

TUGAS PERORANGAN/INDIVIDU

Modul 10

FIREBASE REACT.JS

Oleh:

BAHRUDIN CHOLID RIZKY.P. NIM. 1741720191



PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

PRAKTIKUM

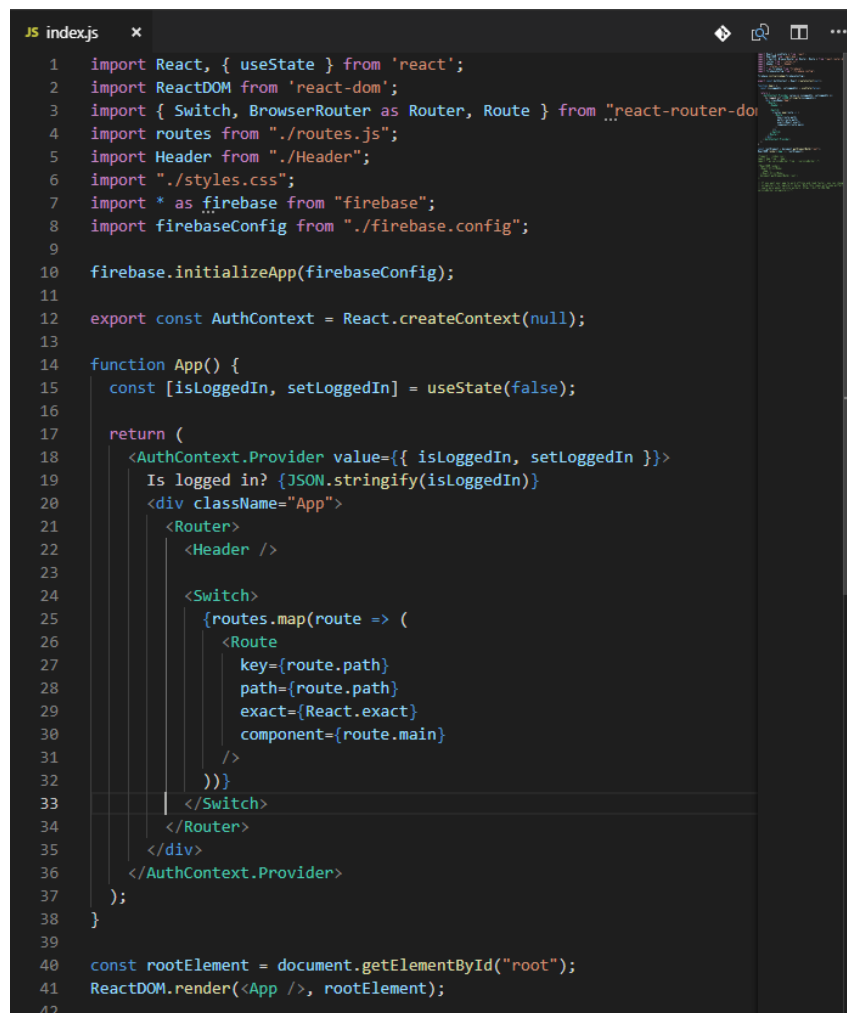
Buatlah project react pada directory yang baru.

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).

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

```
D:\KULIAH\SEMESTER 6\Pemrograman Berbasis Framework\Firebase\FirebaseReactjs\firebase_reactjs>
npm install react-router-dom
[.....] - rollbackFailedOptional: verb npm-session 6623d0e2f5fe0838
```

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



```
JS index.js x
1  import React, { useState } 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  import * as firebase from 'firebase';
8  import firebaseConfig from './firebase.config';
9
10 firebase.initializeApp(firebaseConfig);
11
12 export const AuthContext = React.createContext(null);
13
14 function App() {
15   const [isLoggedIn, setLoggedIn] = useState(false);
16
17   return (
18     <AuthContext.Provider value={{ isLoggedIn, setLoggedIn }}>
19       Is logged in? {JSON.stringify(isLoggedIn)}
20       <div className="App">
21         <Router>
22           <Header />
23
24           <Switch>
25             {routes.map(route => (
26               <Route
27                 key={route.path}
28                 path={route.path}
29                 exact={React.exact}
30                 component={route.main}
31               />
32             ))}
33           </Switch>
34         </Router>
35       </div>
36     </AuthContext.Provider>
37   );
38 }
39
40 const rootElement = document.getElementById("root");
41 ReactDOM.render(<App />, rootElement);
42
```

Login.js

```
JS index.js JS Login.js x
1 import React, { useState, useContext } from "react";
2 import { AuthContext } from "../index";
3 import * as firebase from "firebase";
4
5 var provider = new firebase.auth.GoogleAuthProvider();
6
7 const Login = () => {
8   const [email, setEmail] = useState("");
9   const [password, setPassword] = useState("");
10  const [error, setErrors] = useState("");
11
12  const Auth = useContext(AuthContext);
13  const handleForm = e => {
14    /*e.preventDefault();
15    console.log(Auth);
16    Auth.setLoggedIn(true);*/
17    e.preventDefault();
18    firebase
19      .auth()
20      .signInWithEmailAndPassword(email, password)
21      .then(res => {
22        if (res.user) Auth.setLoggedIn(true);
23      })
24      .catch(e => {
25        setErrors(e.message);
26      });
27  };
28
29  const loginWithGoogle = e => {
30    e.preventDefault();
31    firebase
32      .auth()
33      .signInWithPopup(provider)
34      .then(res => {
35        if (res.user) Auth.setLoggedIn(true);
36      })
37      .catch(e => {
38        setErrors(e.message);
39      })
40  }
41}
```

Join.js

```
JS index.js  JS Login.js  JS Join.js  x
1  import React, { useState, useContext } from "react";
2  import { AuthContext } from "../index";
3  import * as firebase from "firebase";
4
5  var provider = new firebase.auth.GoogleAuthProvider();
6
7  const Join = (method) firebase.auth.Auth.createUserWithEmailAndPassword(email: string, password: string): Promise<firebase.auth.UserCredential>
8  const [email, setEmail] = useState('');
9  const [password, setPassword] = useState('');
10  const [errors, setErrors] = useState('');
11
12  const AuthContext = useContext(AuthContext);
13  const handleJoin = (e) => {
14    e.preventDefault();
15    console.log(email, password);
16    AuthContext.firebase.auth().createUserWithEmailAndPassword(email, password)
17    .then(res => {
18      if (res.user) AuthContext.setLoggedIn(true);
19    })
20    .catch(e => {
21      setErrors(e.message);
22    });
23  };
24
25  const joinWithGoogle = e => {
26    e.preventDefault();
27    firebase.auth().signInWithPopup(provider)
28    .then(res => {
29      if (res.user) AuthContext.setLoggedIn(true);
30    })
31    .catch(e => {
32      setErrors(e.message);
33    });
34  };
35
36  export default Join;
37
38  }
39
40  }
41
```

Header.js

```
JS index.js JS Login.js JS Join.js JS Header.js x
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;
```

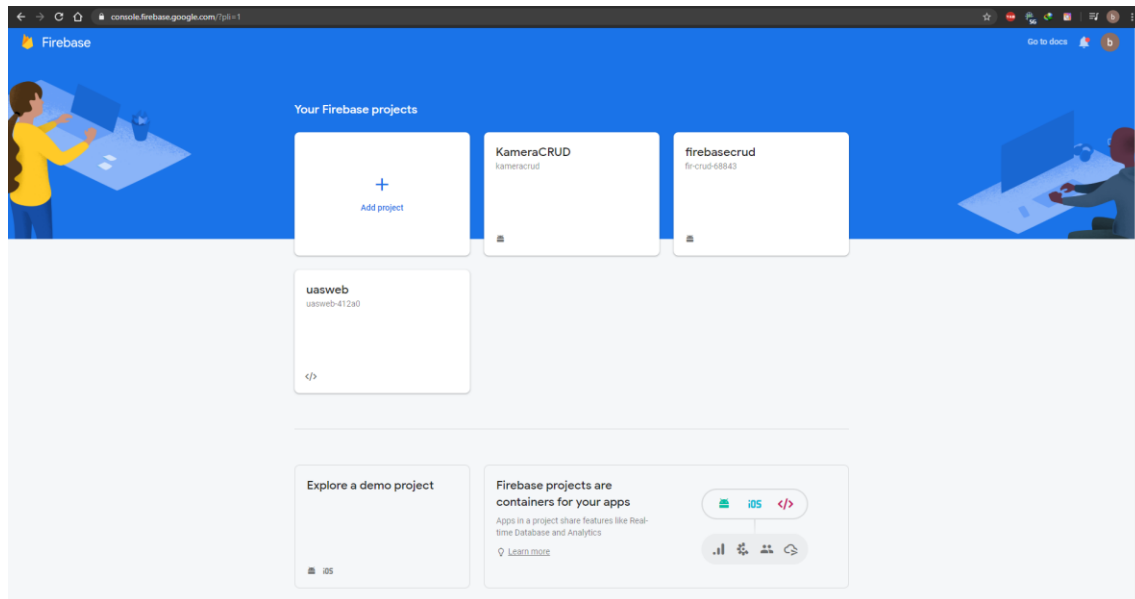
routes.js

```
JS index.js JS Login.js JS Join.js JS Header.js JS routes.js x
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 Login.js JS Join.js JS Header.js # styles.css x
1  .App {
2    font-family: sans-serif;
3    padding: 50px;
4  }
5
6  h1 {
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    box-shadow: 0 1px 3px 0 rgba(0, 0, 0, 0.12), 0 1px 2px 0 rgba(0, 0, 0, 0.23);
18    transition: all 0.3s cubic-bezier(0.25, 0.8, 0.25, 1);
19    padding: 10px;
20  }
21
22  button {
23    width: 125px;
24    box-shadow: 0 1px 3px 0 rgba(0, 0, 0, 0.12), 0 1px 2px 0 rgba(0, 0, 0, 0.23);
25    border: none;
26    padding: 7px;
27    text-align: center;
28    cursor: pointer;
29  }
30
31  button[type="submit"] {
32    background-color: #4c566a;
33    color: white;
34    font-size: 14px;
35    font-weight: 900;
36    border-radius: 5px;
37    text-transform: uppercase;
38    padding: 15px;
39  }
40
41  }
```

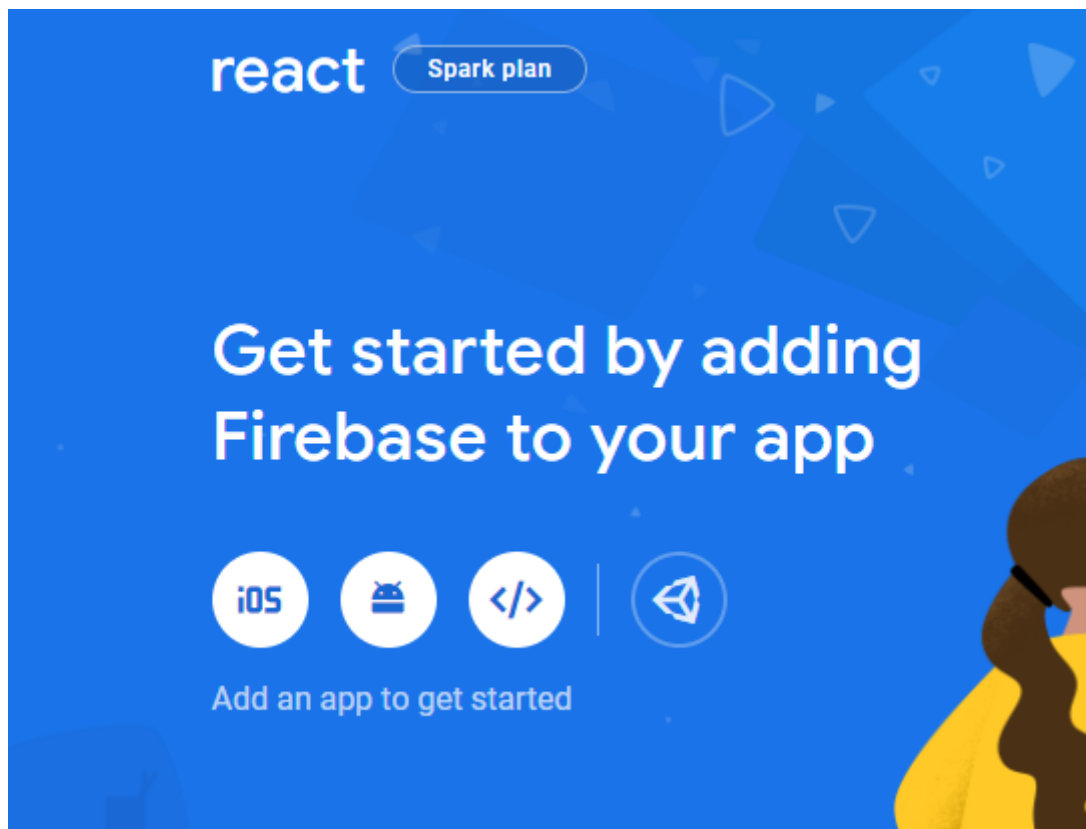
Kemudian buka <https://console.firebase.google.com/> setelah itu buat project baru dengan nama project yang kalian inginkan.



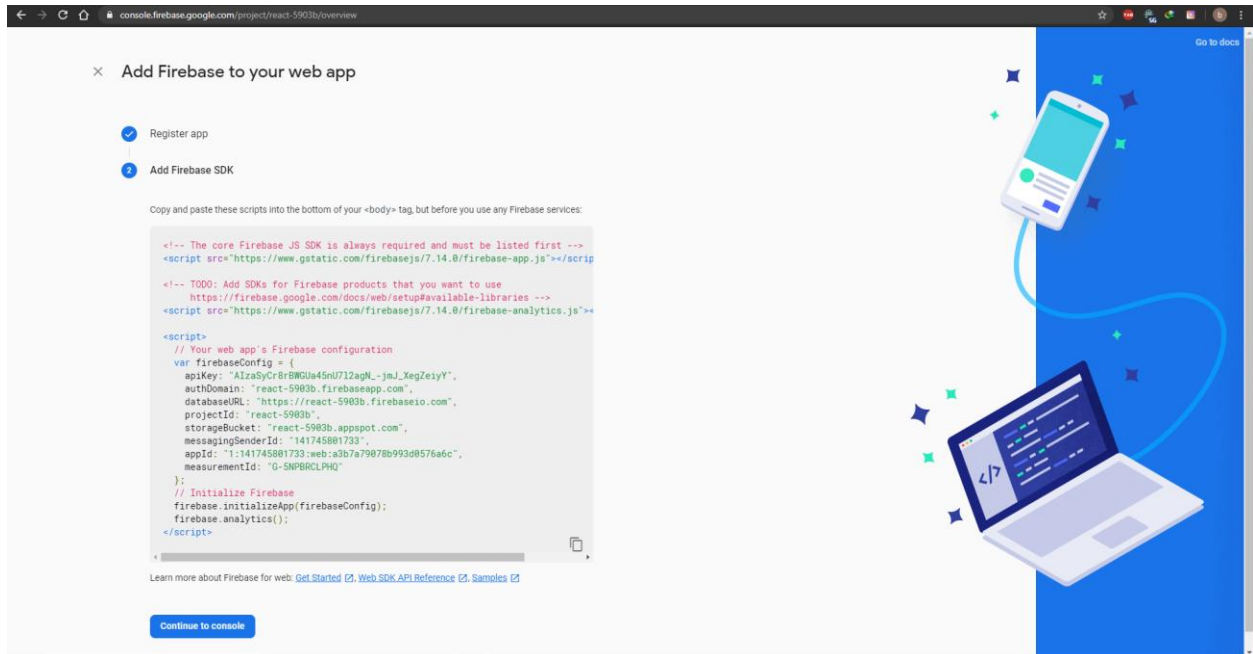
Pada cmd lakukan perintah ***npm install --save firebase*** di directory project react untuk menambahkan dependency firebase kedalam react kita.

```
D:\KULIAH\SEMESTER 6\Pemrograman Berbasis Framework\Firebase\FirebaseReactjs\firebase_reactjs>
npm install --save firebase
[.....] / refresh-package-json:firebase: timing action:finalize Completed in 128
```

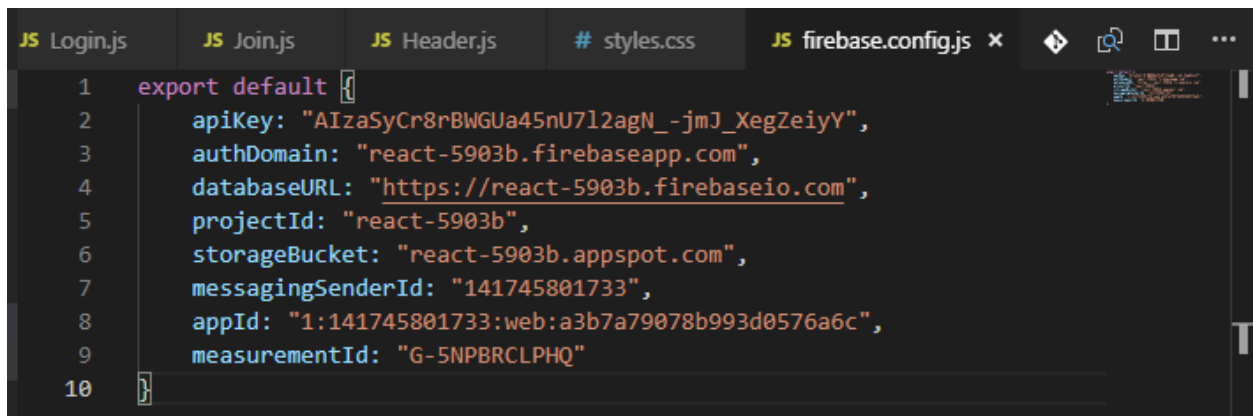
Setelah itu lakukan setting pada project firebase yang kalian buat tadi agar react kalian dapat terhubung dengan firebase. Pada setting get started pilih yang untuk web.



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).



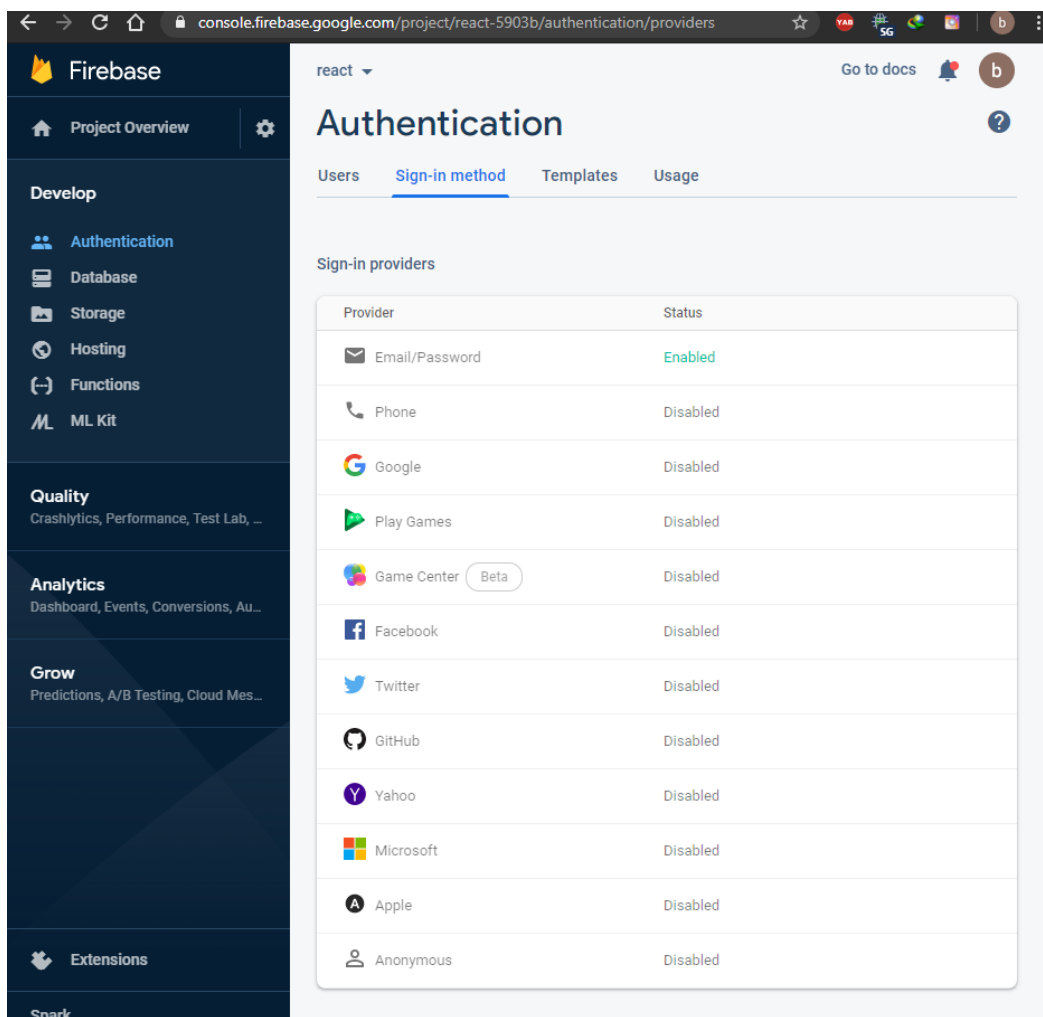
firebase.config.js



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

```
JS index.js x JS Login.js JS Join.js JS Header.js # styles.css JS fi
1 import React, { useState } 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 import * as firebase from 'firebase';
8 import firebaseConfig from './firebase.config';
9
```

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



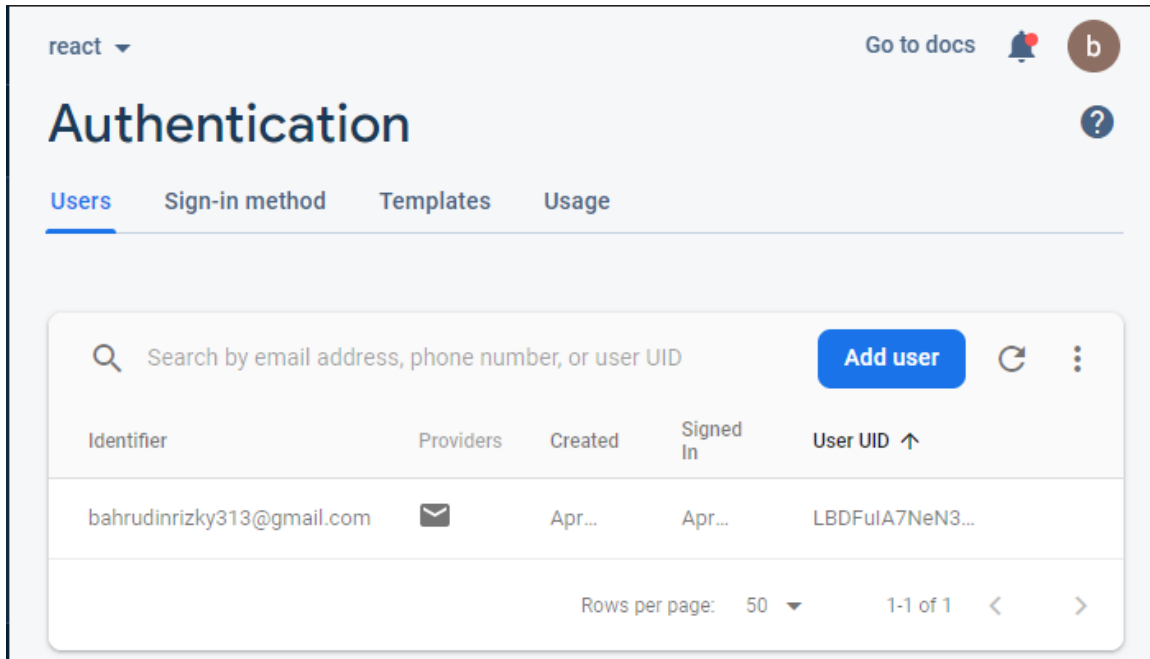
Jika sudah, lakukan edit code pada **Join.js** untuk membuat email dan password untuk user. Pertama import dahulu objek firebase agar bisa digunakan pada **Join.js** dengan code berikut.

```
JS index.js JS Login.js JS Join.js x JS Header.js # styles.css JS fi
1 import React, { useState, useContext } from "react";
2 import { AuthContext } from "../index";
3 import * as firebase from "firebase";
4
```

Kemudian cari method auth dan lakukan edit code seperti dibawah ini.

```
11
12     const Auth = useContext(AuthContext);
13     const handleForm = e => {
14         /*e.preventDefault();
15         console.log(Auth);
16         Auth.setLoggedIn(true);*/
17         e.preventDefault();
18         firebase
19             .auth()
20             .createUserWithEmailAndPassword(email, password)
21             .then(res => {
22                 if (res.user) Auth.setLoggedIn(true);
23             })
24             .catch(e => {
25                 setErrors(e.message);
26             });
27     };
28
```

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



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.

```
JS index.js JS Login.js x JS Join.js JS Header.js # styles.css JS fi
1 import React, { useState, useContext } from "react";
2 import { AuthContext } from "../index";
3 import * as firebase from "firebase";
4
```

Kemudian cari method auth dan lakukan edit code seperti dibawah ini.

```
const Login = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [error, setErrors] = useState("");


  const Auth = useContext(AuthContext);
  const handleForm = e => {
    /*e.preventDefault();
    console.log(Auth);
    Auth.setLoggedIn(true);*/
    e.preventDefault();
    firebase
      .auth()
      .signInWithEmailAndPassword(email, password)
      .then(res => {
        if (res.user) Auth.setLoggedIn(true);
      })
      .catch(e => {
        setErrors(e.message);
      });
  };
};
```

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.

Is logged in? true

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

Join

 Join With Google

TUGAS

Buatlah laporan dengan menyertakan screenshot hasil dan code yang telah dibuat.

Pada button join with google belum terdapat aksi, silahkan tambahkan function agar dapat mendaftar dengan cara klik button tersebut lalu memilih akun gmail yang akan dipakai.


Is logged in? true

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

Join

bahrudinrizky313@gmail.com

.....

 Join With Google

LOGIN

The popup has been closed by the user before

