

# **LAPORAN INDIVIDU**

## ***REALTIME DATABASE***

Disusun sebagai

MATA KULIAH : PEMROGRAMAN BERBASIS FRAMEWORK

Oleh :

**Nisa' Nurrochmah Hani / 1741720164**

**3C / 18**



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

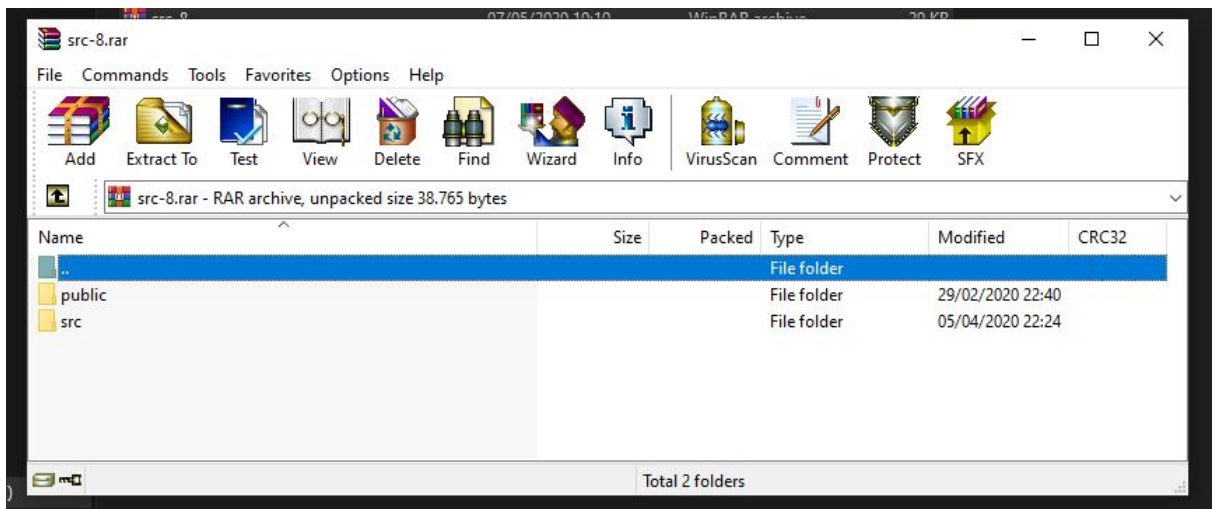
## 1.1 Langkah Pertama

1. Buatlah project reactjs baru

```
PS E:\KULIAH NISA\Semester 6\Pemrograman Berbasis Framework\realtimeDB> create-react-app realtimedb
Creating a new React app in E:\KULIAH NISA\Semester 6\Pemrograman Berbasis Framework\realtimeDB\realtimedb.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

[.....] / refresh-package-json:ripemd160: [sill] refresh-package-json E:\KULIAH NISA\Semester 6\Pemrograman Berbasis Framework\realtimeDB\realtimedb\node_mo
```

2. Backup data public dan src, kemudian hapus file data public dan src
3. Ekstrakt file source modul 8.rar ke folder reactjs



4. Install beberapa package yang dibutuhkan

- npm install react-router-dom

```
PS E:\KULIAH NISA\Semester 6\Pemrograman Berbasis Framework\realtimeDB\realtimedb> npm install react-router-dom
[.....] - rollbackFailedOptional: verb npm-session 21540563a678197e
```

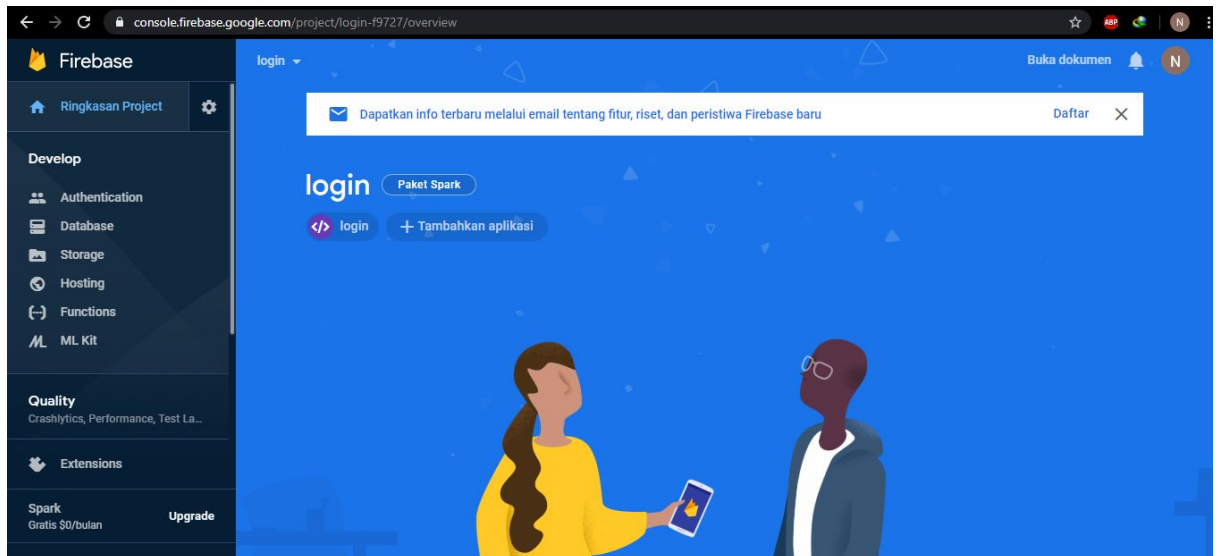
- npm install --save firebase

```
PS E:\KULIAH NISA\Semester 6\Pemrograman Berbasis Framework\realtimeDB\realtimedb> npm install --save firebase
[.....] / rollbackFailedOptional: verb npm-session ad80a60d2c50feb4
```

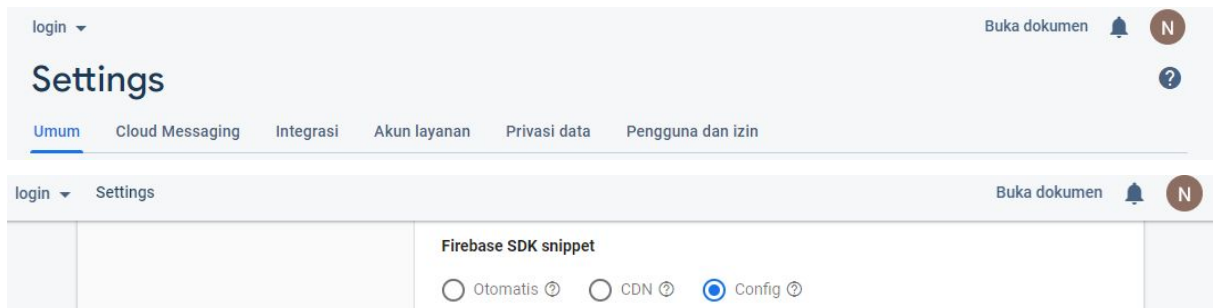
- npm install bootstrap

```
PS E:\KULIAH NISA\Semester 6\Pemrograman Berbasis Framework\realtimeDB\realtimedb> npm install bootstrap
[.....] | loadExtraneous: sill resolveWithNewModule bootstrap@4.4.1 checking installable status
```

5. Pastikan anda tidak lupa konfigurasi dari aplikasi firebase yang telah dibuat. Jika lupa anda bias masuk ke firebase control di <https://console.firebase.google.com>
6. Kemudian klik "Ringkasan project", dan pilih app yang sudah dbuat. Sesuai dengan materi sebelumnya di app "firebase-login-app", kemudian klik setting



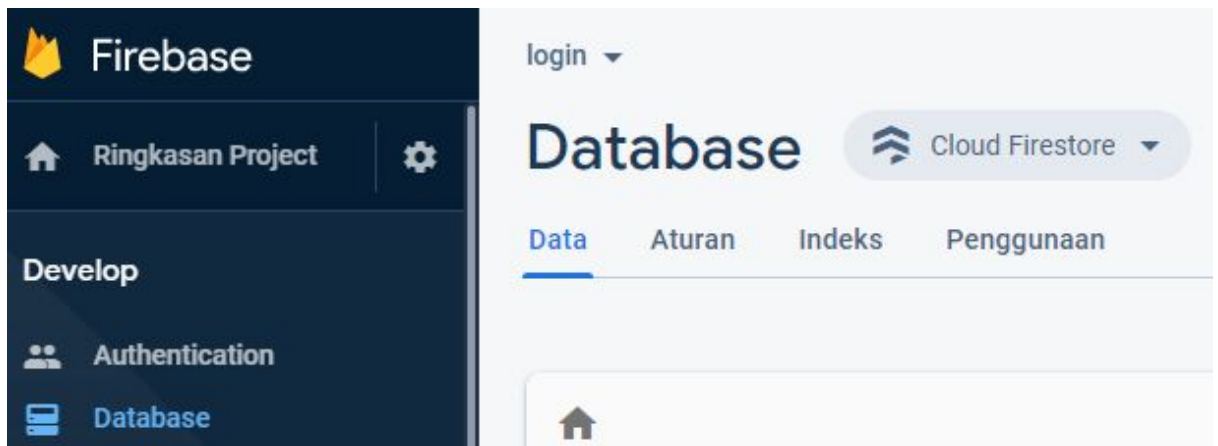
7. Pada tab umum/common di menu setting, scroll kebawah dan pilih opsi config pada firebase SDK Snippet



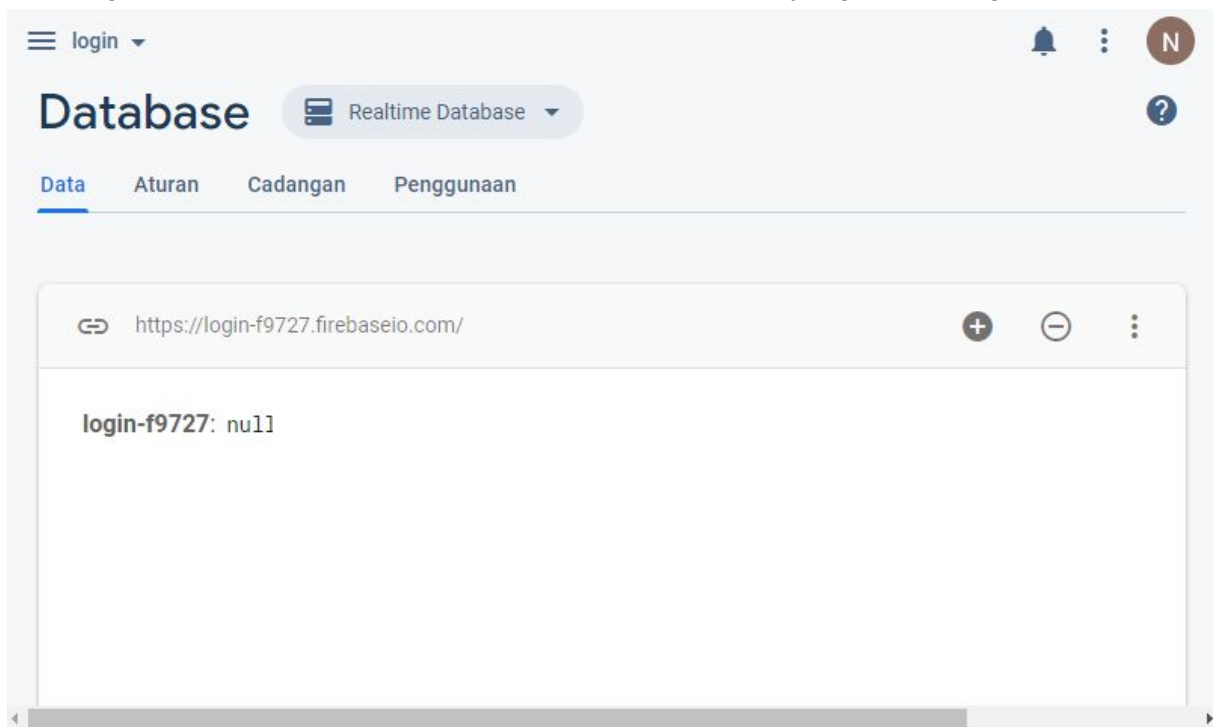
8. Konfigurasi inilah yang akan terus kita gunakan saat berinteraksi dengan API firebase yang telah kita buat



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



10. Seperti gambar berikut tampilan database real time firebase yang akan kita gunakan



## 1.2 Langkah Kedua

1. Pada project reactjs, buat file dan folder firebase/config.js dalam folder src dan simpan konfigurasi code program diatas (config dari langkah 8) pada file config.js

```
realtimedb > src > firebase > config.js > ...
1  const firebaseConfig = {
2    apiKey: "AIzaSyDhoDtC8nkm07dy4AQduWTbnLxFy9cD9G8",
3    authDomain: "login-f9727.firebaseio.com",
4    databaseURL: "https://login-f9727.firebaseio.com",
5    projectId: "login-f9727",
6    storageBucket: "login-f9727.appspot.com",
7    messagingSenderId: "697331420575",
8    appId: "1:697331420575:web:8ea5b30ec44195ecd681aa",
9    measurementId: "G-67P4HQQ6E5"
10 };
11 export default firebaseConfig;
```

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

```
realtimeadb > src > container > BlogPost > BlogPost.jsx > ...
1  import React, {Component} from "react";
2  import './BlogPost.css';
3  import Post from "../../component/BlogPost/Post";
4  import firebase from "firebase";
5  import firebaseConfig from "../../firebase/config";
6
7  class BlogPost extends Component{
8      constructor(props) {
9          super(props);
10         firebase.initializeApp(firebaseConfig);
11
12         this.state = {
13             listArtikel: []
14         }
15     }
16     ambilDataDariServerAPI = () => {
17         let ref = firebase.database().ref("/");
18         ref.on("value", snapshot => {
19             const state= snapshot.val();
20             this.setState(state);
21         });
22     }
23
24     simpanDataServerAPI = () => {
25         firebase.database()
26             .ref("/")
27             .set(this.state);
28     }
29
30     componentDidMount() {
31         this.ambilDataDariServerAPI()
32     }
```



```

realtimeDb > src > container > Blogspot > BlogPost.jsx > BlogPost > ambliDataDanServerAPI
33   componentDidUpdate(prevProps, prevState) {
34     if (prevState !== this.state) {
35       this.simpanDataakeServerAPI();
36     }
37   }
38
39   handleHapusArtikel = (idArtikel) => {
40     const {listArtikel} = this.state;
41     const newState = listArtikel.filter(data => {
42       return data.uid !== idArtikel;
43     });
44     this.setState({listArtikel: newState});
45   }
46
47   handleTambahArtikel = (event) => {
48     let formInsertArtikel = {...this.state.insert.formInsertArtikel};
49     let timestamp = new Date().getTime();
50     formInsertArtikel['id'] = timestamp;
51     formInsertArtikel[event.target.name] = event.target.value;
52     this.setState({
53       insertArtikel: formInsertArtikel
54     });
55   }
56
57   handleTombolSimpan = (event) => {
58     let title = this.refs.judulArtikel.value;
59     let body = this.refs.isiArtikel.value;
60     let uid = this.refs.uid.value;

```

```

realtimeDb > src > container > Blogspot > BlogPost.jsx > BlogPost > ambliDataDanServerAPI
61
62     if (uid && title && body) {
63       const {listArtikel} = this.state;
64       const indeksArtikel = listArtikel.findIndex(data => {
65         return data.uid === uid;
66       });
67       listArtikel[indeksArtikel].title = title;
68       listArtikel[indeksArtikel].body = body;
69       this.setState({listArtikel});
70     } else if (title && body) {
71       const uid = new Date().getTime().toString();
72       const {listArtikel} = this.state;
73       listArtikel.push({uid, title, body});
74       this.setState({ listArtikel });
75     }
76
77     this.refs.judulArtikel.value = "";
78     this.refs.isiArtikel.value = "";
79     this.refs.uid.value = "";
80   });

```

```

81
82   render() {
83     return(
84       <div className="post-artikel">
85         <div className="form pb-2 border-bottom">
86           <div className="form-group row">
87             <label htmlFor="title" className="col-sm-2 col-form-label">Judul</label>
88             <div className="col-sm-10">
89               <input type="text" className="form-control" id="title" name="title" ref="judulArtikel"/>
90             </div>
91           </div>
92           <div className="form-group row">
93             <label htmlFor="body" className="col-sm-2 col-form-label">Isi</label>
94             <div className="col-sm-10">
95               <textarea className="form-control" id="body" name="body" rows="3" ref="isiArtikel"></textarea>
96             </div>
97           </div>
98           <input type="hidden" name="uid" ref="uid"/>
99           <button type="submit" className="btn btn-primary" onClick={this.handleTomboSimpan}>Simpan</button>
100         </div>
101         <h2>Daftar Artikel</h2>
102         {
103           this.state.listArtikel.map(artikel => {
104             return <Post key={artikel.uid} judul={artikel.title} isi={artikel.body}
105                       idArtikel={artikel.uid} hapusArtikel={this.handleHapusArtikel}/>
106           })
107         }
108       </div>
109     )
110   }
111 }
112
113 export default BlogPost;

```

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

```

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" onClick={() => {
13           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;

```

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







### 1.3 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?

**jawaban:** hampir keseluruhan terjadi perubahan pada blogpost.jsx dikarenakan pada modul 8 menggunakan server api, sedangkan pada modul 12 menggunakan realtime database pada firebase.

- 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?

**jawaban:** menambahkan source code pada baris dikarenakan menggunakan realtime database

```
<button className="btn btn-sm btn-warning" onClick={() => {
  if (window.confirm('Apakah Anda Yakin Menghapus Artikel Ini?')) props.hapusArtikel(props.idArtikel)}}>
  Hapus
</button>
</div>
</div>
)
```

- c. Apakah Global API service yang kita buat pada Modul-8 kemarin kita pakai lagi pada praktikum kali ini? Kenapa alasannya?

**jawaban:** Tidak, karena pada modul ini menggunakan realtime database pada firebase.

- 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.

**jawaban:** data tersimpan pada database yang ada di firebase.



- 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.

**jawaban:** sangat lebih mudah dan praktis menggunakan firebase realtime database, dikarenakan tidak perlu menyalakan server secara manual pada cmd seperti modul 4 dan modul 8 apabila setiap kita akan menjalankan project yang telah dibuat.