

**TUGAS PERORANGAN/INDIVIDU**

**LAPORAN JOBSHEET MINGGU 13**  
**FIREBASE REALTIME DATABASE**

**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. Langkah Persiapan

- a. Buat project reactjs baru.

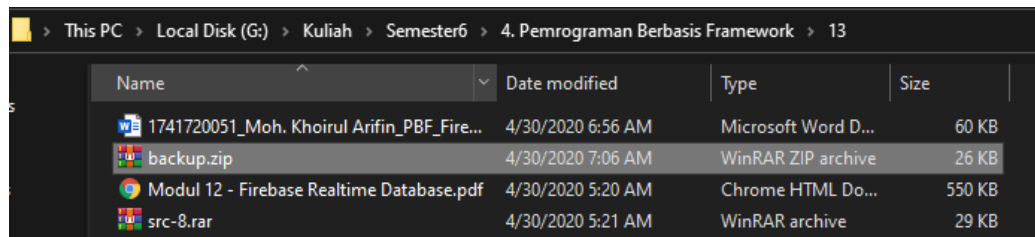
```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu13> npx create-react-app praktikum

Creating a new React app in G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu13\praktikum.

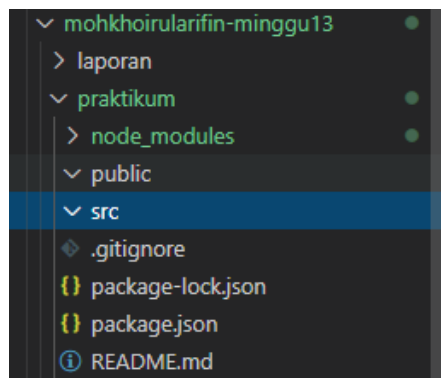
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

- b. Backup folder public dan src, kemudian hapus semua file dan folder yang ada di dalam folder public dan src.

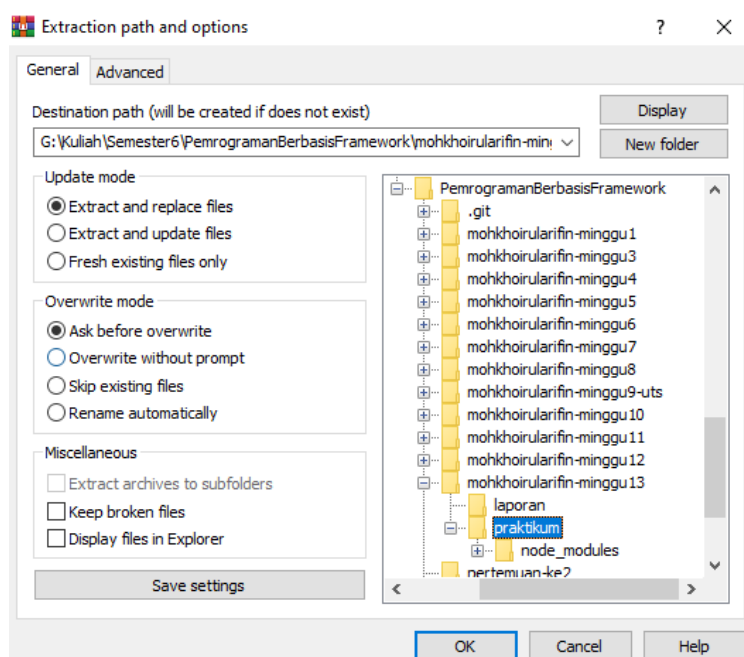
- 1) Backup Folder.



- 2) Hapus isi Folder.



- c. Ekstrak file Sorce Modul 8.rar ke dalam project reactjs.



d. Install beberapa package yang dibutuhkan.

1) npm install react-router-dom

```
PS G:\Kuliah\Semester6\PemrogramanBasisFramework\mohkhoirularifin-minggu13\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"})
```

2) npm install --save firebase

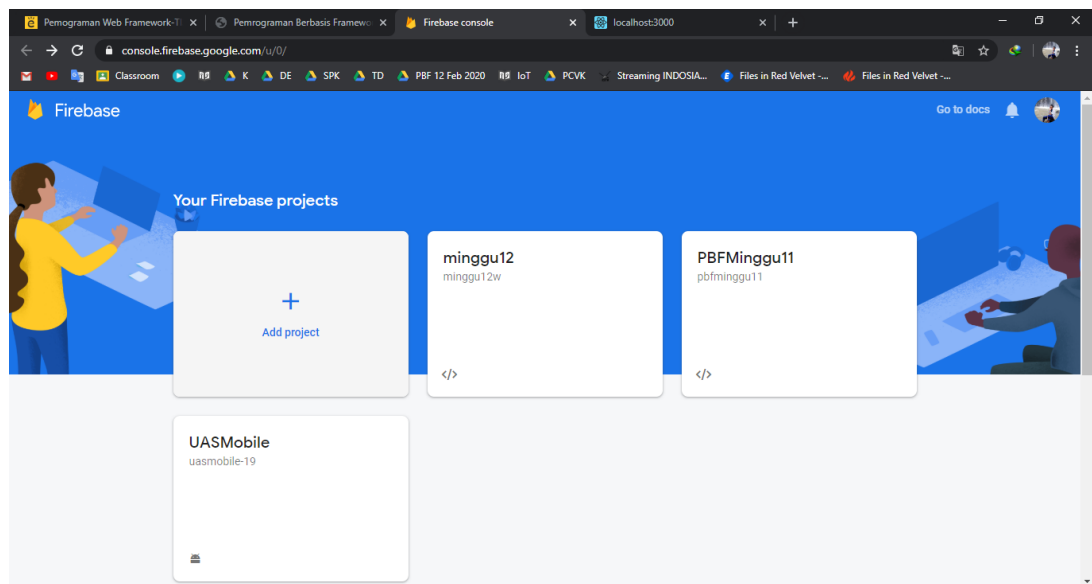
```
PS G:\Kuliah\Semester6\PemrogramanBasisFramework\mohkhoirularifin-minggu13\praktikum> npm install --save firebase
> protobufjs@6.9.0 postinstall G:\Kuliah\Semester6\PemrogramanBasisFramework\mohkhoirularifin-minggu13\praktikum\node_modules\protobufjs
> node scripts/postinstall

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.
```

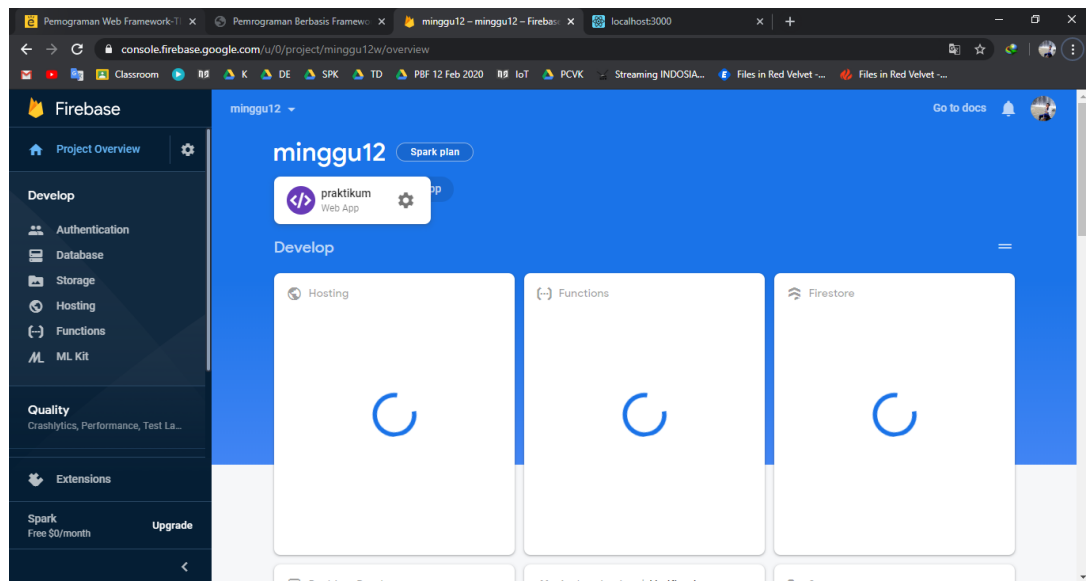
3) npm install bootstrap

```
PS G:\Kuliah\Semester6\PemrogramanBasisFramework\mohkhoirularifin-minggu13\praktikum> npm install bootstrap
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 bootstrap@4.4.1 requires a peer of jquery@1.9.1 - 3 but none is installed. You must install peer dependencies yourself.
npm WARN bootstrap@4.4.1 requires a peer of popper.js@^1.16.0 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"})
```

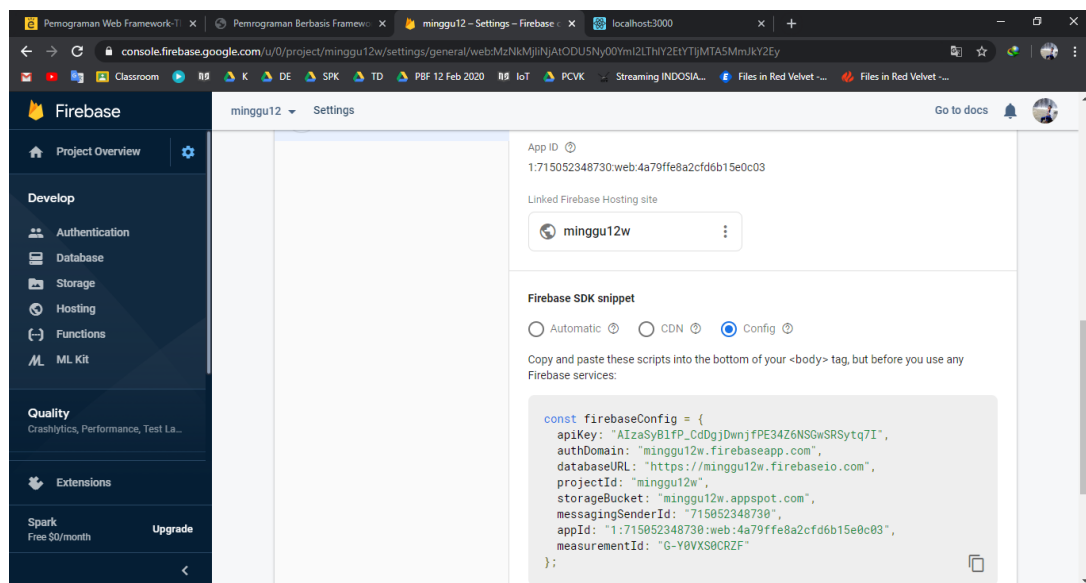
e. Masuk project firebase konsol. Karena minggu kemarin saya membuat project dengan nama minggu12, maka saya memilih project minggu12.



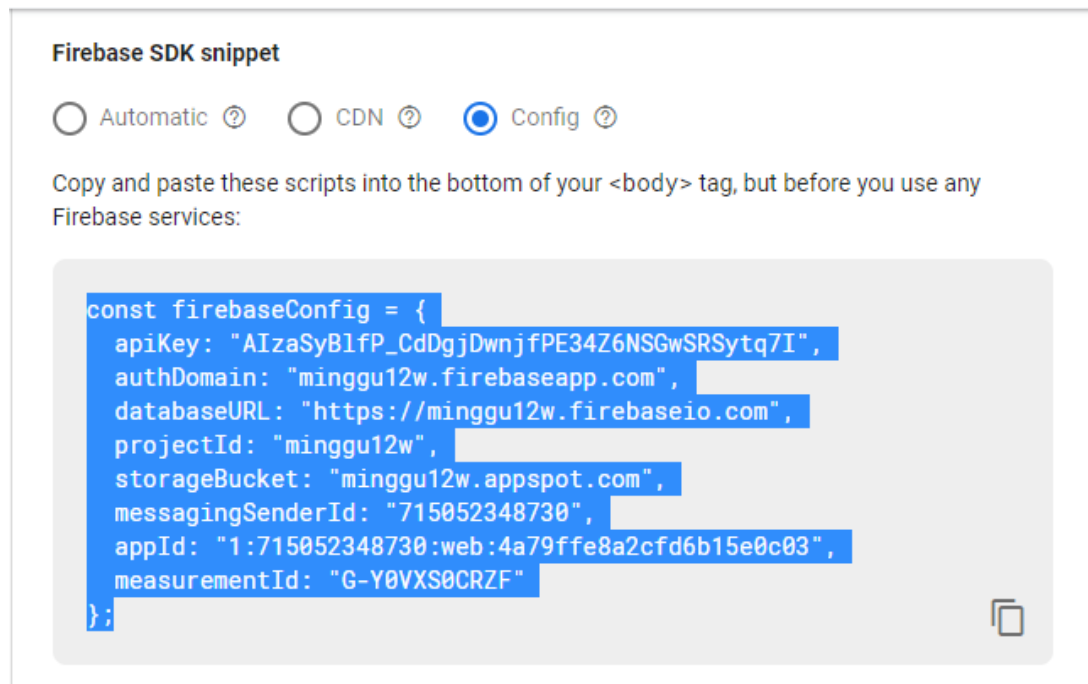
- f. Klik “Project Overview”, dan pilih app praktikum dan klik icon setting.



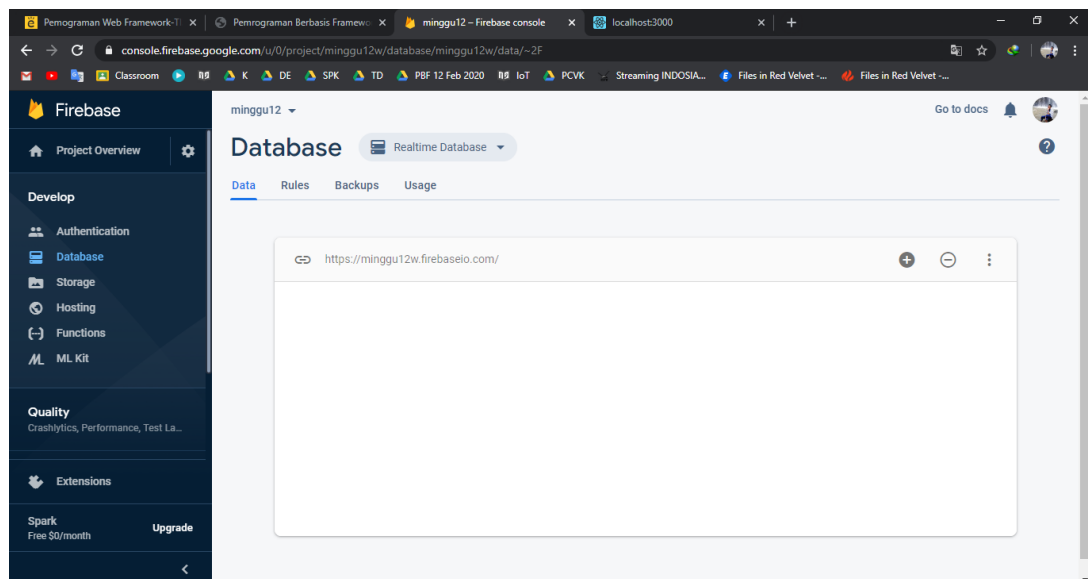
- g. Pada tab umum/common di menu Setting, scroll ke bawah, dan pilih opsi config pada Firebase SDK snippet.



- h. Konfigurasi inilah yang akan terus kita gunakan saat berinteraksi dengan API Firebase yang telah kita buat.



- i. Masuk pada menu database pada firebase. Pada materi kali ini kita menggunakan real time database firebase jadi pilih menu Real time database.



## 2. Langkah Praktikum

- a. Membuat file dan folder firebase/config.js dalam folder src.

```
const firebaseConfig = {
  apiKey: "AIzaSyBlfP_CdDgjDwnjfPE34Z6NSGwSRStq7I",
  authDomain: "minggul2w.firebaseio.com",
  databaseURL: "https://minggul2w.firebaseio.com",
  projectId: "minggul2w",
  storageBucket: "minggul2w.appspot.com",
  messagingSenderId: "715052348730",
  appId: "1:715052348730:web:4a79ffe8a2cfd6b15e0c03",
  measurementId: "G-Y0VXS0CRZF"
};

export default firebaseConfig;
```

- b. Modifikasi statefull component BlogPost.jsx.

```
import React, {Component} from "react";
import './BlogPost.css';
import Post from "../../component/BlogPost/Post";
// import API from "../../services";
import firebase from "firebase";
import firebaseConfig from "../../firebase/config";

class BlogPost extends Component{
  constructor(props) {
    super(props);
    firebase.initializeApp(firebaseConfig);    //
    Inisialisasi Konfigurasi Database Firebase

    this.state = {                                // Komponen
      State dari React Untuk Statefull Component
      listArtikel: []                             // Variabel
      Array yang Digunakan untuk Menyimpan data API
    }
  }

  ambilDataDariServerAPI = () => {                // fungsi
    untuk mengambil data API dari Realtime Database Firebase
    let ref = firebase.database().ref("/");
    ref.on("value", snapshot => {
      const state = snapshot.val();
      this.setState(state);
    });
  }

  simpanDataKeServerAPI = () => {                // Fungsi
    untuk Mengirim/Insert Data ke API Realtime Database Firebase
    firebase.database()
      .ref("/")
      .set(this.state);
  }

  componentDidMount() {                          // Komponen
    untuk Mengecek Ketika Component telah di mount-ing, maka panggil
    API
    this.ambilDataDariServerAPI()                // ambil data
    dari server API lokal
  }
}
```

```

componentDidUpdate(prevProps, prevState) {
  if (prevState !== this.state) {
    this.simpanDataKeServerAPI();
  }
}

handleHapusArtikel = (idArtikel) => { // fungsi
yang meng-handle button action hapus data
  const {listArtikel} = this.state;
  const newState = listArtikel.filter(data => {
    return data.uid !== idArtikel;
  });
  this.setState({listArtikel: newState});
}

handleTombolSimpan = (event) => { // fungsi
untuk meng-handle tombol submit di klik
  let title = this.refs.judulArtikel.value; // this.refs
mengacu pada input field (input text, textarea, dll)
  let body = this.refs.isiArtikel.value;
  let uid = this.refs.uid.value;

  if (uid && title && body) { // cek apakah
semua data memiliki nilai (tidak null)
    const { listArtikel } = this.state;
    const indeksArtikel = listArtikel.findIndex(data => {
      return data.uid === uid;
    });
    listArtikel[indeksArtikel].title = title;
    listArtikel[indeksArtikel].body = body;
    this.setState({ listArtikel });
  } else if (title && body) { // Jika data
belum ada di server
    const uid = new Date().getTime().toString();
    const { listArtikel } = this.state;
    listArtikel.push({ uid, title, body });
    this.setState({ listArtikel });
  }

  this.refs.judulArtikel.value = "";
  this.refs.isiArtikel.value = "";
  this.refs.uid.value = "";
};

render() {
  return(
    <div className="post-artikel">
      <div className="form pb-2 border-bottom">
        <div className="form-group row">
          <label htmlFor="title" className="col-sm-2 col-form-label">Judul</label>
          <div className="col-sm-10">
            <input type="text" className="form-control" id="title" name="title" ref="judulArtikel"/>
          </div>
        </div>
        <div className="form-group row">
          <label htmlFor="body" className="col-sm-2 col-form-label">Isi</label>
          <div className="col-sm-10">

```

```

                                <textarea className="form-control"
id="body" name="body" rows="3" ref="isiArtikel"></textarea>
                                </div>
                                </div>
                                <input type="hidden" name="uid" ref="uid" />
                                <button type="submit" className="btn btn-
primary" onClick={this.handleTombolSimpan}>Simpan</button>
                                </div>
                                <h2>Daftar Artikel</h2>
                                { // Looping dan masukan untuk setiap data yang
                                ada di listArtikel ke variabel artikel
                                  this.state.listArtikel.map(artikel => {
                                    return <Post key={artikel.uid}
                                judul={artikel.title} isi={artikel.body}
                                idArtikel={artikel.uid}
                                hapusArtikel={this.handleHapusArtikel}/>
                                  })
                                }
                                </div>
                                )
                                }
                                }
                                export default BlogPost;

```

### c. Modifikasi stateless component Post.jsx.

```

import React from "react";

const Post = (props) => {
  return (
    <div className="artikel">
      <div className="gambar-artikel">
        
      </div>
      <div className="konten-artikel">
        <div className="judul-
artikel">{props.judul}</div>
        <p className="isi-artikel">{props.isi}</p>
        <button className="btn btn-sm btn-warning"
          onClick={() => { if (window.confirm('Apakah
anda yakin menghapus atikel ini?'))
props.hapusArtikel(props.idArtikel) }}>
          Hapus
        </button>
      </div>
    </div>
  )
}

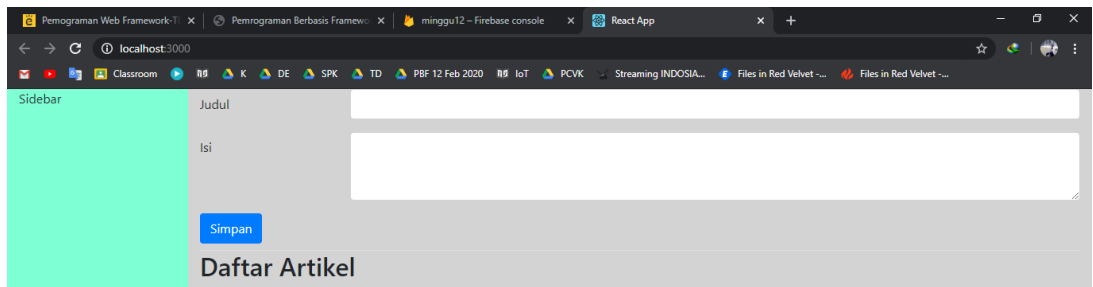
export default Post;

```

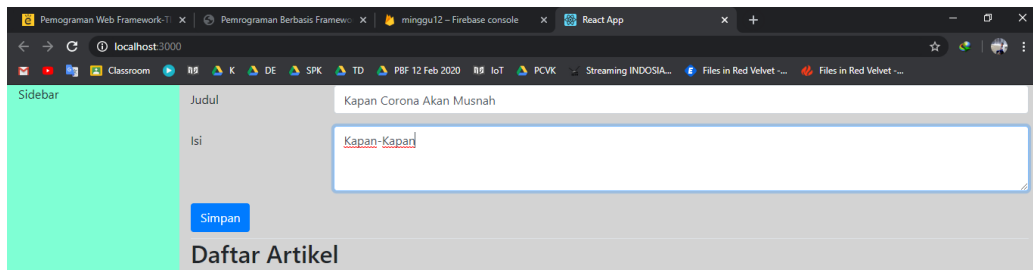


d. Lakukan proses insert data pada browser.

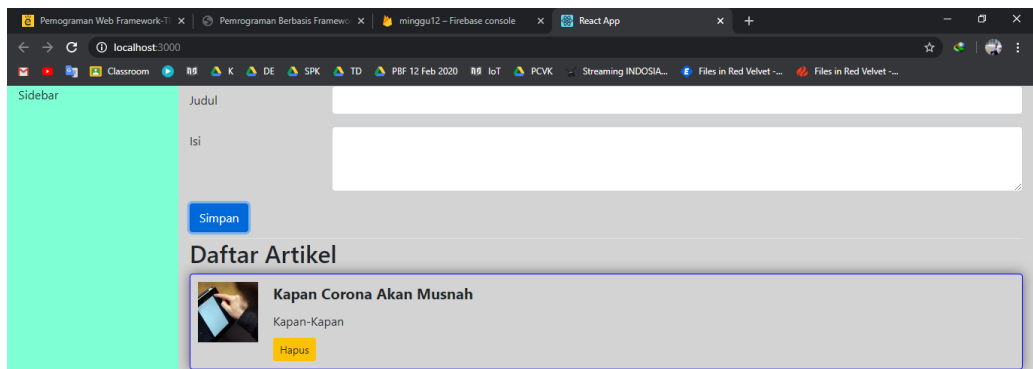
### 1) Tampilan Awal



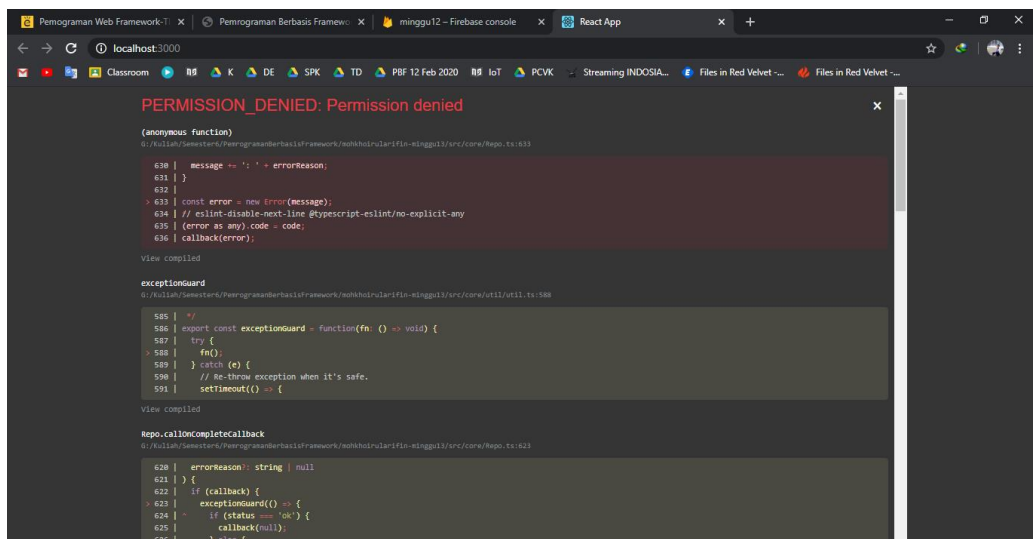
### 2) Proses Penyimpanan



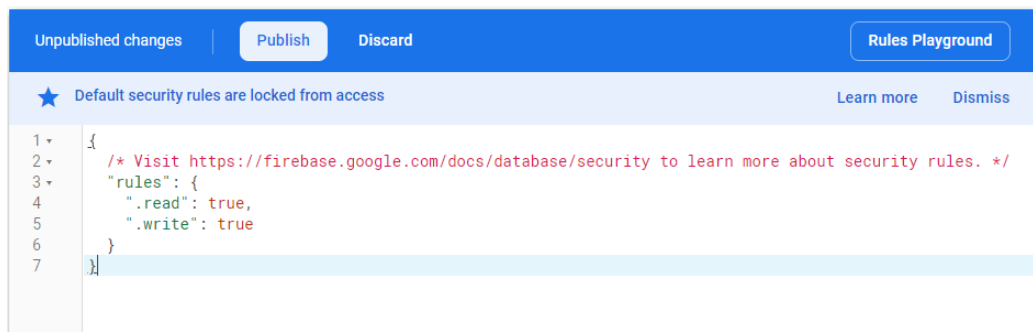
Pada awal-awal proses berjalan dengan lancar



Namun setelah kurang lebih 2 detik terjadi error seperti gambar berikut

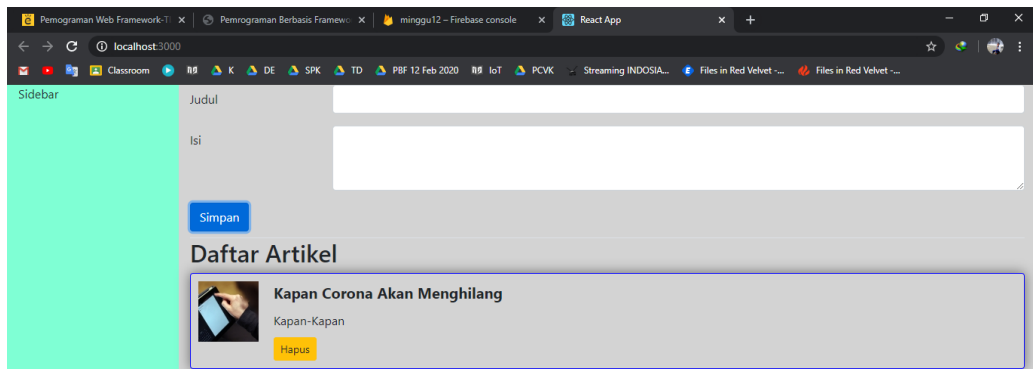


Setelah ditelusuri ternyata pada Rule Database ada yang diubah menjadi true.

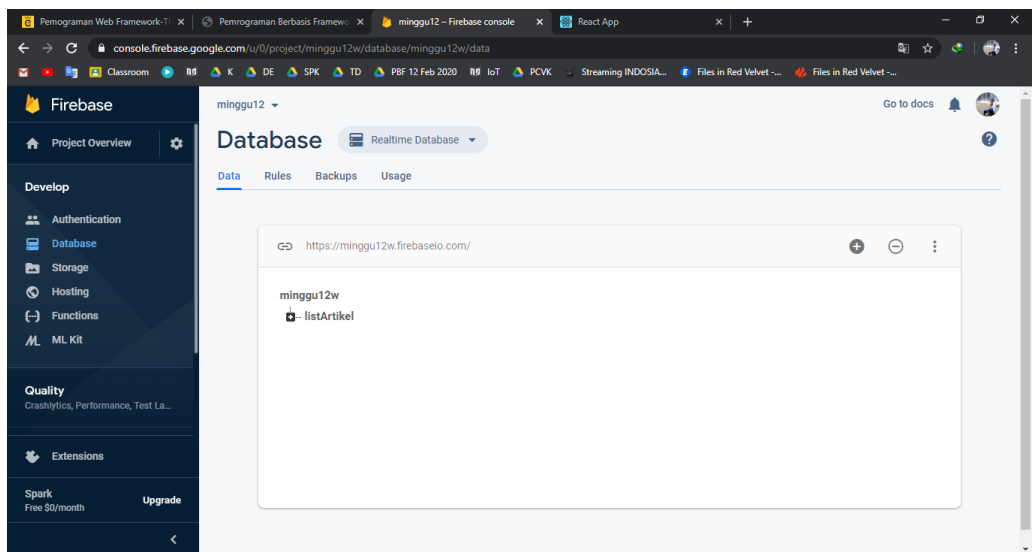


```
1 {
2   /* Visit https://firebase.google.com/docs/database/security to learn more about security rules. */
3   "rules": {
4     ".read": true,
5     ".write": true
6   }
7 }
```

Setelah diubah Rule nya dan melakukan pengisian lagi akan tampil seperti ini



### 3) Tampilan Database Firebase setelah proses penyimpanann

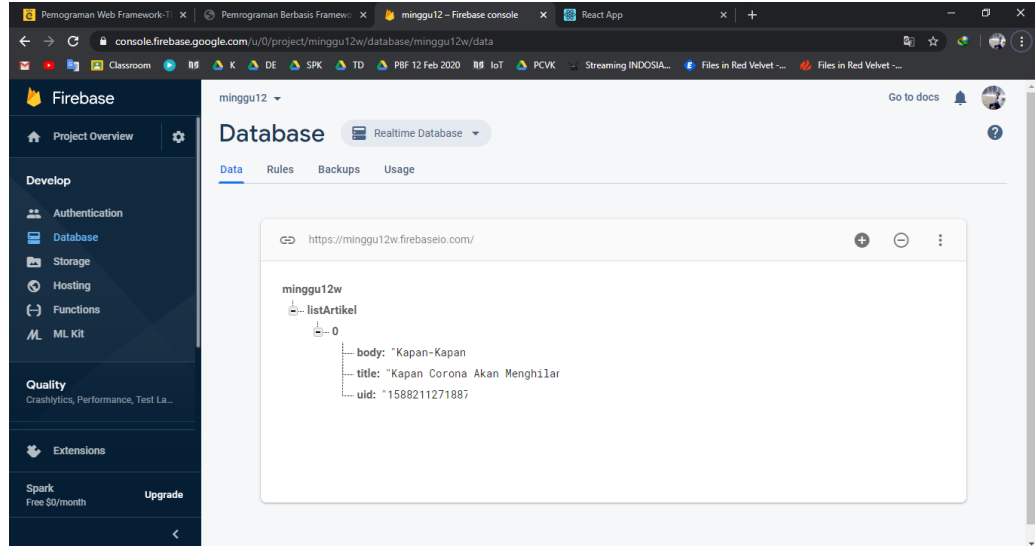


### 3. Pertanyaan Praktikum

- a. Perhatikan file BlogPost.jsx, apa saja kode program yang berubah dari BlogPost.js pada Modul-8 dengan BlogPost.jsx pada praktikum kali ini? Kenapa?
- Penghapusan import API  
Import API dihapus karena pada praktikum sudah tidak menggunakan API, melainkan menggunakan firebase.
  - Perubahan beberapa fungsi  
Perubahan fungsi disini mengacu pada perubahan fungsi yang sebelumnya menggunakan api diubah menjadi fungsi yang menggunakan Firebase.
  - Penambahan simpanDataKeServerAPI  
Fungsi ini ditujukan untuk mengirimkan data yang diinputkan ke database firebase.
  - Penambahan componentDidUpdate  
Pada komponen ini dicek apakah data sama dengan data yang sudah ada pada database, jika data tidak sama maka akan diteruskan ke dalam fungsi simpanDataKeServerAPI.
- b. Perhatikan file Post.jsx, apa saja kode program yang berubah dari Post.js pada Modul-8 dengan Post.jsx pada praktikum kali ini? Kenapa?
- Pada file Post.jsx tidak terlalu banyak perubahan. Yaitu terdapat penambahan munculnya window baru berupa window konfirmasi apakah user mau menghapus.
- c. Apakah Global API service yang kita buat pada Modul-8 kemarin kita pakai lagi pada praktikum kali ini? Kenapa alasannya?
- Pada praktikum ini global API sudah tidak digunakan lagi, karena semua proses diarahkan pada Firebase. Dengan bukti bahwa pada file terdapat import firebase dan menghapus import API.

d. Data yang kita insert bertambah, dan saat kita refresh browser, data masih tetap ada. Dimanakah data-data artikel tersebut disimpan? Tunjukkan hasil screenshot data disimpan tersebut.

- Data-data tersebut akan disimpan dalam database firebase yang telah dibuat.



e. Menurut kalian lebih mudah dan lebih praktis mana aplikasi reactjs menggunakan data API sendiri (seperti Modul-4 dan Modul-8) atau menggunakan Firebase realtime database? Berika alasannya.

- Menurut saya akan lebih mudah dan lebih praktis menggunakan firebase. Dengan menggunakan firebase kita tidak perlu membuka localhost baru untuk mengaksesnya seperti yang dilakukan API.