

**TUGAS PERORANGAN/INDIVIDU**

## **LAPORAN TUGAS MINGGU 13**

Disusun sebagai

**MATA KULIAH : PEMROGRAMAN BERBASIS FRAMEWORK**

Oleh :

**Izmir Ta'is Khasna Lugazi**

**1741720180**

Kelas :

**3B**



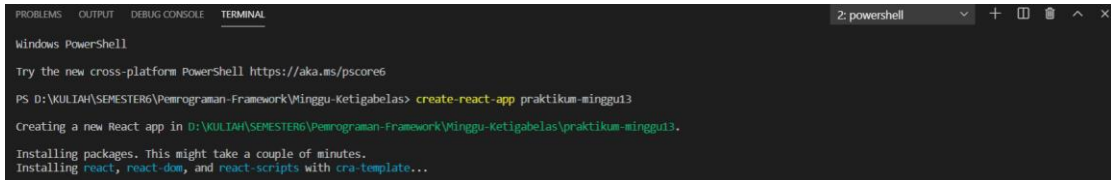
**PROGRAM STUDI D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2020**

# Praktikum

## CRUD pada Firebase

### 1.1 Langkah Persiapan

1. Buat project react js



```
Windows PowerShell

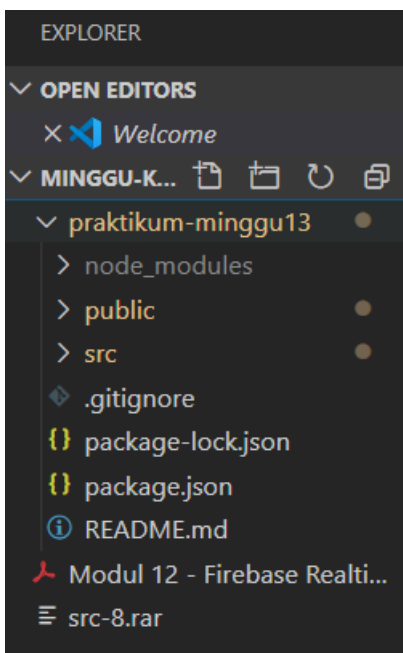
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\KULIAH\SEMESTER6\Pemrograman-Framework\Minggu-Ketigabelas> create-react-app praktikum-minggu13

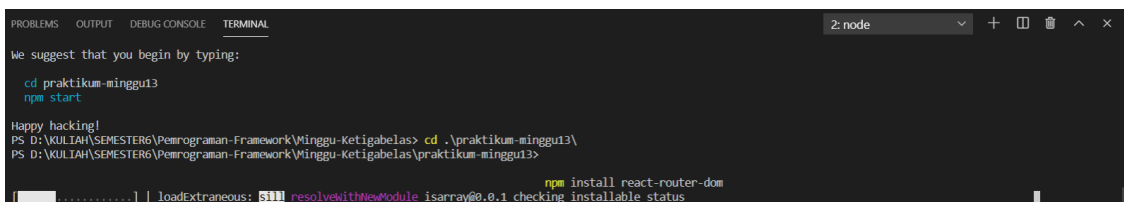
Creating a new React app in D:\KULIAH\SEMESTER6\Pemrograman-Framework\Minggu-Ketigabelas\praktikum-minggu13.

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

2. Backup folder public dan src yang telah di download



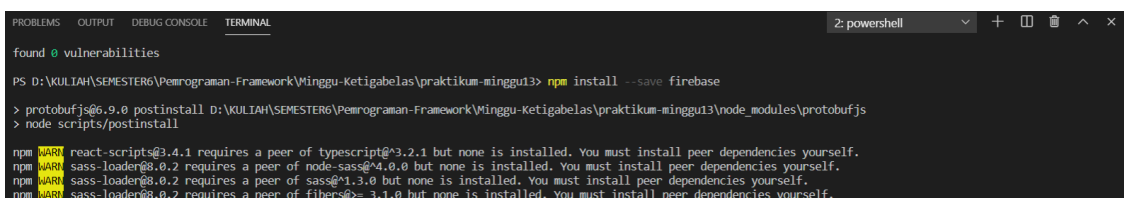
3. Install package yang dibutuhkan melalui CMD atau terminal
  - a. Npm install react-router-dom



```
PS D:\KULIAH\SEMESTER6\Pemrograman-Framework\Minggu-Ketigabelas> cd .\praktikum-minggu13\
PS D:\KULIAH\SEMESTER6\Pemrograman-Framework\Minggu-Ketigabelas\praktikum-minggu13> npm install react-router-dom

[ ] loadExtraneous: 5111 resolveWithNewModule isarray@0.0.1 checking installable status
```

- b. Npm install --save firebase



```
PS D:\KULIAH\SEMESTER6\Pemrograman-Framework\Minggu-Ketigabelas\praktikum-minggu13> npm install --save firebase

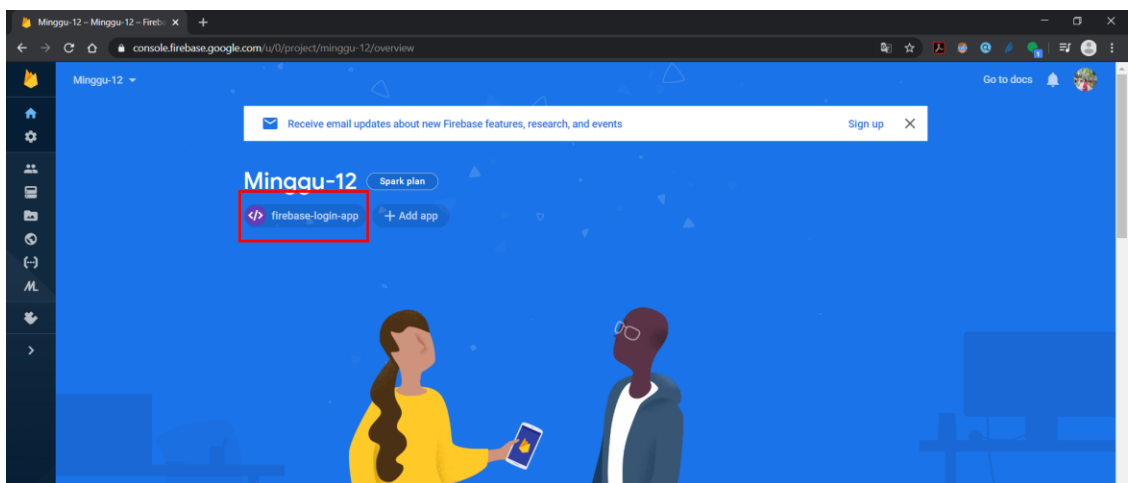
> protobufjs@6.9.0 postinstall D:\KULIAH\SEMESTER6\Pemrograman-Framework\Minggu-Ketigabelas\praktikum-minggu13\node_modules\protobufjs
> node scripts/postinstall

npm WARN react-scripts@3.4.1 requires a peer of typescript@^3.2.1 but none is installed. You must install peer dependencies yourself.
npm WARN sass-loader@8.0.2 requires a peer of node-sass@^4.0.0 but none is installed. You must install peer dependencies yourself.
npm WARN sass-loader@8.0.2 requires a peer of sass@^1.3.0 but none is installed. You must install peer dependencies yourself.
npm WARN sass-loader@8.0.2 requires a peer of fibers@>= 3.1.0 but none is installed. You must install peer dependencies yourself.
```

### c. Npm install bootstrap

```
PS D:\KULIAH\SEMESTER6\Penrograman-Framework\Minggu-Ketigabelas\praktikum-minggu13> npm install bootstrap
npm WARN react-scripts@0.4.1 requires a peer of typescript@>=2.1 but none is installed. You must install peer dependencies yourself.
npm WARN sass-loader@8.0.2 requires a peer of node-sass@>=4.0.0 but none is installed. You must install peer dependencies yourself.
npm WARN sass-loader@8.0.2 requires a peer of sass@>=1.3.0 but none is installed. You must install peer dependencies yourself.
npm WARN sass-loader@8.0.2 requires a peer of fibers@>= 3.1.0 but none is installed. You must install peer dependencies yourself.
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-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"})
```

4. Pastikan tidak lupa konfigurasi dari aplikasi firebase yang telah dibuat. jika lupa bisa masuk project firebase konsol di <https://console.firebase.google.com>
5. Kemudian klik “ringkasan project”, dan pilih app yang sudah dibuat. Sesuai dengan materi sebelumnya adalah app “firebase-login-app”, kemudian klik setting.



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

#### Firebase SDK snippet



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
&ltscript src="/__/firebase/7.14.2/firebase-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
https://firebase.google.com/docs/web/setup#available-libraries
&ltscript src="/__/firebase/7.14.2/firebase-analytics.js"></script>

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

This snippet uses [SDK auto-configuration in Firebase Hosting](#).

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

#### Firebase SDK snippet

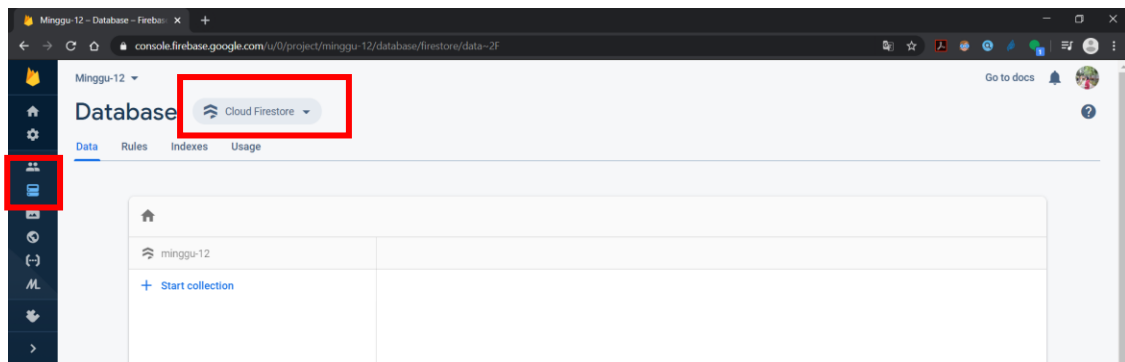
☐ Automatic ☐ CDN ☒ Config

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

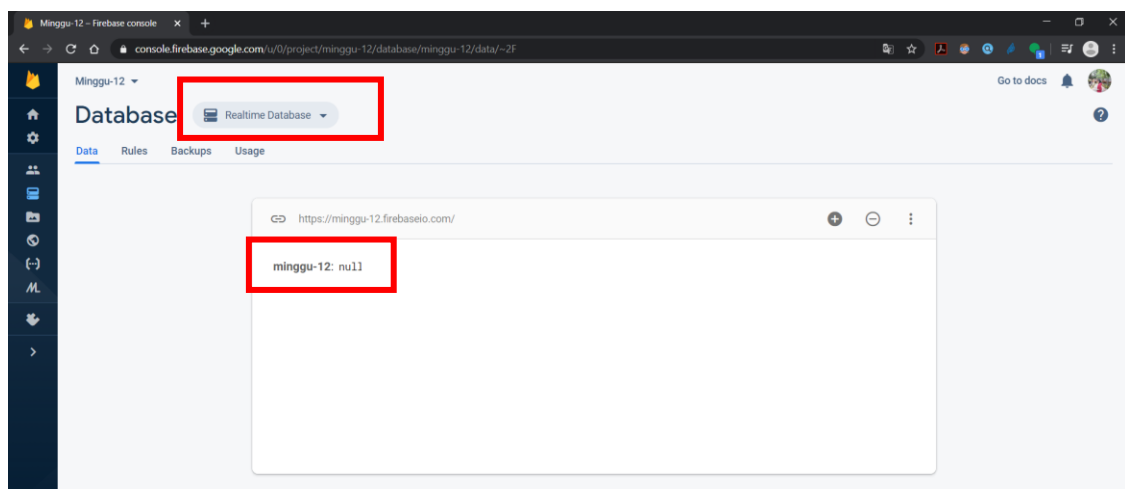
```
const firebaseConfig = {
  apiKey: "AIzaSyAiRs_ov91081pgR00p_YowyDZeBjq-xqQ",
  authDomain: "minggu-12.firebaseio.com",
  databaseURL: "https://minggu-12.firebaseio.com",
  projectId: "minggu-12",
  storageBucket: "minggu-12.appspot.com",
  messagingSenderId: "208265740640",
  appId: "1:208265740640:web:2576b536d198cc29e21e05",
  measurementId: "G-8T5RFD81H"
};
```

[Remove this app](#)

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



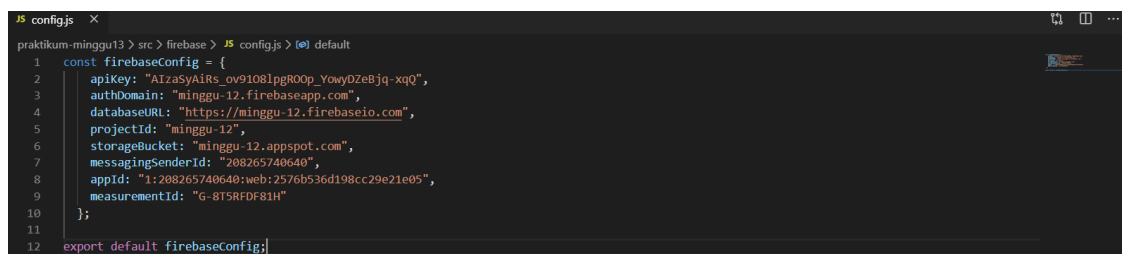
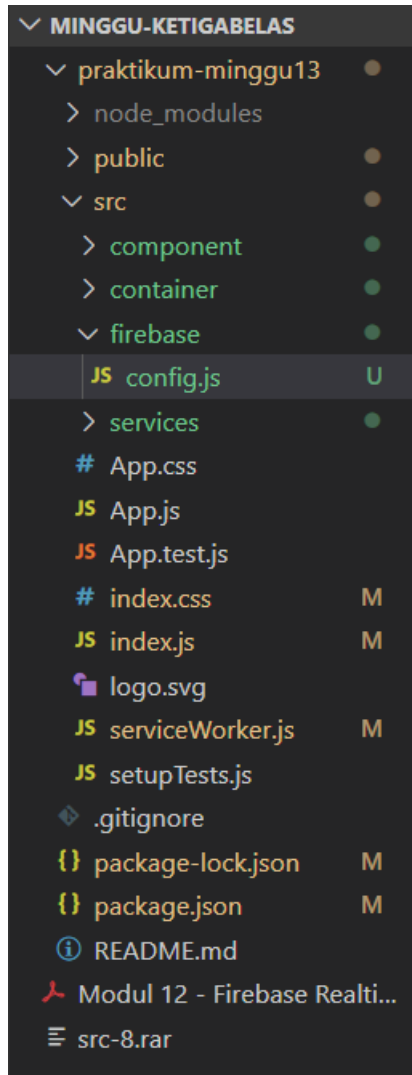
9. Seperti Gambar berikut tampilan database real time firebase yang akan kita gunakan



10. Langkah Persiapan telah selesai. Kita mulai praktikumnya.

## 1.2 Langkah Praktikum

11. Pada project ReactJS kalian, buat file dan folder firebase/config.js dalam folder src dan simpan konfigurasi code program di atas (seperti Langkah 8) pada file config.js.



12. Kita modifikasi statefull component BlogPost.jsx yang dulu pernah kita kerjakan, menjadi seperti ini

```
BlogPost.jsx X
praktikum-minggu13 > src > container > BlogPost > BlogPost.jsx > BlogPost > render
1  import React, {Component} from "react";
2  import './BlogPost.css';
3  import Post from "../../component/BlogPost/Post";
4  // import API from "../../services";
5  import firebase from "firebase";
6  import firebaseConfig from "../../firebase/config";
7
8  class BlogPost extends Component{
9    constructor(props) {
10      super(props);
11      firebase.initializeApp(firebaseConfig); // inisialisasi konfigurasi database firebase
12
13      this.state = {           // komponen state dari React untuk statefull component
14        listArtikel: []       // variabel array yang digunakan untuk menyimpan data API
15      }
16    }
17
18    ambilDataDariServerAPI = () => {           // fungsi untuk mengambil data API dari Realtime Database Firebase
19      let ref = firebase.database().ref("/");
20      ref.on("value", snapshot => {
21        const state = snapshot.val();
22        this.setState(state);
23      });
24    }
25
26    simpanDataDariServerAPI = () => { // fungsi untuk mengirim/insert data ke API Realtime Database Firebase
27      firebase.database()
28        .ref("/")
29        .set(this.state);
30    }
31
32    componentDidMount() {           // komponen untuk mengecek ketika component telah di-mount-ing, maka panggil API
33      this.ambilDataDariServerAPI() // ambil data dari server API lokal
34    }
35  }
```

```
BlogPost.jsx X
praktikum-minggu13 > src > container > BlogPost > BlogPost.jsx > BlogPost > componentDidMount
35
36  componentDidUpdate(prevProps, prevState) {
37    if (prevState !== this.state) {
38      this.simpanDataDariServerAPI();
39    }
40  }
41
42  handlehapusArtikel = (idArtikel) => {           // fungsi yang meng-handle button action hapus data
43    const {listArtikel} = this.state;
44    const newState = listArtikel.filter(data => {
45      return data.uid !== idArtikel;
46    });
47    this.setState({listArtikel: newState});
48  }
49
50  handleTombolSimpan = (event) => {           // fungsi untuk meng-handle tombol submit di klik
51    let title = this.refs.judulArtikel.value; // this.refs mengacu pada input field (input text, textarea, dll)
52    let body = this.refs.isiArtikel.value;
53    let uid = this.refs.uid.value;
54
55    if (uid && title && body) { // Cek apakah semua data memiliki nilai (tidak null)
56      const { listArtikel } = this.state;
57      const indeksArtikel = listArtikel.findIndex(data => {
58        return data.uid === uid;
59      });
60      listArtikel[indeksArtikel].title = title;
61      listArtikel[indeksArtikel].body = body;
62      this.setState({ listArtikel });
63    } else if (title && body) { // jika data belum ada di server
64      const uid = new Date().getTime().toString();
65      const { listArtikel } = this.state;
66      listArtikel.push({ uid, title, body });
67      this.setState({ listArtikel });
68    }
69  }
```

```

BlogPost.jsx
praktikum-minggu13 > src > container > BlogPost > BlogPost.jsx > BlogPost > handleTombolSimp
69
70   this.refs.judulArtikel.value = "";
71   this.refs.isiArtikel.value = "";
72   this.refs.uid.value = "";
73 };
74
75 render() {
76   return(
77     <div className="post-artikel">
78       <div className="form pb-2 border-bottom">
79         <div className="form-group row">
80           <label htmlFor="title" className="col-sm-2 col-form-label">Judul</label>
81           <div className="col-sm-10">
82             <input type="text" className="form-control" id="title" name="title" ref="judulArtikel"/>
83           </div>
84         </div>
85         <div className="form-group row">
86           <label htmlFor="body" className="col-sm-2 col-form-label">Isi</label>
87           <div className="col-sm-10">
88             <textarea className="form-control" id="body" name="body" rows="3" ref="isiArtikel"></textarea>
89           </div>
90         </div>
91         <input type="hidden" name="uid" ref="uid" />
92         <button type="submit" className="btn btn-primary" onClick={this.handleTombolSimp}>Simpan</button>
93       </div>
94       <h2>Daftar Artikel</h2> { // looping dan masukkan untuk setiap data yang ada di listArtikel ke variabel artikel
95         this.state.listArtikel.map(artikel => {
96           return <Post key={artikel.id} judul={artikel.title} isi={artikel.body}
97             idArtikel={artikel.id} hapusArtikel={this.handleHapusArtikel}/>
98         })
99       </div>
100     </div>
101   )
102 }
103
104
105 export default BlogPost;

```

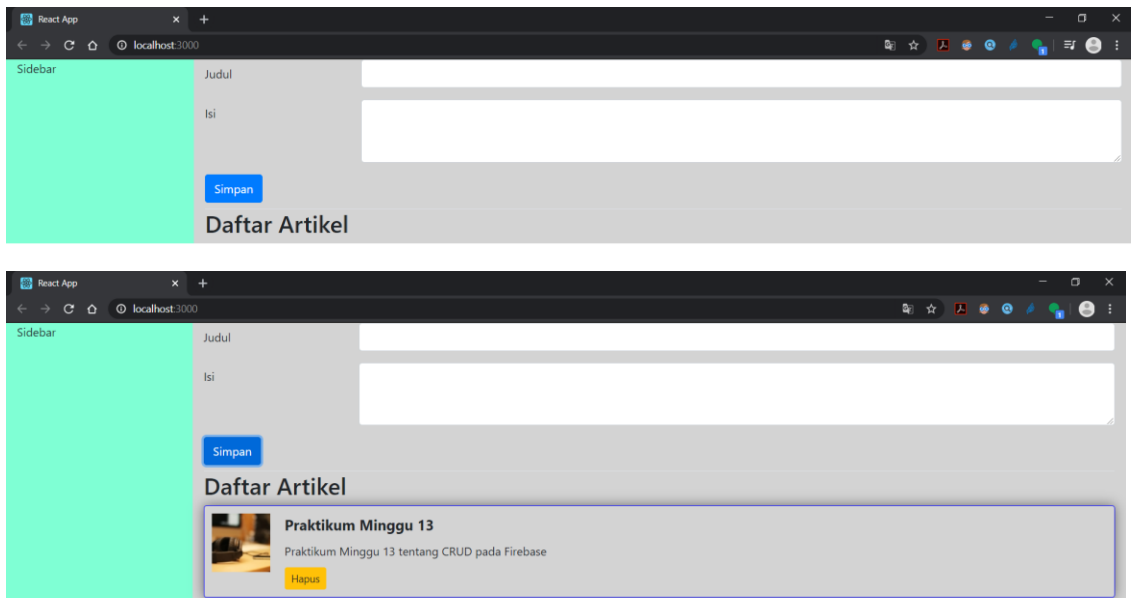
13. Edit pula stateless component Post.jsx menjadi seperti gambar berikut

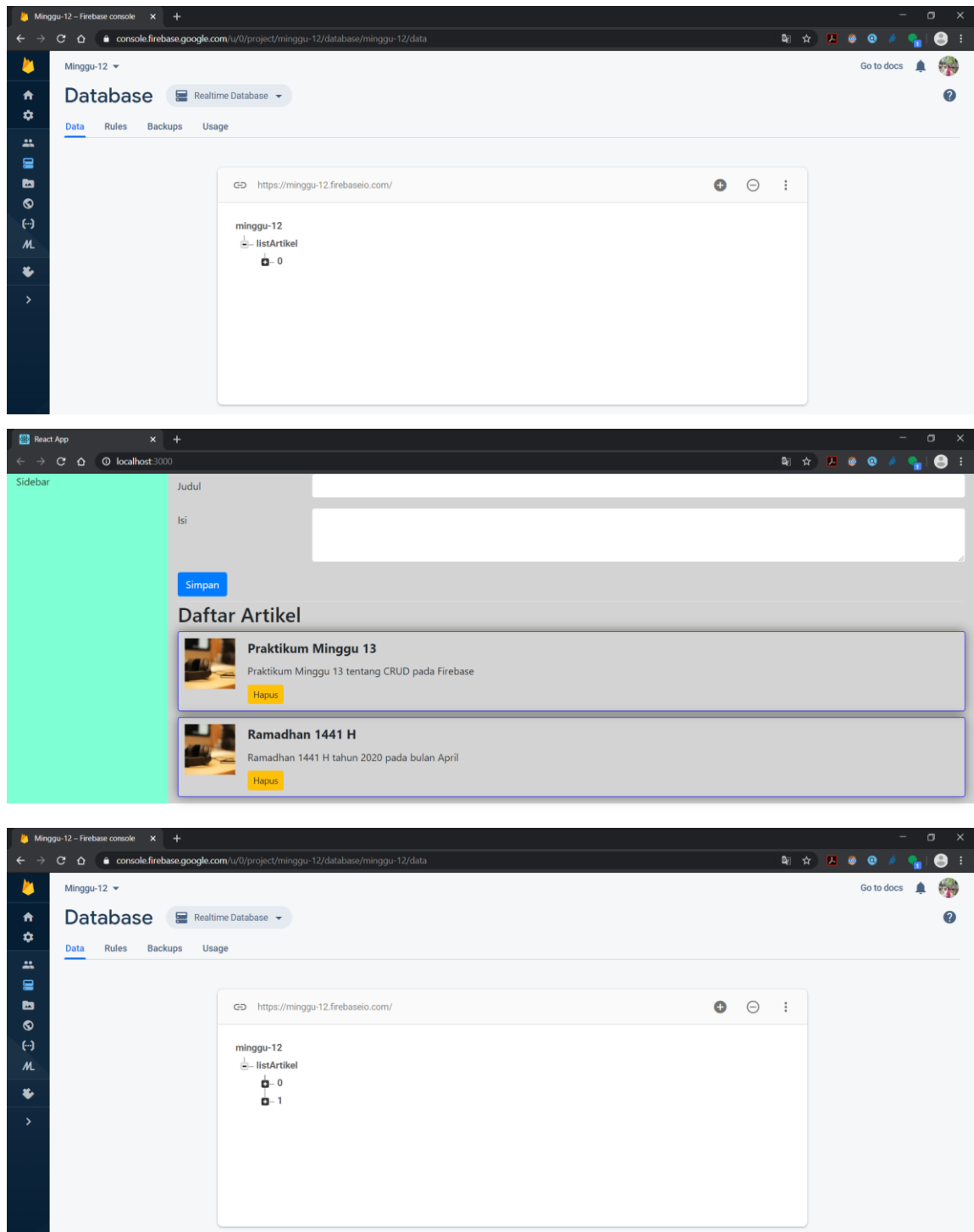
```

Post.jsx
praktikum-minggu13 > src > component > BlogPost > Post.jsx > Post
1   import React from "react";
2
3   const Post = (props) => {
4     return (
5       <div className="artikel">
6         <div className="gambar-artikel">
7           
8         </div>
9         <div className="konten-artikel">
10          <div className="judul-artikel">{props.judul}</div>
11          <p className="isi-artikel">{props.isi}</p>
12          <button className="btn btn-sm btn-warning"
13            onClick={() => { if (window.confirm('Apakah anda yakin menghapus artikel ini?')) props.hapusArtikel(props.idArtikel)}}>
14            Hapus
15          </button>
16        </div>
17      </div>
18    )
19  }
20
21  export default Post;

```

14. Silahkan lakukan proses insert data pada browser, lihat apa yang terjadi





### 1.3 Pertanyaan Praktikum

1. Perhatikan file BlogPost.jsx, apa saja kode program yang berubah dari BlogPost.js pada Modul-8 dengan BlogPost.jsx pada praktikum kali ini? Kenapa?
  - Penambahan import firebase dan menghapus import API, karena pada materi ini menjelaskan tentang penggunaan CRUD pada firebase

```
4 // import API from "../services";  
5 import firebase from "firebase";
```



- Menginisialisasi konfigurasi database firebase agar terhubung dengan project pada firebase
  - Menambahkan fungsi untuk mengirim data ke Realtime Database Firebase. Data akan tersimpan pada realtime database firebase
  - Menambahkan fungsi `componentDidUpdate` yang digunakan untuk melakukan operasi DOM setelah data telah diperbaharui.
  - Penambahan source code pada `handleTombolSimpan`
  - Menambahkan refs untuk menampung value
2. Perhatikan file `Post.jsx`, apa saja kode program yang berubah dari `Post.jsx` pada Modul-8 dengan `Post.jsx` pada praktikum kali ini? Kenapa?
- Adanya penambahan source code pada window konfirmasi

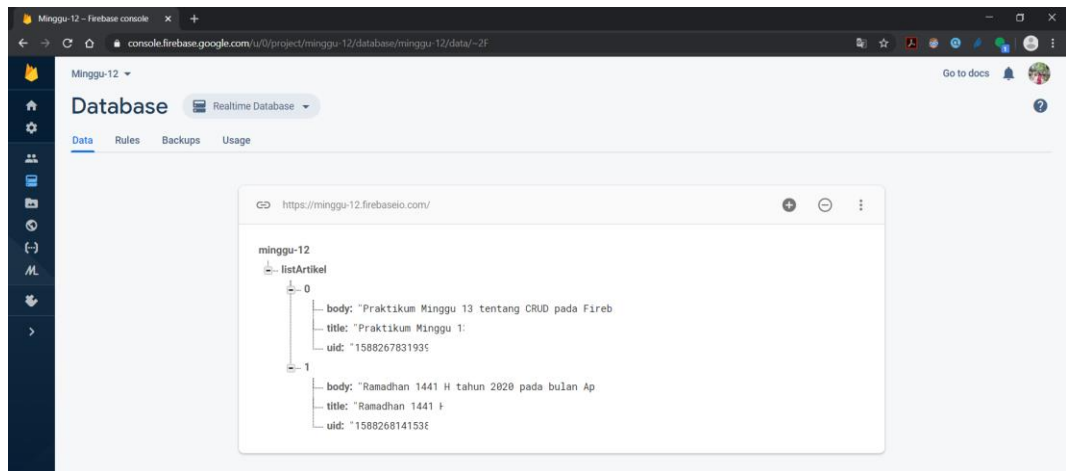
```

12 <button className="btn btn-sm btn-warning"
13   onClick={() => { if (window.confirm('Apakah anda yakin menghapus artikel ini ?'))props.hapusArtikel(props.idArtikel)}}>
14   Hapus
15 </button>

```

3. Apakah Global API service yang kita buat pada Modul-8 kemarin kita pakai lagi pada praktikum kali ini? Kenapa alasannya?
- Tidak karena pada materi ini menggunakan Realtime Database Firebase
4. 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 disimpan pada firebase database





5. 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 lebih efisien menggunakan Firebase, karena tidak perlu membuat server untuk file.json.