LAPORAN PRAKTIKUM
MATA KULIAH PEMEROGRAMAM BERBASIS FRAMEWORK

Nama : Alenovan Wiradhika Putra

Kelas : TI-3C D-IV T.INFORMATIKA

NIM : 1741720065

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

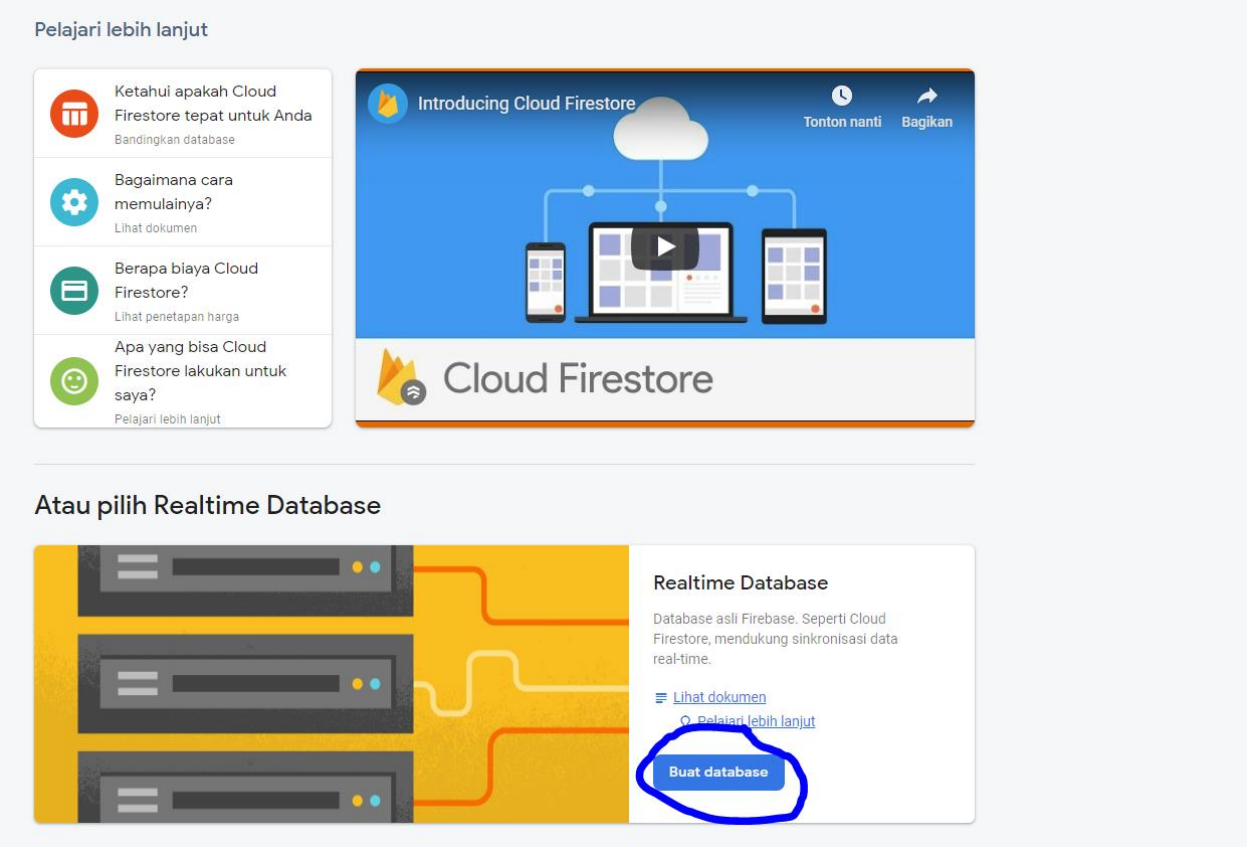
POLITEKNIK NEGERI MALANG

2020

1.Setting Fire Base

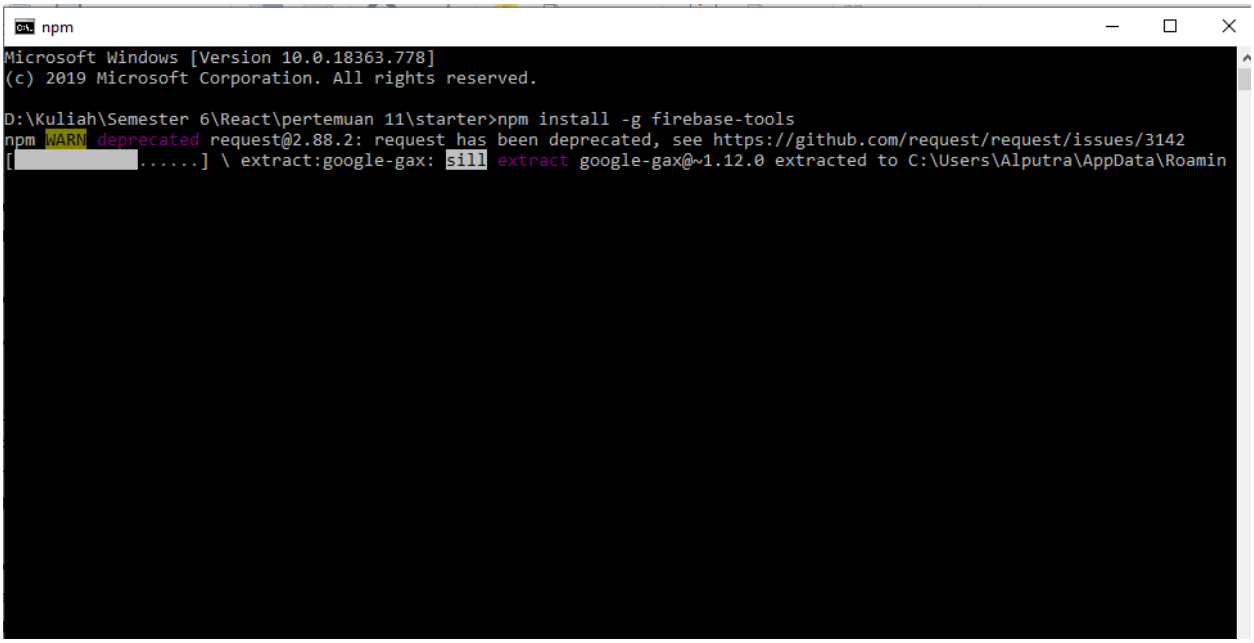
Login ke firebase console dan add a new project. Pilih authentification dan pilih tab menu Set up sign in method

Setelah enabling athentification pilih menu Database dan pilih create database

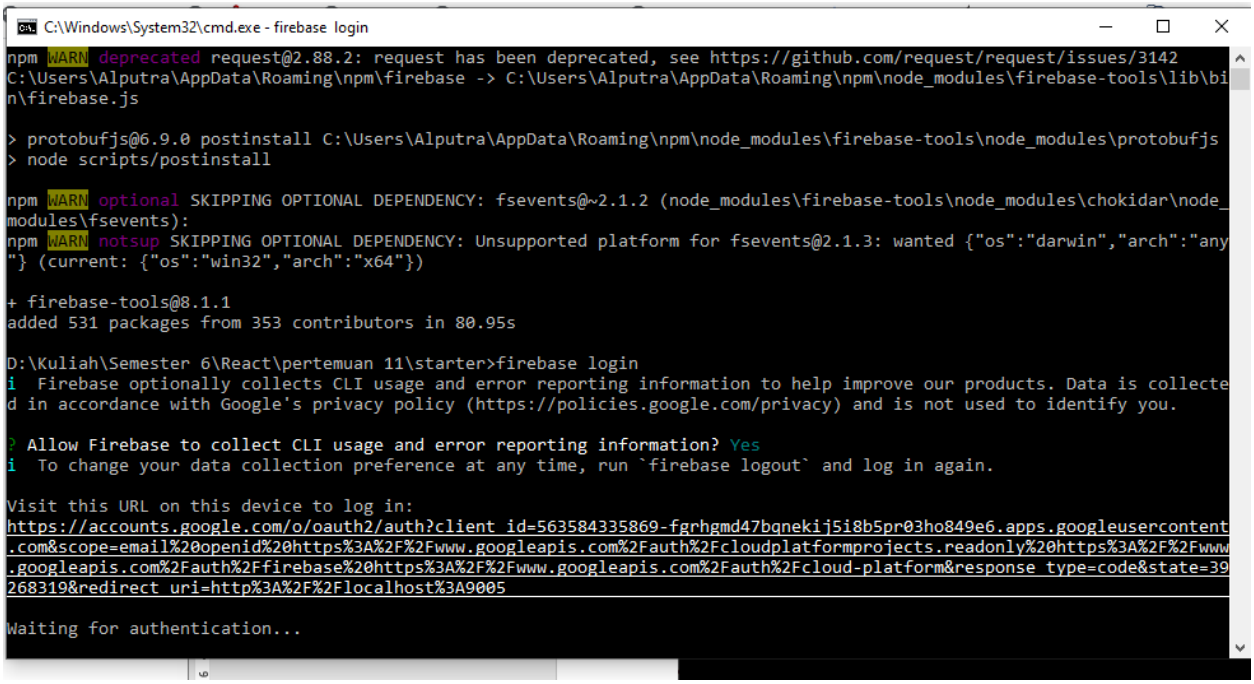


Buat Data Base dalam metode pengujian

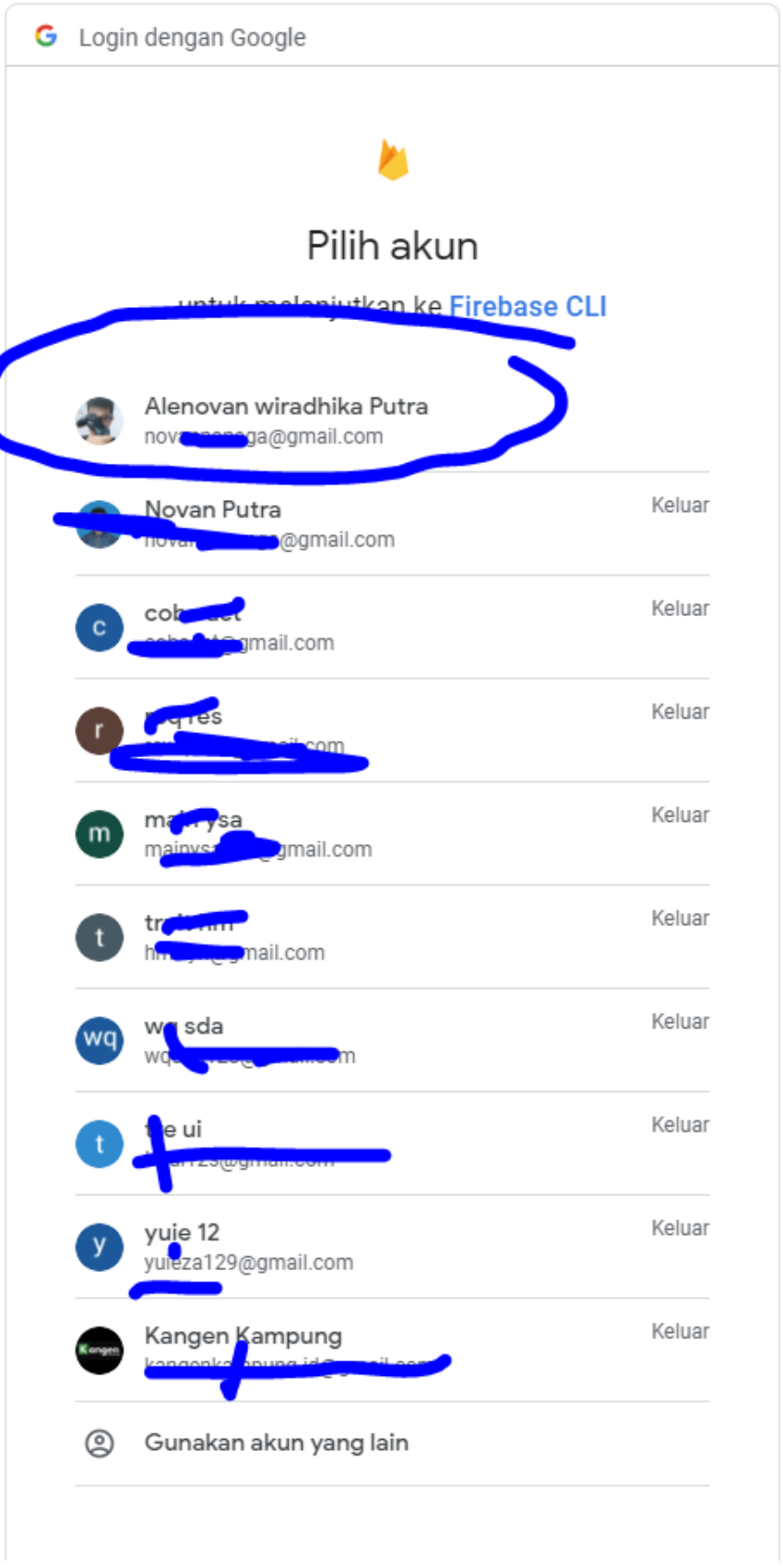
Saya menggunakan Tamplate Project kosongan yang sudah saya buat, saya memualai dengan menginstall firebase-tools di project react saya



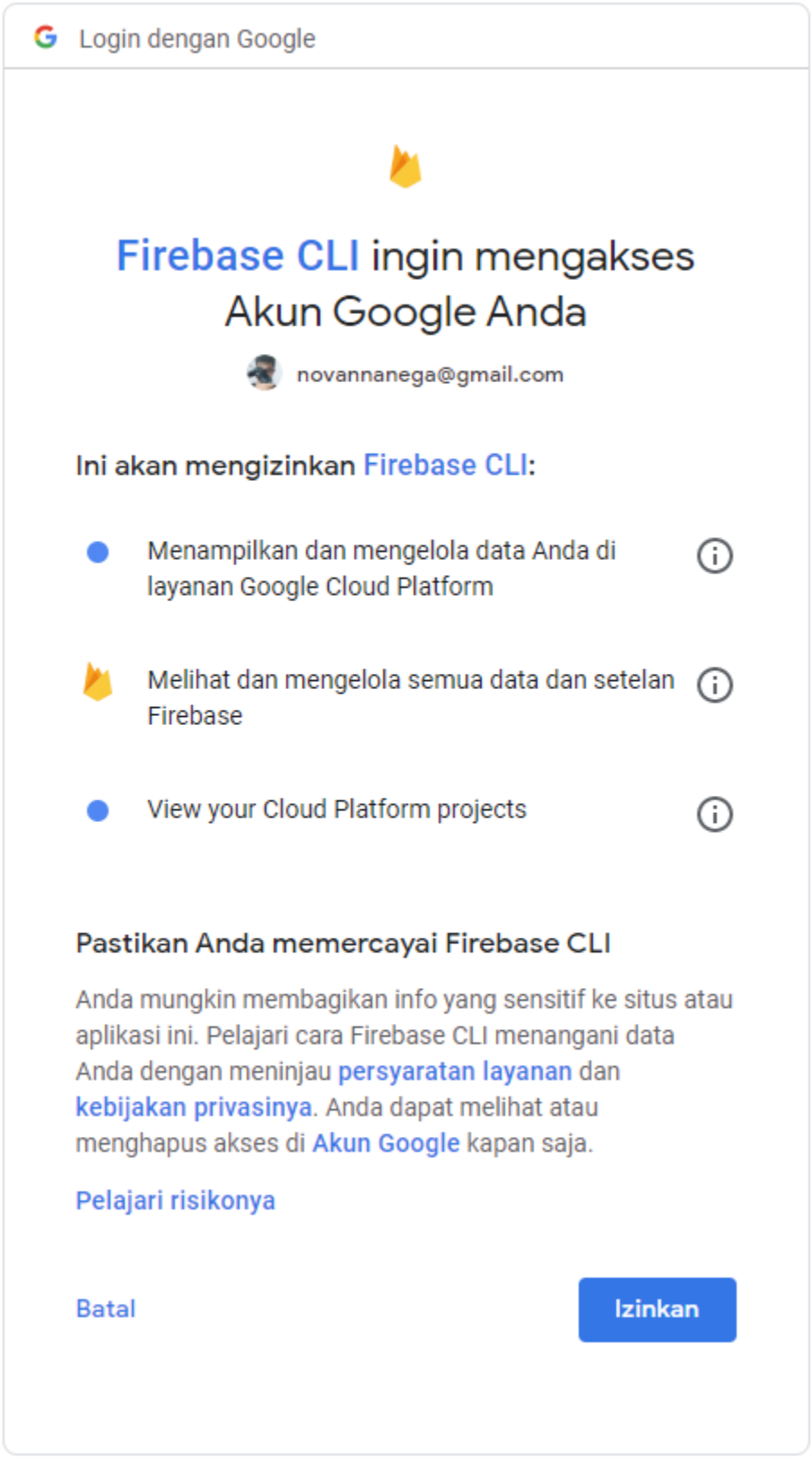
Kemudian kita login ke firebase kita menggunakan akun kita masing-masing



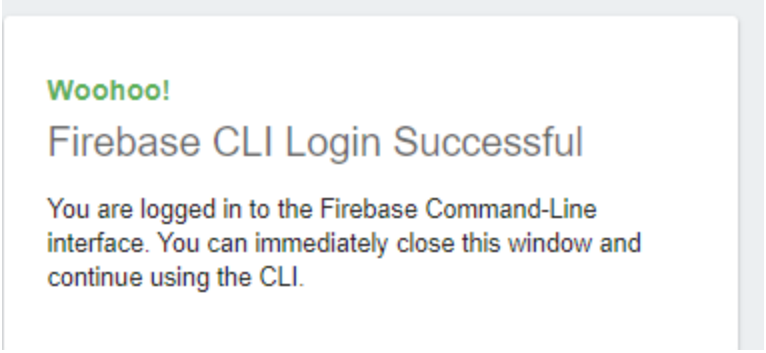
Browser akan terbuka dan kita akan di suruh memilih kaun mana yang akan kita gunakan


Login dengan Google
Pilih akun
untuk melanjutkan ke Firebase CLI
Alenovan wiradhika Putra
Novan Putra
Keluar
Keluar
Keluar
Keluar
Keluar
Keluar
Keluar
yuie 12
Keluar
Kangen Kampung
Keluar
Gunakan akun yang lain

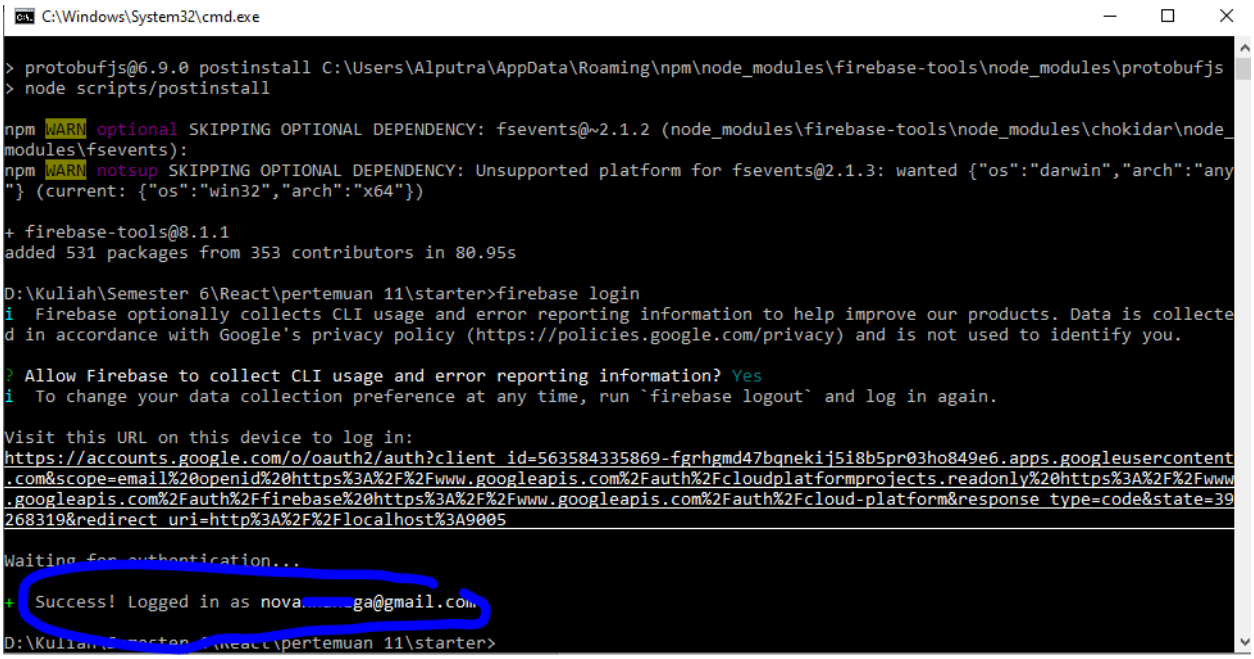
Dan jangan lupa di “allow” firebasenya



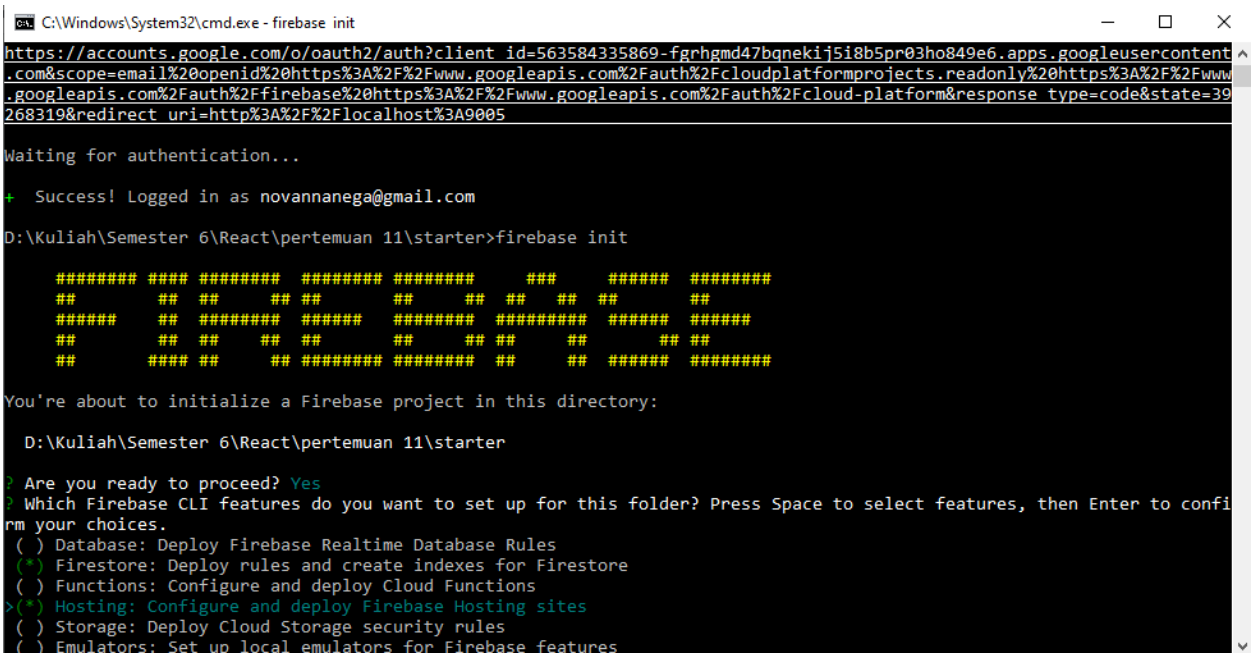
Apablia berhasil mentyambungkan aka nada tampilan seperti ini yang menandakan project anda telah tersambung



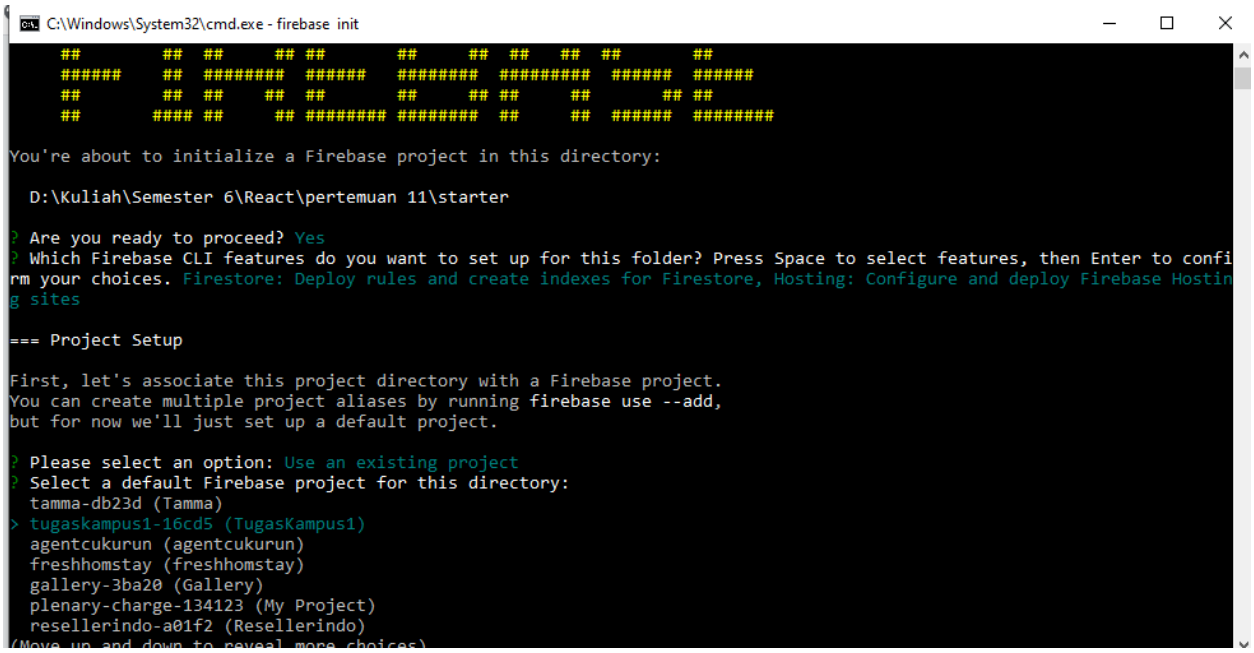
Dan kita cek juga di bagian terminal akan terdapat tampilan seperti ini



Dan jangan lupa firebase init dan setting firebasenya



Firebase di sambungkan dengan project yang sudah kita buat di awal” tadi , disini saya sambaungkan dengan tugaskampus1

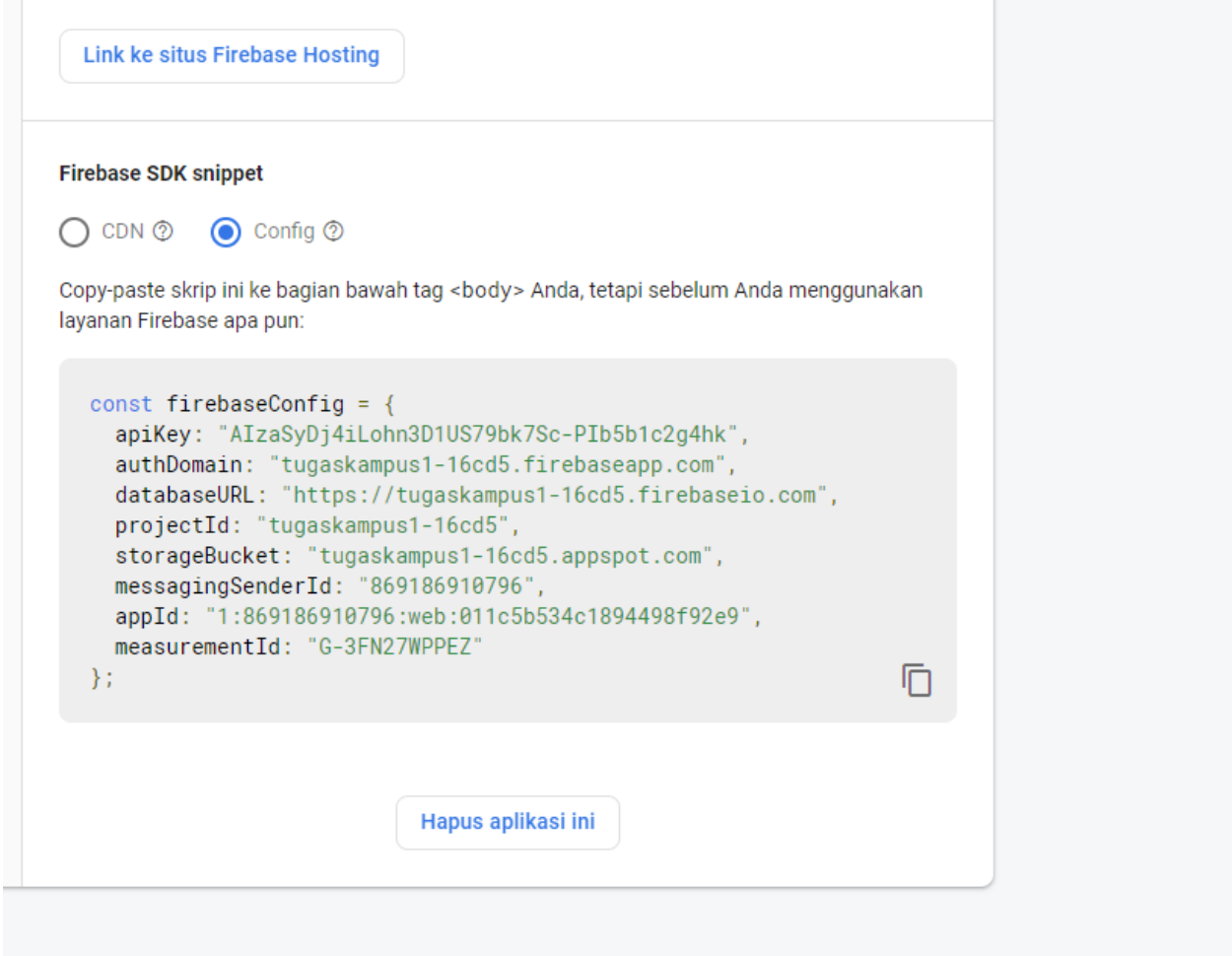


Npm Run Build dan firebase server untuk membangun firebase

```
D:\Kuliah\Semester 6\React\pertemuan 11\starter>npm run build

> starter@0.1.0 build D:\Kuliah\Semester 6\React\pertemuan 11\starter
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

  39.84 KB  build\static\js\2.b1620f7f.chunk.js
  773 B     build\static\js\runtime-main.eaf72a32.js
  631 B     build\static\js\main.44db3576.chunk.js
  556 B     build\static\css\main.d1b05096.chunk.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.
```

Kemudian Buat firebase/firebase.js

Ambil firebase config anda sendiri” di firebase console anda



Dan berikan import firebase , dan jangan lupa export

```
import firebase from "firebase/app";
import "firebase/auth";
import "firebase/firestore";
const firebaseConfig = {
  apiKey: "AIzaSyDj4iLohn3D1US79bk7Sc-PIb5b1c2g4hk",
  authDomain: "tugaskampus1-16cd5.firebaseapp.com",
  databaseURL: "https://tugaskampus1-16cd5.firebaseio.com",
  projectId: "tugaskampus1-16cd5",
  storageBucket: "tugaskampus1-16cd5.appspot.com",
  messagingSenderId: "869186910796",
  appId: "1:869186910796:web:011c5b534c1894498f92e9",
  measurementId: "G-3FN27WPPEZ"
};
//
// firebase.analytics();
export const myFirebase = firebase.initializeApp(firebaseConfig);
const baseDb = myFirebase.firestore();
export const db = baseDb;
```

Kemuadi action/auth.js

```
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());
    });
};
```

Kemudian kita edit reducernya

Reducerst/auth.js

```
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;
        }
    };
```

Edit juga reducers/index.js

```
import { combineReducers } from "redux";
import auth from "./auth";
export default combineReducers({ auth });
```

kemudian setting redux plumbing

pilih file configureStore.js di dalam 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;
}
```

Edit juga bagian root.js

```
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 data src/index.js

```
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 App.js

```
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);
```

kemudian kita buat tampilan login nya , di components/login.js

```
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));
```

buat juga file home.js nya

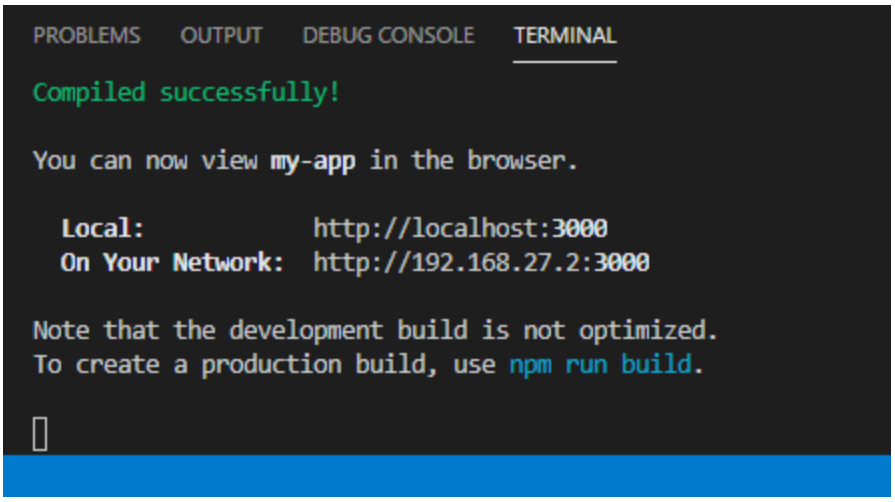
```
import React, { Component } from "react";
import { connect } from "react-redux";
import { logoutUser } from "../actions/auth";

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);
```

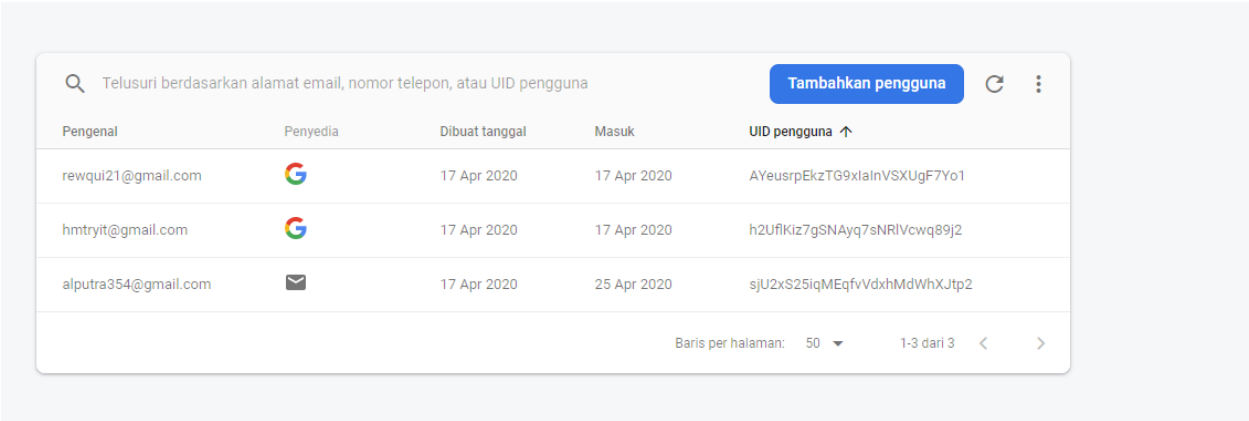
Dan buat file dalam folder components dengan nama protectedroute.js

```
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;
```

Start project

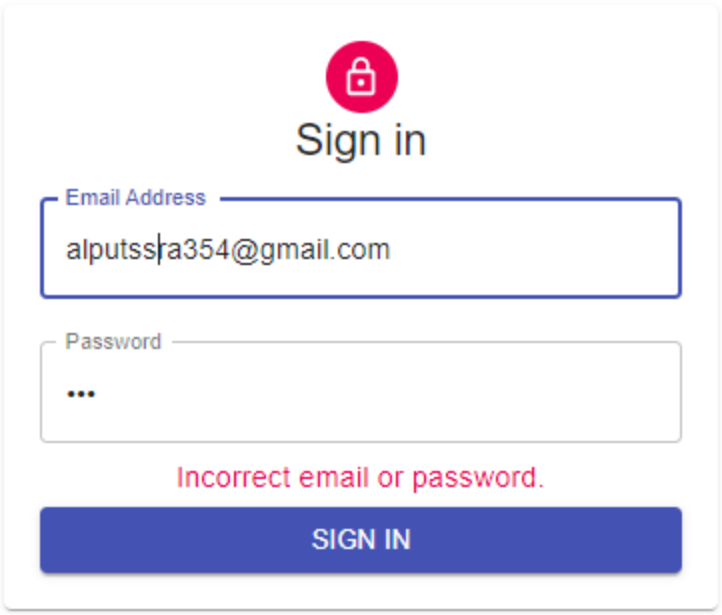


Kemudian tambah atau cek user di firebase anda
Di authentication



Dan coba masukkan di web login

Apablia login salah



Dan apabila login anda benar , akan masuk kehalaman home