framework\minggu5\react-router&gt; npm install react-router-dom npm WARN tsutils@3.21.0 requires a peer of typescript@&gt;=2.8.0    &gt;= 3.2.0-dev    &gt;= 3.3.0-dev    &gt;= 3.4.0-dev    &gt;= 3.5.0-dev    &gt;= 3.6.0-dev    &gt;= 3.7.0-beta    &gt;= 3.7.0-beta but none is installed. You must install peer dependencies yourself. npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents): npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}) npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\watchpack-chokidar2\node_modules\fsevents): npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}) npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents): npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})  + react-router-dom@5.2.0 added 11 packages from 6 contributors and audited 1960 packages in 19.753s  131 packages are looking for funding   run `npm fund` for details  found 0 vulnerabilities  PS C:\Users\ASUS\Pemrograman-Framework\minggu5\react-router&gt;  </pre>
2	Kemudian buka app.js pada dan ketikkan code dibawah ini.

```
react-router > src > JS App.js > ...
```

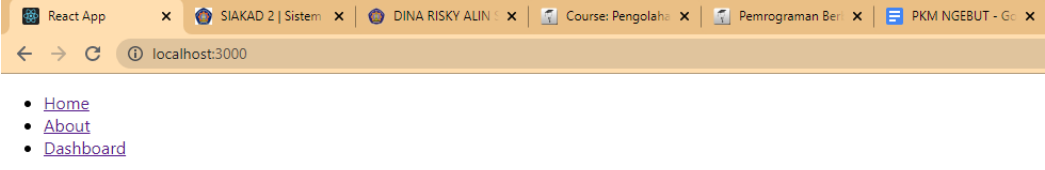
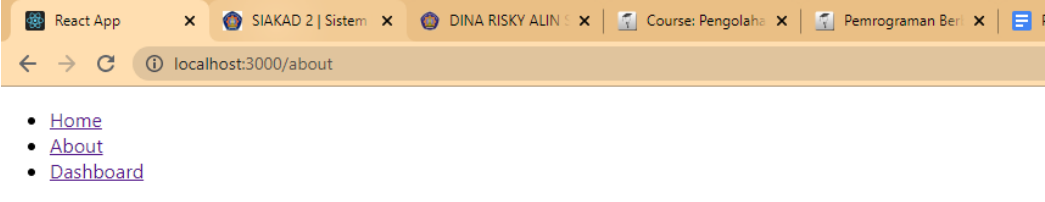
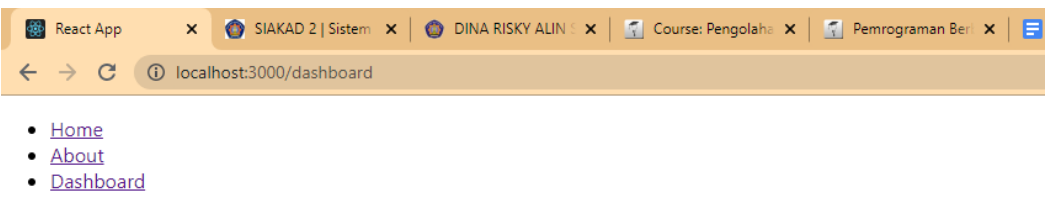
```
1 import React from "react";
2 import {
3   BrowserRouter as Router,
4   Switch,
5   Route,
6   Link
7 } from "react-router-dom";
8
9 // Situs ini memiliki 3 halaman, yang semuanya dirender dinamis di browser.
10 // Meskipun halaman tidak pernah di-refresh, perhatikan bagaimana React Router
11 // membuat URL selalu terbaru saat Anda menavigasi situs.
12 // Ini menjaga riwayat browser, memastikan hal-hal seperti tombol kembali
13 // dan bookmark berfungsi dengan baik.
```

```
14
15 export default function BasicExample() {
16   return (
17     <Router>
18       <div>
19         <ul>
20           <li>
21             <Link to="/">Home</Link>
22           </li>
23           <li>
24             <Link to="/about">About</Link>
25           </li>
26           <li>
27             <Link to="/dashboard">Dashboard</Link>
28           </li>
29         </ul>
30         <hr />

```

```
31       <Switch>
32         <Route exact path="/">
33           <Home />
34         </Route>
35         <Route path="/about">
36           <About />
37         </Route>
38         <Route path="/dashboard">
39           <Dashboard />
40         </Route>
41       </Switch>
42     </div>
43   </Router>
44 );
45 }
46 // Anda dapat menganggap komponen ini sebagai "halaman(konten)" di situs Anda.
47 function Home() {
48   return (
49     <div>
50       <h2>Home</h2>
51     </div>
52   );
53 }
54 function About() {
55   return (
56     <div>
57       <h2>About</h2>
58     </div>

```

	<pre> 59   }); 60   } 61   function Dashboard() { 62     return ( 63       &lt;div&gt; 64         &lt;h2&gt;Dashboard&lt;/h2&gt; 65       &lt;/div&gt; 66     ); 67   } </pre>
3	<p>Lakukan npm start, lalu cek hasilnya akan seperti dibawah ini.</p>  <p>Home</p>
	<p>Pilih setiap menu lalu amati bagaimana perubahannya.</p>  <p>About</p>  <p>Dashboard</p>

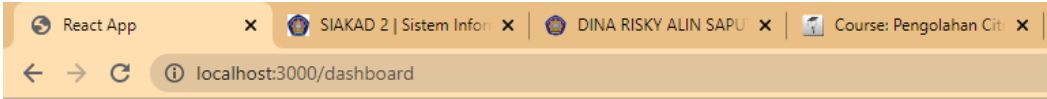
## b. URL Parameters

Langkah	Keterangan
1	Pada percobaan kali ini kita mencoba membuat link react router dengan menggunakan params, sehingga kita hanya membuat satu template yang dapat berubah kontennya sesuai

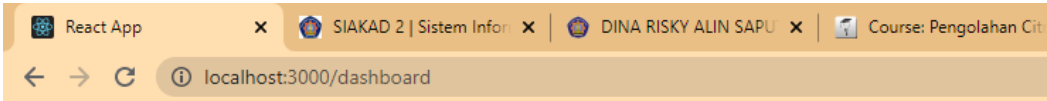
dengan apa yang kita klik.

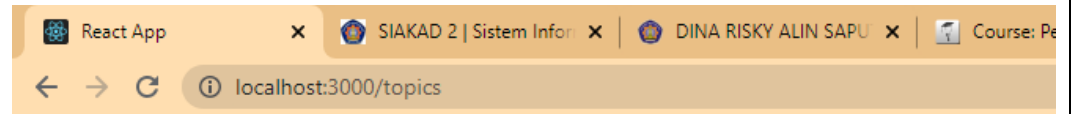
```
react-router > src > JS App.js > Child
1  import React from "react";
2  import {
3    BrowserRouter as Router,
4    Switch,
5    Route,
6    Link,
7    useParams
8  } from "react-router-dom";
9
10 // Params adalah placeholder di URL yang dimulai dengan titik dua,
11 // seperti param `:id` yang didefinisikan dalam route dalam contoh ini.
12
13 export default function ParamsExample() {
14   return (
15     <Router>
16       <div>
17         <h2>Accounts</h2>
18         <ul>
19           <li>
20             <Link to="/netflix">Netflix</Link>
21           </li>
22           <li>
23             <Link to="/gmail">Gmail</Link>
24           </li>
25           <li>
26             <Link to="/yahoo">Yahoo</Link>
27           </li>
28           <li>
29             <Link to="/amazon">Amazon</Link>
30           </li>
```

```
31         </ul>
32         <Switch>
33           <Route path="/:id" children={<Child />} />
34         </Switch>
35       </div>
36     </Router>
37   );
38 }
39
40 function Child() {
41   let { id } = useParams();
42
43   return (
44     <div>
45       <h3>ID: {id}</h3>
46     </div>
47   );
48 }
49
```

2	Ketikkan seperti code dibawah ini.
3	<p>Cek hasilnya maka akan seperti dibawah ini.</p>  <p><b>Accounts</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Netflix</a></li> <li>• <a href="#">Gmail</a></li> <li>• <a href="#">Yahoo</a></li> <li>• <a href="#">Amazon</a></li> </ul> <p><b>ID: dashboard</b></p>
	Secara tampilan memang hasilnya hampir sama dengan praktikum sebelumnya, namun dengan metode yang berbeda.

### c. Use Nesting Router

Langkah	Keterangan
1	Jika kita ingin melakukan percabangan link di dalam react router, maka kita dapat menggunakan cara ini sebagai solusinya.
2	Ketikkan seperti code dibawah ini.
3	<p>Cek hasilnya dan coba klik setiap link nya.</p>  <ul style="list-style-type: none"> <li>• <a href="#">Home</a></li> <li>• <a href="#">Topics</a></li> </ul>



- [Home](#)
- [Topics](#)

## Topics

- [Kuliner](#)
- [Travelling](#)
- [Review Hotel](#)

Please select a topic.



- [Home](#)
- [Topics](#)

## Topics

- [Kuliner](#)
- [Travelling](#)
- [Review Hotel](#)

Sate, Nasi goreng

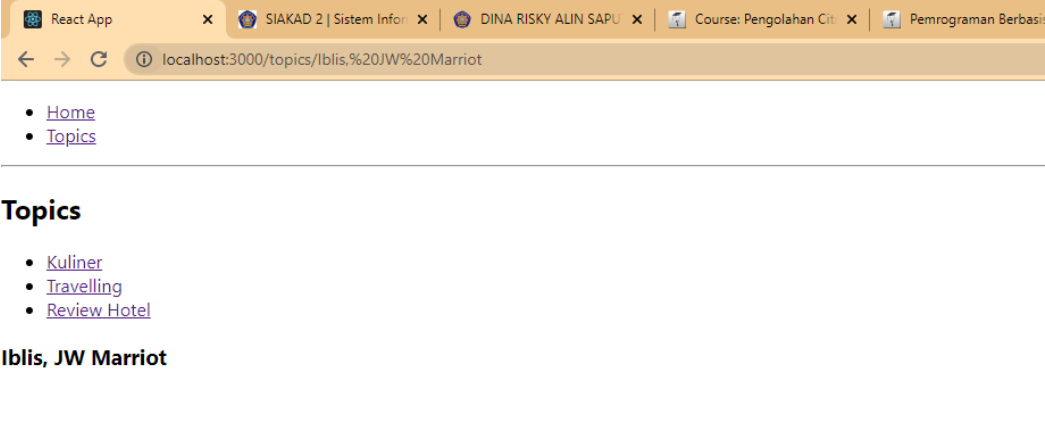


- [Home](#)
- [Topics](#)

## Topics

- [Kuliner](#)
- [Travelling](#)
- [Review Hotel](#)

Wisata alam, Museum

	
	<p>Bisa kita lihat bahwa link induk yang diatas akan menentukan hasil yang keluar dibawahnya, dan link yang dibawah dapat memunculkan hasil yg berbeda pula.</p>

#### d. Use Redirects (Auth)

Langkah	Keterangan
1	<p>Cara ini dapat anda gunakan untuk masuk ke halaman yang membutuhkan authentication. Setelah masuk dan melakukan klik ke menu lain termasuk masuk ke halaman yang memerlukan authentication, posisi user tidak perlu login ulang (karena posisi sudah logged) sampai user malakukan sign out.</p>
2	<p>Ketikkan seperti code dibawah ini.</p> <pre> react-router &gt; src &gt; JS App.js &gt; AuthExample 1  import React from "react"; 2  import { 3    BrowserRouter as Router, 4    Switch, 5    Route, 6    Link, 7    Redirect, 8    useHistory, 9    useLocation 10 } from "react-router-dom"; 11 12 // Pada aplikasi ini memiliki 3 halaman: public page, private page, dan halaman login 13 // untuk masuk ke private page, Anda harus login terlebih dahulu. 14 15 // Pertama, klik public page. Kemudian, kunjungi private page. 16 // karena anda belum login , jadi Anda diarahkan ke halaman login. 17 // setelah login, Anda akan diarahkan kembali ke private page. 18 19 // Perhatikan perubahan setiap URL. Jika Anda mengklik tombol kembali, 20 // apakah anda kembali ke halaman login? Tidak, karena anda sudah login. 21 // Cobalah, maka anda akan kembali ke halaman yang Anda kunjungi sebelum login, yaitu public page. 22 23 export default function AuthExample() { 24   return ( 25     &lt;Router&gt; 26     &lt;div&gt; </pre>

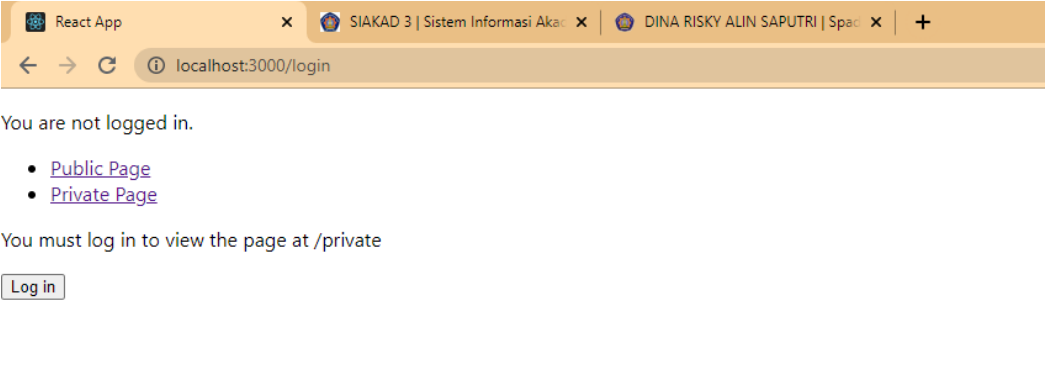
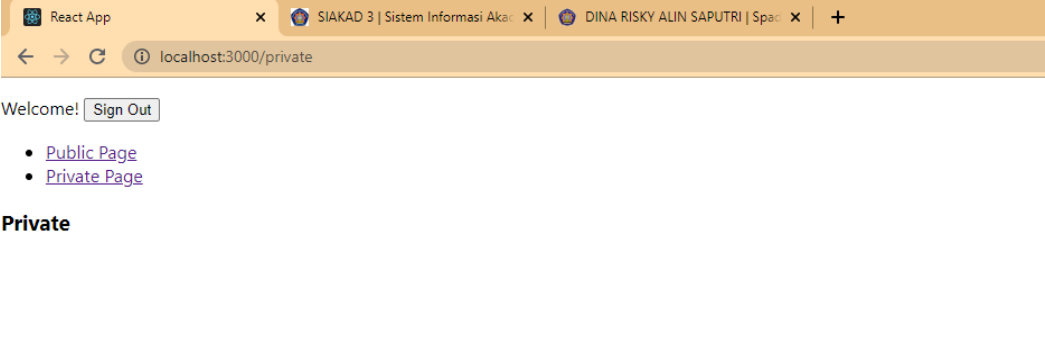


```

27     <Switch>
28     |   <AuthButton />
29     | </Switch>
30     <ul>
31     |   <li>
32     |     <Link to="/public">Public Page</Link>
33     |   </li>
34     |   <li>
35     |     <Link to="/private">Private Page</Link>
36     |   </li>
37     </ul>
38
39     <Switch>
40     |   <Route exact path="/public">
41     |     <PublicPage />
42     |   </Route>
43     |   <Route path="/login">
44     |     <LoginPage />
45     |   </Route>
46     |   <PrivateRoute path="/private">
47     |     <ProtectedPage />
48     |   </PrivateRoute>
49     </Switch>
50   </div>
51 </Router>
52 );
53 }
54
55 const fakeAuth = {
56   isAuthenticated: false,
57   authenticated(cb) {
58     fakeAuth.isAuthenticated = true;
59     setTimeout(cb, 100); // fake async
60   },
61   signout(cb) {
62     fakeAuth.isAuthenticated = false;
63     setTimeout(cb, 100);
64   },
65 };
66
67 function AuthButton() {
68   let history = useHistory();
69
70   return fakeAuth.isAuthenticated ? (
71     <p>
72       Welcome!{" " }
73     <button
74       onClick={() => {


```

	<pre> 75         fakeAuth.signout(() =&gt; history.push("/")); 76     }} 77     &gt; 78     Sign Out 79     &lt;/button&gt; 80 &lt;/p&gt; 81 ) : ( 82     &lt;p&gt;You are not logged in.&lt;/p&gt; 83 ); 84 } 85 // Pembungkus untuk &lt;Route&gt; yang mengarahkan ke login 86 // tampilkan jika Anda belum terkonfirmasi. 87 88 function PrivateRoute({ children, ...rest }) { 89     return ( 90         &lt;Route 91             {...rest} 92             render={({ location }) =&gt; 93                 fakeAuth.isAuthenticated ? ( 94                     children 95                 ) : ( 96                     &lt;Redirect 97                         to={{ 98                             pathname: "/login", 99                             state: { from: location } 100                         }} 101                     /&gt; 102                 ) 103             ) 104         /&gt; 105     ); 106 } 107 function PublicPage() { 108     return &lt;h3&gt;Public&lt;/h3&gt; 109 } 110 111 function ProtectedPage() { 112     return &lt;h3&gt;Private&lt;/h3&gt;; 113 } 114 115 function LoginPage() { 116     let history = useHistory(); 117     let location = useLocation(); 118 119     let { from } = location.state    { from: { pathname: "/" } }; 120     let login = () =&gt; { 121         fakeAuth.authenticated(() =&gt; { 122             history.replace(from); 123         }); 124     }; 125 126     return ( 127         &lt;div&gt; 128             &lt;p&gt;You must log in to view the page at {from.pathname}&lt;/p&gt; 129             &lt;button onClick={login}&gt;Log in&lt;/button&gt; 130         &lt;/div&gt; 131     ); 132 } </pre>
3	<p>Cek hasilnya, ketika klik private page maka diharuskan untuk login terlebih dahulu, ketika sudah masuk dan klik private page kembali kita tidak perlu login kembali karena kita berada dalam posisi logged.</p>

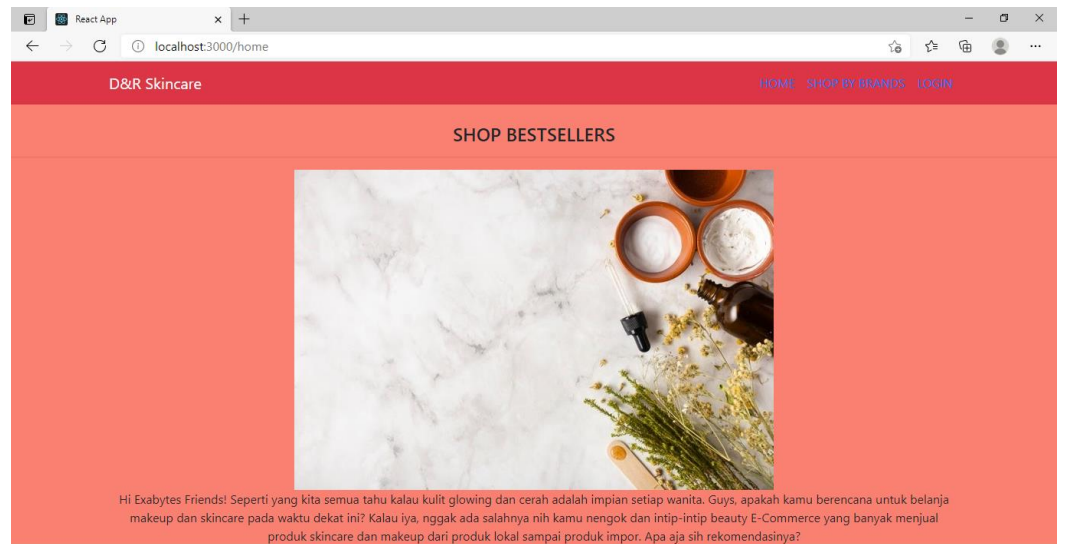
	
	

## Soal Langkah Praktikum

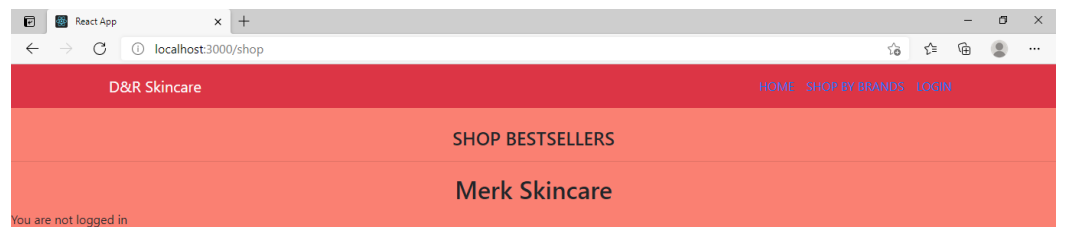
1. Buatlah sebuah situs marketplace(online shop) sederhana, dengan menerapkan konsep SPA (Single Page Application) React Router dimana beberapa menu nya jika di klik terdapat fitur “Nesting” dan “Redirect (Auth)”.

Langkah	Keterangan
1	<p>Tampilan halaman awal</p> 

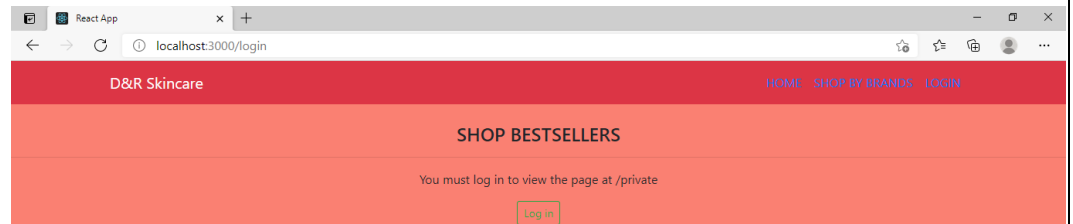
## Tampilan pada halaman home



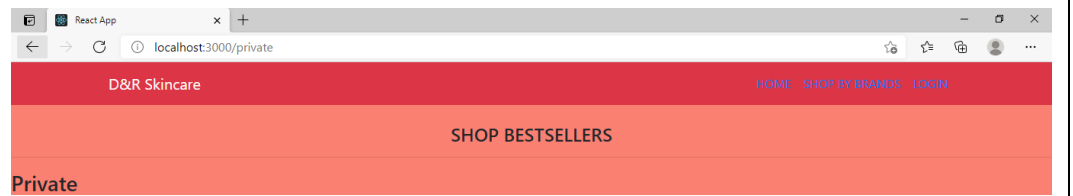
## Tampilan pada halaman shop by brands sebelum login



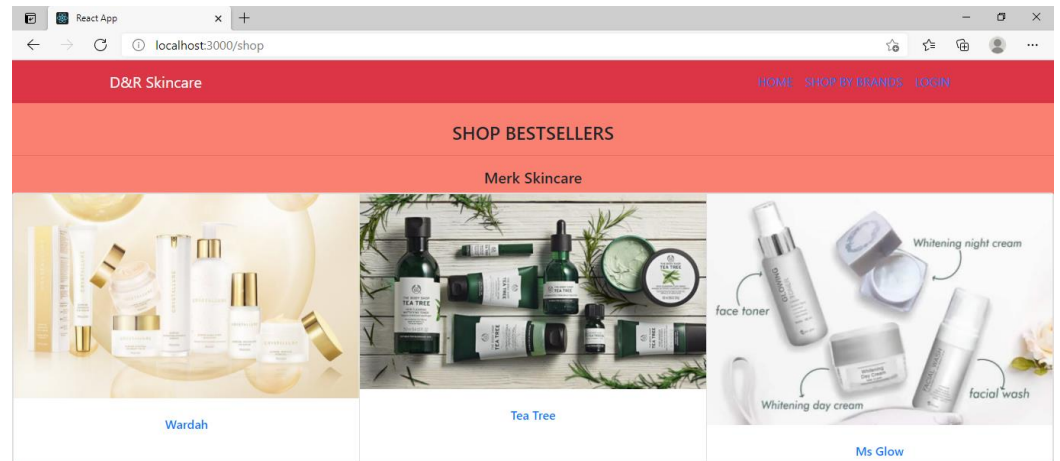
## Tampilan pada halaman login



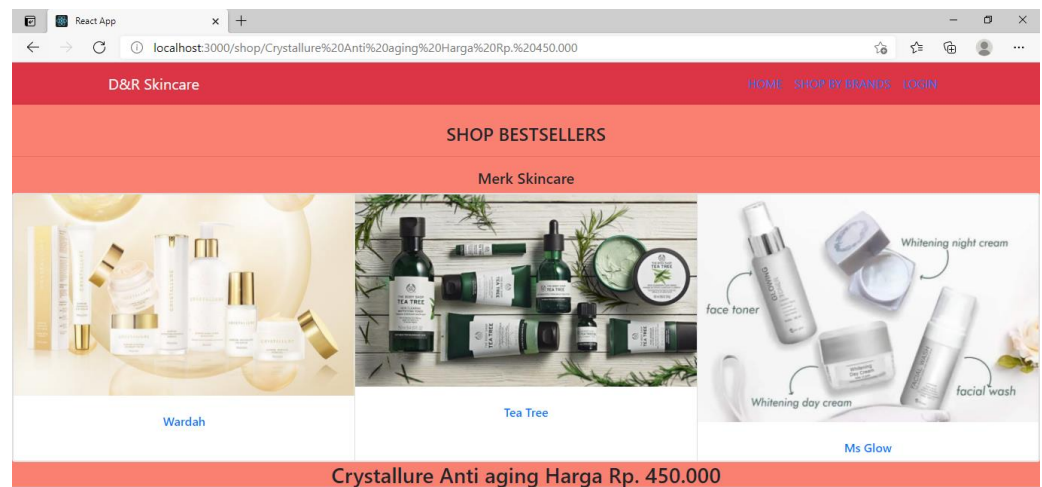
Tampilan pada halaman login setelah melakukan event click button login



Tampilan pada halaman shop by brands setelah login maka akan muncul beberapa link merk skincare



Tampilan pada halaman shop by brands ketika memiliki merk wardah



Link Github : <https://github.com/dinariskyas/PemrogramanFramework2021/tree/master/minggu5>

Link Youtube : <https://youtu.be/-zSAEiQgVts>