

**TUGAS INDIVIDU**  
**PEMROGRAMAN BERBASIS FRAMEWORK**

**“Firebase Login”**



**oleh:**

**Tika Yulianti / 1741720153**

**TI-3A**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

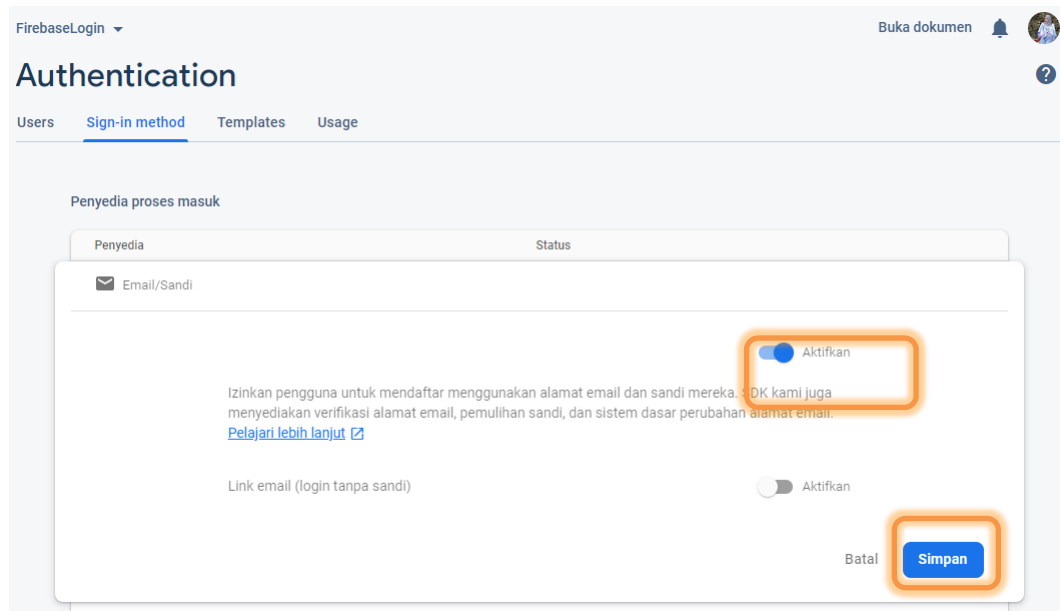
**2020**

## Praktikum

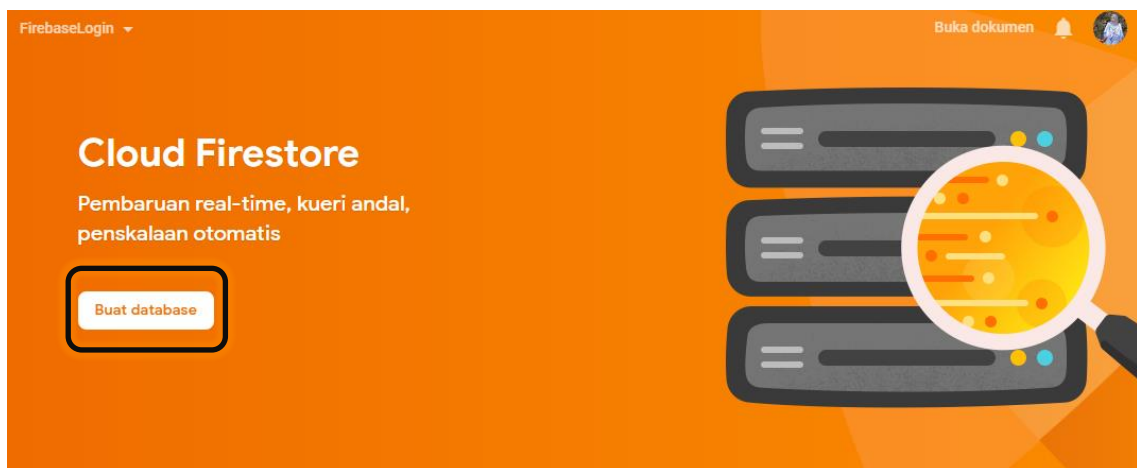
### Setting Firebase

Langkah-langkah praktikum:

1. Melakukan setting firebase
  - a. Melakukan login ke firebase console
  - b. Menambahkan proyek baru dengan cara **add a new project** dan menamainya **FirestoreLogin**
  - c. Memilih menu **Authentication** lalu memilih tab menu **Sign-in method** untuk melakukan aktivasi Email/Sandi. Kemudian klik **Simpan**



2. Memilih menu **Database** dan klik **Buat Database**



3. Pada langkah pertama memilih **Mulai dalam mode pengujian** lalu klik **Berikutnya**

**Buat database** ✕

1 Amankan aturan untuk Cloud Firestore


2 Tetapkan lokasi Cloud Firestore

Setelah menentukan struktur data, Anda harus menulis aturan untuk mengamankan data Anda.  
[Pelajari lebih lanjut](#)

☐ **Mulai dalam mode produksi**  
Data Anda akan dijadikan pribadi secara default. Akses baca/tulis klien hanya akan diberikan sesuai yang ditetapkan oleh aturan keamanan Anda.

☒ **Mulai dalam mode pengujian**  
Data Anda akan terbuka secara default untuk mengaktifkan penyiapan cepat. Akses baca/tulis klien akan ditolak setelah 30 hari jika aturan keamanan tidak diperbarui.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2020, 5, 23);
    }
  }
}
```

 **Siapa saja yang memiliki referensi database Anda akan dapat membaca dan menulis ke database Anda selama 30 hari**

Mengaktifkan Cloud Firestore akan mencegah Anda menggunakan Cloud Datastore dengan project ini, khususnya dari aplikasi App Engine yang terkait

Batal **Berikutnya**


4. Kemudian untuk langkah kedua mengeklik button **Selesai**

**Buat database** ✕

1 Amankan aturan untuk Cloud Firestore

2 Tetapkan lokasi Cloud Firestore

Setelan lokasi Anda adalah tempat data Cloud Firestore akan disimpan.

 **Setelah ditetapkan, Anda tidak lagi dapat mengubah setelan lokasi ini nanti. Selain itu, setelan lokasi ini akan menjadi lokasi untuk bucket Cloud Storage default Anda.**

[Pelajari lebih lanjut](#)

Lokasi Cloud Firestore

nam5 (us-central) ▼

Mengaktifkan Cloud Firestore akan mencegah Anda menggunakan Cloud Datastore dengan project ini, khususnya dari aplikasi App Engine yang terkait

Batal **Selesai**

## Setting react app dan firebase

Langkah-langkah praktikum:

1. Menambahkan aplikasi react dengan memasukkan perintah sebagai berikut:

```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> create-react-app login

Creating a new React app in E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\login.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

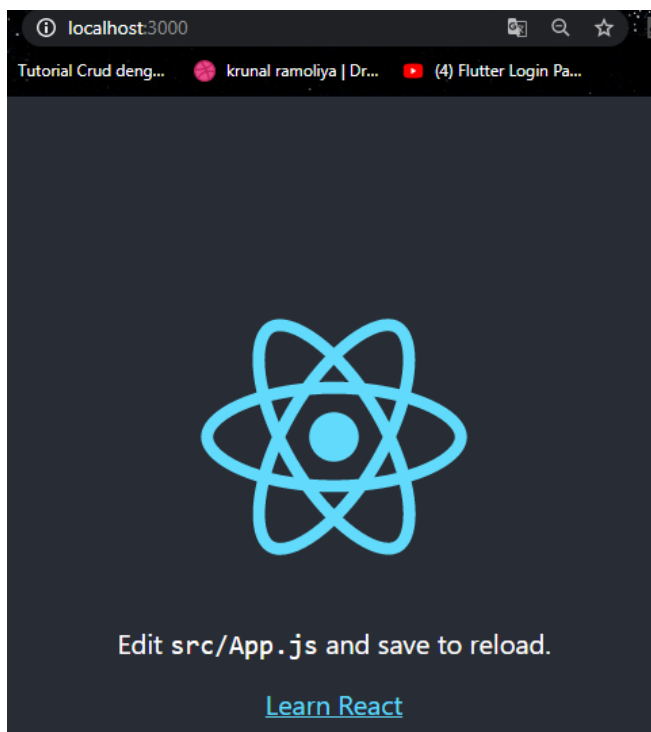
2. Menginstall project baru dengan perintah seperti berikut:

```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\login> npm run start

> login@0.1.0 start E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\login
> react-scripts start

i [wds]: Project is running at http://192.168.232.1/
i [wds]: webpack output is served from
i [wds]: Content not from webpack is served from E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\login\public
i [wds]: 404s will fallback to /
Starting the development server...
```

Pada web browser akan tampil seperti berikut:



3. Melakukan instalasi firebase seperti berikut:

a. Menginstal firebase tools

```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> npm install -g firebase-tools
```

b. Melakukan firebase login

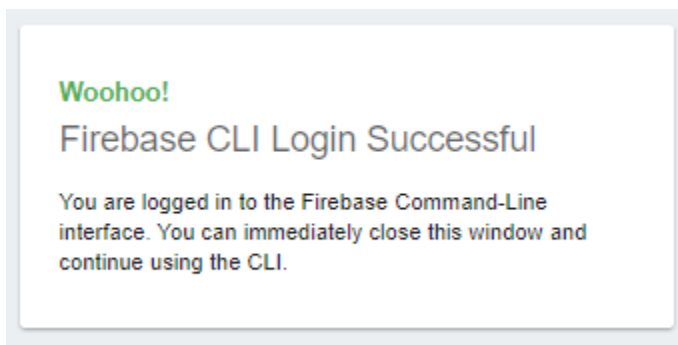
```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> firebase login
i Firebase optionally collects CLI usage and error reporting information to help improve our products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin>
? Allow Firebase to collect CLI usage and error reporting information? Yes
i To change your data collection preference at any time, run `firebase logout` and log in again.

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhgmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response_type=code&state=938749423&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...

+ Success! Logged in as tikayulian27@gmail.com
```

Lalu akan muncul



c. Memasukkan **firebase init** pada terminal

```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> firebase init

#####  #####  #####  #####  #####  #####  #####  #####
##      ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##
#####  ##  #####  #####  #####  #####  #####  #####
##      ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##
##      #####  ##  #####  #####  ##  ##  #####  #####

You're about to initialize a Firebase project in this directory:

E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin
```

4. Memilih setting untuk firebase nya firebase dan Hosting

```
? Are you ready to proceed? Yes
? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices.
( ) Database: Deploy Firebase Realtime Database Rules
(*) Firestore: Deploy rules and create indexes for Firestore
( ) Functions: Configure and deploy Cloud Functions
>(*) Hosting: Configure and deploy Firebase Hosting sites
( ) Storage: Deploy Cloud Storage security rules
( ) Emulators: Set up local emulators for Firebase features
```

5. Memilih Use an existing project dan memilih firebase yang telah dibuat yaitu FirebaseLogin

```
? Please select an option: Use an existing project
? Select a default Firebase project for this directory:
  android-add08 (Android)
  androidtika-dc30c (androidTika)
  fashionshop-4af96 (FashionShop)
> fir-login-cde0c (FirebaseLogin)
  pbfmodul10 (PBFModul10)
  webuas-6d48b (webuas)
```

6. Memilih Y lalu N untuk overwriting index.html

```
? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
+ Wrote public/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
i Writing gitignore file to .gitignore...

+ Firebase initialization complete!
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin>
```

7. Langkah selanjutnya adalah membangun firebase

- a. Memasukkan perintah **npm run build** pada terminal

```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\login> npm run build

> login@0.1.0 build E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\login
> react-scripts build

Creating an optimized production build...
Compiled successfully.
```

- b. Lalu **firebase serve**

```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\login> firebase serve

=== Serving from 'E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin'...

i hosting: Serving hosting files from: public
+ hosting: Local server: http://localhost:5000
```

Welcome

## Firebase Hosting Setup Complete

You're seeing this because you've successfully setup Firebase Hosting. Now it's time to go build something extraordinary!

[OPEN HOSTING DOCUMENTATION](#)

Firebase SDK loaded with auth, database, messaging, storage

## Instalasi Redux

### 1. Menginstall Redux dan react redux, redux thunk

#### a. Memasukkan **npm install redux** pada terminal

```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> npm install redux
npm WARN saveError ENOENT: no such file or directory, open 'E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\package.json'
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No description
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No repository field.
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No README data
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No license field.

+ redux@4.0.5
added 4 packages from 3 contributors and audited 4 packages in 10.553s
found 0 vulnerabilities
```

#### b. Lalu **npm install react-redux**

```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> npm install react-redux
npm WARN saveError ENOENT: no such file or directory, open 'E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\package.json'
npm WARN react-redux@7.2.0 requires a peer of react@^16.8.3 but none is installed. You must install peer dependencies yourself.
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No description
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No repository field.
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No README data
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No license field.

+ react-redux@7.2.0
added 7 packages from 5 contributors and audited 21 packages in 7.037s
found 0 vulnerabilities
```

#### c. Lalu **npm install redux-thunk**

```
PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> npm install redux-thunk
npm WARN saveError ENOENT: no such file or directory, open 'E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\package.json'
npm WARN react-redux@7.2.0 requires a peer of react@^16.8.3 but none is installed. You must install peer dependencies yourself.
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No description
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No repository field.
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No README data
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No license field.

+ redux-thunk@2.3.0
added 1 package from 1 contributor and audited 34 packages in 5.762s
found 0 vulnerabilities
```

#### d. Npm install react-router-dom



```

PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> npm install react-router-dom
npm WARN saveError ENOENT: no such file or directory, open 'E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\package.json'
npm WARN react-redux@7.2.0 requires a peer of react@^16.8.3 but none is installed. You must install peer dependencies yourself.
npm WARN react-router-dom@5.1.2 requires a peer of react@>=15 but none is installed. You must install peer dependencies yourself.
npm WARN react-router@5.1.2 requires a peer of react@>=15 but none is installed. You must install peer dependencies yourself.
npm WARN mini-create-react-context@0.3.2 requires a peer of react@^0.14.0 || ^15.0.0 || ^16.0.0 but none is installed. You must install peer dependencies yourself.
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No description
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No repository field.
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No README data
npm WARN 28-TikaYulianti-PBF-FirebaseLogin No license field.

+ react-router-dom@5.1.2
added 11 packages from 5 contributors and audited 86 packages in 6.765s
found 0 vulnerabilities

```

## 2. Menginstall firebase

```

PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> npm install firebase

> core-js@3.6.5 postinstall E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin\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!

```

## 3. Memilih UI library

```

PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> npm install @material-ui/core

```

## 4. Memasukkan code berikut ke dalam file public → index.html

```

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"/>

```

## 5. Menginstall material icon

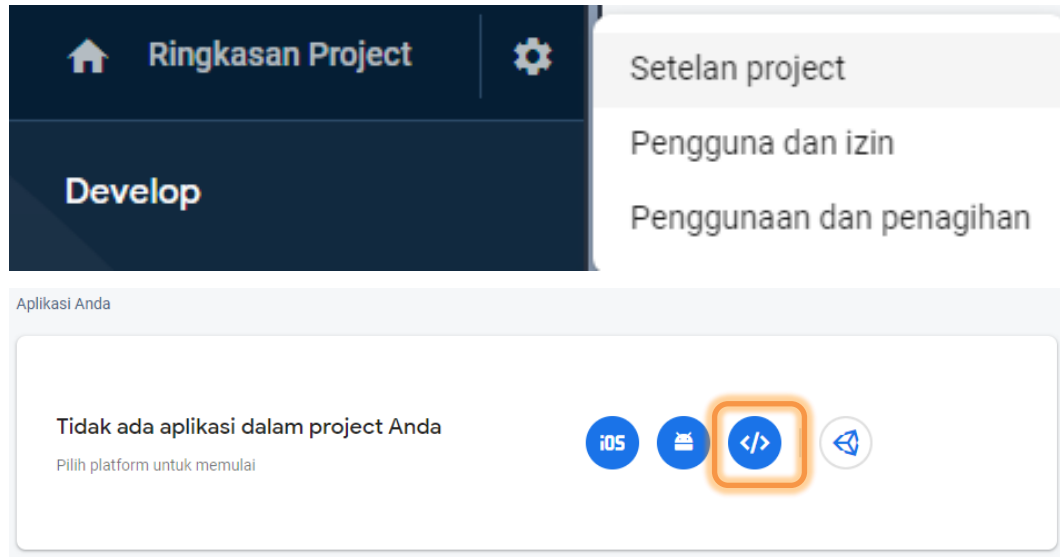
```

PS E:\POLINEMA\Semester 6\PBF\28-TikaYulianti-PBF-FirebaseLogin> npm install @material-ui/icons

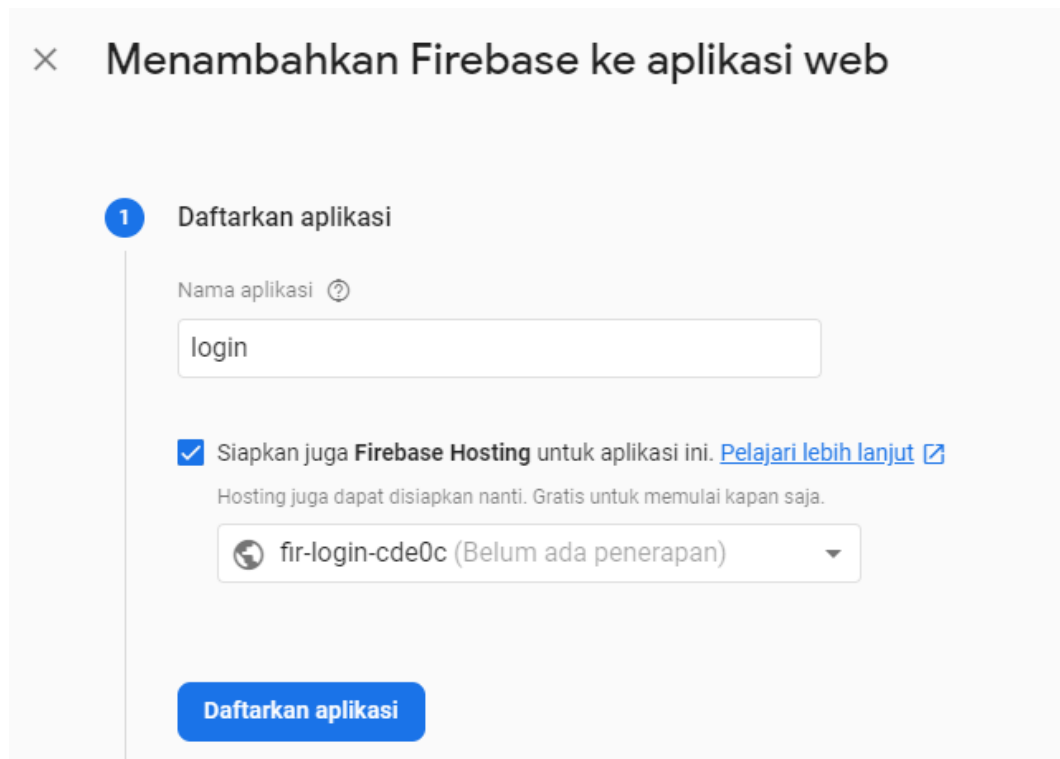
```

## Setting Firebase Config

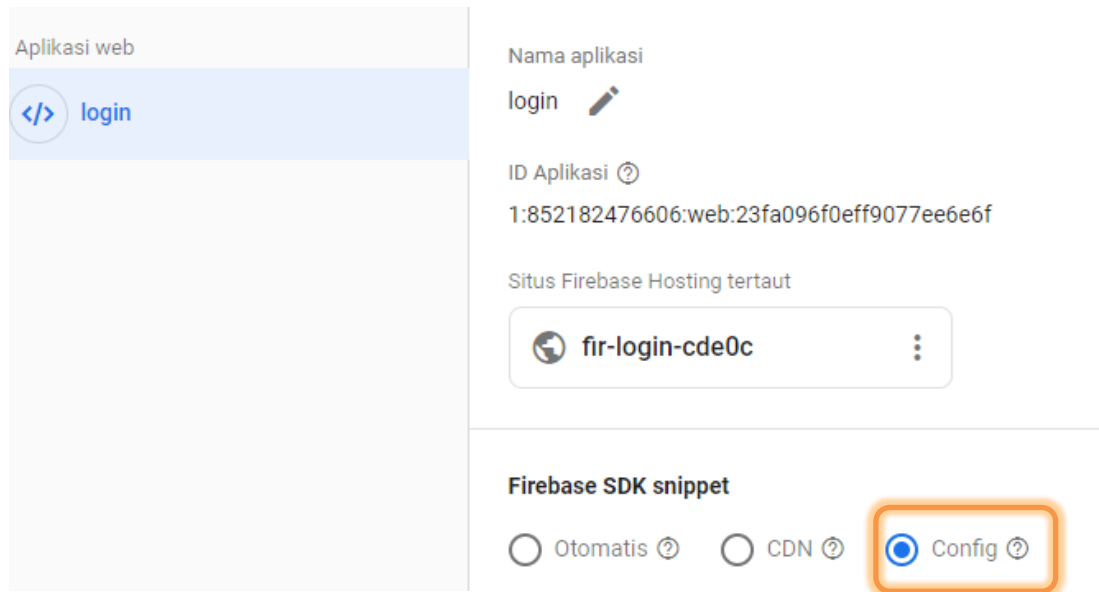
1. Memilih **Project setting** pada firebase



Lalu mendaftarkan aplikasi



## 2. Memilih config pada **Firestore SDK snippet**



## 3. Mengedit file **src** → **firebase** → **firebase.js**

```
login > src > firebase > JS firebase.js > ...
1  import firebase from "firebase/app";
2  import "firebase/auth";
3  import "firebase/firestore";
4
5  const firebaseConfig = {
6    apiKey: "AIzaSy8UIbUJjOL9TCJCpMpUOcQT59C8q1HXmsk",
7    authDomain: "fir-login-cde0c.firebaseio.com",
8    databaseURL: "https://fir-login-cde0c.firebaseio.com",
9    projectId: "fir-login-cde0c",
10   storageBucket: "fir-login-cde0c.appspot.com",
11   messagingSenderId: "852182476606",
12   appId: "1:852182476606:web:23fa096f0eff9077ee6e6f",
13   measurementId: "G-BSHDFKPJ7Z"
14 };
15
16 export const myFirebase = firebase.initializeApp(firebaseConfig);
17 const baseDb = myFirebase.firestore();
18 export const db = baseDb;
```

4. Menambahkan file **src** → **actions** → **auth.js**

```
login > src > actions > JS auth.js > ...
1  import { myFirebase } from "../firebase/firebase";
2
3  export const LOGIN_REQUEST = "LOGIN_REQUEST";
4  export const LOGIN_SUCCESS = "LOGIN_SUCCESS";
5  export const LOGIN_FAILURE = "LOGIN_FAILURE";
6
7  export const LOGOUT_REQUEST = "LOGOUT_REQUEST";
8  export const LOGOUT_SUCCESS = "LOGOUT_SUCCESS";
9  export const LOGOUT_FAILURE = "LOGOUT_FAILURE";
10
11 export const VERIFY_REQUEST = "VERIFY_REQUEST";
12 export const VERIFY_SUCCESS = "VERIFY_SUCCESS";
13
14 const requestLogin = () => {
15   ...return {
16     ...type: LOGIN_REQUEST
17   };
18 };
19
20 const receiveLogin = user => {
21   ...return {
22     ...type: LOGIN_SUCCESS,
23     ...user
24   };
25 };
26
27 const loginError = () => {
28   ...return {
29     ...type: LOGIN_FAILURE
30   };
31 };
32
33 const requestLogout = () => {
34   ...return {
35     ...type: LOGOUT_REQUEST
36   };
37 }
```

```
37 ..};
38 ..
39 ..const receiveLogout = () => {
40 ..  return {
41 ..    type: LOGOUT_SUCCESS
42 ..  };
43 ..};
44 ..
45 ..const logoutError = () => {
46 ..  return {
47 ..    type: LOGOUT_FAILURE
48 ..  };
49 ..};
50 ..
51 ..const verifyRequest = () => {
52 ..  return {
53 ..    type: VERIFY_REQUEST
54 ..  };
55 ..};
56 ..
57 ..const verifySuccess = () => {
58 ..  return {
59 ..    type: VERIFY_SUCCESS
60 ..  };
61 ..};
62 ..
63 ..export const loginUser = (email, password) => dispatch => {
64 ..  dispatch(requestLogin());
65 ..  myFirebase
66 ..    .auth()
67 ..    .signInWithEmailAndPassword(email, password)
68 ..    .then(user => {
69 ..      dispatch(receiveLogin(user));
70 ..    })
```

```

71     ...    .catch(error => {
72     ...    //Do something with the error if you want!
73     ...    dispatch(loginError());
74     ...    });
75     };
76     ...
77     export const logoutUser = () => dispatch => {
78     ...    dispatch(requestLogout());
79     ...    myFirebase
80     ...    .auth()
81     ...    .signOut()
82     ...    .then(() => {
83     ...    ...    dispatch(receiveLogout());
84     ...    })
85     ...    .catch(error => {
86     ...    //Do something with the error if you want!
87     ...    dispatch(logoutError());
88     ...    });
89     };
90     ...
91     export const verifyAuth = () => dispatch => {
92     ...    dispatch(verifyRequest());
93     ...    myFirebase.auth().onAuthStateChanged(user => {
94     ...    ...    if (user !== null) {
95     ...    ...    dispatch(receiveLogin(user));
96     ...    ...    }
97     ...    ...    dispatch(verifySuccess());
98     ...    ...    });
99     ...    };

```

## Setting Reducer

1. Membuat file `src → reducers → auth.js`

```
login > src > 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/auth";
11 export default (
12   state = {
13     ...isLoggingIn: false,
14     ...isLoggingOut: false,
15     ...isVerifying: false,
16     ...loginError: false,
17     ...logoutError: false,
18     ...isAuthenticated: false,
19     ...user: {}
20   },
21   action
22 ) => {
23   switch (action.type) {
24     case LOGIN_REQUEST:
25       return {
26         ...state,
27         ...isLoggingIn: true,
28         ...loginError: false
29       };
30     case LOGIN_SUCCESS:
31       return {
32         ...state,
33         ...isLoggingIn: false,
34         ...isAuthenticated: true,
35         ...user: action.user
36       };
37     case LOGIN_FAILURE:
```

```

login > src > reducers > JS auth.js > default
38  ...return {
39    ...state,
40    ...isLoggingIn: false,
41    ...isAuthenticated: false,
42    ...loginError: true
43  };
44  ...case LOGOUT_REQUEST:
45  ...return {
46    ...state,
47    ...isLoggingOut: true,
48    ...logoutError: false
49  };
50  ...case LOGOUT_SUCCESS:
51  ...return {
52    ...state,
53    ...isLoggingOut: false,
54    ...isAuthenticated: false,
55    ...user: {}
56  };
57  ...case LOGOUT_FAILURE:
58  ...return {
59    ...state,
60    ...isLoggingOut: false,
61    ...logoutError: true
62  };
63  ...case VERIFY_REQUEST:
64  ...return {
65    ...state,
66    ...isVerifying: true,
67    ...verifyingError: false
68  };
69  ...case VERIFY_SUCCESS:
70  ...return {
71    ...state,
72    ...isVerifying: false
73  };
74  ...default:
75  ...return state;
76  ...}
77  ...};

```

2. Membuat file `src → reducers → index.js`

```

login > src > reducers > JS index.js > default
1  import { combineReducers } from "redux";
2  import auth from "../auth";
3  export default combineReducers({ auth });

```

3. Menyetting **Redux plumbing**



4. Membuat file **src** → **configureStore.js**

```
login > src > JS configureStore.js > ...
1  import { applyMiddleware, createStore } from "redux";
2  import thunkMiddleware from "redux-thunk";
3
4  import { verifyAuth } from "../actions/auth";
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 }
```

5. Membuat file **src** → **Root.js**

```
login > src > JS 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 "../configureStore";
6  const store = configureStore();
7  function Root() {
8    return (
9      <Provider store={store}>
10        <Router>
11          <App />
12        </Router>
13      </Provider>
14    ); } export default Root;
15
```

6. Membuat file **src** → **index.js**

```
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
6  ReactDOM.render(<Root />, document.getElementById("root"));
7
8  // If you want your app to work offline and load faster, you can change
9  // unregister() to register() below. Note this comes with some pitfalls.
10 // Learn more about service workers: https://bit.ly/CRA-PWA
11 serviceWorker.unregister();
```



7. Menambahkan file `src` → `App.js`

```
login > src > JS App.js > ...
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
```

8. Membuat file `src` → `components` → `Login.js`

```
login > src > components > JS Login.js > [🔍] styles > [🔍] "@global" > [🔍] body
1  ✓ import React, { Component } from "react";
2  import { connect } from "react-redux";
3  import { Redirect } from "react-router-dom";
4  import { loginUser } from "../actions/auth";
5  import { withStyles } from "@material-ui/styles";
6
7  import Avatar from "@material-ui/core/Avatar";
8  import Button from "@material-ui/core/Button";
9  import TextField from "@material-ui/core/TextField";
10 import LockOutlinedIcon from "@material-ui/icons/LockOutlined";
11 import Typography from "@material-ui/core/Typography";
12 import Paper from "@material-ui/core/Paper";
13 import Container from "@material-ui/core/Container";
14
15  ✓ const styles = () => ({
16  ✓    "@global": {
17  ✓      body: {
18  ✓        backgroundColor: "#fff"
19  ✓      }
20  ✓    },
21  ✓    paper: {
22  ✓      marginTop: 100,
23  ✓      display: "flex",
24  ✓      padding: 20,
25  ✓      flexDirection: "column",
26  ✓      alignItems: "center"
27  ✓    },
28  ✓    avatar: {
29  ✓      marginLeft: "auto",
30  ✓      marginRight: "auto",
31  ✓      backgroundColor: "#f50057"
32  ✓    },
33  ✓    form: {
34  ✓      marginTop: 1
35  ✓    },
36  ✓    errorText: {
37  ✓      color: "#f50057",
38  ✓      marginBottom: 5,
39  ✓      textAlign: "center"
40  ✓    }
  }
```

```

41   });
42
43   class Login extends Component {
44     state = { email: "", password: "" };
45
46     handleEmailChange = ({ target }) => {
47       this.setState({ email: target.value });
48     };
49
50     handlePasswordChange = ({ target }) => {
51       this.setState({ password: target.value });
52     };
53
54     handleSubmit = () => {
55       const { dispatch } = this.props;
56       const { email, password } = this.state;
57
58       dispatch(loginUser(email, password));
59     };
60
61     render() {
62       const { classes, loginError, isAuthenticated } = this.props;
63       if (isAuthenticated) {
64         return <Redirect to="/" />;
65       } else {
66         return (
67           <Container component="main" maxWidth="xs">
68             <Paper className={classes.paper}>
69               <Avatar className={classes.avatar}>
70                 <LockOutlinedIcon />
71               </Avatar>
72               <Typography component="h1" variant="h5">
73                 Sign in
74               </Typography>
75               <TextField
76                 variant="outlined"
77                 margin="normal"
78                 fullWidth
79                 id="email"

```

```

79     id="email"
80     label="Email Address"
81     name="email"
82     onChange={this.handleEmailChange}
83   />
84   <TextField
85     variant="outlined"
86     margin="normal"
87     fullWidth
88     name="password"
89     label="Password"
90     type="password"
91     id="password"
92     onChange={this.handlePasswordChange}
93   />
94   {loginError && (
95     <Typography component="p" className={classes.errorText}>
96       Incorrect email or password.
97     </Typography>
98   )}
99   <Button
100     type="button"
101     fullWidth
102     variant="contained"
103     color="primary"
104     className={classes.submit}
105     onClick={this.handleSubmit}
106   >
107     Sign In
108   </Button>
109 </Paper>
110 </Container>
111 );
112 }
113 }
114 }

```

```

116 function mapStateToProps(state) {
117   return {
118     isLoggingIn: state.auth.isLoggingIn,
119     loginError: state.auth.loginError,
120     isAuthenticated: state.auth.isAuthenticated
121   };
122 }
123
124 export default withStyles(styles)(connect(mapStateToProps)(Login));

```

9. Membuat file `src → components → Home.js`

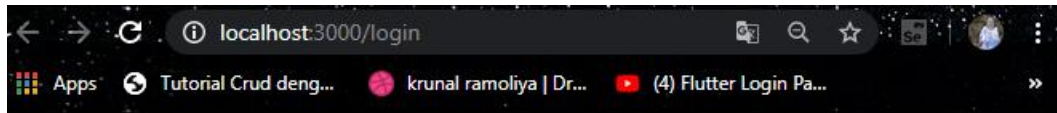
```
login > src > components > JS Home.js > ...
1  import React, { Component } from "react";
2  import { connect } from "react-redux";
3  import { logoutUser } from "../actions/auth"; class Home extends Component {
4    handleLogout = () => {
5      const { dispatch } = this.props;
6      dispatch(logoutUser());
7    };
8    render() {
9      const { isLoggingOut, logoutError } = this.props; return (
10        <div>
11          <h1>This is your app's protected area.</h1>
12          <p>Any routes here will also be protected</p>
13          <button onClick={this.handleLogout}>Logout</button>
14          {isLoggingOut && <p>Logging Out...</p>}
15          {logoutError && <p>Error logging out</p>}
16        </div>
17      );
18    }
19  }function mapStateToProps(state) {
20    return {
21      isLoggingOut: state.auth.isLoggingOut,
22      logoutError: state.auth.logoutError
23    };
24  }export default connect(mapStateToProps)(Home);
25
```

10. Membuat file `src → components → ProtectedRoute.js`

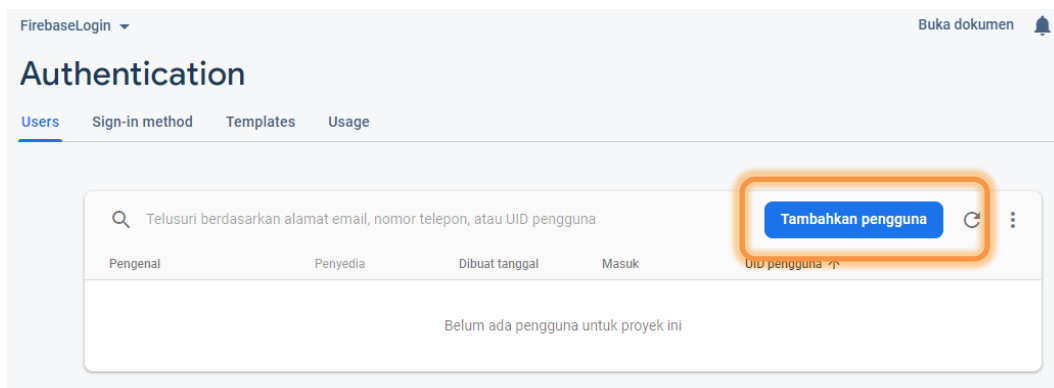
```
login > src > components > JS ProtectedRoute.js > ...
1  import React from "react";
2  import { Route, Redirect } from "react-router-dom";
3  const ProtectedRoute = ({
4    component: Component,
5    isAuthenticated,
6    isVerifying,
7    ...rest
8  }) => (
9    <Route
10     {...rest}
11     render={props => (
12       isVerifying ? (
13         <div />
14       ) : isAuthenticated ? (
15         <Component {...props} />
16       ) : (
17         <Redirect
18           to={{
19             pathname: "/login",
20             state: { from: props.location }
21           }} />
22       )
23     )
24   />
25 );
26 export default ProtectedRoute;
27
28
```



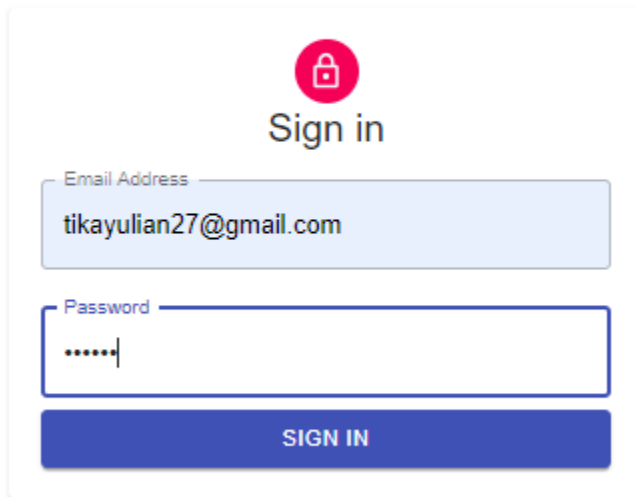
11. Melakukan running, maka pada browser akan muncul seperti berikut:

A 'Sign in' form with a pink lock icon at the top. Below the icon is the text 'Sign in'. There are two input fields: 'Email Address' and 'Password'. At the bottom is a blue button labeled 'SIGN IN'.

12. Masuk ke firebase console kemudian masuk ke menu **Authentication** → **add user**

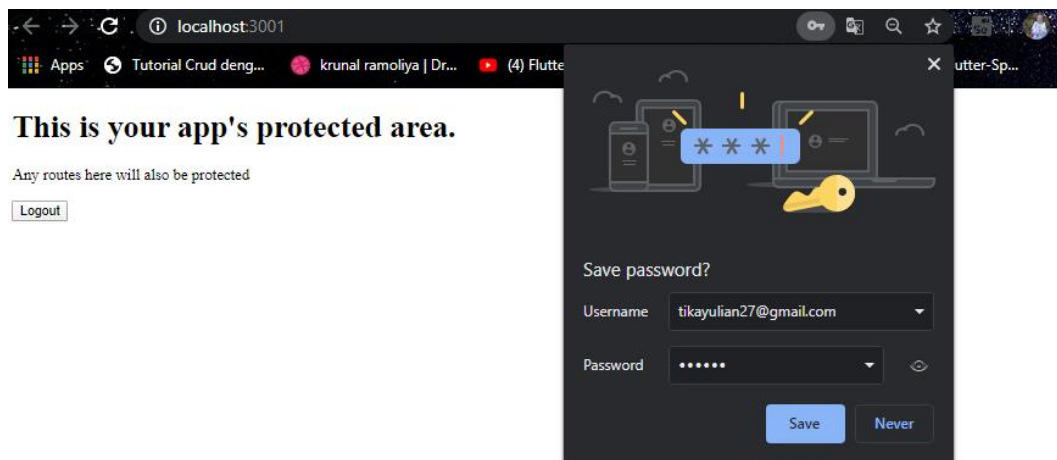


13. Memilih **npm start** kemudian masukkan username dan password



A sign-in form with a red lock icon and the text "Sign in". It contains two input fields: "Email Address" with the value "tikayulian27@gmail.com" and "Password" with masked characters. Below the fields is a blue button labeled "SIGN IN".

14. Lalu masuk ke halaman Beranda



15. Selesai