

LAPORAN PEMROGRAMAN FRAMEWORK

“SESSION 9 GLOBAL API”



Oleh:

Nama : Bagus Satria Putra

Kelas : 3F

Absen/NIM : 8 / 1841720146

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

22 APRIL 2021

Link Github :

<https://github.com/BagusSatria123/PemrogramanFrameworkReact/tree/master/Pertemuan10/Script2>

Link Youtube : <https://youtu.be/Jv1V5Wb6Sdo>

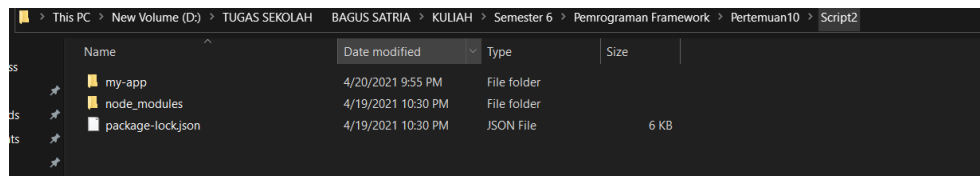
NO.

KETERANGAN

1.1 Langkah Praktikum

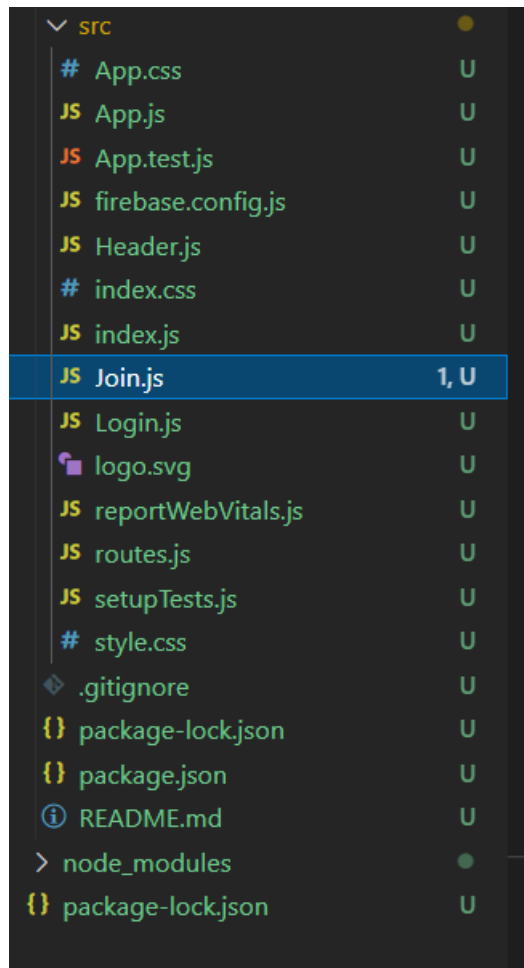
- 1.** Buatlah project react pada directory yang baru

Jawaban :



- 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).

Jawaban :



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

install react-router-dom dahulu.

Jawaban :

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
rograman Framework\Pertemuan10\Script2\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 'D:\TUGAS SEKOLAH BAGUS SATRIA\KU
LIAH\Semester 6\Pemrograman Framework\Pertemuan10\Script2\package.json'
npm WARN react-router-dom@5.2.0 requires a peer of react@>=15 but none is installed. You must
install peer dependencies yourself.
npm WARN react-router@5.2.0 requires a peer of react@>=15 but none is installed. You must inst
all peer dependencies yourself.
npm WARN mini-create-react-context@0.4.1 requires a peer of react@^0.14.0 || ^15.0.0 || ^16.0.
0 || ^17.0.0 but none is installed. You must install peer dependencies yourself.
npm WARN Script2 No description
npm WARN Script2 No repository field.
npm WARN Script2 No license field.

+ react-router-dom@5.2.0
added 18 packages from 11 contributors and audited 18 packages in 4.958s
found 0 vulnerabilities

PS D:\TUGAS SEKOLAH BAGUS SATRIA\KULIAH\Semester 6\Pemrograman Framework\Pertemuan10\Script2>
```

4. Jika sudah, isilah code seperti dibawah ini sesuai dengan nama file masingmasing.

Jawaban :

index.js

```
JS firebase.config.js U JS Login.js U # style.css U JS index.js 1, U X JS Join.js U JS routes.js U JS Header.js U
my-app > src > JS index.js > App > routes.map() callback
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';
5 import Header from './Header';
6 import './style.css';
7 import 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           <Switch>
24             {routes.map(route => (
25               <Route
26                 key={route.path}
27                 path={route.path}
28                 exact={route.exact}
29                 component={route.main}
30               />
31             ))}
32           </Switch>
25         </Router>
26       </div>
27     </AuthContext.Provider>
28   );
29 }
```

Join.js

```
JS firebase.config.js U JS Login.js U # style.css U JS index.js U JS Join.js 1, U X JS routes.js U JS Header.js U
my-app > src > JS Join.js > [0] Join > [0] handleForm
40   .then((res) => {
17       if(res.user) Auth.setLoggedIn(true)
18   })
19   .catch(e => {
20       setErrors(e.message);
21   })
22   })
23
24   const onJoin = () => {
25       var provider = new firebase.auth.GoogleAuthProvider();
26       firebase.auth()
27         .signInWithPopup(provider)
28         .then((result) => {
29             /** @type {firebase.auth.OAuthCredential} */
30             var credential = result.credential;
31
32             // This gives you a Google Access Token. You can use it to access the Google API.
33             var token = credential.accessToken;
34             // The signed-in user info.
35             var user = result.user;
36             if(result.user) Auth.setLoggedIn(true);
37         }).catch((error) => {
38             console.log(error)
39         });
40     }
41
42     return (
43         <div>
44             <h1>Join</h1>
45             <form onSubmit={e => handleForm(e)}>
46                 <input
47                     value={email}
```

Login.js

```
View Go Run Terminal Help Loginjs - Script2 - Visual Studio Code
... JS firebase.config.js U JS Login.js 1, U X # style.css U JS index.js U JS Join.js U JS routes.js U JS Header.js U
my-app > src > JS Login.js > [0] Login
1   import React, {useState, useContext} from 'react';
2   import {AuthContext} from './index';
3   import firebase from 'firebase';
4
5
6   const Login = () => {
7       const [email, setEmail] = useState("");
8       const [password, setPassword] = useState("");
9       const [error, setErrors] = useState("");
10
11       const Auth = useContext(AuthContext);
12       const handleForm = e => {
13         e.preventDefault();
14         firebase
15           .auth()
16           .signInWithEmailAndPassword(email, password)
17           .then(res => {
18             if(res.user) Auth.setLoggedIn(true)
19           })
20           .catch(e => {
21             setErrors(e.message);
22           })
23       }
24
25       return (
26         <div>
27             <h1>Login</h1>
28             <form onSubmit={e => handleForm(e)}>
29                 <input
30                     value={email}
31                     onChange={e => setEmail(e.target.value)}
32                 />
33             </div>
34         )
35     }
36
37     export default Login;
```

Header.js

```
Go Run Terminal Help
Header.js - Script2 - Visual Studio Code

my-app > src > JS Header.js > routes.map() callback
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 export default Header;
```

Routes.js

```
routes.js - Script2 - Visual Studio Code

my-app > src > JS routes.js > routes
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 export default routes;
```

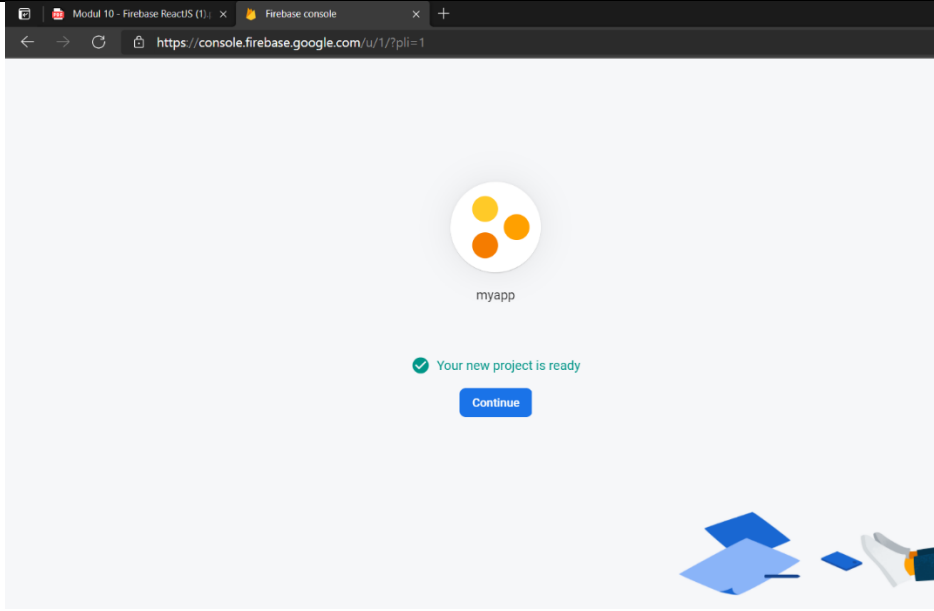
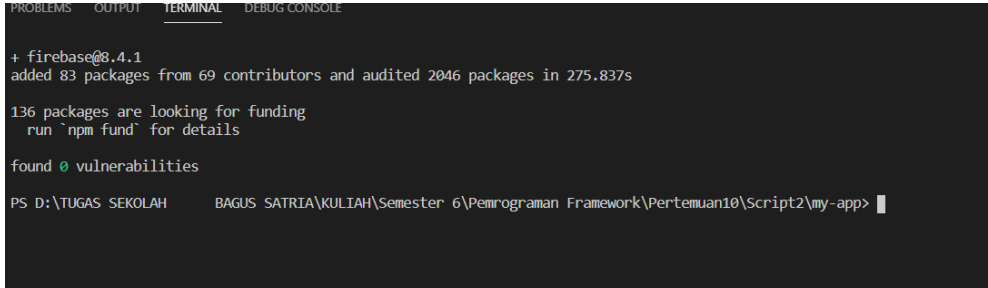
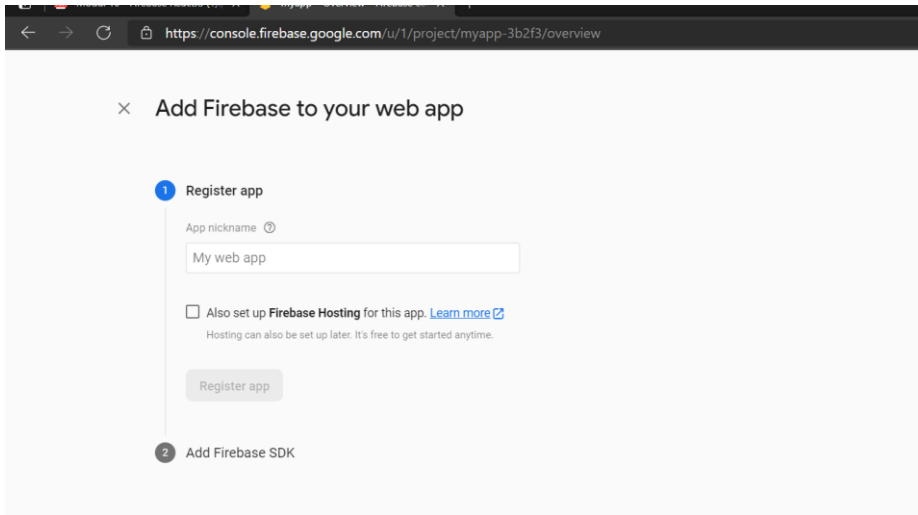
Style.css

```
Go Run Terminal Help
style.css - Script2 - Visual Studio Code

my-app > src > # style.css > .nav li
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   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   margin: 0 0 25px 0;
31   padding: 7px;
```

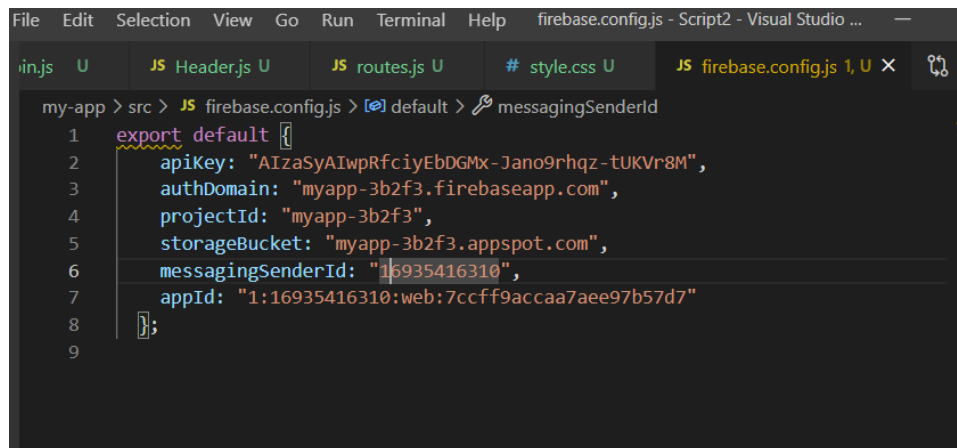
5. Kemudian buka <https://console.firebase.google.com/> setelah itu buat project baru dengan nama project yang kalian inginkan

Jawaban :

	
<p>6.</p>	<p>Pada cmd lakukan perintah npm install --save firebase di directory project react untuk menambahkan dependency firebase kedalam react kita.</p> <p>Jawaban ;</p> 
<p>7.</p>	<p>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</p> <p>Jawaban ;</p> 

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).

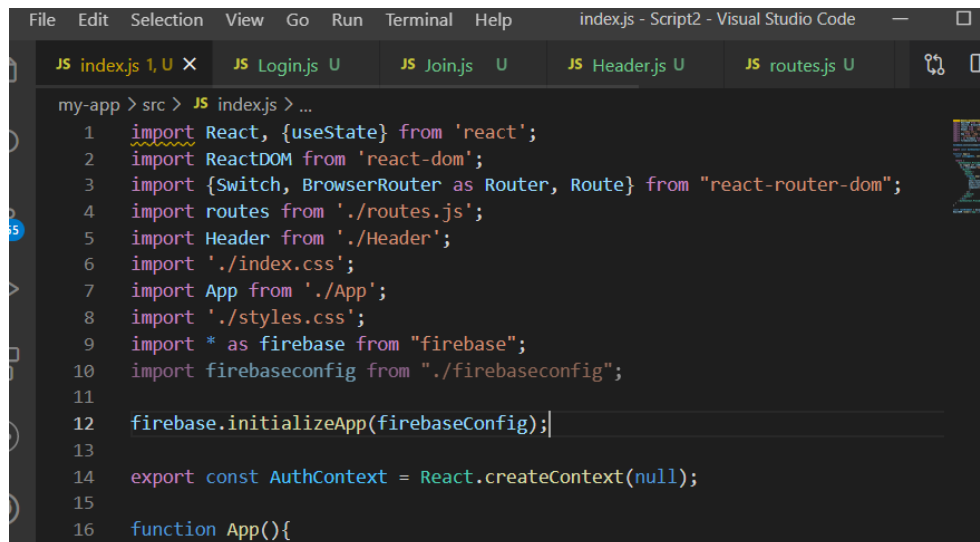
Jawaban :



```
File Edit Selection View Go Run Terminal Help firebase.config.js - Script2 - Visual Studio ...
in.js U JS Header.js U JS routes.js U # style.css U JS firebase.config.js 1, U X
my-app > src > JS firebase.config.js > default > messagingSenderId
1 export default {
2   apiKey: "AIzaSyAIwPRfcyEbdGMx-Jano9rhqz-tUKVr8M",
3   authDomain: "myapp-3b2f3.firebaseio.com",
4   projectId: "myapp-3b2f3",
5   storageBucket: "myapp-3b2f3.appspot.com",
6   messagingSenderId: "16935416310",
7   appId: "1:16935416310:web:7ccff9accaa7aee97b57d7"
8 };
9
```

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

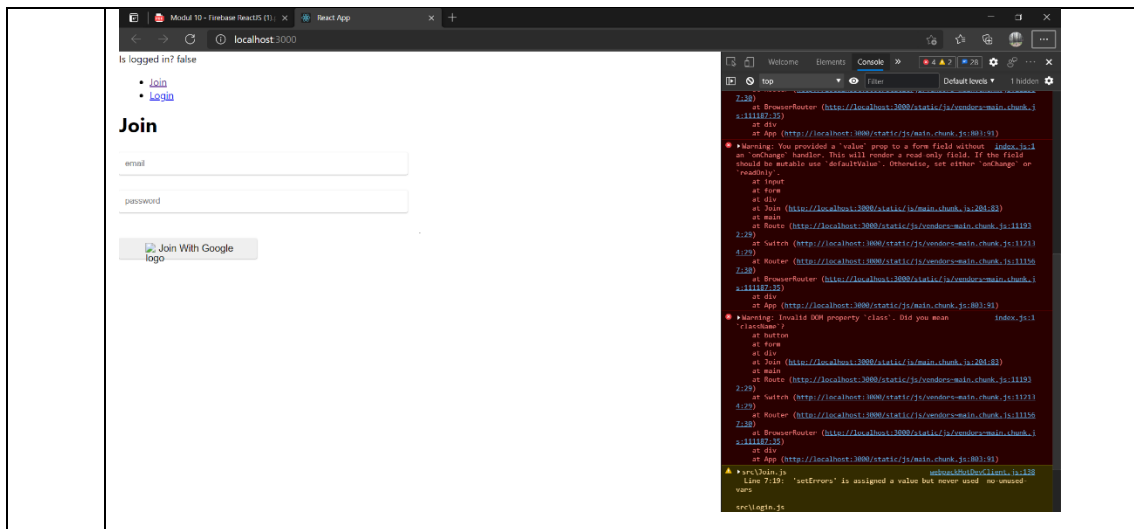
Jawaban :



```
File Edit Selection View Go Run Terminal Help index.js - Script2 - Visual Studio Code
JS index.js 1, U X JS Login.js U JS Join.js U JS Header.js U JS routes.js U
my-app > src > JS index.js > ...
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 './index.css';
7 import App from './App';
8 import './styles.css';
9 import * as firebase from "firebase";
10 import firebaseconfig from "./firebaseconfig";
11
12 firebase.initializeApp(firebaseConfig);
13
14 export const AuthContext = React.createContext(null);
15
16 function App(){
```

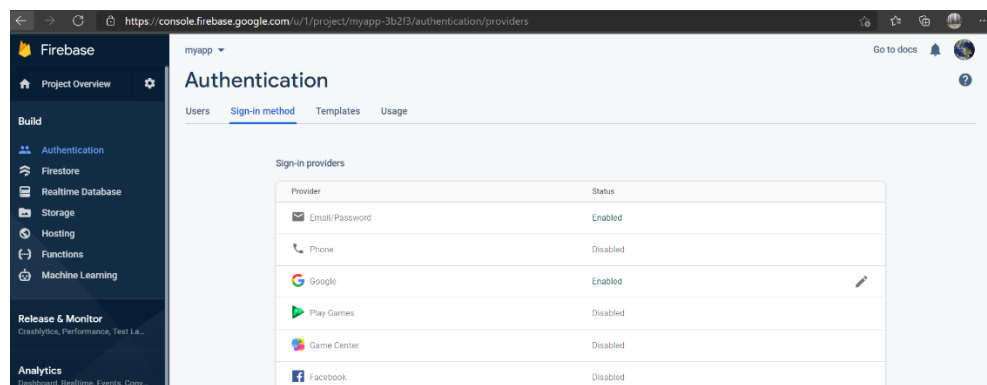
10. Jika sudah berhasil maka coba jalankan, jika tidak terdapat error maka project anda telah terhubung dengan firebase

Jawaban :



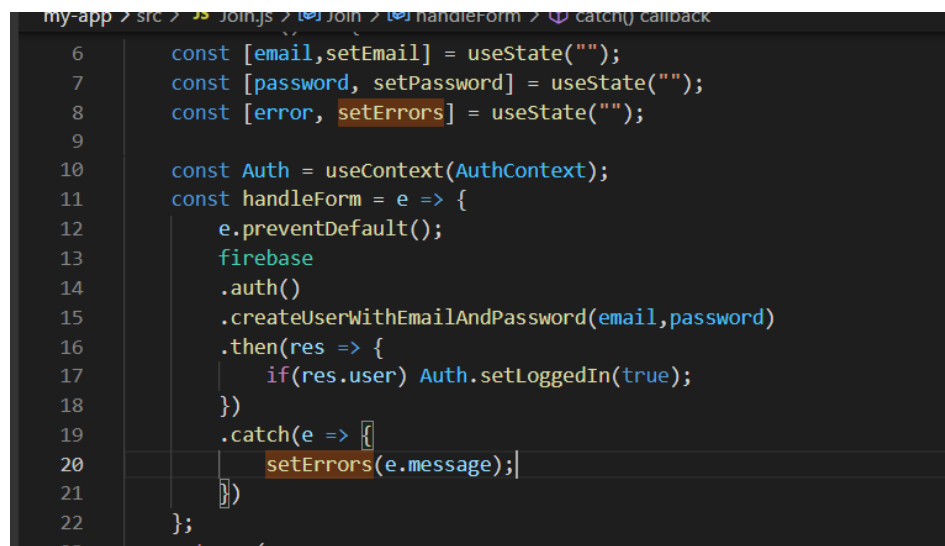
11. Selanjutnya aktifkan terlebih dahulu authentication email dan google pada firebase console.

Jawaban :



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

Jawaban :



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

Jawaban :


Is logged in? true

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

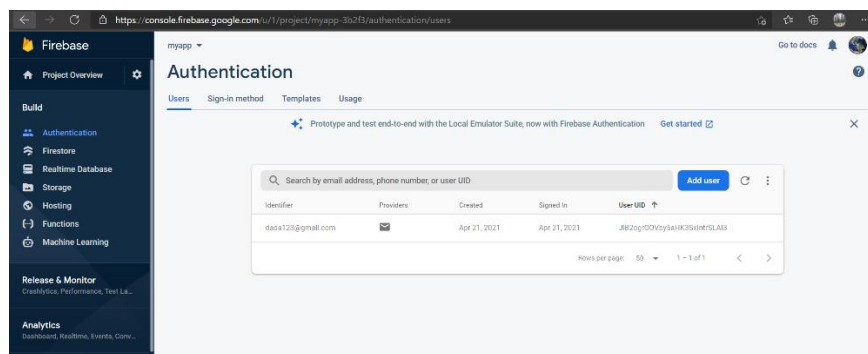
Join

dada123@gmail.com

.....

 Join with Google

JOIN



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.

Jawaban :

```
JS index.js U JS firebase.config.js U JS Login.js 1, U X # style.css U JS Join.js U JS Header.js U JS routes.js U
my-app > src > JS Login.js > Login

1 import React, {useState, useContext} from 'react';
2 import {AuthContext} from "../index";
3 import firebase from "firebase";
4
5
6 const Login = () => {
7   const [email, setEmail] = useState("");
8   const [password, setPassword] = useState("");
9   const [error, setErrors] = useState("");
10
11   const Auth = useContext(AuthContext);
12   const handleForm = e => {
13     e.preventDefault();
14     firebase
15       .auth()
16       .signInWithEmailAndPassword(email, password)
17       .then(res => {
18         if(res.user) Auth.setLoggedIn(true)
19       })
20       .catch(e => {
21         setErrors(e.message);
22       })
23   }
24 }
```


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

Jawaban :

Is logged in? false

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

Login

 Login With Google

LOGIN


Is logged in? true

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

Login

dada123@gmail.com

.....

 Login With Google

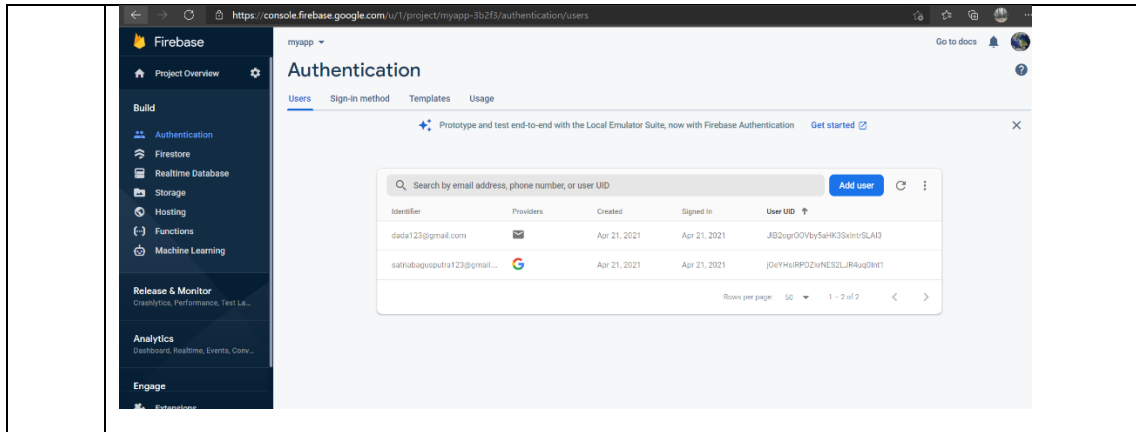
LOGIN

1.2 Soal Latihan Praktikum

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

Jawaban :

```
JS firebase.config.js U JS Login.js U # style.css U JS index.js U JS Join.js 1, U X JS routes.js U JS Header.js U
my-app > src > JS Join.js > [0] Join > [0] handleForm
16 .then(res => {
17   if(res.user) Auth.setLoggedIn(true)
18 })
19 .catch(e => {
20   setErrors(e.message);
21 })
22
23
24 const onJoin = () => {
25   var provider = new firebase.auth.GoogleAuthProvider();
26   firebase.auth()
27     .signInWithPopup(provider)
28     .then((result) => {
29       /** @type {firebase.auth.OAuthCredential} */
30       var credential = result.credential;
31
32       // This gives you a Google Access Token. You can use it to access the Google API.
33       var token = credential.accessToken;
34       // The signed-in user info.
35       var user = result.user;
36       if(result.user) Auth.setLoggedIn(true);
37     }).catch((error) => {
38       console.log(error)
39     });
40   }
41 }
```



--Alhamdulillah--