

MODUL 5
REACT ROUTER

PEMROGRAMAN BERBASIS FRAMEWORK



Disusun oleh:
Livia Yurike Khuril Maula
D4 TI – 3F / 15
NIM : 1841720025

POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D4 TEKNIK INFORMATIKA

MARET 2021

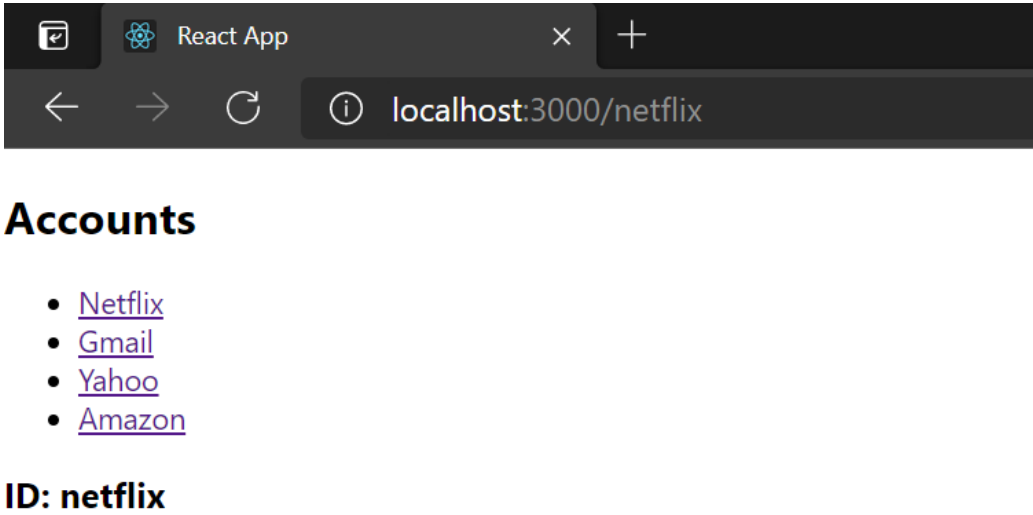
PRAKTIKUM

1.1 Langkah Praktikum

a. Basic Router

No	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 D:\Kuliah\SEMESTER 6\Pemrograman Berbasis Framework\Pertemuan 5> cd react-router PS D:\Kuliah\SEMESTER 6\Pemrograman Berbasis Framework\Pertemuan 5> npm install react-router-dom npm WARN tsutils@3.20.0 requires a peer of typescript@>=2.8.0 >= 3.2.0-dev >= 3.3.0-dev >= 3.4.0-dev >= 3.5.0-dev >= 3.6.0-dev >= 3.6.0-beta >= 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\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"}) 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"}) + react-router-dom@5.2.0 added 11 packages from 6 contributors and audited 1952 packages in 68.285s 127 packages are looking for funding run `npm fund` for details found 0 vulnerabilities PS D:\Kuliah\SEMESTER 6\Pemrograman Berbasis Framework\Pertemuan 5> react-router</pre>
2	<p>Kemudian buka app.js pada dan ketikkan code dibawah ini.</p> <pre>react-router > src > 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 secara 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 14 export default function BasicExample() { 15 return (16 <Router> 17 <div> 18 19 20 <Link to="/">Home</Link> 21 22 23 <Link to="/about">About</Link> 24 25 26 <Link to="/dashboard">Dashboard</Link> 27 28 29 <hr /> 30 <Switch> 31 <Route exact path="/"> 32 <Home /> 33 </Route> 34 <Route path="/about"> 35 <About /> 36 </Route> 37 <Route path="/dashboard"></pre> <pre>react-router > src > App.js 38 <Dashboard /> 39 </Route> 40 </Switch> 41 </div> 42 </Router> 43); 44 } 45 46 // Anda dapat menganggap komponen ini sebagai "halaman(konten)" di situs Anda. 47 48 function Home() { 49 return (50 <div> 51 <h2>Home</h2> 52 </div> 53); 54 } 55 56 function About() { 57 return (58 <div> 59 <h2>About</h2> 60 </div> 61); 62 } 63 64 function Dashboard() { 65 return (66 <div> 67 <h2>Dashboard</h2> 68 </div> 69); 70 }</pre>
3	<p>Lakukan npm start, lalu cek hasilnya akan seperti dibawah ini.</p>  <ul style="list-style-type: none">HomeAboutDashboard <p>Home</p>

b. URL Parameters

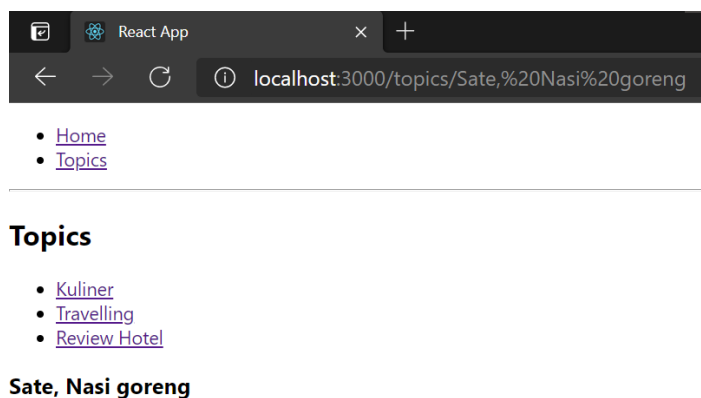
1	Pada percobaan kali ini kita mencoba membuat link react router dengan menggunakan params, sehingga kita hanya membuat satu template yang dapat berubah kontennya sesuai dengan apa yang kita klik.
2	<p>Ketikkan seperti code dibawah ini</p> <div><pre>react-router > src > JS App.js > ParamsExample 1 import 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 export default function ParamsExample() { 13 return (14 <Router> 15 <div> 16 <h2>Accounts</h2> 17 18 19 <Link to="/netflix">Netflix</Link> 20 21 22 <Link to="/gmail">Gmail</Link> 23 24 25 <Link to="/yahoo">Yahoo</Link> 26 27 28 <Link to="/amazon">Amazon</Link> 29 30 31 </div> 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 }</pre></div>
3	<p>Cek hasilnya maka akan seperti dibawah ini.</p> <div></div>

c. Use Nesting Router

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.

```
JS Appjs X
react-router > src > JS Appjs > Topics
1 import React from "react";
2 import {
3   BrowserRouter as Router,
4   Switch,
5   Route,
6   Link,
7   useParams,
8   useRouteMatch
9 } from "react-router-dom";
10
11 // Karena route adalah komponen regular react,
12 // sehingga dapat ditampilkan di mana saja dalam penempatannya,
13 // termasuk dalam child element.
14
15 // Ini akan membantu anda untuk memecah menjadi beberapa bundel
16 // karena pemisahan kode pada aplikasi React Router sama dengan
17 // pemecahan kode pada aplikasi react lainnya.
18
19 export default function NestingExample() {
20   return (
21     <Router>
22       <div>
23         <ul>
24           <li>
25             <Link to="/">Home</Link>
26           </li>
27           <li>
28             <Link to="/topics">Topics</Link>
29           </li>
30         </ul>
31         <hr />
32         <Switch>
33           <Route exact path="/">
34             <Home />
35           </Route>
36           <Route path="/topics">
37
38             <Topics />
39           </Route>
40         </Switch>
41       </div>
42     </Router>
43   );
44 }
45
46 function Home() {
47   return (
48     <div>
49       <h2>Home</h2>
50     </div>
51   );
52 }
53
54 function Topics() {
55   // path untuk membuat jalur <Route> yang terhadap rute induk,
56   // sedangkan url untuk membuat link
57   let { path, url } = useRouteMatch();
58   return (
59     <div>
60       <h2>Topics</h2>
61       <ul>
62         <li>
63           <Link to={`/${url}/Sate, Nasi goreng`}>Kuliner</Link>
64         </li>
65         <li>
66           <Link to={`/${url}/Wisata alam, Museum`}>Travelling</Link>
67         </li>
68         <li>
69           <Link to={`/${url}/Ibis, JW Marriot`}>Review Hotel</Link>
70         </li>
71       </ul>
72       <Switch>
73         <Route exact path={path}>
74
75           <h3>Please select a topic.</h3>
76         </Route>
77         <Route path={`${path}/${topicId}`}>
78           <Topic />
79         </Route>
80       </Switch>
81     </div>
82   );
83 }
84
85 function Topic() {
86   let { topicId } = useParams();
87
88   return (
89     <div>
90       <h3>{topicId}</h3>
91     </div>
92   );
93 }
94
```

3 Cek hasilnya dan coba klik setiap link nya.



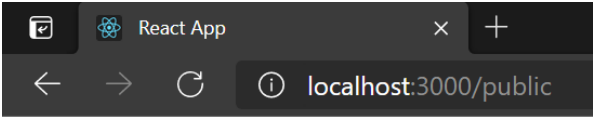
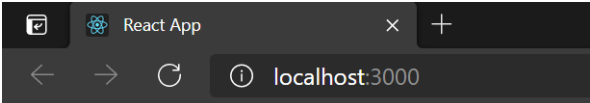
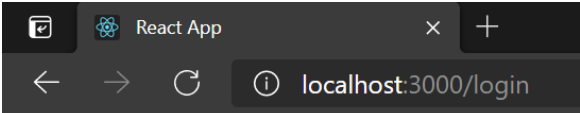
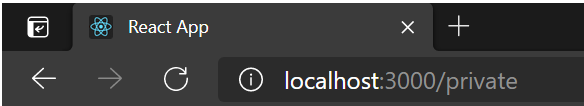
d. Use Redirects (Auth)

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 melakukan sign out.</p>
2	<p>Ketikkan seperti code dibawah ini</p> <pre>JS App.js x react-router > src > JS App.js > AuthButton > <function> 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 // Setelah login, anda akan diarahkan kembali ke private page. 17 18 // Perhatikan perubahan setiap URL. Jika anda mengklik tombol kembali, 19 // apakah anda kembali ke halaman login? Tidak, karena anda sudah login. 20 // Cobalah, maka anda akan kembali ke halaman yang anda kunjungi sebelum login, yaitu public page 21 22 export default function AuthExample() { 23 return (24 <Router> 25 <div> 26 <Switch> 27 <AuthButton /> 28 </Switch> 29 30 31 32 <Link to="/public">Public Page</Link> 33 34 35 <Link to="/private">Private Page</Link> 36 37 38 39 <Switch> 40 <Route 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 authenticate(cb) { 58 fakeAuth.isAuthenticated = true; 59 setTimeout(cb, 100); //fake async 60 }, 61 signout(cb) { 62 fakeAuth.isAuthenticated = false; 63 setInterval(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(() => history.push("/"));
76     }
77 }
78     Sign out
79 </button>
80 </p>
81 ) : (
82     <p>You are not logged in.</p>
83 );
84 }
85
86 // Pembungkus untuk <Route> yang mengarahkan ke login
87 // tampilan jika anda belum terkonfirmasi.
88
89 function PrivateRoute({ children, ...rest }) {
90     return (
91         <Route
92             {...rest}
93             render={() =>
94                 fakeAuth.isAuthenticated ? (
95                     children
96                 ) : (
97                     <Redirect
98                         to={{
99                             pathname: "/login",
100                             state: { from: location }
101                         }}
102                     />
103                 )
104             )
105         />
106     );
107 }
108
109 function PublicPage() {
110     return <h3>Public</h3>;
111 }
112
113 function ProtectedPage() {
114     return <h3>Private</h3>;
115 }
116
117 }
118
119 function LoginPage() {
120
121     let history = useHistory();
122     let location = useLocation();
123
124     let { from } = location.state || { from: { pathname: "/" } };
125     let login = () => {
126         fakeAuth.authenticate(() => {
127             history.replace(from);
128         });
129     };
130
131     return (
132         <div>
133             <p>You must log in to view the page at {from.pathname}</p>
134             <button onClick={login}>Log in</button>
135         </div>
136     );
137 }
138

```

3	<p>Cek hasilnya, ketika klik private page maka diharuskan untuk login terlebih dahulu, ketikasudah masuk dan klik private page kembali kita tidak perlu login kembali karena kita berada dalam posisi logged.</p> <div data-bbox="338 338 933 454">  </div> <p>You are not logged in.</p> <ul style="list-style-type: none"> • Public Page • Private Page <p>Public</p> <div data-bbox="338 698 930 799">  </div> <p>You are not logged in.</p> <ul style="list-style-type: none"> • Public Page • Private Page <div data-bbox="338 974 920 1086">  </div> <p>You are not logged in.</p> <ul style="list-style-type: none"> • Public Page • Private Page <p>You must log in to view the page at /private</p> <div data-bbox="338 1299 414 1330"> <input type="button" value="Log in"/> </div> <div data-bbox="338 1377 924 1482">  </div> <p>Welcome! <input type="button" value="Sign out"/></p> <ul style="list-style-type: none"> • Public Page • Private Page <p>Private</p>
---	---

1.2 Soal Latihan Praktikum

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)”.

No	Keterangan
1	Lakukan instalasi react (<i>npm install</i>) terlebih dahulu
2	<p>Tambahkan library react router dengan menambahkan <i>npm install react-router-dom</i></p> 
2	<p>Import css bootstrap.min.css (css bootstrap yang sudah dikompresi) ke dalam App.js Jika css tidak ditemukan, install lewat cmd dengan perintah “npm install bootstrap”</p>
3	<p>Ketikan code pada App.js seperti dibawah</p> 


```

75     [...rest]
76     render={props =>
77       fakeAuth.isAuthenticated ? (
78         <Component {...props} />
79       ) : (
80         <Redirect
81           to={{
82             pathname: "/login",
83             state: { from: props.location }
84           }}
85         />
86       )
87     }
88   };
89 }
90
91
92 function Home() {
93   return (
94     <div className="home">
95       <header class="masthead">
96         <div class="container">
97           <div class="masthead-subheading">Welcome to our Store!</div>
98           <div class="masthead-heading text-uppercase">It's Nice To Meet You!</div>
99         </div>
100       </header>
101     </div>
102   );
103 }
104
105 function Product() {
106   let { path, url } = useRouteMatch();
107   return (
108     <div className="header-2">
109       {/* <h2>Data Savings</h2> */}
110     </div>
111   );

```

```

112   <div>
113     <li className="topbar-2">
114       <link to={` ${url}/glasses`}>Glasses</Link>
115     </li>
116     <li className="topbar-2">
117       <link to={` ${url}/clothes`}>Clothes</Link>
118     </li>
119   </ul>
120
121   <Switch>
122     <Route exact path={path}>
123       <Route>
124         <Route path={` ${path}/${dataid}`}
125           <dataBody />
126         </Route>
127       </Route>
128     </Switch>
129   </div>
130 )
131
132
133 function DataBody() {
134   let { dataid } = useParams();
135
136   if (dataid === "glasses") {
137     return (
138       <div className="glasses">
139         <div className="grid">
140           <div className="row">
141             <div className="col-sm">
142               <img src={glasses} alt="Gambar Thumbnail Artikel" />
143               <center><h3>Glassesess Haviva Black</h3>
144               <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor</p></center>
145             </div>
146             <div className="col-sm">
147               <img src={glasses} alt="Gambar Thumbnail Artikel" />
148               <center><h3>Glassesess Haviva Black</h3>

```

```

149       <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor</p></center>
150     </div>
151     <div className="col-sm">
152       <img src={glasses} alt="Gambar Thumbnail Artikel" />
153       <center><h3>Glassesess Haviva Black</h3>
154       <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor</p></center>
155     </div>
156   </div>
157   <div className="row">
158     <div className="col-sm">
159       <img src={glasses} alt="Gambar Thumbnail Artikel" />
160       <center><h3>Glassesess Haviva Black</h3>
161       <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor</p></center>
162     </div>
163     <div className="col-sm">
164       <img src={glasses} alt="Gambar Thumbnail Artikel" />
165       <center><h3>Glassesess Haviva Black</h3>
166       <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor</p></center>
167     </div>
168     <div className="col-sm">
169       <img src={glasses} alt="Gambar Thumbnail Artikel" />
170       <center><h3>Glassesess Haviva Black</h3>
171       <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor</p></center>
172     </div>
173   </div>
174   </div>
175   </div>
176 )
177 } else if (dataid === "Clothes") {
178   return (
179     <div className="clothes">
180       <header class="masthead3">
181         <div class="container">
182           </div>
183       </header>
184     </div>
185   );

```

```

186   }
187 }
188
189 class Login extends Component {
190   state = { redirectToReferrer: false };
191
192   login = () => {
193     fakeAuth.authenticate() => {
194       this.setState({ redirectToReferrer: true });
195     });
196   };
197
198   render() {
199     const { from } = this.props.location.state || { from: { pathname: "/home" } };
200     const { redirectToReferrer } = this.state;
201
202     if (redirectToReferrer) {
203       return <Redirect to={from} />
204     }
205
206     return (
207       <div className="Login">
208         <header class="masthead">
209           <div class="container">
210             <div class="masthead-subheading">You must log in to view the page</div>
211             <button onClick={this.login}>Login</button>
212           </div>
213         </header>
214       </div>
215     );
216   }
217 }
218
219 export default Marketplace;

```

```

1  header.masthead {
2    padding-top: 10.5rem;
3    padding-bottom: 6rem;
4    text-align: center;
5    color: ■#fff;
6    background-image: url("../assets/header-bg.jpg");
7    background-repeat: no-repeat;
8    background-attachment: scroll;
9    background-position: center center;
10   background-size: cover;
11 }
12
13 header.masthead3 {
14   padding-top: 10.5rem;
15   padding-bottom: 20rem;
16   text-align: center;
17   color: ■#fff;
18   background-image: url("../assets/clothes.jpg");
19   background-repeat: no-repeat;
20   background-attachment: scroll;
21   background-position: center center;
22   background-size: cover;
23 }
24
25 .container {
26   width: 100%;
27   padding-right: 0.75rem;
28   padding-left: 0.75rem;
29   margin-right: auto;
30   margin-left: auto;
31 }
32
33 header.masthead .masthead-subheading {
34   font-size: 1.5rem;
35   font-style: italic;
36   line-height: 1.5rem;
37   margin-bottom: 25px;

```

```

38   font-family: "Droid Serif", -apple-system, BlinkMacSystemFont, "Segoe UI",
39   Roboto, "Helvetica Neue", Arial, sans-serif, "Apple Color Emoji",
40   "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";
41 }
42
43 @media (min-width: 768px) {
44   header.masthead {
45     padding-top: 17rem;
46     padding-bottom: 12.5rem;
47   }
48   header.masthead .masthead-subheading {
49     font-size: 2.25rem;
50     font-style: italic;
51     line-height: 2.25rem;
52     margin-bottom: 2rem;
53   }
54   header.masthead .masthead-heading {
55     font-size: 4.5rem;
56     font-weight: 700;
57     line-height: 4.5rem;
58     margin-bottom: 4rem;
59   }
60 }
61
62 .header {
63   overflow: hidden;
64 }
65
66 .header img {
67   width: 40px;
68   float: left;
69   display: flex;
70   align-items: center;
71   margin-top: 15px;
72   margin-left: 15px;
73 }
74

```

```

75 .TopBar-1 {
76   float: left;
77   text-align: center;
78   text-decoration: ■brown;
79   padding: 14px 16px;
80   list-style: none;
81   font-size: 23px;
82   font-weight: 500;
83   font-style: oblique;
84 }
85
86 #TB1 {
87   margin-left: 25px;
88 }
89
90 a:any-link {
91   color: □#212121;
92   text-decoration: none;
93   font-style: oblique;
94 }
95
96 #logout {
97   color: □#212121;
98   float: right;
99   margin-bottom: 0;
100  font-style: oblique;
101  font-size: 23px;
102 }
103
104 #logout button {
105   width: 75px;
106   color: □#000;
107   background-color: ■#fec810;
108   border-radius: 10px;
109   border: none;
110   margin-left: 10px;
111   margin-right: 10px;

```

```

112   font-style: oblique;
113 }
114
115 #warning-log {
116   color: □#000;
117   float: right;
118   border: none;
119   border-radius: 10px;
120   margin-top: 30px;
121   margin-right: 20px;
122   font-size: 23px;
123   font-style: oblique;
124 }
125
126 /* Home */
127
128 .Home {
129   text-transform: uppercase;
130   border-top: 1px solid □rgba(139, 136, 136, 0.425);
131   margin-top: 75px;
132   height: 745px;
133 }
134
135 .Home h2 {
136   color: □#212121;
137 }
138
139 .row {
140   margin-top: 10px;
141   display: flex;
142   justify-content: center;
143 }
144
145 .col-sm {
146   border: 1px solid □#212121;
147   width: 20%;
148   float: left;

```

```

149     margin: 0 10px;
150     border-radius: 10px;
151     padding-top: 25px;
152 }
153
154 .col-sm p {
155     font-size: 10px;
156     padding: 0px 10px 0px 10px;
157     text-align: justify;
158 }
159
160 .col-sm img {
161     width: 250px;
162 }
163
164 /* Login */
165
166 .Login {
167     text-transform: uppercase;
168     border-top: 1px solid #139191;
169     margin-top: 100px;
170     height: 745px;
171     font-style: oblique;
172 }
173
174 .Login button {
175     width: 100px;
176     height: 50px;
177     background-color: #f8d7da;
178     color: #000;
179     border: none;
180     border-radius: 10px;
181     position: absolute;
182     margin-top: 75px;
183     font-size: 20px;
184     margin-bottom: 4rem;
185     font-style: oblique;

```

```

186 }
187
188 /* Data Page */
189
190 .header-2 {
191     overflow: hidden;
192     margin-top: 75px;
193     border-top: 1px solid #139191;
194     height: 745px;
195 }
196
197 .header-2 ul {
198     margin-left: 60px;
199 }
200
201 .TopBar-2 {
202     float: left;
203     text-align: center;
204     text-decoration: #800000;
205     padding: 14px 16px;
206     list-style: none;
207     font-size: 25px;
208     font-weight: 500;
209     font-style: oblique;
210 }
211
212 /* Glasses */
213
214 .Glasses {
215     height: 650px;
216     margin-top: 100px;
217 }
218
219 .table-glasses {
220     display: flex;
221     justify-content: center;
222 }

```

```

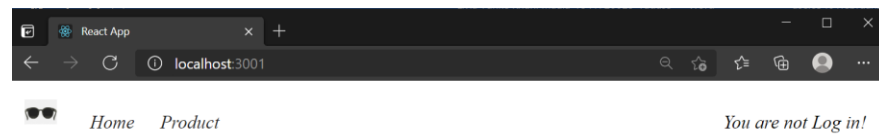
223
224     table,
225     th,
226     td {
227         border: 1px solid #212121;
228         border-collapse: collapse;
229     }
230
231     th,
232     td {
233         padding: 15px;
234     }
235
236     /* clothes */
237
238     .Clothes {
239         height: 650px;
240         margin-top: 100px;
241     }
242
243     .table-clothes {
244         display: flex;
245         justify-content: center;
246     }
247

```

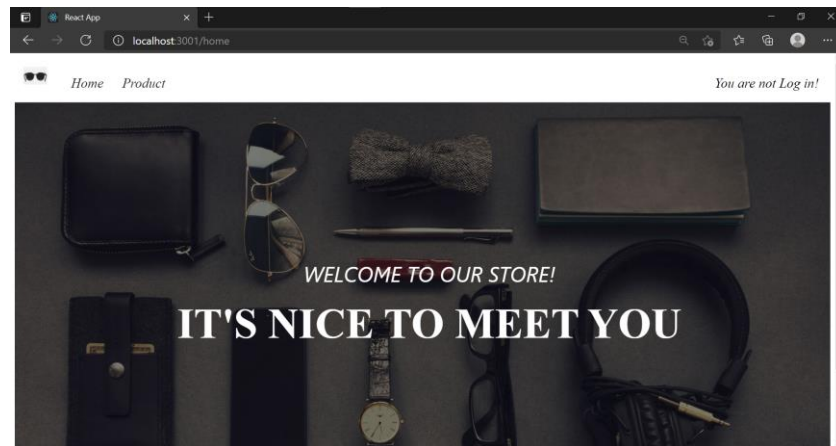
5

Hasil

Jalankan dengan perintah `npm start`

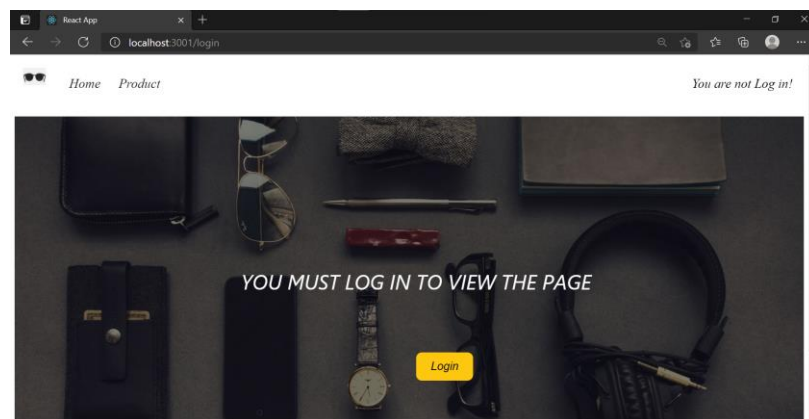


Halaman Home

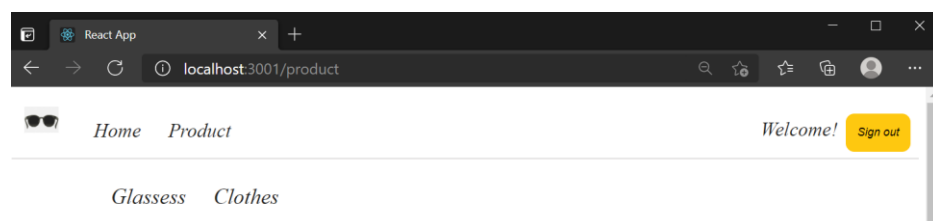


Halaman Produk

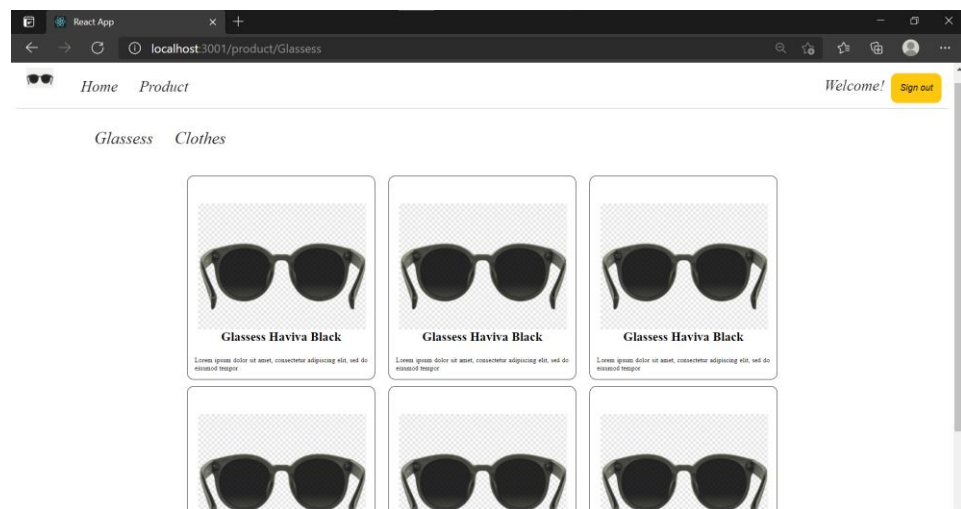
User diminta untuk melakukan login terlebih dahulu agar bisa masuk pada halaman produk



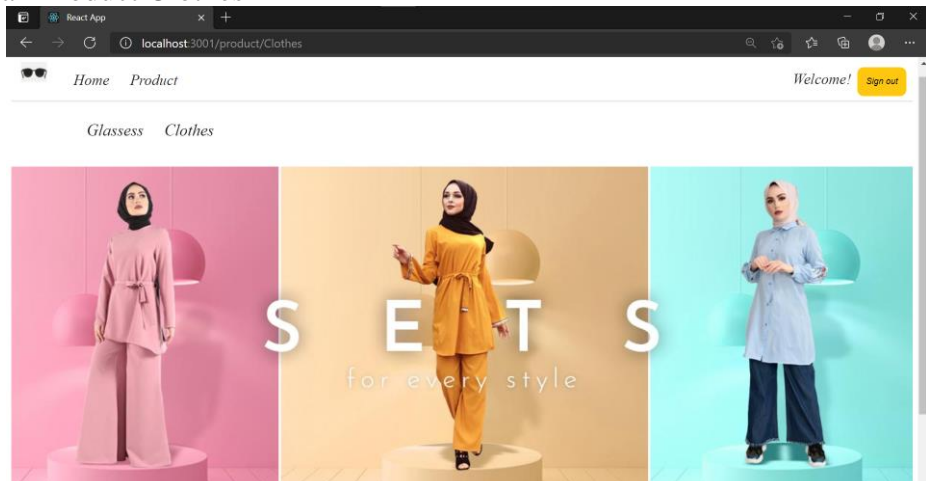
Setelah klik button Login status pada Pojok Kanan Atas Berubah
Dan tampilan halaman produk berupa nested seperti ini :



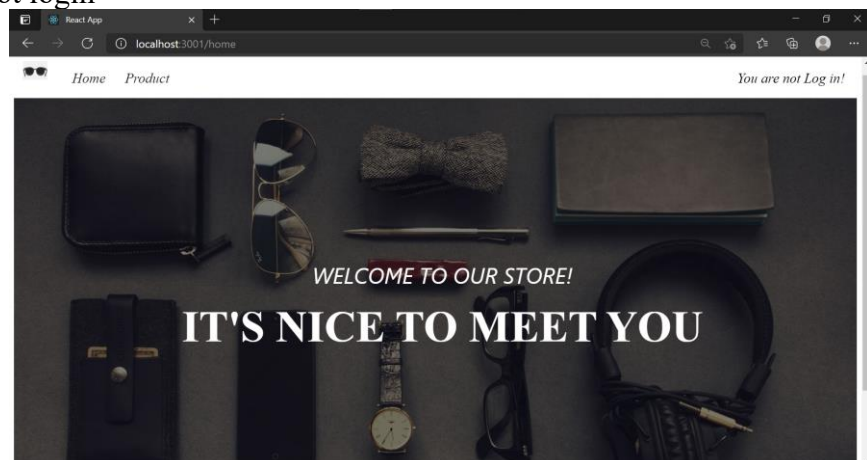
Halaman Product – Glassess



Halaman Product Clothes



Jika button signup diklik maka akan Kembali ke halaman home dengan kondisi 'you're not login'



Link Github :

<https://github.com/liviayurike/Pemrograman-Berbasis-Framework/tree/master/Pertemuan%205>

Link Youtube :

<https://youtu.be/T-go1oWrfsI>