LAPORAN TUGAS PEMROGRAMAN BERBASIS FRAMEWORK

Modul 11

Firebase Login

****

Disusun Oleh :

Nama : Hunayn Risatayn

NIM : 1841720148

Kelas : TI-3E

Program Studi D-IV Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

April 2021

# Praktikum – Bagian 1: Firebase Console Setup

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| **1** | Setting Firebase Console  Login ke firebase console dan add a new project. Pilih authentification dan pilih tab menu Set up sign in method      Setelah enabling athentification pilih menu Database dan pilih create database        **Setting react app dan firebase**  Buat aplikasi baru dengan perintah      Run project baru dengan perintah    Install firebase    Pilih setting untuk firebase nya firebase dan Hosting            Untuk membangun firebase      **Instalasi Redux**  Install Redux dan react redux, redux thunk          Install firebase    Pilih UI library    Paste kan file public/index.html    Install material icon    **Setting firebase config**  Pilih project setting            Pilih config    Edit pada src/firebase/firebase.js  import firebase from "firebase/app";  import "firebase/auth";  import "firebase/firestore";  // For Firebase JS SDK v7.20.0 and later, measurementId is optional  const firebaseConfig = {    apiKey: "AIzaSyAIzw2L6\_fC6V3pg\_FV\_56RyZbjLHG6vhk",    authDomain: "fir-login-hunayn.firebaseapp.com",    databaseURL: "https://fir-login-hunayn-default-rtdb.firebaseio.com",    projectId: "fir-login-hunayn",    storageBucket: "fir-login-hunayn.appspot.com",    messagingSenderId: "189410526567",    appId: "1:189410526567:web:f68f97d0d41bbe0999dbc2",    measurementId: "G-FL6MGT379N",  };  export const myFirebase = firebase.initializeApp(firebaseConfig);  const baseDb = myFirebase.firestore();  export const db = baseDb;  Edit pada src/actions/auth.js sebagaimana berikut  import { myFirebase } from "../firebase/firebase";  export const LOGIN\_REQUEST = "LOGIN\_REQUEST";  export const LOGIN\_SUCCESS = "LOGIN\_SUCCESS";  export const LOGIN\_FAILURE = "LOGIN\_FAILURE";  export const LOGOUT\_REQUEST = "LOGOUT\_REQUEST";  export const LOGOUT\_SUCCESS = "LOGOUT\_SUCCESS";  export const LOGOUT\_FAILURE = "LOGOUT\_FAILURE";  export const VERIFY\_REQUEST = "VERIFY\_REQUEST";  export const VERIFY\_SUCCESS = "VERIFY\_SUCCESS";  const requestLogin = () => {    return {      type: LOGIN\_REQUEST,    };  };  const receiveLogin = (user) => {    return {      type: LOGIN\_SUCCESS,      user,    };  };  const loginError = () => {    return {      type: LOGIN\_FAILURE,    };  };  const requestLogout = () => {    return {      type: LOGOUT\_REQUEST,    };  };  const receiveLogout = () => {    return {      type: LOGOUT\_SUCCESS,    };  };  const logoutError = () => {    return {      type: LOGOUT\_FAILURE,    };  };  const verifyRequest = () => {    return {      type: VERIFY\_REQUEST,    };  };  const verifySuccess = () => {    return {      type: VERIFY\_SUCCESS,    };  };  export const loginUser = (email, password) => (dispatch) => {    dispatch(requestLogin());    myFirebase      .auth()      .signInWithEmailAndPassword(email, password)      .then((user) => {        dispatch(receiveLogin(user));      })      .catch((error) => {        //Do something with the error if you want!        dispatch(loginError());      });  };  export const logoutUser = () => (dispatch) => {    dispatch(requestLogout());    myFirebase      .auth()      .signOut()      .then(() => {        dispatch(receiveLogout());      })      .catch((error) => {        //Do something with the error if you want!        dispatch(logoutError());      });  };  export const verifyAuth = () => (dispatch) => {    dispatch(verifyRequest());    myFirebase.auth().onAuthStateChanged((user) => {      if (user !== null) {        dispatch(receiveLogin(user));      }      dispatch(verifySuccess());    });  };  **Setting Reducer**  Buka src/reducers/auth.js kemudian tambahkan file berikut  import {    LOGIN\_REQUEST,    LOGIN\_SUCCESS,    LOGIN\_FAILURE,    LOGOUT\_REQUEST,    LOGOUT\_SUCCESS,    LOGOUT\_FAILURE,    VERIFY\_REQUEST,    VERIFY\_SUCCESS,  } from "../actions/auth";  export default (    state = {      isLoggingIn: false,      isLoggingOut: false,      isVerifying: false,      loginError: false,      logoutError: false,      isAuthenticated: false,      user: {},    },    action  ) => {    switch (action.type) {      case LOGIN\_REQUEST:        return {          ...state,          isLoggingIn: true,          loginError: false,        };      case LOGIN\_SUCCESS:        return {          ...state,          isLoggingIn: false,          isAuthenticated: true,          user: action.user,        };      case LOGIN\_FAILURE:        return {          ...state,          isLoggingIn: false,          isAuthenticated: false,          loginError: true,        };      case LOGOUT\_REQUEST:        return {          ...state,          isLoggingOut: true,          logoutError: false,        };      case LOGOUT\_SUCCESS:        return {          ...state,          isLoggingOut: false,          isAuthenticated: false,          user: {},        };      case LOGOUT\_FAILURE:        return {          ...state,          isLoggingOut: false,          logoutError: true,        };      case VERIFY\_REQUEST:        return {          ...state,          isVerifying: true,          verifyingError: false,        };      case VERIFY\_SUCCESS:        return {          ...state,          isVerifying: false,        };      default:        return state;    }  };  Tambahkan source code pada src/reducers/index.js seperti berikut  import { combineReducers } from "redux";  import auth from "./auth";  export default combineReducers({ auth });  Buatlah file baru bernama configureStore.js pada folder src  import { applyMiddleware, createStore } from "redux";  import thunkMiddleware from "redux-thunk";  import { verifyAuth } from "./actions/auth";  import rootReducer from "./reducers";  export default function configureStore(persistedState) {    const store = createStore(      rootReducer,      persistedState,      applyMiddleware(thunkMiddleware)    );    store.dispatch(verifyAuth());    return store;  }  Buatlah file baru bernama Root.js pada folder src kemudian tambahkan source code seperti berikut  import React from "react";  import { Provider } from "react-redux";  import { BrowserRouter as Router } from "react-router-dom";  import App from "./App";  import configureStore from "./configureStore";  const store = configureStore();  function Root() {    return (      <Provider store={store}>        <Router>          <App />        </Router>      </Provider>    );  }  export default Root;  Edit source code pada src/index.js seperti berikut  import React from "react";  import ReactDOM from "react-dom";  import Root from "./Root";  import \* as serviceWorker from "./serviceWorker";  ReactDOM.render(<Root />, document.getElementById("root"));  // If you want your app to work offline and load faster, you can change  // unregister() to register() below. Note this comes with some pitfalls.  // Learn more about service workers: https://bit.ly/CRA-PWA  serviceWorker.unregister();  Edit source code pada src/App.js seperti berikut  import React from "react";  import { Route, Switch } from "react-router-dom";  import { connect } from "react-redux";  import ProtectedRoute from "./components/protectedRoute";  import Home from "./components/home";  import Login from "./components/login";  function App(props) {    const { isAuthenticated, isVerifying } = props;    return (      <Switch>        <ProtectedRoute          exact          path="/"          component={Home}          isAuthenticated={isAuthenticated}          isVerifying={isVerifying}        />        <Route path="/login" component={Login} />      </Switch>    );  }  function mapStateToProps(state) {    return {      isAuthenticated: state.auth.isAuthenticated,      isVerifying: state.auth.isVerifying,    };  }  export default connect(mapStateToProps)(App);  Buatlah file baru pada src/components/login.js kemudian tambahkan source code berikut  import React, { Component } from "react";  import { connect } from "react-redux";  import { Redirect } from "react-router-dom";  import { loginUser } from "../actions/auth";  import { withStyles } from "@material-ui/styles";  import Avatar from "@material-ui/core/Avatar";  import Button from "@material-ui/core/Button";  import TextField from "@material-ui/core/TextField";  import LockOutlinedIcon from "@material-ui/icons/LockOutlined";  import Typography from "@material-ui/core/Typography";  import Paper from "@material-ui/core/Paper";  import Container from "@material-ui/core/Container";  const styles = () => ({    "@global": {      body: {        backgroundColor: "#fff",      },    },    paper: {      marginTop: 100,      display: "flex",      padding: 20,      flexDirection: "column",      alignItems: "center",    },    avatar: {      marginLeft: "auto",      marginRight: "auto",      backgroundColor: "#f50057",    },    form: {      marginTop: 1,    },    errorText: {      color: "#f50057",      marginBottom: 5,      textAlign: "center",    },  });  class Login extends Component {    state = { email: "", password: "" };    handleEmailChange = ({ target }) => {      this.setState({ email: target.value });    };    handlePasswordChange = ({ target }) => {      this.setState({ password: target.value });    };    handleSubmit = () => {      const { dispatch } = this.props;      const { email, password } = this.state;      dispatch(loginUser(email, password));    };    render() {      const { classes, loginError, isAuthenticated } = this.props;      if (isAuthenticated) {        return <Redirect to="/" />;      } else {        return (          <Container component="main" maxWidth="xs">            <Paper className={classes.paper}>              <Avatar className={classes.avatar}>                <LockOutlinedIcon />              </Avatar>              <Typography component="h1" variant="h5">                Sign in              </Typography>              <TextField                variant="outlined"                margin="normal"                fullWidth                id="email"                label="Email Address"                name="email"                onChange={this.handleEmailChange}              />              <TextField                variant="outlined"                margin="normal"                fullWidth                name="password"                label="Password"                type="password"                id="password"                onChange={this.handlePasswordChange}              />              {loginError && (                <Typography component="p" className={classes.errorText}>                  Incorrect email or password.                </Typography>              )}              <Button                type="button"                fullWidth                variant="contained"                color="primary"                className={classes.submit}                onClick={this.handleSubmit}              >                Sign In              </Button>            </Paper>          </Container>        );      }    }  }  function mapStateToProps(state) {    return {      isLoggingIn: state.auth.isLoggingIn,      loginError: state.auth.loginError,      isAuthenticated: state.auth.isAuthenticated,    };  }  export default withStyles(styles)(connect(mapStateToProps)(Login));  Buatlah file baru pada src/components/home.js kemudian tambahkan source code seperti berikut  import React, { Component } from "react";  import { connect } from "react-redux";  import { logoutUser } from "../actions/auth";  class Home extends Component {    handleLogout = () => {      const { dispatch } = this.props;      dispatch(logoutUser());    };    render() {      const { isLoggingOut, logoutError } = this.props;      return (        <div>          <h1>This is your app's protected area.</h1>          <p>Any routes here will also be protected</p>          <button onClick={this.handleLogout}>Logout</button>          {isLoggingOut && <p>Logging Out....</p>}          {logoutError && <p>Error logging out</p>}        </div>      );    }  }  function mapStateToProps(state) {    return {      isLoggingOut: state.auth.isLoggingOut,      logoutError: state.auth.logoutError,    };  }  export default connect(mapStateToProps)(Home);  Buatlah file baru pada src/components/protectedRoute.js kemudian tambahkan source code berikut  import React from "react";  import { Route, Redirect } from "react-router-dom";  const ProtectedRoute = ({    component: Component,    isAuthenticated,    isVerifying,    ...rest  }) => (    <Route      {...rest}      render={(props) =>        isVerifying ? (          <div />        ) : isAuthenticated ? (          <Component {...props} />        ) : (          <Redirect            to={{              pathname: "/login",              state: { from: props.location },            }}          />        )      }    />  );  export default ProtectedRoute;  Running projet dengan syntax “npm start” pada terminal    Masuk ke firebase console kemudian add user    Execute “npm start” pada terminal kemudian masukkan username dan password    Kemudian masuk ke dalam halaman beranda    Link GitHub : <https://github.com/hunaynr/framework_prog/tree/main/11thWeek>  Link Youtube : https://www.youtube.com/channel/UCP53BJw16j\_nt4Y0sETg1rA |