TUGAS PERORANGAN/INDIVIDU

LAPORAN JOBSHEET MINGGU 12

FIREBASE LOGIN

Disusun Sebagai:

MATA KULIAH: PEMROGRAMAN BERBASIS FRAMEWORK

Oleh:

Moh. Khoirul Arifin/1741720051

TI-3B

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

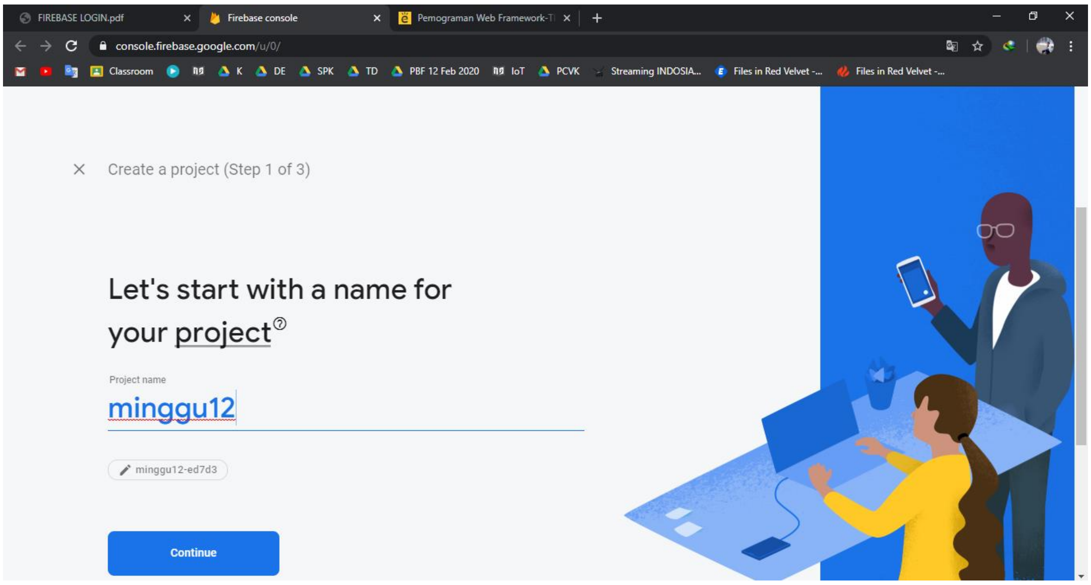
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

1. Setting Firebase Console

a. Create Project



Create a project (Step 1 of 3)

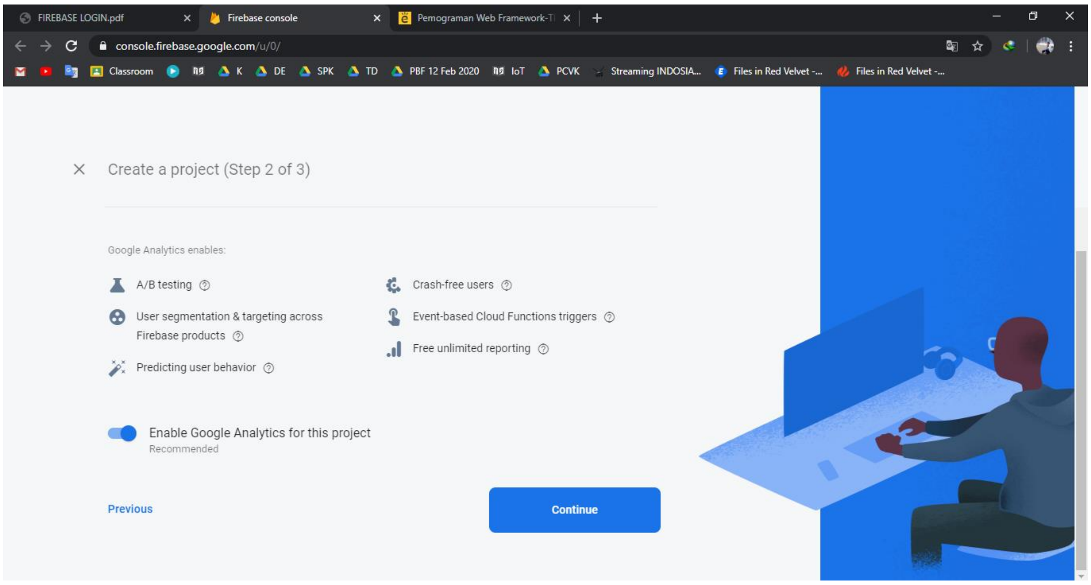
Let's start with a name for your project

Project name

minggu12

minggu12-ed7d3

Continue



Create a project (Step 2 of 3)

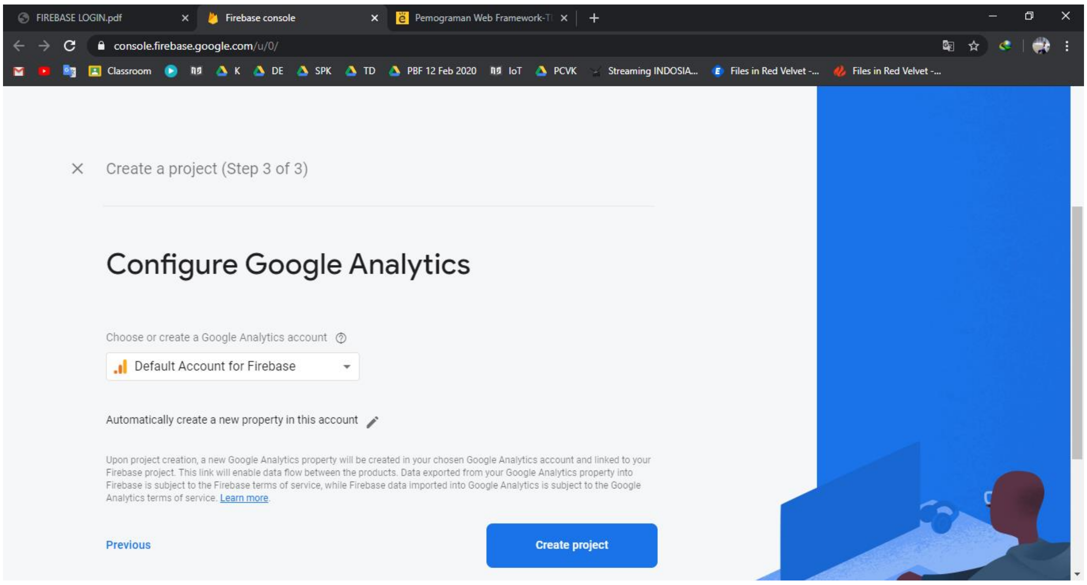
Google Analytics enables:

- A/B testing
- User segmentation & targeting across Firebase products
- Predicting user behavior
- Crash-free users
- Event-based Cloud Functions triggers
- Free unlimited reporting

● Enable Google Analytics for this project
Recommended

Previous

Continue



Create a project (Step 3 of 3)

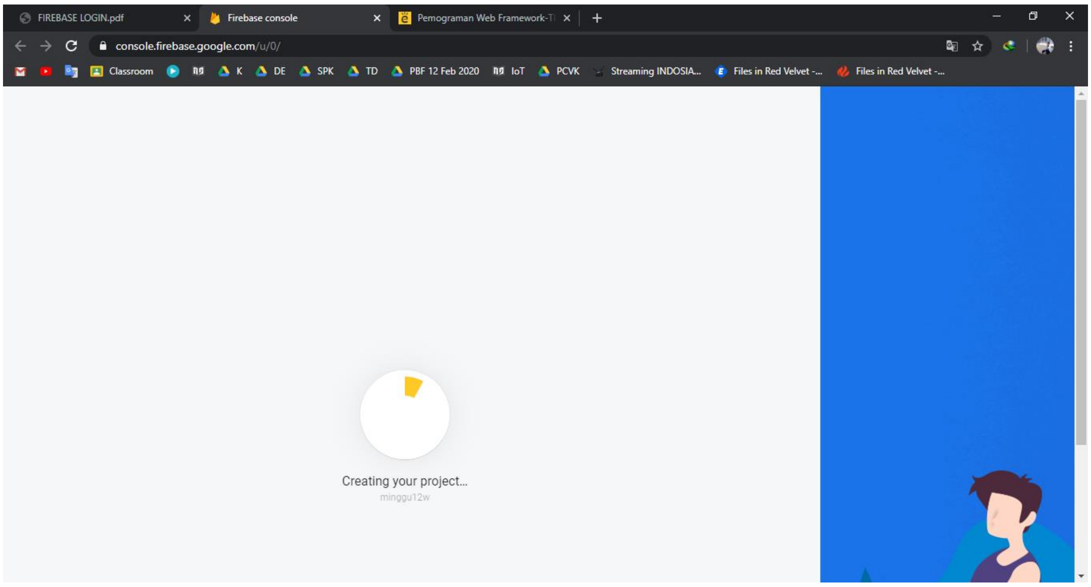
Configure Google Analytics

Choose or create a Google Analytics account

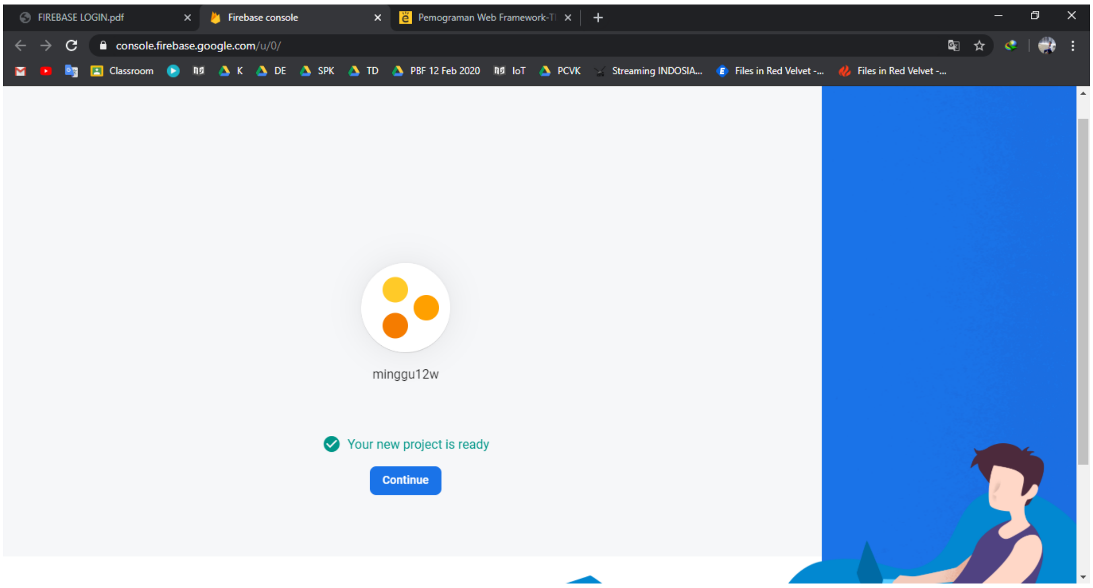
Default Account for Firebase

Automatically create a new property in this account

Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. Learn more.



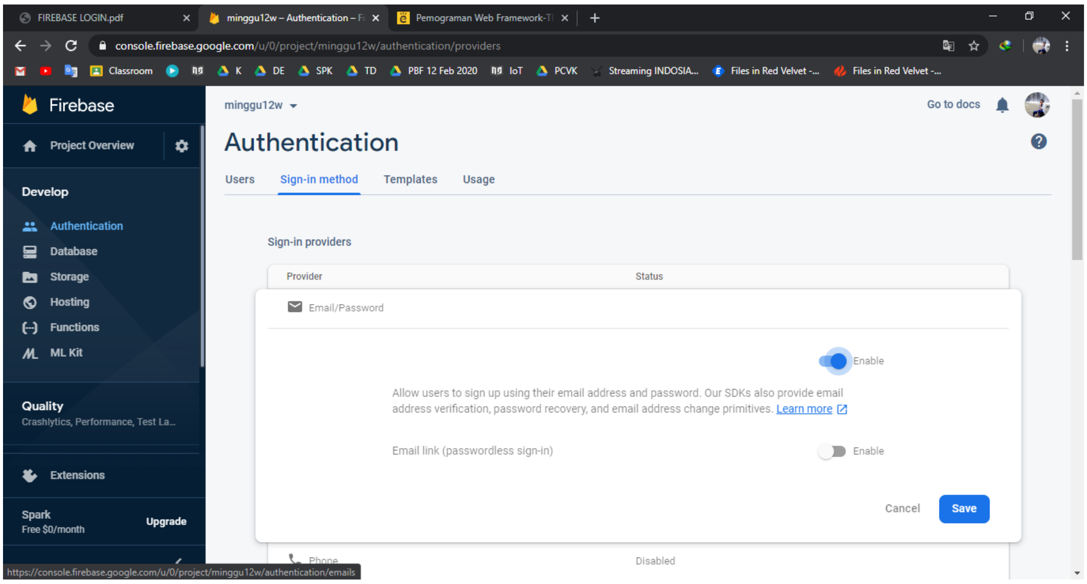
Creating your project...
minggu12w



minggu12w

Your new project is ready

b. Setting Authentification


FIREBASE LOGIN.pdf
minggu12w – Authentication – F
Pemograman Web Framework-T
console.firebase.google.com/u/0/project/minggu12w/authentication/providers
Firebase
Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics, Performance, Test La...
Extensions
Spark
Free $0/month
Upgrade
minggu12w
Go to docs
Users
Sign-in method
Templates
Usage
https://console.firebase.google.com/u/0/project/minggu12w/authentication/emails

Authentication

Sign-in providers

Provider | Status

Email/Password

Enable

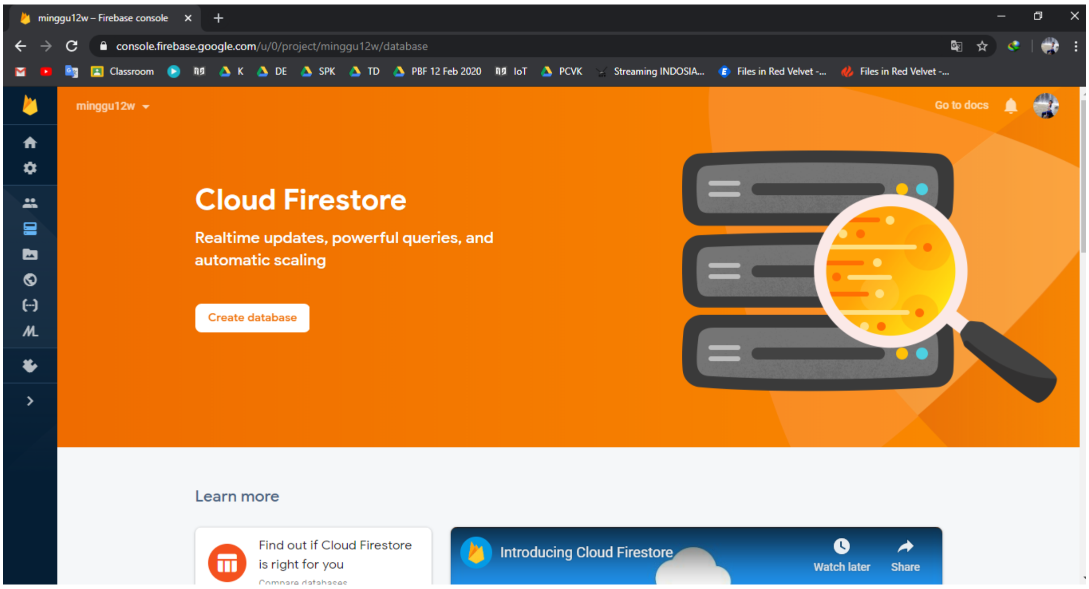
Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. Learn more

Email link (passwordless sign-in) Enable

Cancel Save

Phone Disabled

c. Create Database


minggu12w – Firebase console
console.firebase.google.com/u/0/project/minggu12w/database
minggu12w
Go to docs
Create database
Watch later
Share

Cloud Firestore

Realtime updates, powerful queries, and automatic scaling

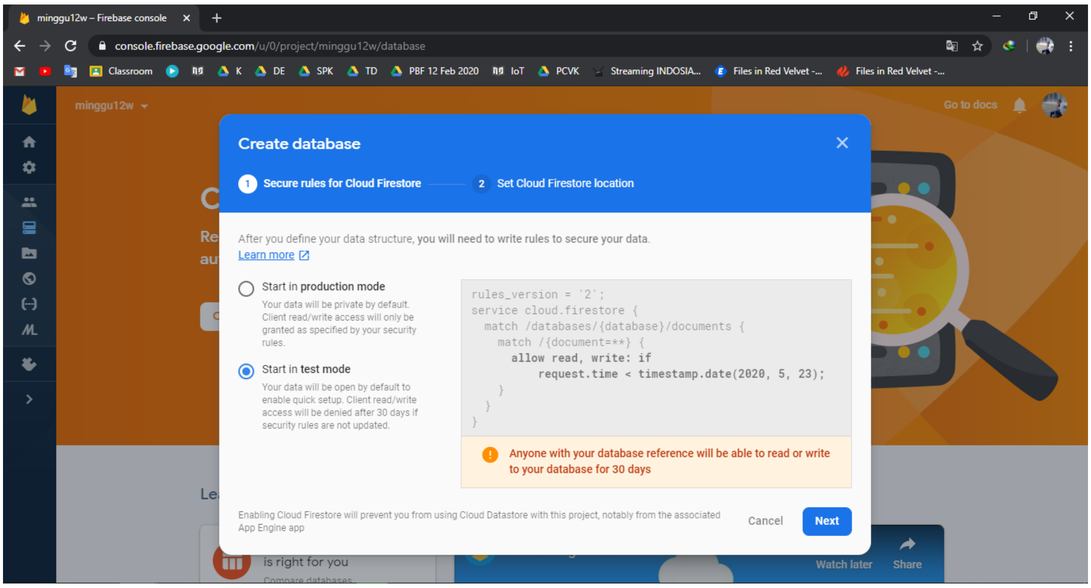
Learn more

Find out if Cloud Firestore is right for you

Compare databases

Introducing Cloud Firestore

d. Pilih Start in Test Mode


minggu12w – Firebase console
console.firebase.google.com/u/0/project/minggu12w/database
minggu12w
Go to docs

Create database

1 Secure rules for Cloud Firestore — 2 Set Cloud Firestore location

After you define your data structure, you will need to write rules to secure your data.
Learn more

○ Start in production mode
Your data will be private by default. Client read/write access will only be granted as specified by your security rules.

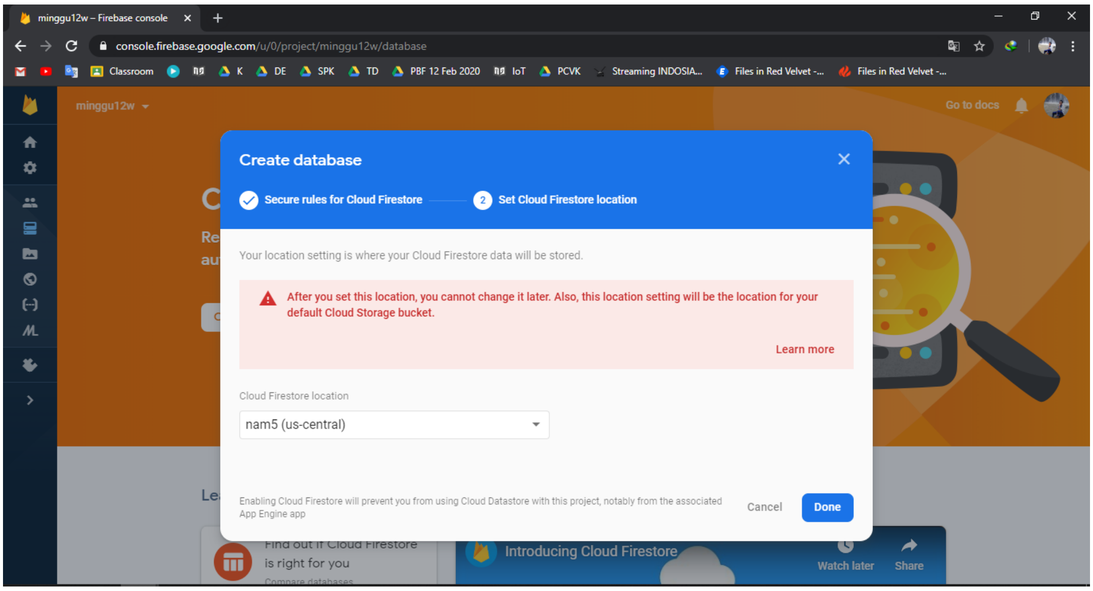
● Start in test mode
Your data will be open by default to enable quick setup. Client read/write access will be denied after 30 days if security rules are not updated.

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

Anyone with your database reference will be able to read or write to your database for 30 days

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

e. Set Cloud Firestore Location


minggu12w – Firebase console
console.firebase.google.com/u/0/project/minggu12w/database
minggu12w
Go to docs

Create database

✓ Secure rules for Cloud Firestore — 2 Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.

After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

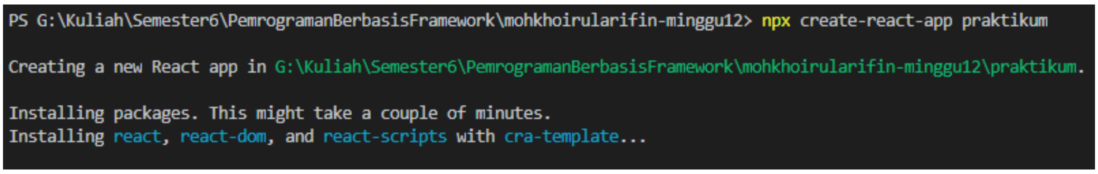
Learn more

Cloud Firestore location: nam5 (us-central)

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

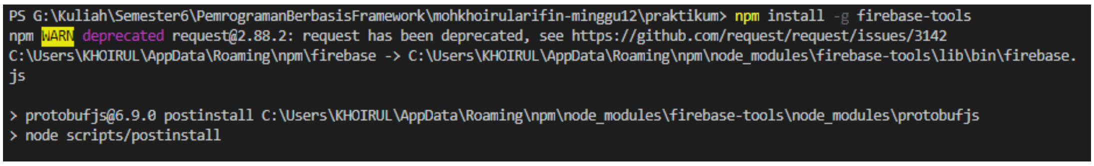
2. Setting React app dan firebase

a. Membuat Aplikasi Baru

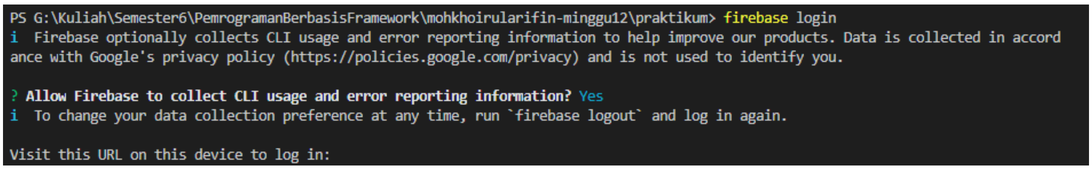


```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12> npx create-react-app praktikum

Creating a new React app in G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum.

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

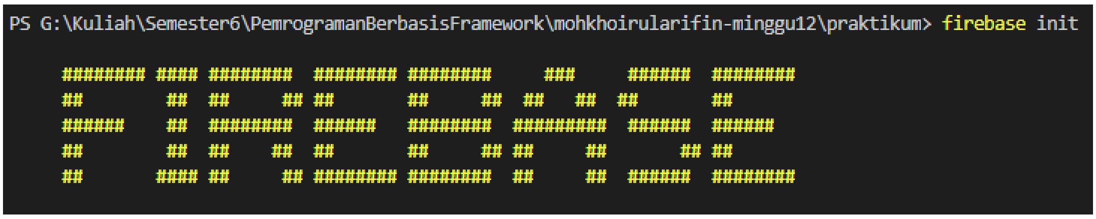
b. Install Firebase



```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> npm install -g firebase-tools
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
C:\Users\KHOIRUL\AppData\Roaming\npm\firebase -> C:\Users\KHOIRUL\AppData\Roaming\npm\node_modules\firebase-tools\lib\bin\firebase.
js

> protobufjs@6.9.0 postinstall C:\Users\KHOIRUL\AppData\Roaming\npm\node_modules\firebase-tools\node_modules\protobufjs
> node scripts/postinstall
```

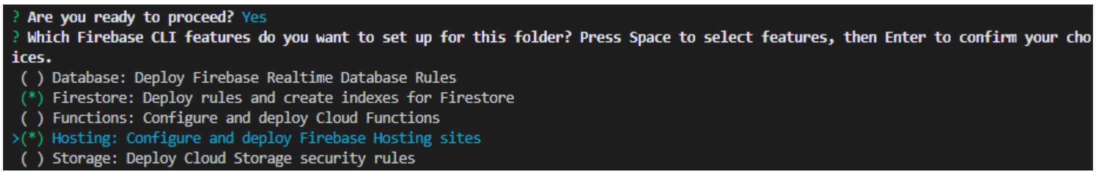


```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> firebase login
i  Firebase optionally collects CLI usage and error reporting information to help improve our products. Data is collected in accord
ance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

? 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:
```



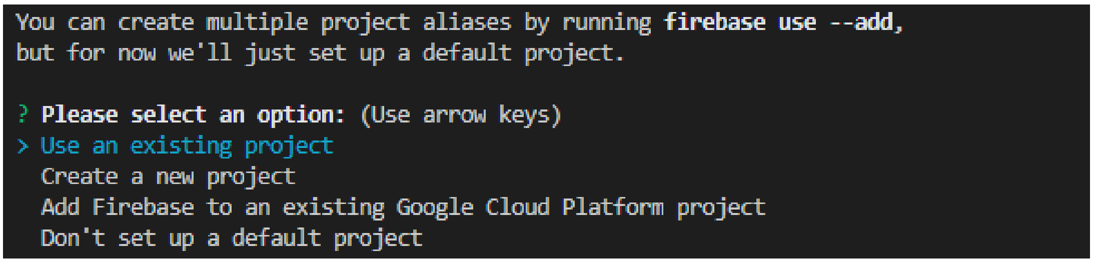
```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> firebase init

     ######## #### ########  ######## ########     ###     ######  ########
     ##        ##  ##     ## ##       ##     ##  ##   ##  ##       ##
     ######    ##  ########  ######   ########  #########  ######  ######
     ##        ##  ##    ##  ##       ##     ## ##     ##       ## ##
     ##       #### ##     ## ######## ########  ##     ##  ######  ########
```

c. Pilih firestore dan hosting



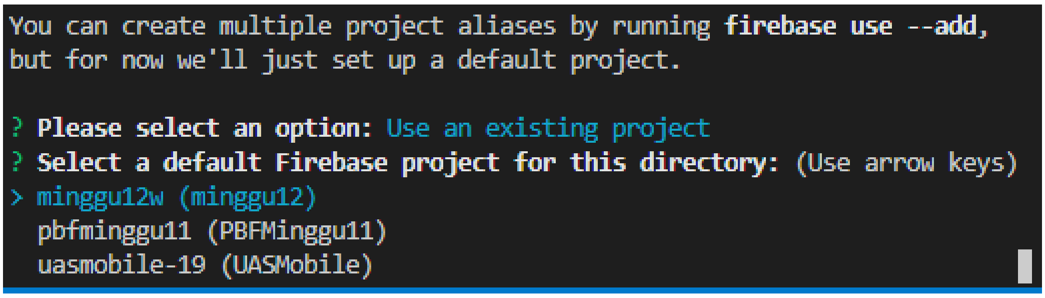
```
? Are you ready to proceed? Yes
? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your cho
ices.
 ( ) Database: Deploy Firebase Realtime Database Rules
 (*) Firestore: Deploy rules and create indexes for Firestore
 ( ) Functions: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

d. Pilih Use existing project



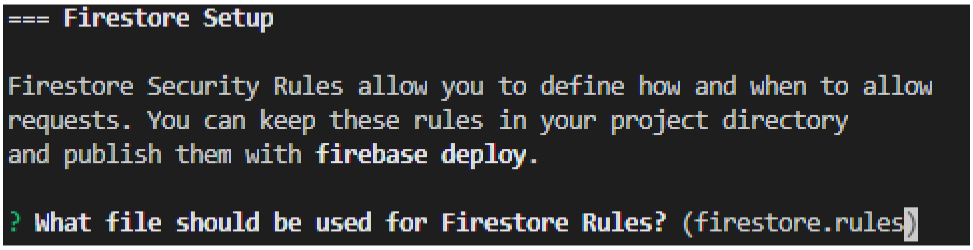
```
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: (Use arrow keys)
> Use an existing project
  Create a new project
  Add Firebase to an existing Google Cloud Platform project
  Don't set up a default project
```

e. Pilih Project yang telah dibuat pada firebase



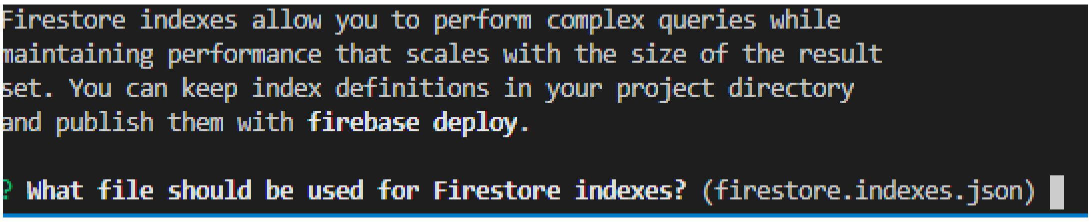
```
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory: (Use arrow keys)
> minggu12w (minggu12)
  pbfminggu11 (PBFMinggu11)
  uasmobile-19 (UASMobile)
```

f. Enter firestore.rules



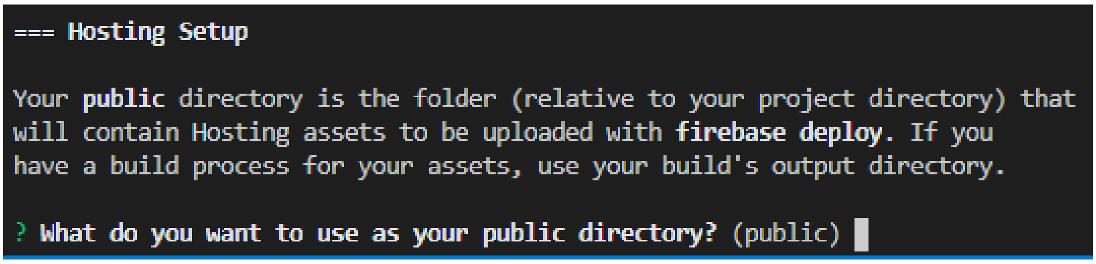
```
=== Firestore Setup

Firestore Security Rules allow you to define how and when to allow
requests. You can keep these rules in your project directory
and publish them with firebase deploy.

? What file should be used for Firestore Rules? (firestore.rules)
```

g. Enter firestore.index.jsos



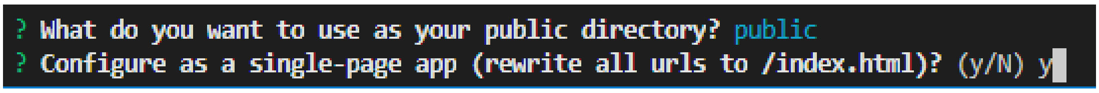
```
Firestore indexes allow you to perform complex queries while
maintaining performance that scales with the size of the result
set. You can keep index definitions in your project directory
and publish them with firebase deploy.

? What file should be used for Firestore indexes? (firestore.indexes.json)
```

h. Enter public



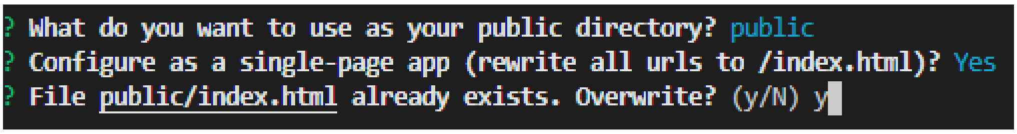
```
=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? (public)
```

i. Pilih y

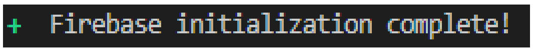


```
? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? (y/N) y
```

j. Pilih y untuk overwrite

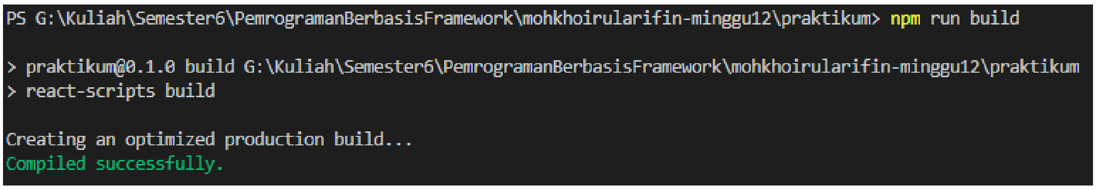


```
? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? File public/index.html already exists. Overwrite? (y/N) y
```



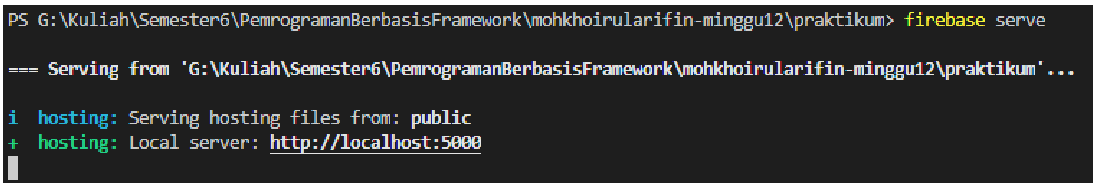
```
+  Firebase initialization complete!
```

k. Ketik npm run Build



```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> npm run build

> praktikum@0.1.0 build G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum
> react-scripts build

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

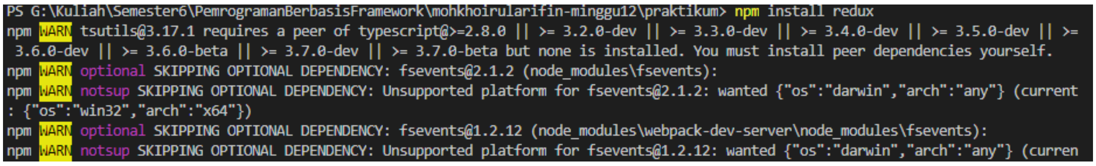
l. Ketik firebase serve



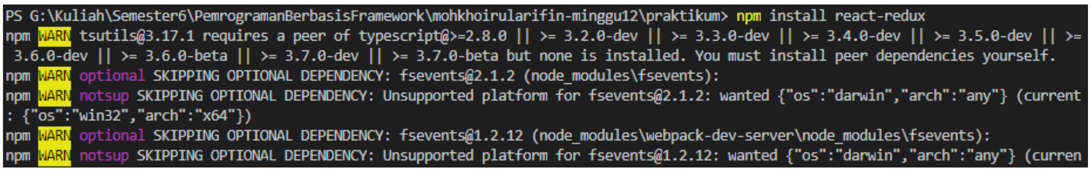
```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> firebase serve

=== Serving from 'G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum'...

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

3. Instalasi Redux

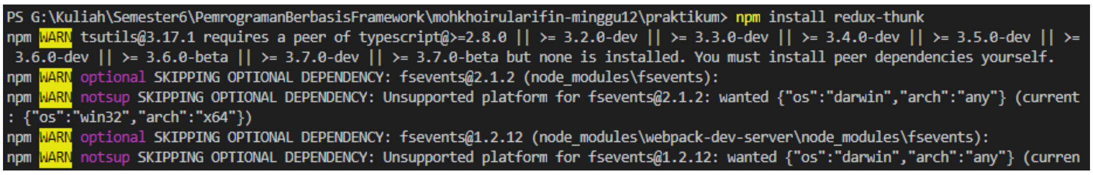
a. Install Redux dan react redux, redux thunk



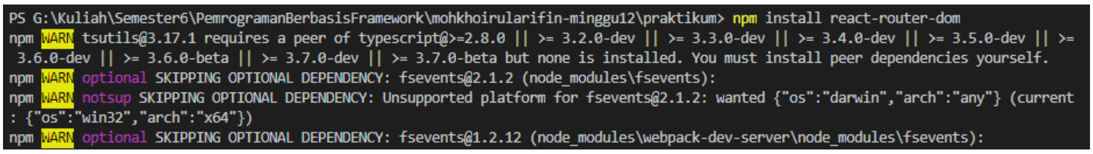
```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> npm install redux
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >=
 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current
: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (curren
```



```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> npm install react-redux
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >=
 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current
: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (curren
```

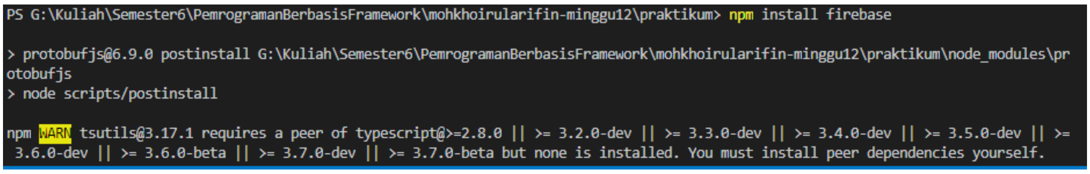


```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> npm install redux-thunk
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >=
 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current
: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (curren
```



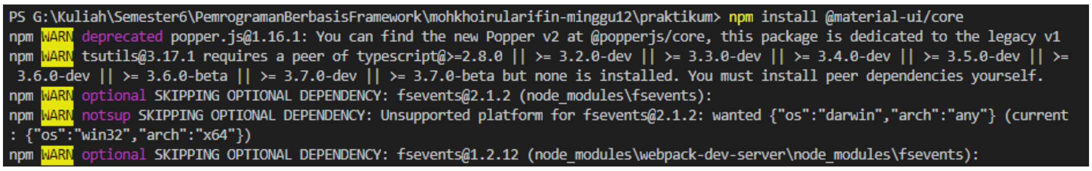
```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> npm install react-router-dom
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >=
 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current
: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\webpack-dev-server\node_modules\fsevents):
```

b. Install firebase



```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> npm install firebase

> protobufjs@6.9.0 postinstall G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum\node_modules\pr
otobufjs
> node scripts/postinstall

npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >=
 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
```

c. Pilih UI library

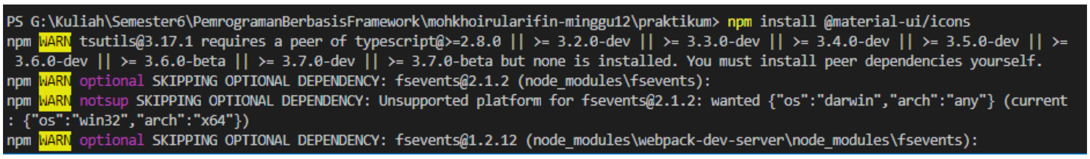


```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> npm install @material-ui/core
npm WARN deprecated popper.js@1.16.1: You can find the new Popper v2 at @popperjs/core, this package is dedicated to the legacy v1
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >=
 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current
: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\webpack-dev-server\node_modules\fsevents):
```

d. Paste kan file public/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Welcome to Firebase Hosting</title>

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

     (...................................)
</html>
```

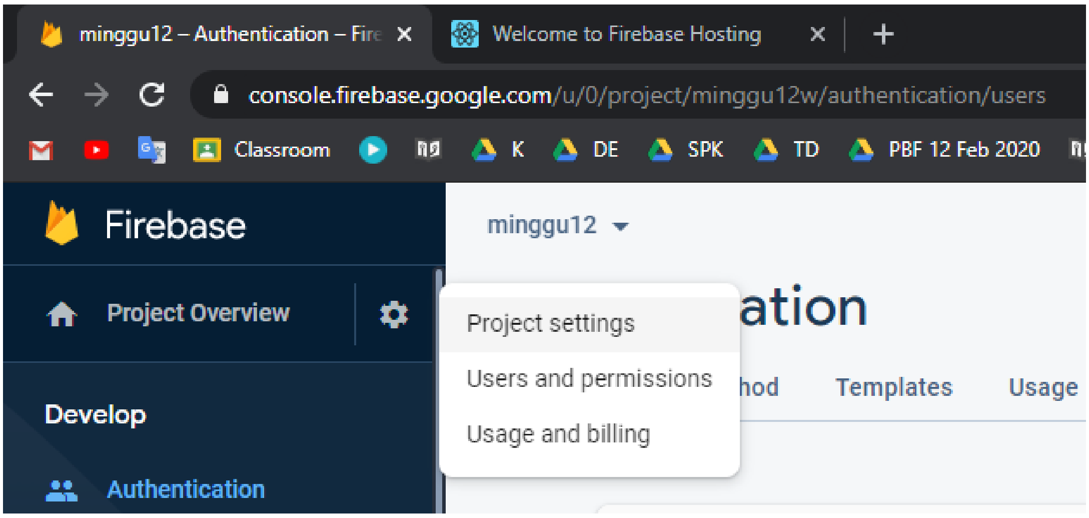
e. Install material icon

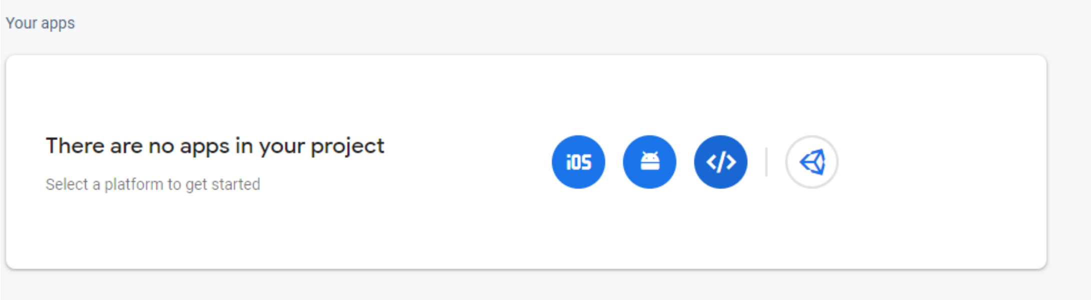


```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu12\praktikum> npm install @material-ui/icons
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >=
 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current
: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\webpack-dev-server\node_modules\fsevents):
```

4. Setting Firebase config

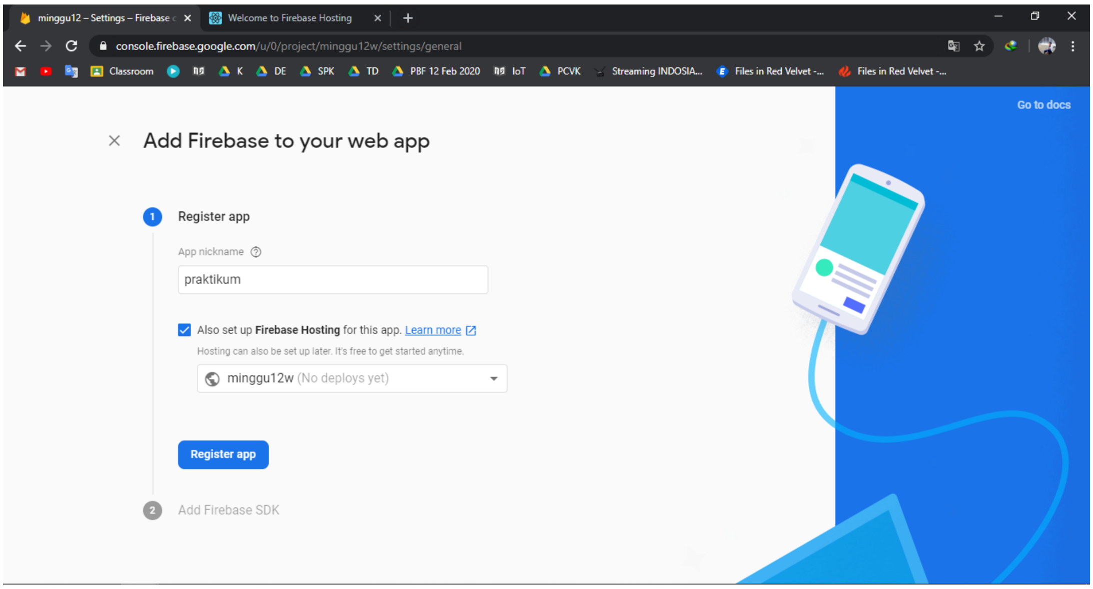
a. Pilih project setting





There are no apps in your project

Select a platform to get started



Add Firebase to your web app

1 Register app

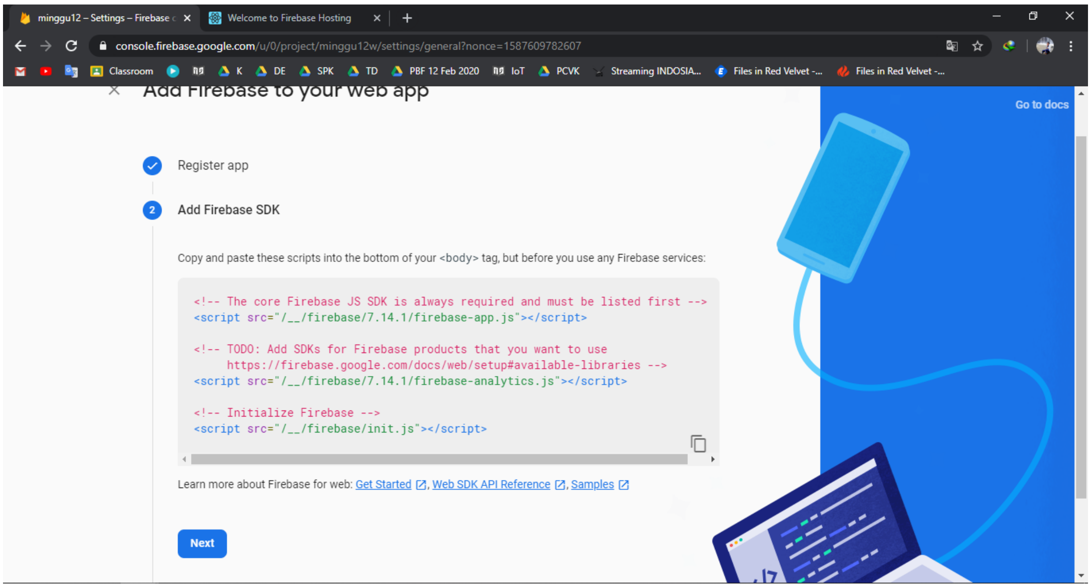
App nickname: praktikum

☑ Also set up **Firebase Hosting** for this app. Learn more

Hosting can also be set up later. It's free to get started anytime.

minggu12w (No deploys yet)

2 Add Firebase SDK


minggu12 – Settings – Firebase c
Welcome to Firebase Hosting
console.firebase.google.com/u/0/project/minggu12w/settings/general?nonce=1587609782607
Classroom
K
DE
SPK
TD
PBF 12 Feb 2020
IoT
PCVK
Streaming INDOSIA...
Files in Red Velvet -...
Files in Red Velvet -...
Go to docs

Add Firebase to your web app

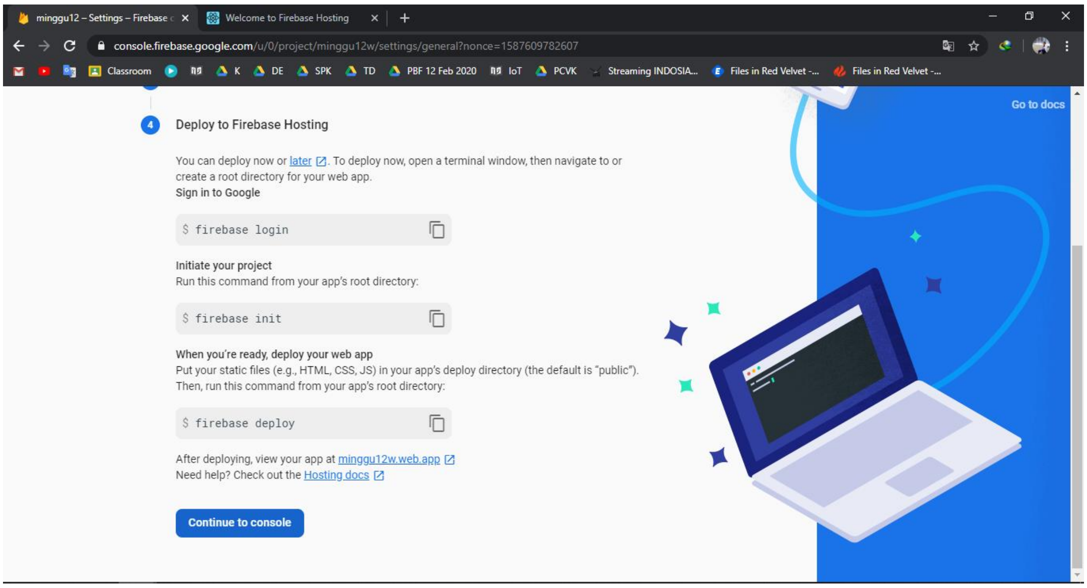
Register app

2 Add Firebase SDK

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="/__/firebase/7.14.1/firebase-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
     https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="/__/firebase/7.14.1/firebase-analytics.js"></script>

<!-- Initialize Firebase -->
<script src="/__/firebase/init.js"></script>
```

Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples


minggu12 – Settings – Firebase c
Welcome to Firebase Hosting
console.firebase.google.com/u/0/project/minggu12w/settings/general?nonce=1587609782607
Classroom
K
DE
SPK
TD
PBF 12 Feb 2020
IoT
PCVK
Streaming INDOSIA...
Files in Red Velvet -...
Files in Red Velvet -...
Go to docs

4 Deploy to Firebase Hosting

You can deploy now or later. To deploy now, open a terminal window, then navigate to or create a root directory for your web app.

Sign in to Google

```
$ firebase login
```

Initiate your project

Run this command from your app's root directory:

```
$ firebase init
```

When you're ready, deploy your web app

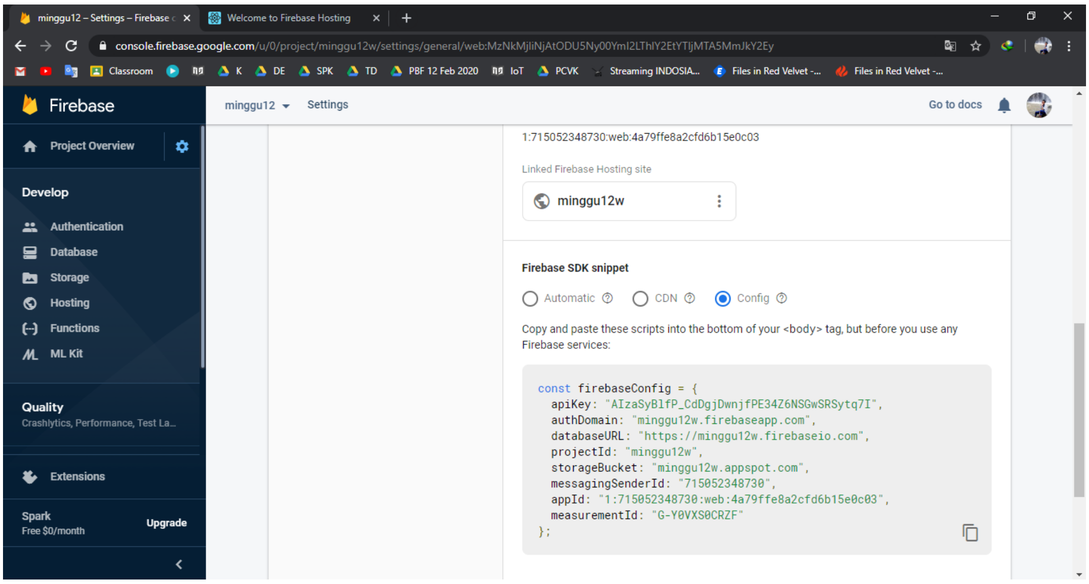
Put your static files (e.g., HTML, CSS, JS) in your app's deploy directory (the default is "public"). Then, run this command from your app's root directory:

```
$ firebase deploy
```

After deploying, view your app at minggu12w.web.app

Need help? Check out the Hosting docs

b. Pilih config



1:715052348730:web:4a79ffe8a2cfd6b15e0c03

Linked Firebase Hosting site

minggu12w

Firebase SDK snippet

○ Automatic ○ CDN ● Config

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
const firebaseConfig = {
  apiKey: "AIzaSyBlfP_CdDgjDwnjfPE34Z6NSGwSRSytq7I",
  authDomain: "minggu12w.firebaseapp.com",
  databaseURL: "https://minggu12w.firebaseio.com",
  projectId: "minggu12w",
  storageBucket: "minggu12w.appspot.com",
  messagingSenderId: "715052348730",
  appId: "1:715052348730:web:4a79ffe8a2cfd6b15e0c03",
  measurementId: "G-Y0VXS0CRZF"
};
```

c. Edit firebase/firebase.js

```
import firebase from "firebase/app";
import "firebase/auth";
import "firebase/firestore";

const firebaseConfig = {
    //Your Config Values
    apiKey: "AIzaSyBlfP_CdDgjDwnjfPE34Z6NSGwSRSytq7I",
    authDomain: "minggu12w.firebaseapp.com",
    databaseURL: "https://minggu12w.firebaseio.com",
    projectId: "minggu12w",
    storageBucket: "minggu12w.appspot.com",
    messagingSenderId: "715052348730",
    appId: "1:715052348730:web:4a79ffe8a2cfd6b15e0c03",
    measurementId: "G-Y0VXS0CRZF"
};

export const myFirebase = firebase.initializeApp(firebaseConfig);
const baseDb = myFirebase.firestore();
export const db = baseDb;
```

d. Edit actions/auth.js

```
import { myFirebase } from "../firebase/firebase";

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());
    });
};


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";
```

5. Setting Reducer

   a. Buka reducers/auth.js

```
import {
    LOGIN_REQUEST,
    LOGIN_SUCCESS,
    LOGIN_FAILURE,
    LOGOUT_REQUEST,
    LOGOUT_SUCCESS,
    LOGOUT_FAILURE,
    VERIFY_REQUEST,
    VERIFY_SUCCESS
} from "../actions/";

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;
    }
};
```

b. Edit reducers/index.js

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

6. Setting Redux plumbing

a. Pilih src folder called configureStore.js dan edit

```
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;
}
```

b. Pilih dan edit folder src dan 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;
```

c. Pilih dan edit folder 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"));

serviceWorker.unregister();
```

d. Buat lah coding 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);
```

e. Buatlah file dan simpan components/Login.js

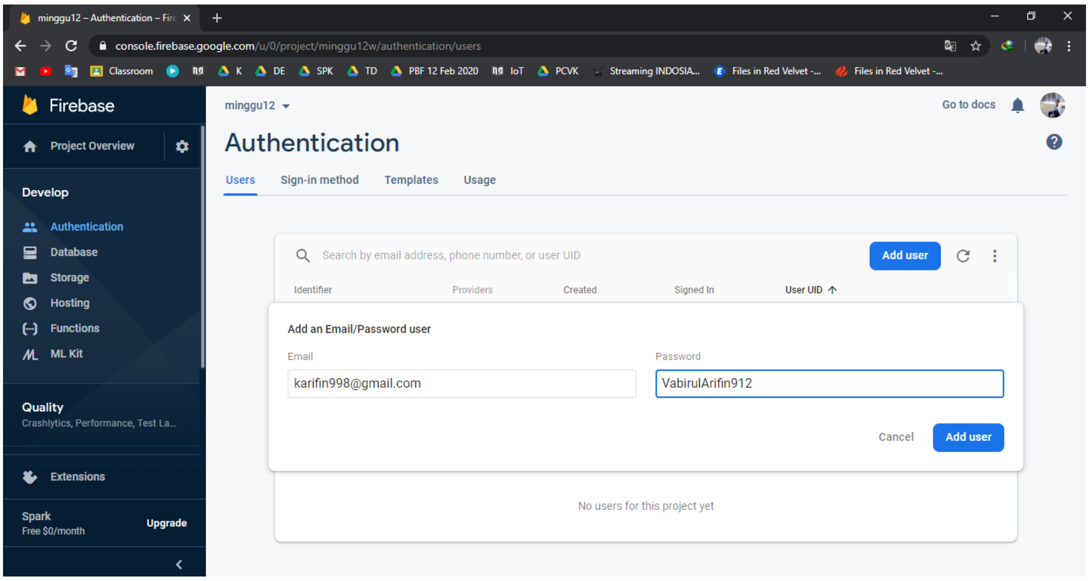
```
import React, { Component } from "react";
import { connect } from "react-redux";
import { Redirect } from "react-router-dom";
import { loginUser } from "../actions/auth";
import { withStyles } from "@material-ui/styles";
import Avatar from "@material-ui/core/Avatar";
import Button from "@material-ui/core/Button";
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));
```

f. Buka component/Home.js

```
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);
```

g. Masuk ke firebase console kemudian add user



console.firebase.google.com/u/0/project/minggu12w/authentication/users

Add an Email/Password user

Email: karifin998@gmail.com

Password: VabirulArifin912

No users for this project yet

h. Pilih npm start kemudian masukkan username dan password

Hasil masih eror pak, mohon maaf