

**TUGAS PERORANGAN/INDIVIDU**

**LAPORAN JOBSHEET MINGGU 11**  
**FIREBASE REACT**

**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**

## A. Menggunakan Email

### 1. Membuat Project React

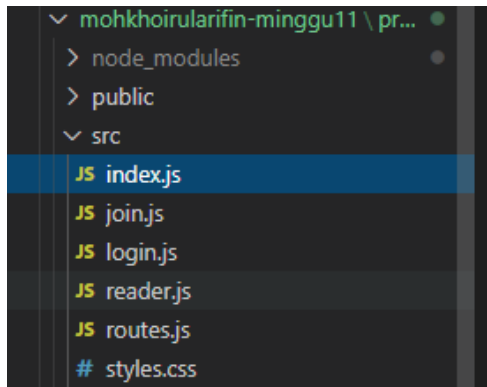
```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu11> npx create-react-app praktikum

Creating a new React app in G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu11\praktikum.

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

> core-js@2.6.11 postinstall G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu11\praktikum\node_modules\babel-runtime\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"
```

### 2. Membuat File



### 3. Install react-router-dom

```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu11\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):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\watchpack\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
```

### 4. Mengubah Code Tiap-Tiap File yang Dibuat

#### a. index.js

```
import React, { useState } from "react";
import ReactDOM from "react-dom";
import { Switch, BrowserRouter as Router, Route } from "react-router-dom";
import routes from "../routes.js";
import Header from "../Header.js";
import "../styles.css"

export const AuthContext = React.createContext(null);

function App() {
  const [isLoggedIn, setLoggedIn] = useState(false)

  return (
```

```

    <AuthContext.Provider value = {{ isLoggedIn, setLoggedIn }}
  >>
    Is Logged in? {JSON.stringify(isLoggedIn)}
    <div className = "App">
      <Router>
        <Header/>

        <Switch>
          {routes.map(route => (
            <Route
              key = {route.path}
              path = {route.path}
              exact = {route.exact}
              component = {route.main}
            />
          ))}
        </Switch>
      </Router>
    </div>
  </AuthContext.Provider>
);
}

const rootElement = document.getElementById("root");
ReactDOM.render(<App />, rootElement);

```

## b. Login.js

```

import React, { useState, useContext } from "react";
import { AuthContext } from "../index";

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);
  };

  return (
    <div>
      <h1>Login</h1>
      <form onSubmit = {e => handleForm(e)}>
        <input
          value = {email}
          onChange = {e => setEmail(e.target.value)}
          name = "email"
          type = "email"
          placeholder = "email"
        />
        <input
          onChange = {e => setPassword(e.target.value)}

          name = "password"
          value = {password}
          type = "password"
          placeholder = "password"
        />
      </form>
    </div>
  );
};

```

```

        <hr />
        <button class = "googleBtn" type = "button">
            <img
                src = "https://upload.wikimedia.org/wikipedia/commons/5/53/Google_%22G%22_Logo.svg"
                alt = "logo"
            />
            Login With Google
        </button>
        <button type = "submit"> Login </button>
        <span>{error}</span>
    </form>
</div>
    );
};

export default Login;

```

### c. Join.js

```

import React, { useState, useContext } from "react";
import { AuthContext } from "../index"

const Join = () => {
    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);
    };

    return (
        <div>
            <h1>Join</h1>
            <form onSubmit = {e => handleForm(e)}>
                <input
                    value = {email}
                    onChange = {e => setEmail(e.target.value)}
                    name = "email"
                    type = "email"
                    placeholder = "email"
                />
                <input
                    onChange = {e => setPassword(e.target.value)}
                    name = "password"
                    value = {password}
                    type = "password"
                    placeholder = "password"
                />
                <hr />
                <button class = "googleBtn" type = "button">
                    <img
                        src = "https://upload.wikimedia.org/wikipedia/commons/5/53/Google_%22G%22_Logo.svg"
                        alt = "logo"
                    />
                    Join With Google
                </button>
            </form>
        </div>
    );
};

```

```

        <button type = "submit"> Login </button>

        <span>{error}</span>
    </form>
</div>
    );
};

export default Join;

```

d. Header.js

```

import React from "react";
import routes from "../routes";
import { Link } from "react-router-dom";

const Header = () => (
    <ul className = "nav">
        {routes.map((route, i) => (
            <li key = {i}>
                <Link to = {route.path}>{route.name}</Link>
            </li>
        ))}
    </ul>
);

export default Header;

```

e. routes.js

```

import React from "react";
import Login from "../Login";
import Join from "../Join";

const routes = [
    { name: "Join", path: "/", exact: true, main: () => <Join /> },
    { name: "Login", path: "/login", exact: true, main: () => <Login /> }
];

export default routes;

```

f. style.css

```

.App {
    font-family: sans-serif;
    padding: 50px;
}

h1 {
    margin: 0 0 25px 0;
    text-align: left;
}

form {
    display: flex;
    flex-direction: column;
}

input {
    max-width: 450px;
}

```

```
        width: auto;
        border: none;
        margin: 0 0 25px 0;
        box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px
        rgba(0, 0, 0, 0.24);
        transition: all 0.3s cubic-bezier(0.25, 0.8, 0.25, 1);
        padding: 10px;
    }

    button {
        width: 125px;
        box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px
        rgba(0, 0, 0, 0.24);
        border: none;
        padding: 7px;
        text-align: center;
        cursor: pointer;
    }

    button[type = "submit"] {
        background-color: #4c566a;
        color: white;
        font-size: 14px;
        font-weight: 900;
        border-radius: 5px;
        text-transform: uppercase;
        padding: 15px;
    }

    .googleBtn {
        width: 225px;
        margin: 0 0 50px 0;
        display: flex;
        justify-content: center;
        font-size: 16px;
    }

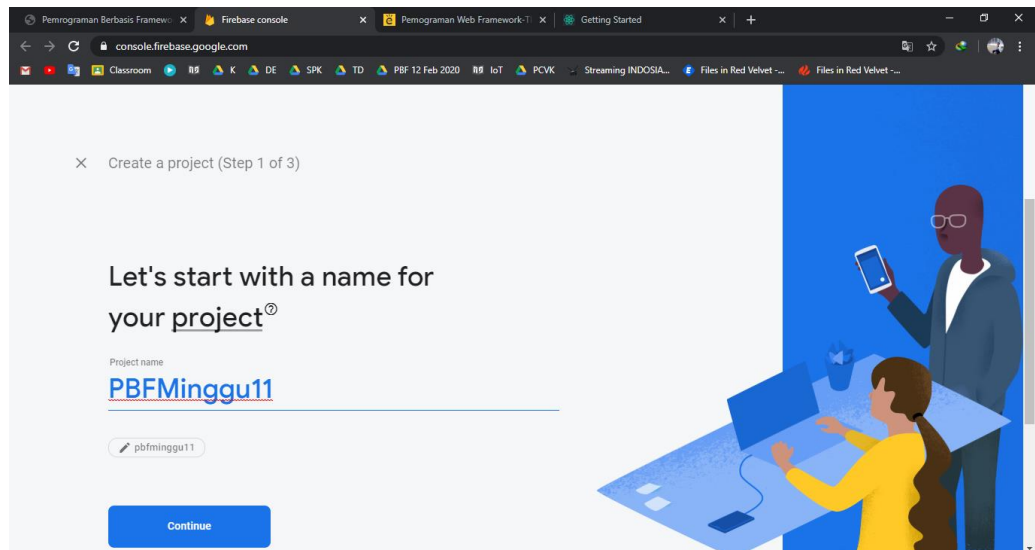
    .googleBtn img {
        width: 16px;
        height: 16px;
        padding: 0;
        margin: 0 5px;
        vertical-align: middle;
    }

    .nav {
        display: flex;
        text-decoration: none;
        list-style-type: none;
        justify-content: flex-start;
        margin: 0 0 50px;
        padding: 0;
    }

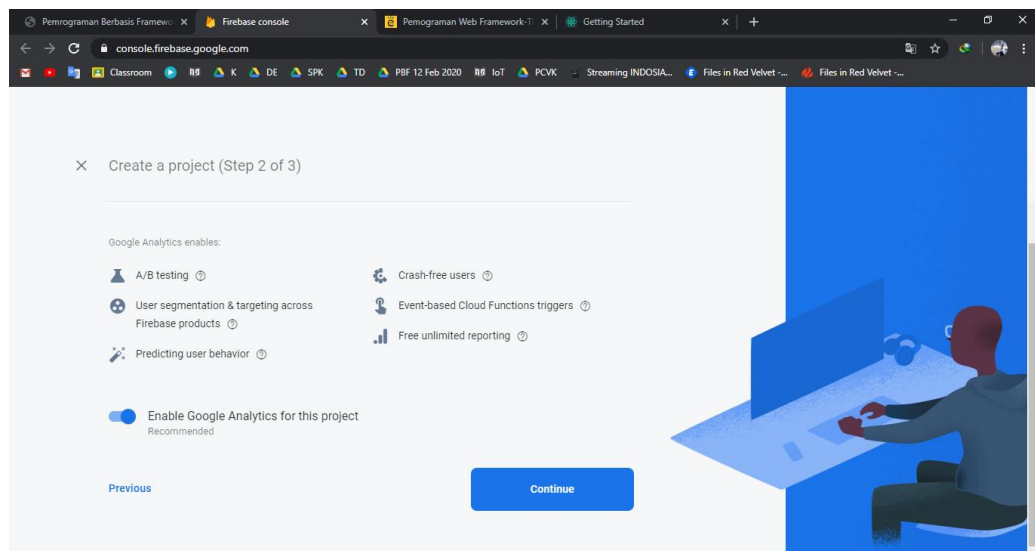
    .nav li {
        margin: 0 10px;
    }
}
```

## 5. Buat Project Baru Pada Firebase

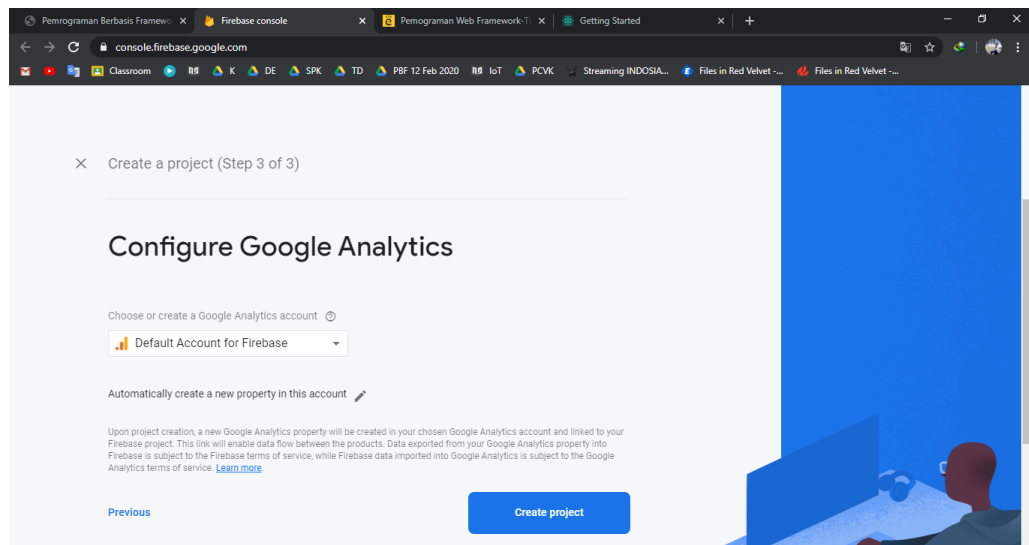
### a. Pengisian Nama Project



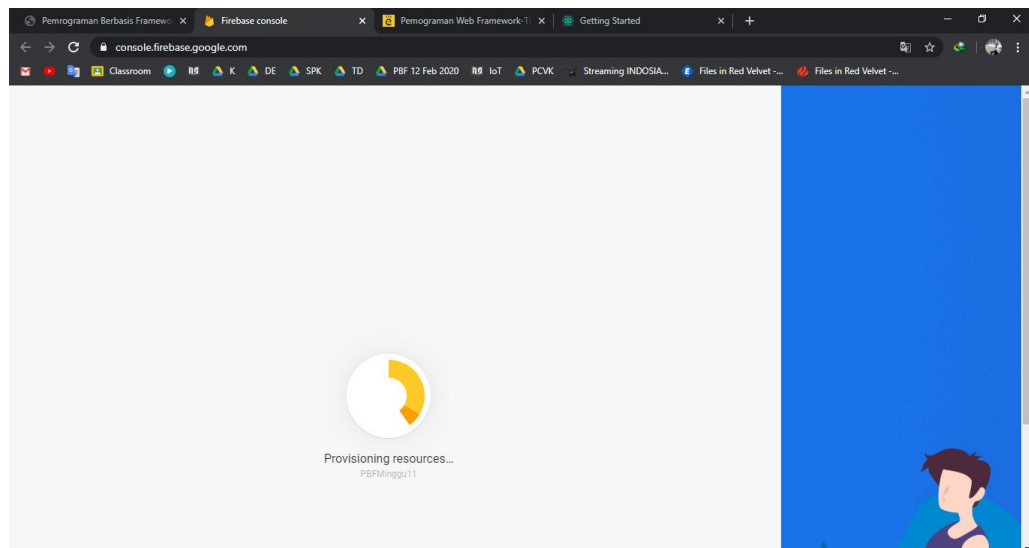
### b. Klik Continue



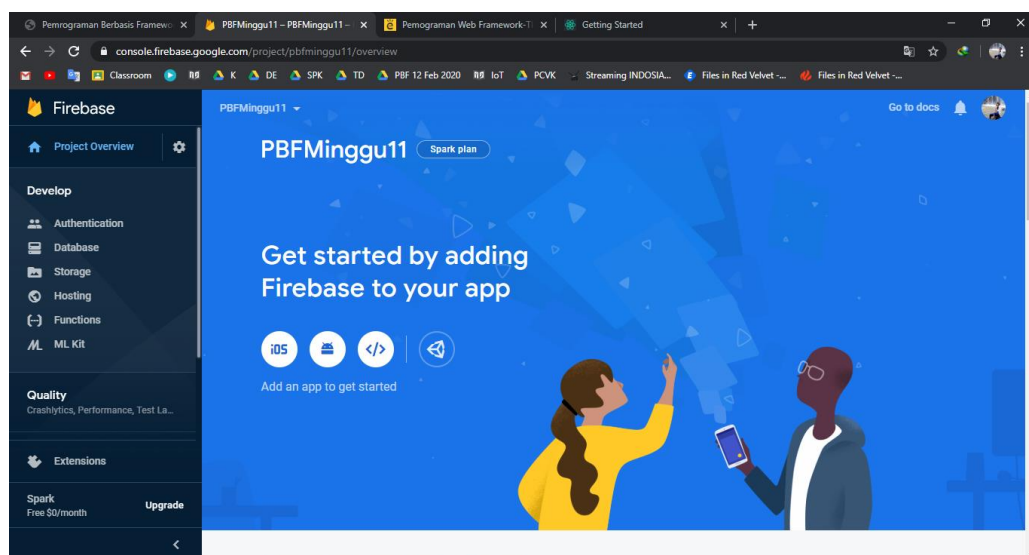
c. Klik Create Project



d. Tunggu Proses Selesai



e. Project Baru Telah Dibuat





## 6. Install firebase Pada Project

```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu11\praktikum> npm install --save firebase

> core-js@3.6.4 postinstall G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu11\praktikum\node_modules\firebase\polyfill\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!

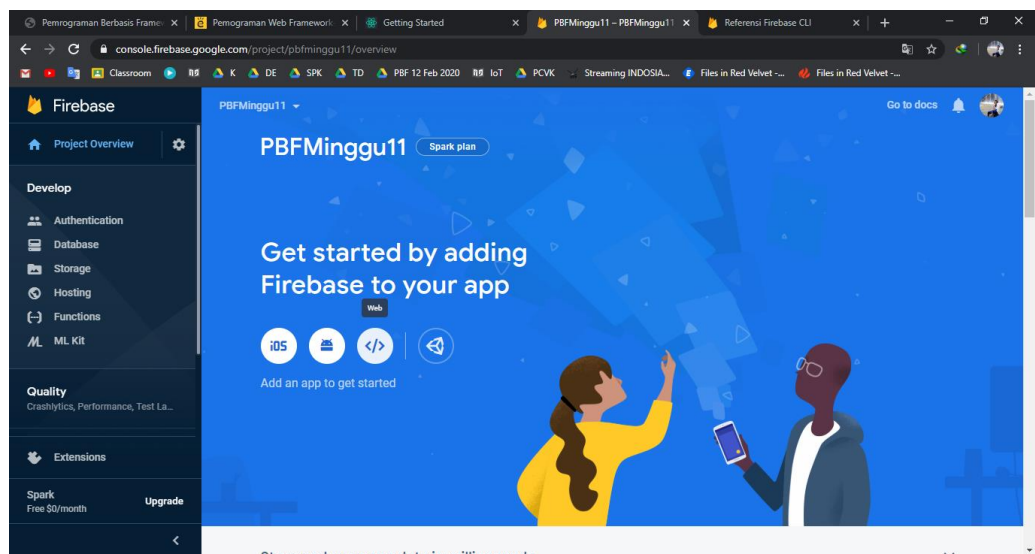
The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

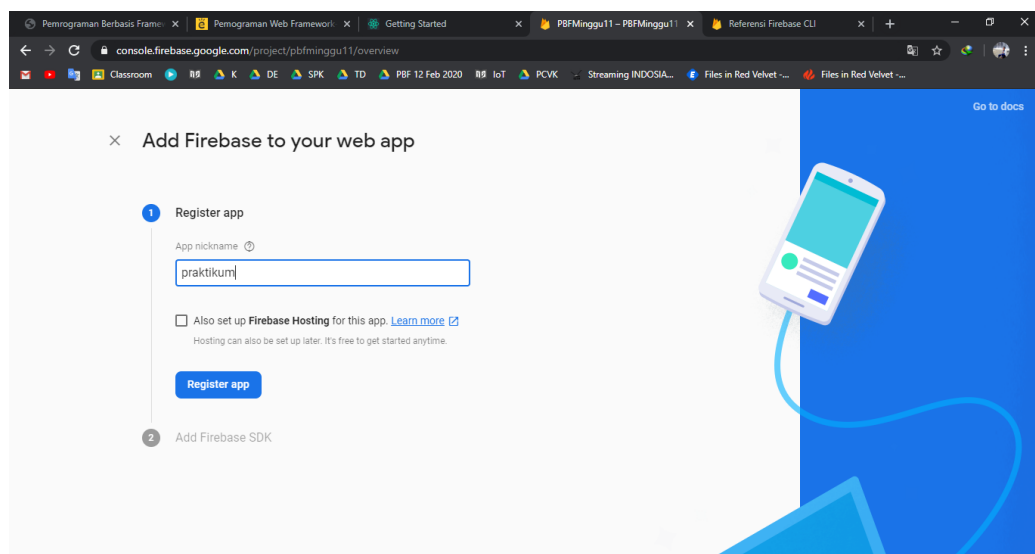
> protobufjs@6.8.9 postinstall G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu11\praktikum\node_modules\pr
```

## 7. Pilih Web pada Get Started Firebase

### a. Pilih Icon Web

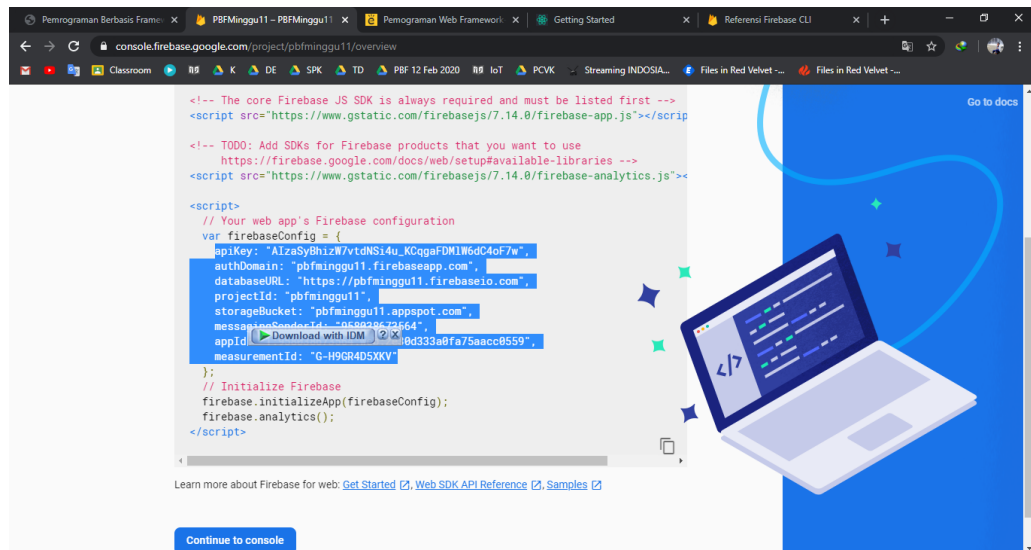


### b. Beri Nama Sesuai Project



## 8. Salin Script yang ada pada Firebase ke File baru

### a. Script Pada Firebase



### b. Paste Ke firebase.config.js yang baru dibuat

```
export default {
  apiKey: "AIzaSyBhizW7vtdNSi4u_KCqgaFDm1W6dC4oF7w",
  authDomain: "pbfminggu11.firebaseio.com",
  databaseURL: "https://pbfminggu11.firebaseio.com",
  projectId: "pbfminggu11",
  storageBucket: "pbfminggu11.appspot.com",
  messagingSenderId: "958038673564",
  appId: "1:958038673564:web:2e30d333a0fa75aacc0559",
  measurementId: "G-H9GR4D5XKV"
};
```

## 9. Inisialisasi Firebase pada index.js

```
import React, { useState } from "react";
import ReactDOM from "react-dom";
import { Switch, BrowserRouter as Router, Route } from "react-router-dom";
import routes from "./routes.js";
import Header from "./Header.js";
import "./styles.css"
import * as firebase from "firebase";
import firebaseConfog from "./firebase.config";

firebase.initializeApp(firebaseConfog);

export const AuthContext = React.createContext(null);

function App() {
  const [isLoggedIn, setLoggedIn] = useState(false)

  return (
    <AuthContext.Provider value = {{ isLoggedIn, setLoggedIn }}>
      Is Logged in? {JSON.stringify(isLoggedIn)}
      <div className = "App">
        <Router>
          <Header/>
```

```

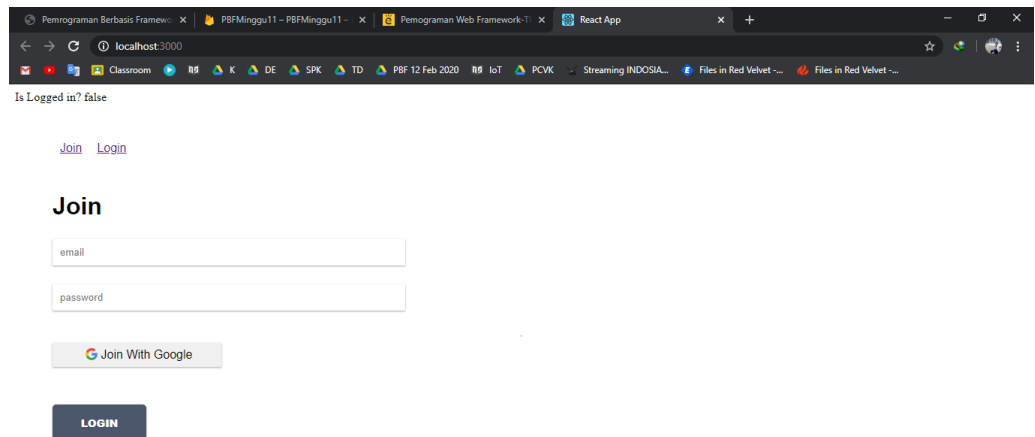
        <Switch>
          {routes.map(route => (
            <Route
              key = {route.path}
              path = {route.path}
              exact = {route.exact}
              component = {route.main}
            />
          ))}
        </Switch>
      </Router>
    </div>
  </AuthContext.Provider>
);
}

const rootElement = document.getElementById("root");
ReactDOM.render(<App />, rootElement);

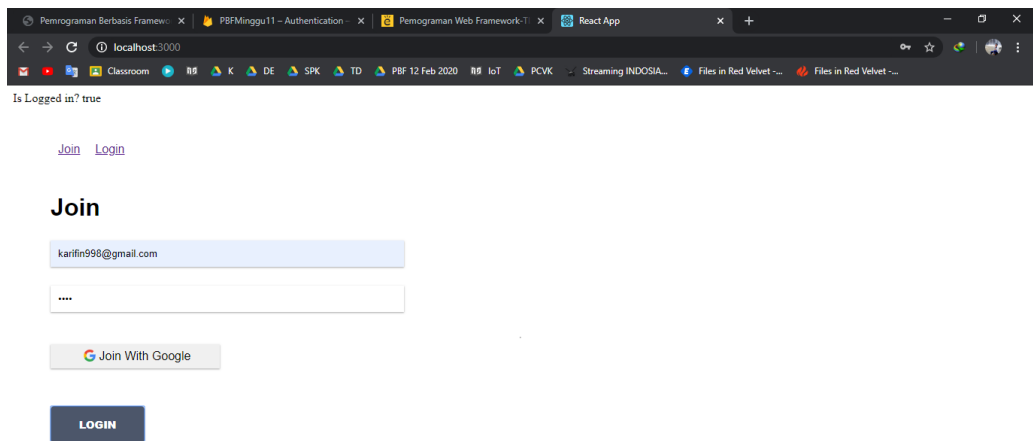
```

## 10. Run Program

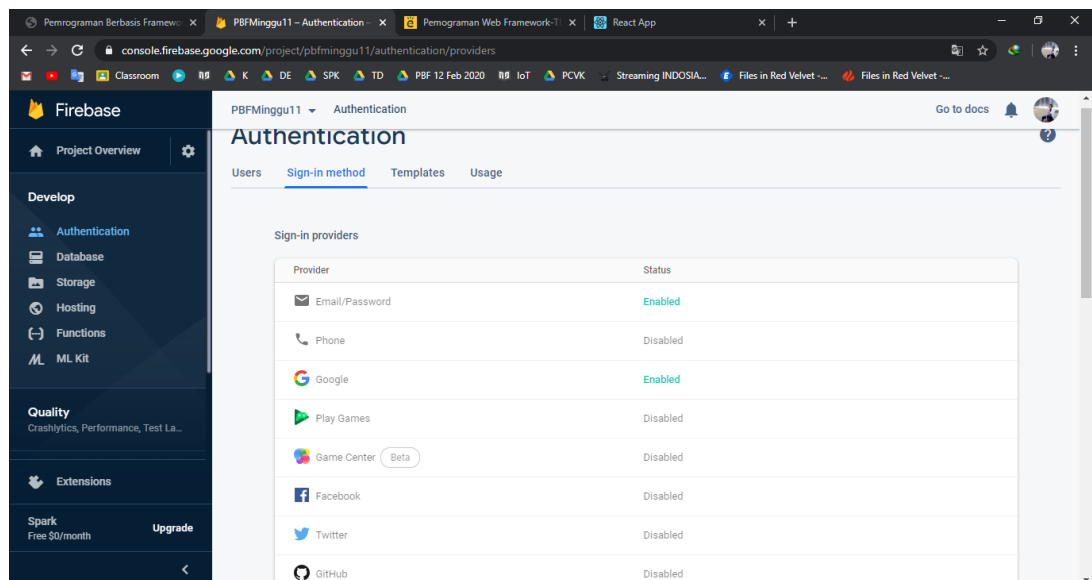
### a. Tampilan Awal



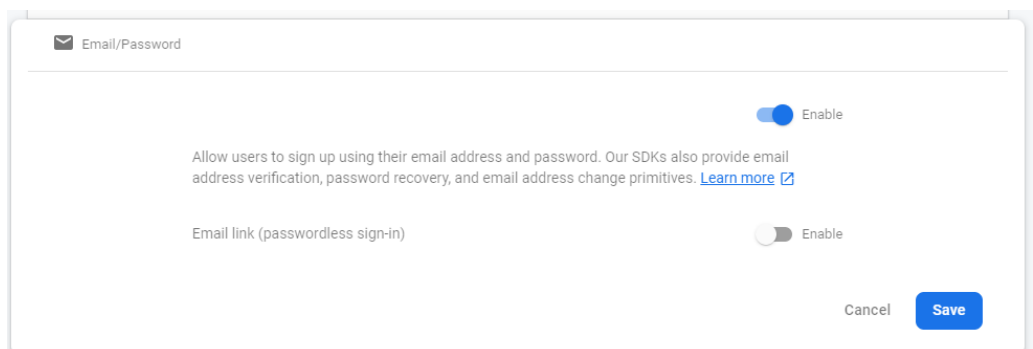
b. Setelah Diklik Join




11. Aktifkan Authentication Email dan Google



a. Email




## b. Google

 Google

Enable

Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the [SHA1 fingerprint](#) for each app on your [Project Settings](#).

 Update the [project-level setting](#) below to continue

Project public-facing name ⓘ

Project support email ⓘ

Whitelist client IDs from external projects (optional) ⓘ

Web SDK configuration ⓘ

## 12. Edit Join.js

```
import React, { useState, useContext } from "react";
import { AuthContext } from "../index";
import * as firebase from "firebase"

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

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

  return (
    <div>
      <h1>Join</h1>
      <form onSubmit = {e => handleForm(e)}>
        <input
          value = {email}
          onChange = {e => setEmail(e.target.value)}
          name = "email"
          type = "email"
          placeholder = "email"
        />
        <input
```

```

        onChange = {e => setPassword(e.target.value)}
        name = "password"
        value = {password}
        type = "password"
        placeholder = "password"
      />
    <hr />
    <button className = "googleBtn" type = "button">
      <img
        src =
"https://upload.wikimedia.org/wikipedia/commons/5/53/Google_%22G%2
2_Logo.svg"
        alt = "logo"
      />
      Join With Google
    </button>

    <button type = "submit"> Join </button>

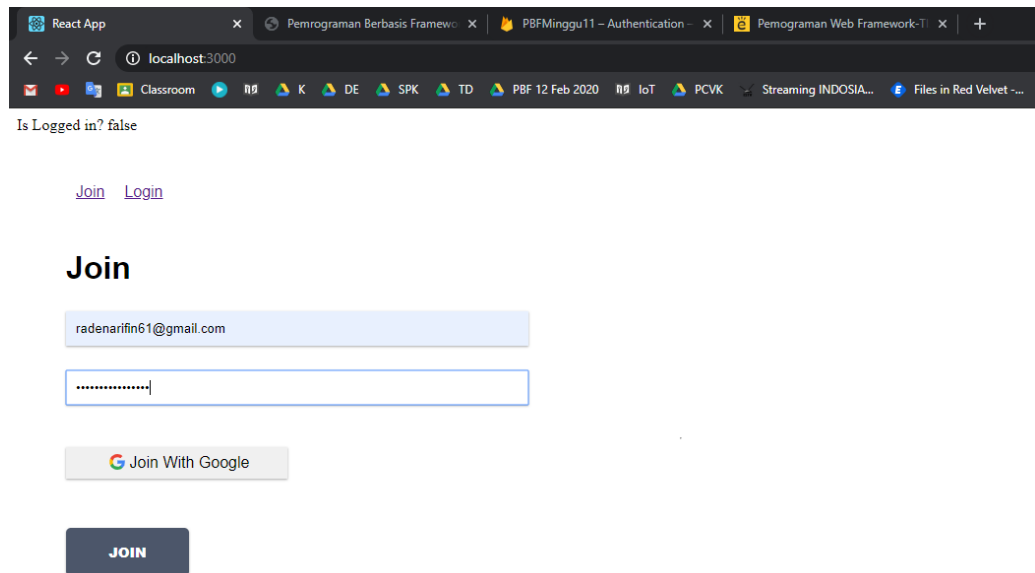
    <span>{error}</span>
  </form>
</div>
  );
};

export default Join;

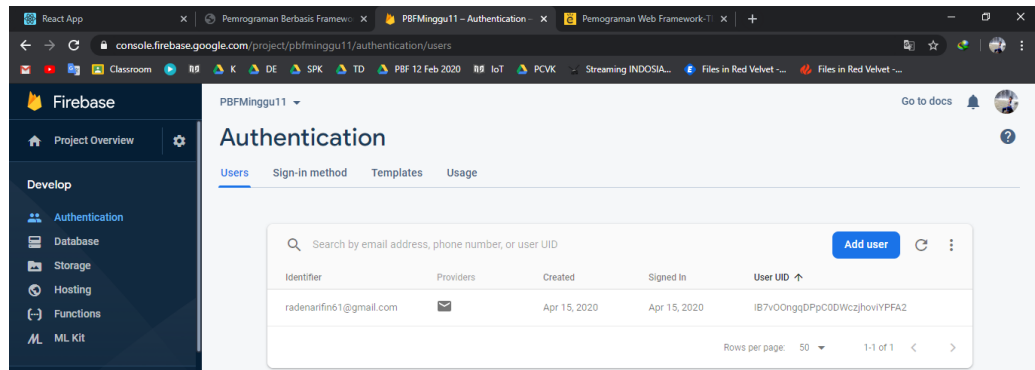
```

### 13. Klik Join pada Aplikasi

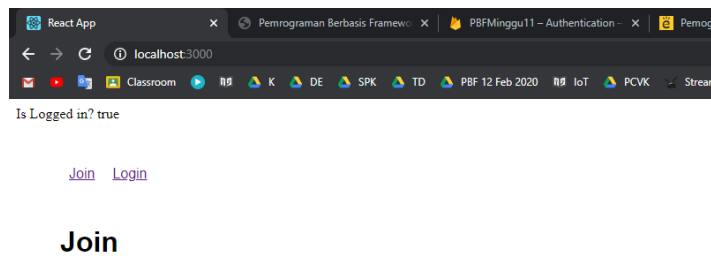
#### a. Klik Join



## b. Hasil Pada Firebase



## c. Hasil Setelah Klik Join



## 14. Ubah Login.js untuk Login

```
import React, { useState, useContext } from "react";
import { AuthContext } from "../index";
import * as firebase from "firebase";

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

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

  return (
    <div>
      <h1>Login</h1>
      <form onSubmit = {e => handleForm(e)}>
        <input
          value = {email}
          onChange = {e => setEmail(e.target.value)}
          name = "email"
        />
      </form>
    </div>
  );
};
```

```

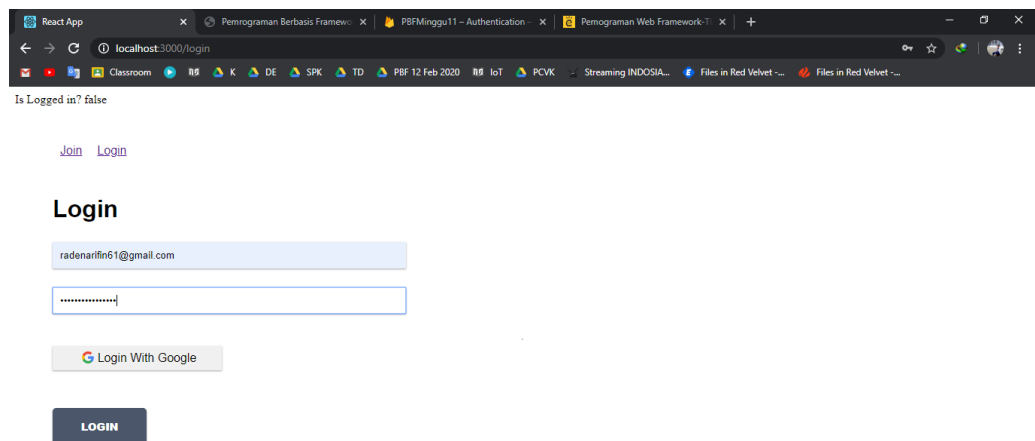
        type = "email"
        placeholder = "email"
      />
      <input
        onChange = {e => setPassword(e.target.value)}
        name = "password"
        value = {password}
        type = "password"
        placeholder = "password"
      />
      <hr />
      <button className = "googleBtn" type = "button">
        <img
          src =
            "https://upload.wikimedia.org/wikipedia/commons/5/53/Google_%22G%2
            2_Logo.svg"
          alt = "logo"
        />
        Login With Google
      </button>
      <button type = "submit"> Login </button>
      <span>{error}</span>
    </form>
  </div>
);
};

export default Login;

```

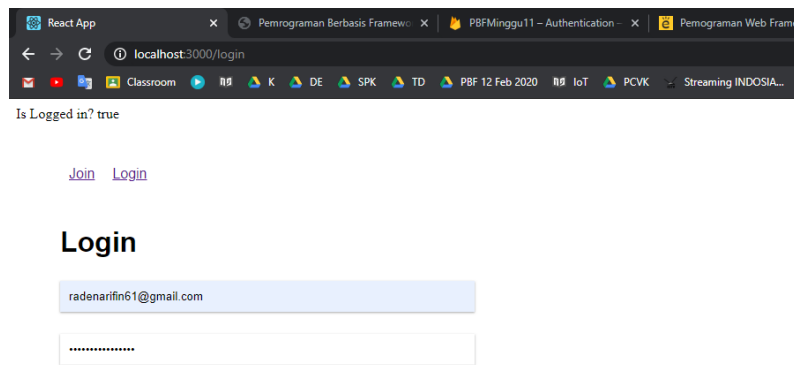
## 15. Klik Login

### a. Tampilan Awal





## b. Tampilan Setelah Login



## B. Menggunakan Akun Google

### 1. Mengubah Join.js

```
import React, { useState, useContext } from "react";
import { AuthContext } from "../index";
import * as firebase from "firebase"

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

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

  const handleGoogleLogin = () => {
    const provider = new firebase.auth.GoogleAuthProvider();

    firebase
      .auth()
      .signInWithPopup(provider)
      .then(res => {
        Auth.setLoggedIn(true)
      })
      .catch(e => {
        setErrors(e.message)
      });
  };

  return (
    <div>
      <h1>Join</h1>
    </div>
  );
};
```

```

        <form onSubmit = {e => handleForm(e)}>
          <input
            value = {email}
            onChange = {e => setEmail(e.target.value)}
            name = "email"
            type = "email"
            placeholder = "email"
          />
          <input
            onChange = {e => setPassword(e.target.value)}
            name = "password"
            value = {password}
            type = "password"
            placeholder = "password"
          />
          <hr />
          <button onClick={() => handleGoogleLogin()}
            className = "googleBtn" type = "button">
            <img
              src =
                "https://upload.wikimedia.org/wikipedia/commons/5/53/Google_%22G%
                22_Logo.svg"
              alt = "logo"
            />
            Join With Google
          </button>

          <button type = "submit"> Join </button>

          <span>{error}</span>
        </form>
      </div>
    );
  };
};

export default Join;

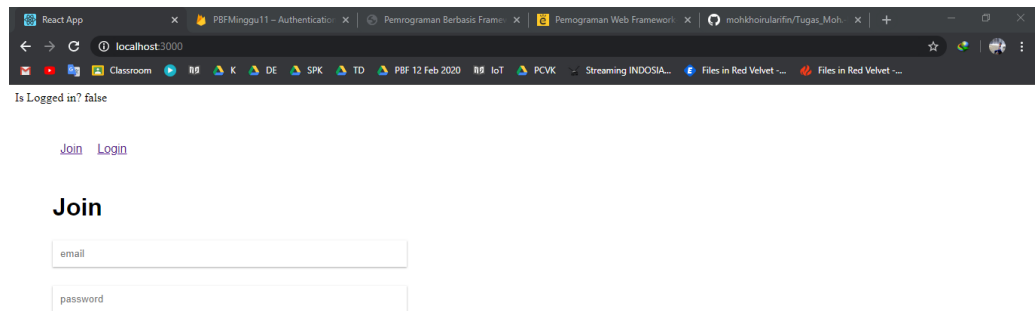
```

Pada code berwarna merah diatas merupakan fungsi untuk Join menggunakan Google. Terdapat fungsi untuk menghandel login menggunakan Google, terdapat provider yang berisi Authentication akun google. Dan SignInWithPopup untuk membuka window baru berupa popup untuk pemilihan akun.

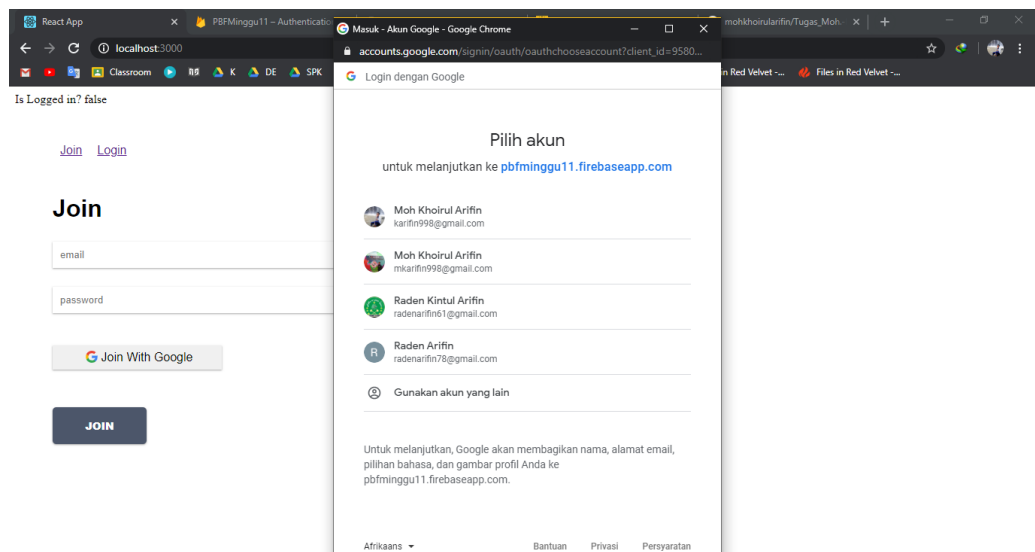
Handel tersebut akan dipanggil pada button, maka saat menekan button maka akan memproses handel.

## 2. Hasil

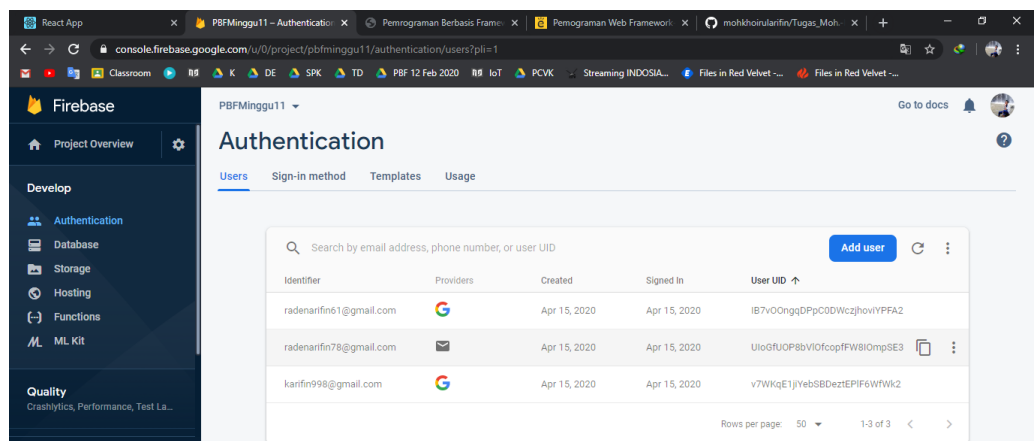
### a. Tampilan Awal



### b. Klik Tombol Join With Google dan Pilih Akun



### c. Tampilan FireBase



### d. Tampilan Setelah Join

