

Pemrograman Berbasis Framework

"Firebase Login"



Oleh :

Angga Maulana Athaariq / 09

TI-3B / 1741720138

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

A. Reducers

1. Index.js

```
index.html ...\public  index.html ...\build  JS index.js ...\reducers X
pertemuan-keduabelas > firebase-login > reducers > JS index.js > default
1  import { combineReducers } from "redux";
2  import auth from "../auth";
3  export default combineReducers({ auth });
```

2. Auth.js

```
JS auth.js X  JS configureStore.js  index.html ...\public
pertemuan-keduabelas > firebase-login > reducers > JS auth.js >
1  import {
2    LOGIN_REQUEST,
3    LOGIN_SUCCESS,
4    LOGIN_FAILURE,
5    LOGOUT_REQUEST,
6    LOGOUT_SUCCESS,
7    LOGOUT_FAILURE,
8    VERIFY_REQUEST,
9    VERIFY_SUCCESS,
10 } from "../actions";
11
12 export default (
13   state = {
14     isLoggingIn: false,
15     isLoggingOut: false,
16     isVerifying: false,
17     loginError: false,
18     logoutError: false,
19     isAuthenticated: false,
20     user: {},
21   },
22   action
23 ) => {
24   switch (action.type) {
25     case LOGIN_REQUEST:
26       return {
27         ...state,
28         isLoggingIn: true,
29         loginError: false,
30       };
31     case LOGIN_SUCCESS:
32       return {
33         ...state,
34         isLoggingIn: false,
35         isAuthenticated: true,
36         user: action.user,
37       };
38     case LOGIN_FAILURE:
39       return {
40         ...state,
41         isLoggingIn: false,
42         isAuthenticated: false,
43         loginError: true,
44       };
45     case LOGOUT_REQUEST:
46       return {
47         ...state,
48         isLoggingOut: true,
49         logoutError: false,
50       };
51     case LOGOUT_SUCCESS:
52       return {
53         ...state,
54         isLoggingOut: false,
55         isAuthenticated: false,
56         user: {},
57       };
58     case LOGOUT_FAILURE:
59       return {
60         ...state,
61         isLoggingOut: false,
62         logoutError: true,
63       };
64     case VERIFY_REQUEST:
65       return {
66         ...state,
67         isVerifying: true,
68         verifyingError: false,
69       };
70     case VERIFY_SUCCESS:
71       return {
72         ...state,
73         isVerifying: false,
74       };
75   }
76 }
```

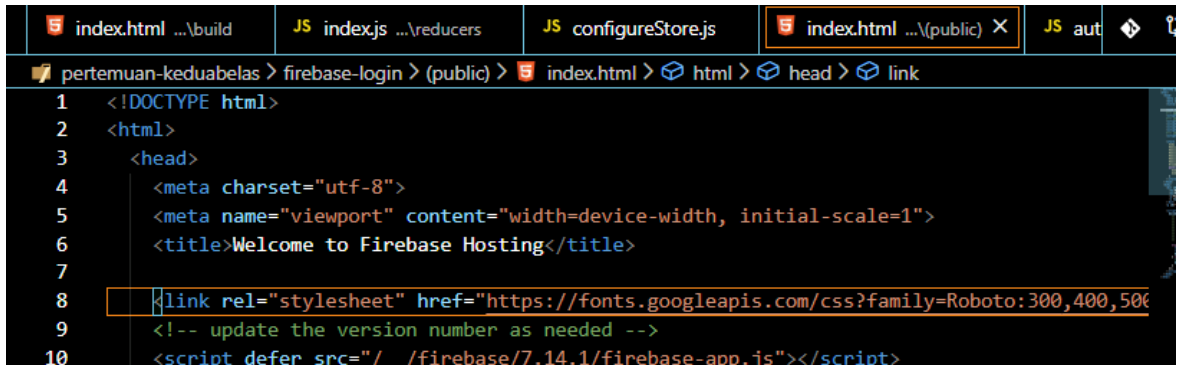
```
JS auth.js X  JS configureStore.js  index.html ...
pertemuan-keduabelas > firebase-login > reducers > JS au
38   case LOGIN_FAILURE:
39     return {
40       ...state,
41       isLoggingIn: false,
42       isAuthenticated: false,
43       loginError: true,
44     };
45   case LOGOUT_REQUEST:
46     return {
47       ...state,
48       isLoggingOut: true,
49       logoutError: false,
50     };
51   case LOGOUT_SUCCESS:
52     return {
53       ...state,
54       isLoggingOut: false,
55       isAuthenticated: false,
56       user: {},
57     };
58   case LOGOUT_FAILURE:
59     return {
60       ...state,
61       isLoggingOut: false,
62       logoutError: true,
63     };
64   case VERIFY_REQUEST:
65     return {
66       ...state,
67       isVerifying: true,
68       verifyingError: false,
69     };
70   case VERIFY_SUCCESS:
71     return {
72       ...state,
73       isVerifying: false,
74     };
75 }
```

B. ConfigureStore.js



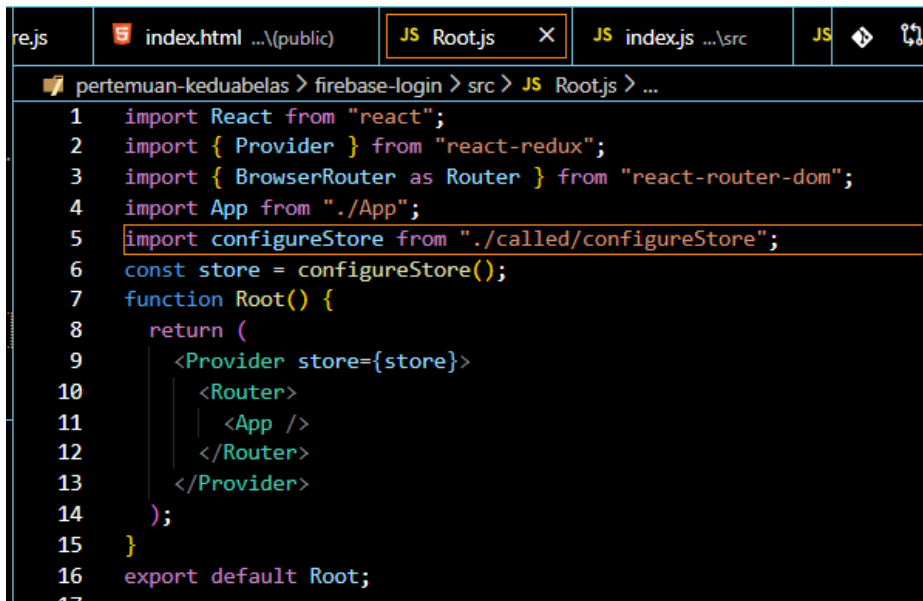
```
1 import { applyMiddleware, createStore } from "redux";
2 import thunkMiddleware from "redux-thunk";
3
4 import { verifyAuth } from "../actions/";
5 import rootReducer from "../reducers";
6
7 export default function configureStore(persistedState) {
8   const store = createStore(
9     rootReducer,
10    persistedState,
11    applyMiddleware(thunkMiddleware)
12  );
13  store.dispatch(verifyAuth());
14  return store;
15 }
16
```

C. Index.html (Public)



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Welcome to Firebase Hosting</title>
7
8     <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:300,400,500">
9     <!-- update the version number as needed -->
10    <script defer src="//firebase/7.14.1/firebase-app.js"></script>
```

D. Root.js



```
1 import React from "react";
2 import { Provider } from "react-redux";
3 import { BrowserRouter as Router } from "react-router-dom";
4 import App from "../App";
5 import configureStore from "../called/configureStore";
6 const store = configureStore();
7 function Root() {
8   return (
9     <Provider store={store}>
10      <Router>
11        <App />
12      </Router>
13    </Provider>
14  );
15 }
16 export default Root;
17
```

E. Index.js (src)

```
Root.js | JS index.html ...\public | JS Root.js | JS index.js ...\src X | JS |
pertemuan-keduabelas > firebase-login > src > JS index.js
1 import React from "react";
2 import ReactDOM from "react-dom";
3 import Root from "../Root";
4 import * as serviceWorker from "../serviceWorker";
5 ReactDOM.render(<Root />, document.getElementById("root"));
6 // If you want your app to work offline and load faster, you c
7 // unregister() to register() below. Note this comes with some
8 // Learn more about service workers: https://bit.ly/CRA-PWA
9 serviceWorker.unregister();
```

F. App.js (src)

```
Root.js | JS index.js ...\src | JS App.js X | JS Login.js | JS H |
pertemuan-keduabelas > firebase-login > src > JS App.js > App
1 import React from "react";
2 import { Route, Switch } from "react-router-dom";
3 import { connect } from "react-redux";
4 import ProtectedRoute from "../components/ProtectedRoute";
5 import Home from "../components/Home";
6 import Login from "../components/Login";
7 function App(props) {
8   const { isAuthenticated, isVerifying } = props;
9   return (
10     <Switch>
11       <ProtectedRoute
12         exact
13         path="/"
14         component={Home}
15         isAuthenticated={isAuthenticated}
16         isVerifying={isVerifying}
17       />
18       <Route path="/login" component={Login} />
19     </Switch>
20   );
21 }
22 function mapStateToProps(state) {
23   return {
24     isAuthenticated: state.auth.isAuthenticated,
25     isVerifying: state.auth.isVerifying,
26   };
27 }
28 export default connect(mapStateToProps)(App);
29
```

G. Components

1. Home.js

```
JS Root.js JS index.js ...src JS App.js JS Login.js JS Home.js X
pertemuan-keduabelas > firebase-login > components > JS Home.js > Home
1 import React, { Component } from "react";
2 import { connect } from "react-redux";
3 import { logoutUser } from "../actions";
4
5 class Home extends Component { handleLogout = () => {
6   const { dispatch } = this.props; dispatch(logoutUser());
7 };
8 render() { const { isLoggingOut, logoutError } = this.props;
9   return ( <div> <h1>This is your app's protected area.</h1> <p>Any routes here will also be
```

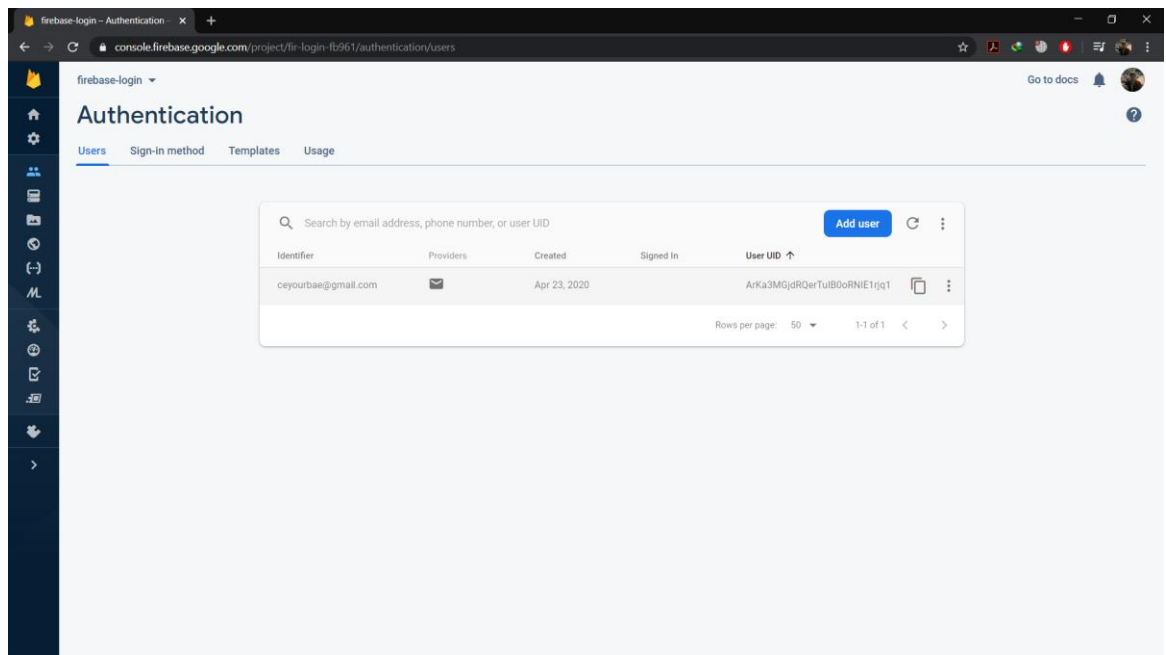
2. Login.js




```
JS Root.js JS index.js ...src JS App.js JS Login.js X
pertemuan-keduabelas > firebase-login > components > JS Login.js > ...
1 import React, { Component } from "react";
2 import { connect } from "react-redux";
3 import { Redirect } from "react-router-dom";
4 import { loginUser } from "../actions";
5 import { withStyles } from "@material-ui/styles";
6 import Avatar from "@material-ui/core/Avatar";
7 import Button from "@material-ui/core/Button";
8 import TextField from "@material-ui/core/TextField";
9 import LockOutlinedIcon from "@material-ui/icons/LockOutlined";
10 import Typography from "@material-ui/core/Typography";
11 import Paper from "@material-ui/core/Paper";
12 import Container from "@material-ui/core/Container";
13 const styles = () => ({
14   "@global": {
15     body: {
16       backgroundColor: "#fff",
17     },
18   },
19   paper: {
20     marginTop: 100,
21     display: "flex",
22     padding: 20,
23     flexDirection: "column",
24     alignItems: "center",
25   },
26   avatar: {
27     marginLeft: "auto",
28     marginRight: "auto",
29     backgroundColor: "#f50057",
30   },
31   form: {
32     marginTop: 1,
33   },
34   errorText: {
35     color: "#f50057",
36     marginBottom: 5,
37     textAlign: "center",
```

```
html ...\public JS Rootjs JS indexjs ...\src JS Appjs JS Loginjs X
pertemuan-keduabelas > firebase-login > components > JS Loginjs > ...
40 class Login extends Component {
41   state = { email: "", password: "" };
42   handleEmailChange = ({ target }) => {
43     this.setState({ email: target.value });
44   };
45   handlePasswordChange = ({ target }) => {
46     this.setState({ password: target.value });
47   };
48   handleSubmit = () => {
49     const { dispatch } = this.props;
50     const { email, password } = this.state;
51     dispatch(loginUser(email, password));
52   };
53   render() {
54     const { classes, loginError, isAuthenticated } = this.props;
55     if (isAuthenticated) {
56       return <Redirect to="/" />;
57     } else {
58       return (
59         <Container component="main" maxWidth="xs">
60           <Paper className={classes.paper}>
61             <Avatar className={classes.avatar}>
62               <LockOutlinedIcon />
63             </Avatar>
64             <Typography component="h1" variant="h5">
65               Sign in
66             </Typography>
67             <TextField
68               variant="outlined"
69               margin="normal"
70               fullWidth
71               id="email"
72               label="Email Address"
73               name="email"
74               onChange={this.handleEmailChange}
75             />
76             <TextField
```

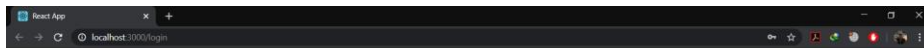
```
html ...\public JS Rootjs JS indexjs ...\src JS Appjs JS Loginjs X JS Home
pertemuan-keduabelas > firebase-login > components > JS Loginjs > ...
76 <TextField
77   variant="outlined"
78   margin="normal"
79   fullWidth
80   name="password"
81   label="Password"
82   type="password"
83   id="password"
84   onChange={this.handlePasswordChange}
85 />
86 {loginError && (
87   <Typography component="p" className={classes.errorText}>
88     Incorrect email or password.
89   </Typography>
90 )}
91 <Button
92   type="button"
93   fullWidth
94   variant="contained"
95   color="primary"
96   className={classes.submit}
97   onClick={this.handleSubmit}
98 >
99   Sign In
100 </Button>
101 </Paper>
102 </Container>
103 );
104 }
105 }
106 }
107 function mapStateToProps(state) {
108   return {
109     isLoggingIn: state.auth.isLoggingIn,
110     loginError: state.auth.loginError,
111     isAuthenticated: state.auth.isAuthenticated,
112   };
113 }
```


H. Authentication Firebase

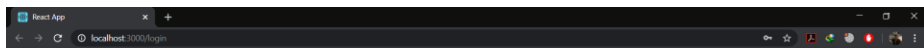



Search by email address, phone number, or user UID					Add user	Refresh	More
Identifier	Providers	Created	Signed in	User UID ↑			
ceyourbae@gmail.com		Apr 23, 2020		ArKa3MGjdRQerTulB0oRNIE1rjq1			
Rows per page: 50					1-1 of 1	<	>

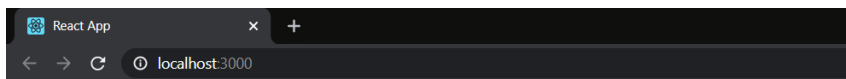
I. Output




Sign in




Sign in



This is your app's protected area.

Any routes here will also be protected

Logout