

# **Laporan Praktikum**

## **“FIREBASE REALTIME DATABASE”**

Disusun sebagai

**MATA KULIAH : PEMROGRAMAN BERBASIS FRAMEWORK**

Oleh :

**Daffa Tahta Akbar**

**1741720152**

**Kelas :**

**3B**



**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2020**

## Pertanyaan:

1. Perhatikan file BlogPost.jsx apa saja kode program yang berubah dari BlogPost.jsx pada modul-8 dengan BlogPost.jsx pada praktikum kali ini, kenapa?
2. Perhatikan file Post.jsx, apa saja kode program yang berubah dari Post.js pada Modul-8 dengan Post.jsx pada praktikum kali ini? Kenapa?
3. Apakah Global API service yang kita buat pada Modul-8 kemarin kita pakai lagi pada praktikum ini? Kenapa alasannya?
4. Data yang kita insert bertambah, dan saat kita refresh browser, data masih tetap ada. Dimanakah data data artikel tersebut di simpan? Tunjukkan hasil screenshot data disimpan tersebut.
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? Berikan alasannya.

**Jawaban:**

1. Komponen state di hapus dan ditambahkan komponen baru, constructor, simpanDataKeServerAPI dan component DidUpdate. Perubahan komponen pada ambilDataDariServerAPI handleHapusArtikel, dan handleTombolSimpan.

Karena untuk mengubah mengambil data dari file json ke firebase.

2. Pada button hapus.

Diberi window confirm untuk meyakinkan apakah dihapus / tidak.

3. Tidak, karena tidak menggunakan data dari file json, sekarang menggunakan database firebase.
4. Di Realtime Firebase
5. Menggunakan firebase realtime, karena tidak perlu susah membuat file untuk datanya, data langsung masuk ke penyimpanan di firebase.

## Praktikum

1. Buat project react js baru
2. Backup folder src dan public
3. Download dan extract folder SourceModul8.rar
4. Install
  - a. React-router-dom
  - b. –save firebase
  - c. Bootstrap
5. Copy konfigurasi config dari console firebase

The screenshot shows the Firebase console interface. On the left, under 'Web apps', there are two entries: 'firebase-login' and 'firebaseApp'. 'firebaseApp' is selected and highlighted in blue. On the right, the configuration for 'firebaseApp' is displayed. It includes an 'App nickname' of 'firebaseApp', an 'App ID' of '1:860451278327:web:b3e9f3427a4e9a939e72f2', and a button to 'Link to a Firebase Hosting site'. Below this, the 'Firebase SDK snippet' is shown with the 'Config' option selected. The snippet contains a JavaScript code block for initializing the Firebase configuration. At the bottom right, there is a 'Remove this app' button.

Web apps

- firebase-login
- firebaseApp**

App nickname  
firebaseApp

App ID  
1:860451278327:web:b3e9f3427a4e9a939e72f2

[Link to a Firebase Hosting site](#)

**Firebase SDK snippet**

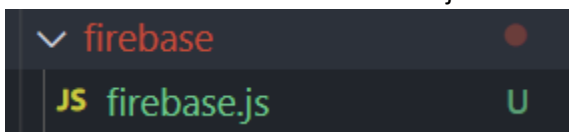
☐ CDN ☒ Config

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

```
const firebaseConfig = {  
  apiKey: "AIzaSyDU5MeojJSqi_7FLP4h0yA9fcNrKqZ71b0",  
  authDomain: "fir-53b35.firebaseio.com",  
  databaseURL: "https://fir-53b35.firebaseio.com",  
  projectId: "fir-53b35",  
  storageBucket: "fir-53b35.appspot.com",  
  messagingSenderId: "860451278327",  
  appId: "1:860451278327:web:b3e9f3427a4e9a939e72f2",  
  measurementId: "G-E662QQD4FK"  
};
```

[Remove this app](#)

6. Buat folder firebase dan file firebase.js

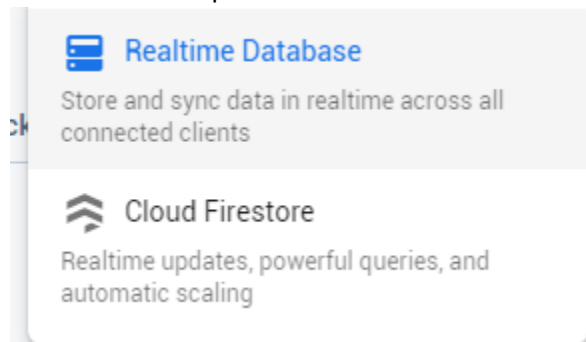


7. Paste config yang telah di copy dari console firebase ke dalam folder firebase.js

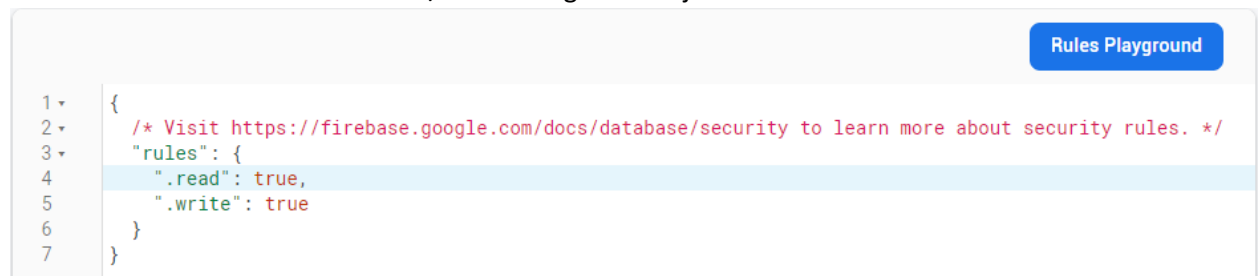
```
const config = {
  apiKey: "AIzaSyDU5MeojJSqi_7FLP4h0yA9fcNrKqZ71b0",
  authDomain: "fir-53b35.firebaseio.com",
  databaseURL: "https://fir-53b35.firebaseio.com",
  projectId: "fir-53b35",
  storageBucket: "fir-53b35.appspot.com",
  messagingSenderId: "860451278327",
  appId: "1:860451278327:web:b3e9f3427a4e9a939e72f2",
  measurementId: "G-E662QQD4FK"
};

export default config;
```

8. Masuk ke dalam pilihan database di firebase dan pilih realtime database



9. Setelah itu masuk ke dalam aturan / rules dan ganti menjadi true



10. Buka BlogPost.jsx dan ubah sehingga menjadi seperti ini :

a. 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 config from "../../firebase/config";

class BlogPost extends Component {
  constructor(props) {
    super(props);
    firebase.initializeApp(config);

    this.state = {
      listArtikel: [],
    };
  }

  ambilDataDariServerAPI = () => {
    let ref = firebase.database().ref("/");
    ref.on("value", (snapshot) => {
      const state = snapshot.val();
      this.setState(state);
    });
  };

  simpanDataKeServerAPI = () => {
    firebase.database().ref("/").set(this.state);
  };

  componentDidMount() {
    // komponen untuk mengecek ketika compnent 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) => {
  const { listArtikel } = this.state;
  const newState = listArtikel.filter((data) => {
    return data.uid !== idArtikel;
  });
  this.setState({ listArtikel: newState });
};

handleTambahArtikel = (event) => {
  // fungsi untuk meng-handle form tambah data
  artikel
  let formInsertArtikel =
  { ...this.state.insertArtikel }; // cloning data
  state insertArtikel ke dalam variabel
  formInsertArtikel
  let timestamp = new Date().getTime(); // digunakan
  untuk menyimpan waktu (sebagai ID artikel)
  formInsertArtikel["id"] = timestamp;
  formInsertArtikel[event.target.name] =
  event.target.value; // menyimpan data onchange ke
  formInsertArtikel sesuai dengan target yg diisi
  this.setState({
    insertArtikel: formInsertArtikel,
  });
};

handleTombolSimpan = (event) => {
  let title = this.refs.judulArtikel.value;
  let body = this.refs.isiArtikel.value;
  let uid = this.refs.uid.value;
  if (uid && title && body) {
    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) {
    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>
            </div>
        </div>
    );
}

```



```

        <input type ='hidden' name='uid'
ref='uid'></input>
        <button
            type="submit"
            className="btn btn-primary"
            onClick={this.handleTombolSimpan}
        >
            Simpan
        </button>
    </div>
    <h2>Daftar Artikel</h2>
    {this.state.listArtikel.map((artikel) => {
        // looping dan masukkan untuk setiap data
yang ada di listArtikel ke variabel artikel
        return (
            <Post
                key={artikel.uid}
                judul={artikel.title}
                isi={artikel.body}
                idArtikel={artikel.uid}
                hapusArtikel={this.handleHapusArtikel}
            />
        ); // mappingkan data json dari API sesuai
dengan kategorinya
    })}
    </div>
    );
}
}

export default BlogPost;

```