

TUGAS INDIVIDU
Redux Hoc
Pemrograman Berbasis Framework

Oleh:

Alfalah Anugara

1741720159

TI-3A / 02



PROGRAM STUDI – D4 TEKNIK INFORMATIKA

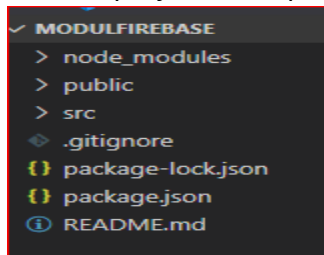
JURUSAN – TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

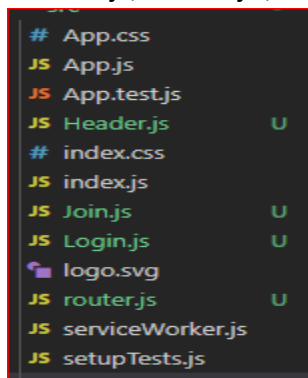
2020/2021

1.1 Langkah Praktikum

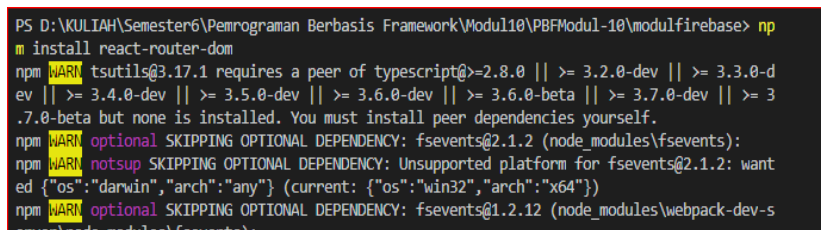
1. Buatlah project react pada directory yang baru



2. Setelah itu pada folder src buatlah file dengan nama index.js ; Login.js ; Join.js ; Header.js; routes.js ; styles.css (file lain yang ada di folder src bisa dihapus saja).



3. Karena pada praktikum ini kita menggunakan react router dom, maka lakukan npm install react-router-dom dahulu



4. Jika sudah, isilah code seperti dibawah ini sesuai dengan nama file masing-masing.
- index.js**

```
JS index.js X
src > JS index.js > ...
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import { Switch, BrowserRouter as Router, Route } from "react-router-dom";
4 import routes from "./routes.js";
5 import Header from "./Header";
6 import "./styles.css";
7
8 export const AuthContext = React.createContext(null);
9
10 function App() {
11   const [isLoggedIn, setLoggedIn] = useState(false);
12
13   return (
14     <AuthContext.Provider value={{ isLoggedIn, setLoggedIn }}>
15       Is Logged in? {JSON.stringify(isLoggedIn)}
16       <div className="App">
17         <Router>
18           <Header/>
19
20           <Switch>
21             {routes.map(route => (
22               <Route
23                 key={route.path}
24                 path={route.path}
25                 exact={route.exact}
26                 component={route.main}
27               />
28             ))}
29           </Switch>
30         </Router>
31       </div>
32     </AuthContext.Provider>
33   );
34 }
35
36 const rootElement = document.getElementById("root");
37 ReactDOM.render(<App />, rootElement);
38
```

Login.js

```
JS index.js X JS routes.js JS Header.js JS Login.js X
src > JS Login.js > default
1 import React, { useState, useContext } from "react";
2 import { AuthContext } from "./index";
3
4 const Login = () => {
5   const [email, setEmail] = useState("");
6   const [password, setPassword] = useState("");
7   const [error, setErrors] = useState("");
8
9   const Auth = useContext(AuthContext);
10   const handleForm = e => {
11     e.preventDefault();
12     console.log(Auth);
13     Auth.setLoggedIn(true);
14   };
15
16   return (
17     <div>
18       <h1>Login</h1>
19       <form onSubmit={e => handleForm(e)}>
20         <input
21           value={email}
22           onChange={e => setEmail(e.target.value)}
23           name="email"
24           type="email"
25           placeholder="email"
26         />
27         <input
28           onChange={e => setPassword(e.target.value)}
29           name="password"
30           value={password}
31           type="password"
32           placeholder="password"
33         />
34         <hr/>
35         <button class="googleBtn" type="button">
36           
40           Login With Google
41         </button>
42         <button type="submit">Login</button>
43         <span>{error}</span>
44       </form>
45     </div>
46   );
47 };
```

Join.js

```
src > JS Joinjs > [⌘] default
1  import React , {useState, useContext} from "react" ;
2  import {AuthContext} from "../index" ;
3
4  const Join = () => {
5    const [email, setEmail] = useState("");
6    const [password, setPassword] = useState("");
7    const [error, setErrors] = useState("");
8
9    const Auth = useContext(AuthContext) ;
10   const handleForm = e => {
11     e.preventDefault();
12     console.log(Auth);
13     Auth.setLoggedIn(true);
14   } ;
15
16   return (
17     <div>
18       <h1>Join</h1>
19       <form onSubmit={e => handleForm(e)}>
20         <input
21           value={email}
22           onChange={e => setEmail(e.target.value)}
23           name="email"
24           type="email"
25           placeholder="email"
26         />
27         <input
28           onChange={e => setPassword(e.target.value)}
29           name = "password"
30           value={password}
31           type="password"
32           placeholder="password"
33         />
34         <hr/>
35         <button class = "googleBtn" type="button">
36           
40           Join With Google
41         </button>
42
43         <button type="submit">Login</button>
44       </form>
45     </div>
46   );
47 }
```

Header.js

```
JS index.js  X  JS routes.js  JS Header.js  X
src > JS Headerjs > [⌘] default
1  import React from "react" ;
2  import routes from "../routes" ;
3  import {link} from "react-router-dom";
4
5  const Header = () => (
6    <ul className="nav">
7      {routes.map((route , i) => (
8        <li key={i}>
9          <Link to={route.path}>{route.name}</Link>
10        </li>
11      ))}
12    </ul>
13  ) ;
14
15  export default Header ;
```

router.js

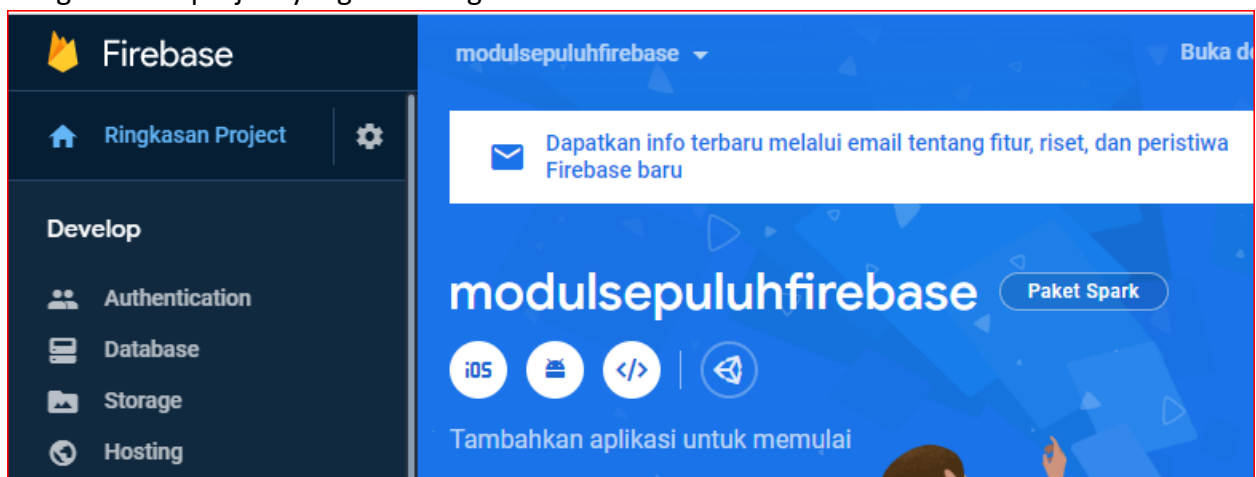
```
JS index.js × JS routes.js ×
src > JS routes.js > [🔍] default
1  import React from "react" ;
2  import Login from "./Login" ;
3  import Join from "./Join" ;
4
5  const routes = [
6    { name : "Join", path : "/" , exact: true , main: () => <Join/>} ,
7    { name : "Login", path : "/login" , exact: true , main: () => <Login/>}
8  ] ;
9
10 export default routes ;
```

style.css

```
JS index.js JS routes.js JS Join.js # styles.css × JS Header.js
src > # styles.css > .googleBtn img
1  .App {
2    font-family: sans-serif;
3    padding: 50px;
4  }
5
6  hi {
7    margin: 0 0 25px 0;
8    text-align: left;
9  }
10
11  form {
12    display: flex;
13    flex-direction: column;
14  }
15
16  input {
17    max-width: 450px;
18    width: auto;
19    border: none;
20    margin: 0 0 25px 0;
21    box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px rgba(0, 0, 0, 0.24);
22    transition: all 0.3s cubic-bezier(0.25, 0.8, 0.25, 1);
23    padding: 10px;
24  }
25
26  button {
27    width: 125px;
28    box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px rgba(0, 0, 0, 0.24);
29    border: none;
30    padding: 7px;
31    text-align: center;
32    cursor: pointer;
33  }
34
35  button[type="submit"] {
36    background-color: #4c566a;
37    color: white;
38    font-size: 14px;
39    font-weight: 900;
40    border-radius: 5px;
41    text-transform: uppercase;
42    padding: 15px;
43  }
44
```

```
JS index.js JS routes.js JS Join.js # styles.css X JS Header.js
src > # styles.css > .googleBtn img
44
45 .googleBtn {
46   width: 255px;
47   margin: 0 0 50px 0;
48   display: flex;
49   justify-content: center;
50   font-size: 16px;
51 }
52
53 .googleBtn img {
54   width: 16px;
55   height: 16px;
56   padding: 0;
57   margin: 0 5px;
58   vertical-align: middle;
59 }
60
61 .nav {
62   display: flex;
63   text-decoration: none;
64   list-style-type: none;
65   justify-content: flex-start;
66   margin: 0 0 50px 0;
67   padding: 0;
68 }
69
70 .nav li {
71   margin: 0 10px;
72 }
```

5. Kemudian buka <https://console.firebase.google.com/> setelah itu buat project baru dengan nama project yang kalian inginkan.



6. Pada cmd lakukan perintah npm install --save firebase di directory project react untuk menambahkan dependency firebase kedalam react kita.

```
PS D:\KULIAH\Semester6\Pemrograman Berbasis Framework\Modul10\PBFModul-10\modulfirebase> npm
install --save firebase

> core-js@3.6.4 postinstall D:\KULIAH\Semester6\Pemrograman Berbasis Framework\Modul10\PBFMod
ul-10\modulfirebase\node_modules\@firebase\polyfill\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript
standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patr
eon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)
```

7. Setelah itu lakukan setting pada project firebase yang kalian buat tadi agar react kalian dapat terhubung dengan firebase. Pada setting get started pilih yang untuk web.



8. Lalu setelah itu anda akan mendapatkan script konfigurasi yang akan menjadi penghubung dengan project react anda. Salin script tersebut lalu tempel pada file baru di dalam folder src yang kita beri nama firebase.config.js (gambar dibawah adalah contoh, sesuaikan script yang anda dapat dari firebase).



firebase.config.js

```
src > JS firebase.config.js > [?] default
1   export default {
2     apiKey: "AIzaSyDo07LQaJ9VV12wTxQj9pS7if6Uzvd4ZtQ",
3     authDomain: "modulesepuluhfirebase.firebaseio.com",
4     databaseURL: "https://modulesepuluhfirebase.firebaseio.com",
5     projectId: "modulesepuluhfirebase",
6     storageBucket: "modulesepuluhfirebase.appspot.com",
7     messagingSenderId: "180531416220",
8     appId: "1:180531416220:web:6de4e620664657a478b159",
9     measurementId: "G-LXR3248JYW"
10  };
```

9. Lalu lakukan inisialisasi firebase config dengan menambahkan code pada index.js seperti dibawah ini.


```
4   import routes from "./routes.js";
5   import Header from "./Header";
6   import "./styles.css";
7   import * as firebase from "firebase" ;
8   import firebaseConfig from "./firebase.config"
9   |
10  firebase.initializeApp(firebaseConfig)
11
```

10. Jika sudah berhasil maka coba jalankan, jika tidak terdapat error maka project anda telah terhubung dengan firebase.

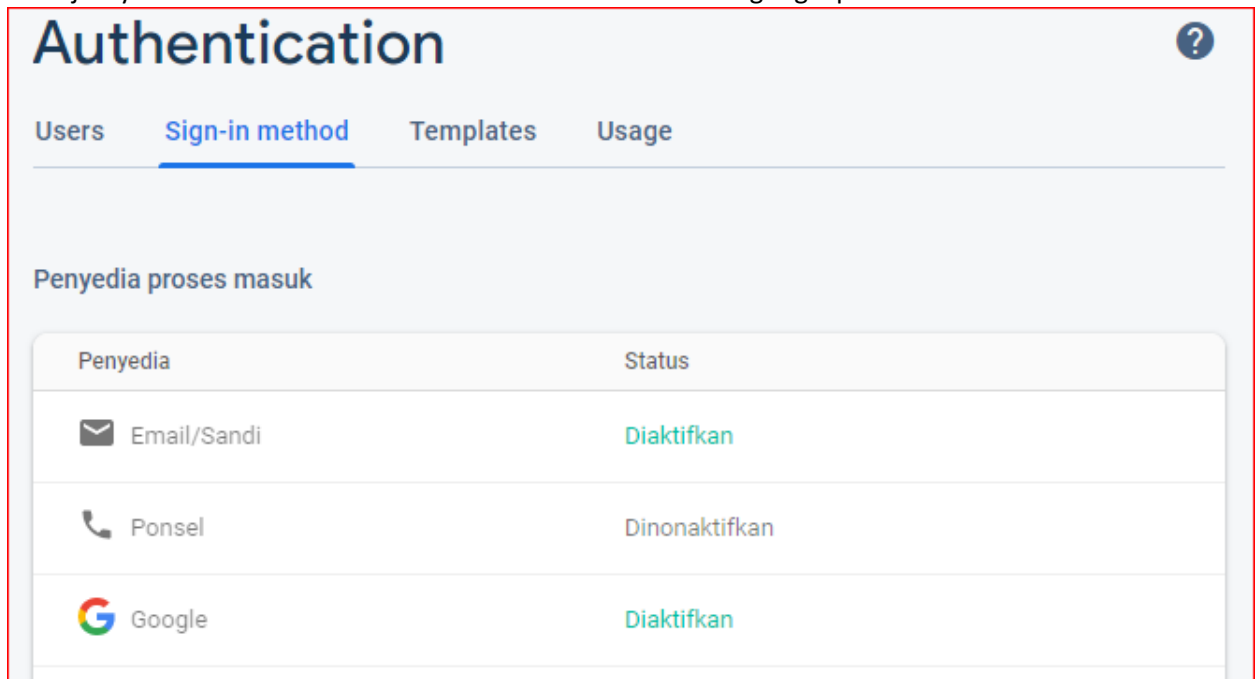
Is Logged in? false

[Join](#) [Login](#)

Join

 Join With Google

11. Selanjutnya aktifkan terlebih dahulu authentication email dan google pada firebase console.



12. Jika sudah, lakukan edit code pada Join.js untuk membuat email dan password untuk user. Pertama import dahulu objek firebase agar bisa digunakan pada Join.js dengan code berikut

```
src > JS Join.js > [?] Join
1  import React , {useState , useContext} from "react" ;
2  import {AuthContext} from "../index" ;
3  import * as firebase from "firebase" ;
4
5  const Join = () => {
6      const [email , setEmail] = useState("") ;
7      const [password , setPassword] = useState("") ;
8      const [error , setErrors] = useState("") ;
9
10     const Auth = useContext(AuthContext) ;
11     const handleForm = e => {
12         e.preventDefault();
13         firebase
14         .auth()
15         .createUserWithEmailAndPassword(email , password)
16         .then(res => {
17             if (res.user) Auth.setLoggedIn(true) ;
18         })
19         .catch(e => {
20             setErrors(e.message);
21         });
22     };
23 }
```

13. Silahkan coba pada aplikasi dengan klik join, lalu isikan email dan password. Coba cek pada firebase maka user telah terbuat seperti dibawah ini

Users Sign-in method Templates Usage				
<div>Telusuri berdasarkan alamat email, nomor telep... Tambahkan pengguna ↻ ⋮</div>				
Pengenal	Penyedia	Dibuat tanggal	Masuk	UID pengguna ↑
alfalah190299@gmail.com		16 ...		amVtCISKfabXn...
Baris per halaman: 50 1-1 dari 1 < >				


14. Langkah selanjutnya adalah bagaimana caranya agar kita dapat sign in/login dengan akun yang telah dibuat, yaitu dengan cara melakukan edit code pada Login.js. Pertama import dahulu objek firebase agar bisa digunakan pada Login.js dengan code berikut.

```
src > JS Login.js > Login
1  import React , {useState , useContext} from "react" ;
2  import {AuthContext} from "../index" ;
3  import * as firebase from "firebase" ;
4
5  const Login = () => {
6      const [email , setEmail] = useState("") ;
7      const [password , setPassword] = useState("") ;
8      const [error, setErrors] = useState("") ;
9
10     const Auth = useContext(AuthContext) ;
11     const handleForm = e => {
12         e.preventDefault();
13         firebase
14         .auth()
15         .createUserWithEmailAndPassword(email , password)
16         .then(res => {
17             if (res.user) Auth.setLoggedIn(true) ;
18         })
19         .catch(e => {
20             setErrors(e.message);
21         });
22     };
```

15. Silahkan coba pada aplikasi dengan cara klik login kemudian masukkan email dan password yang telah didaftarkan, kemudian perhatikan keterangan login pada pojok kiri atas dengan perubahan kondisi dari false menjadi true yang menandakan anda telah berhasil login

[Join](#) [Login](#)

Join

 Join With Google

LOGIN