Pemrograman Berbasis Framework

"Firebase Realtime Database"

Oleh :

Angga Maulana Athaariq / 09

TI-3B / 1741720138

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

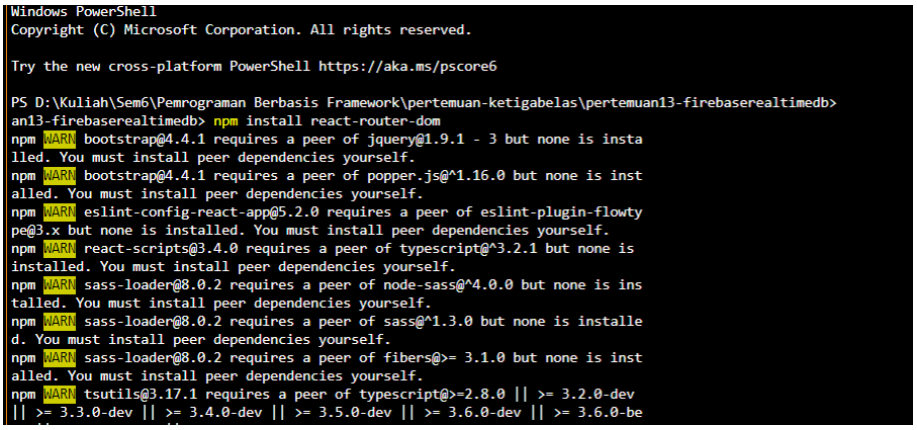
2020

A. Langkah Persiapan

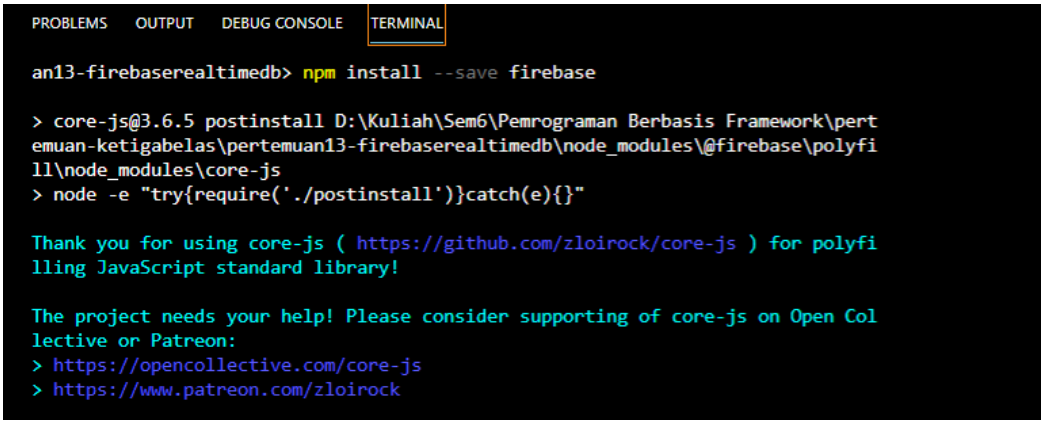
1. Buat project reactjs baru
   Saya disini menggunakan project yang lama kemudian saya mengubah pada folder public & src dengan yang baru.

2. Install package yang dibutuhkan.

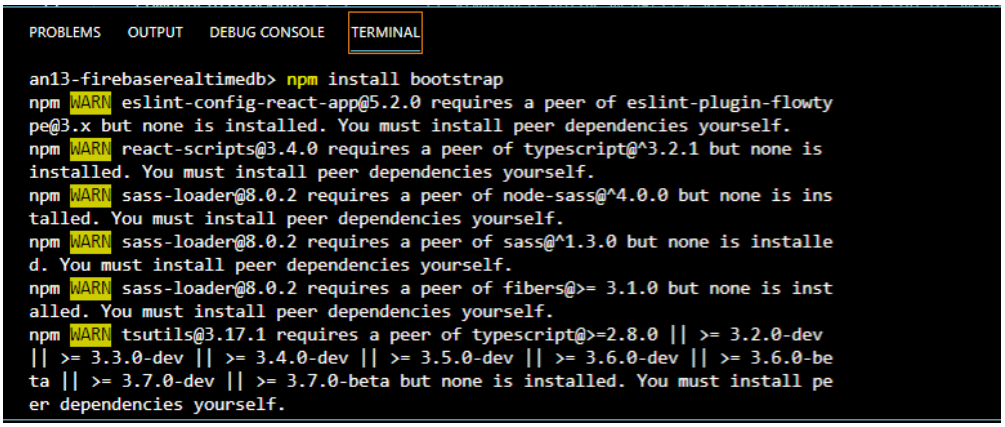
   - npm install react-router-dom



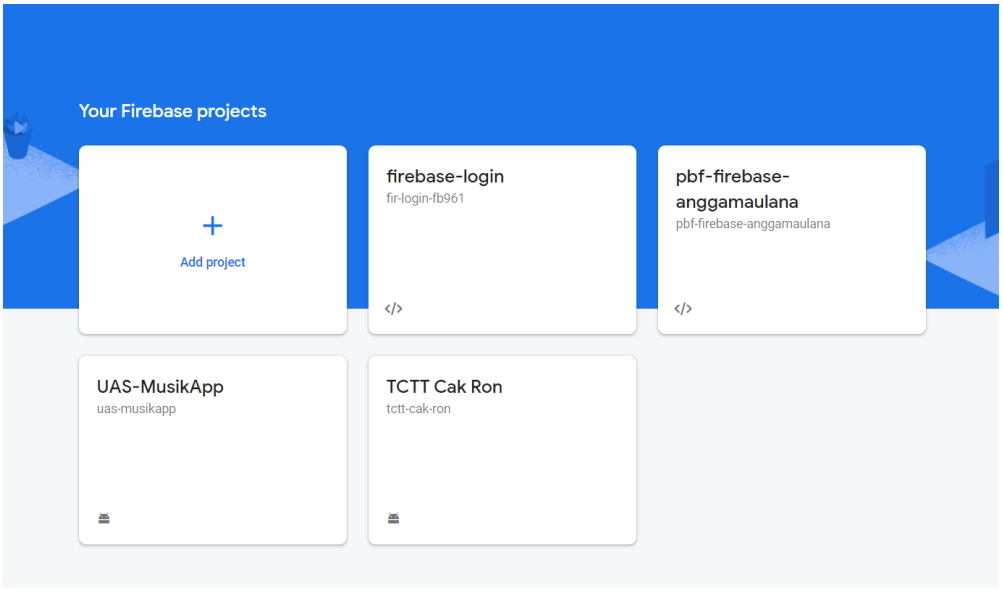
   - npm install --save firebase



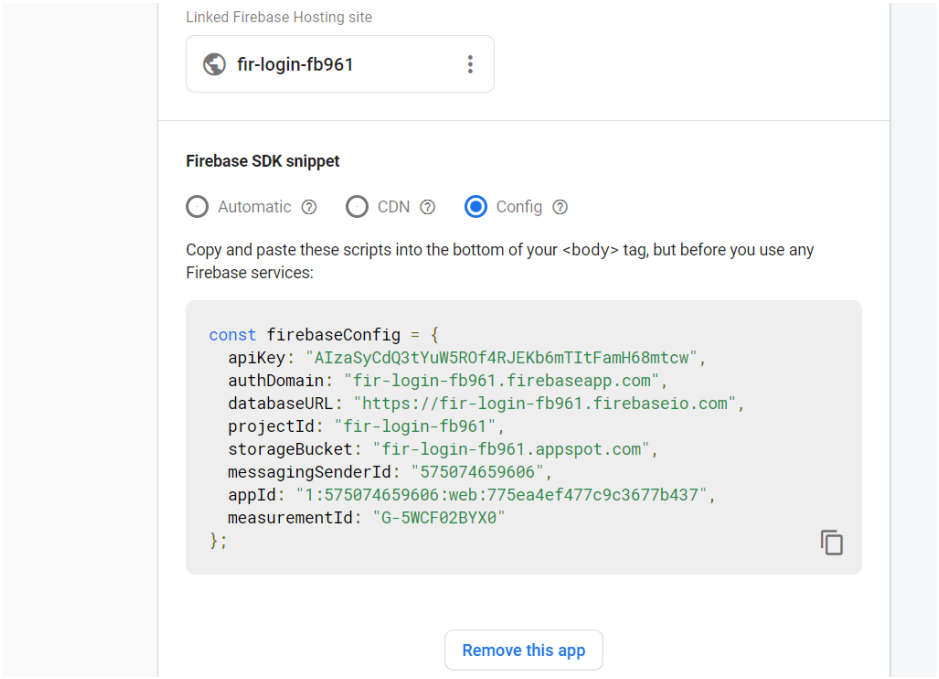
   - npm install bootstrap



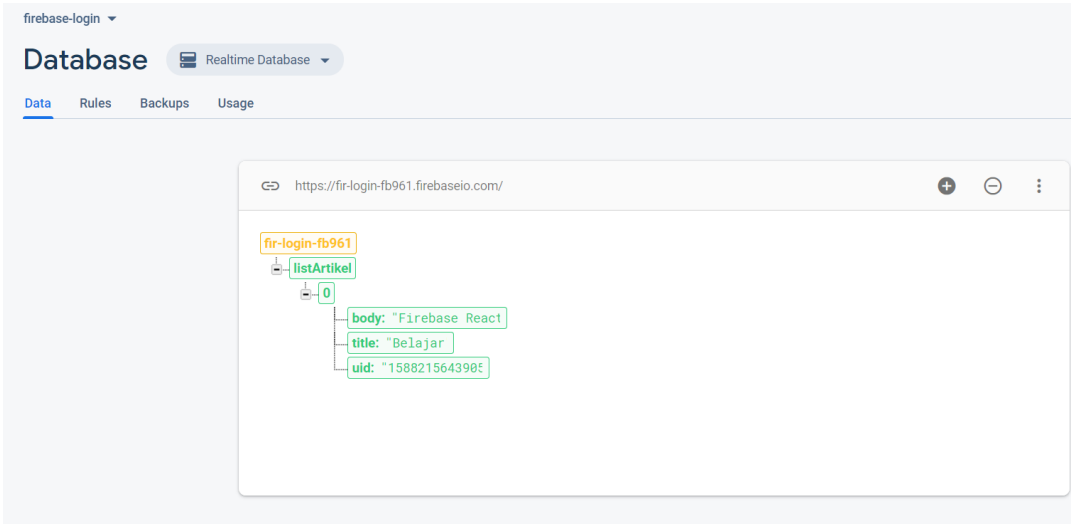
3. Masuk project firebase console, saya menggunakan firebase-login seperti minggu kemarin.



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



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



B. LangkahPraktikum

1. Membuat file dan folder firebase/config.js dalam folder src.

```
const firebaseConfig = {
    apiKey: "AIzaSyCdQ3tYuW5ROf4RJEKb6mTItFamH68mtcw",
    authDomain: "fir-login-fb961.firebaseapp.com",
    databaseURL: "https://fir-login-fb961.firebaseio.com",
    projectId: "fir-login-fb961",
    storageBucket: "fir-login-fb961.appspot.com",
    messagingSenderId: "575074659606",
    appId: "1:575074659606:web:775ea4ef477c9c3677b437",
    measurementId: "G-5WCF02BYX0"
  };
  export default firebaseConfig;
```

2. Modifikasi statefull component BlogPost.jsx.

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


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

      this.state = {
          listArtikel: []
      }
  }

    ambilDataDariServerAPI = () => {                // fungsi
untuk mengambil data dari API dengan penambahan sort dan order
        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 ket
ika compnent telah di-mount-ing, maka panggil API
        this.ambilDataDariServerAPI()  // ambil data dari serv
er API lokal
    }

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

    handleHapusArtikel = (idArtikel) => {        // fungsi yan
g 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 unt
uk meng-handle tombol simpan
       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 });
```

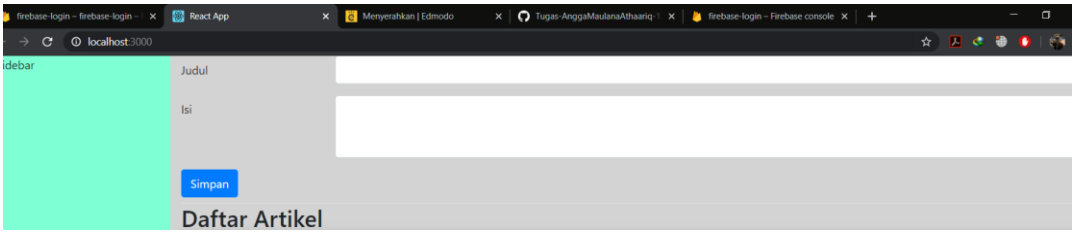
```
        }else if (title && body) {
            const uid = new Date().getTime().toString();
            const { listArtikel } = this.state;
            listArtikel.push({ uid, title, body });
            this.setState({listArtikel});
        }
        this.refs.judulArtikel.value= "";
        this.refs.isiArtikel.value= "";
        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"></tex
tarea>
                        </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>
                {
```

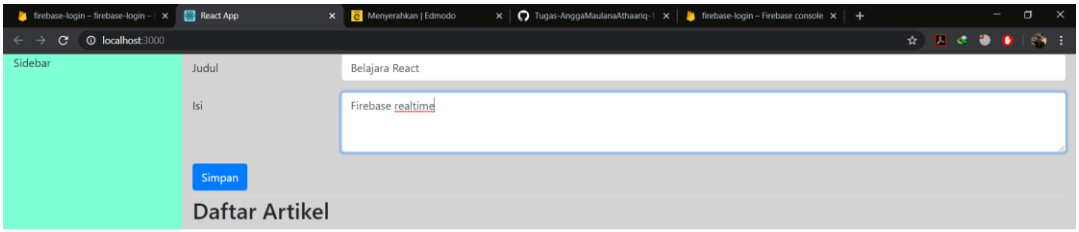
```
this.state.listArtikel.map(artikel => {  // looping dan masukk
an untuk setiap data yang ada di listArtikel ke variabel artik
el

return <Post key={artikel.uid} judul={artikel.title} isi={arti
kel.body} idArtikel={artikel.uid} hapusArtikel={this.handleHap
usArtikel}/>     // mappingkan data json dari API sesuai denga
n kategorinya

                    })

                }

            </div>

        )

    }

}

export default BlogPost;
```

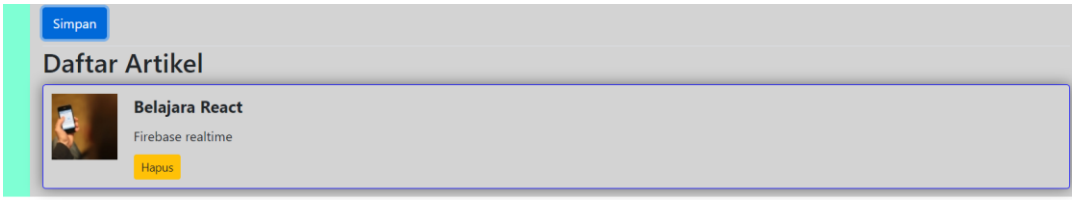
3. Modifikasi stateless component Post.jsx.

```
import React from "react";


const Post = (props) => {

    return (

        <div className="artikel">

            <div className="gambar-artikel">

                <img src="http://placeimg.com/80/80/tech" alt=
"Gambar Tumbnail 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 and
a yakin menghapus artikel ini?')) props.hapusArtikel(props.idA
rtikel)}}>Hapus</button>

            </div>

        </div>

    )

}

export default Post;
```

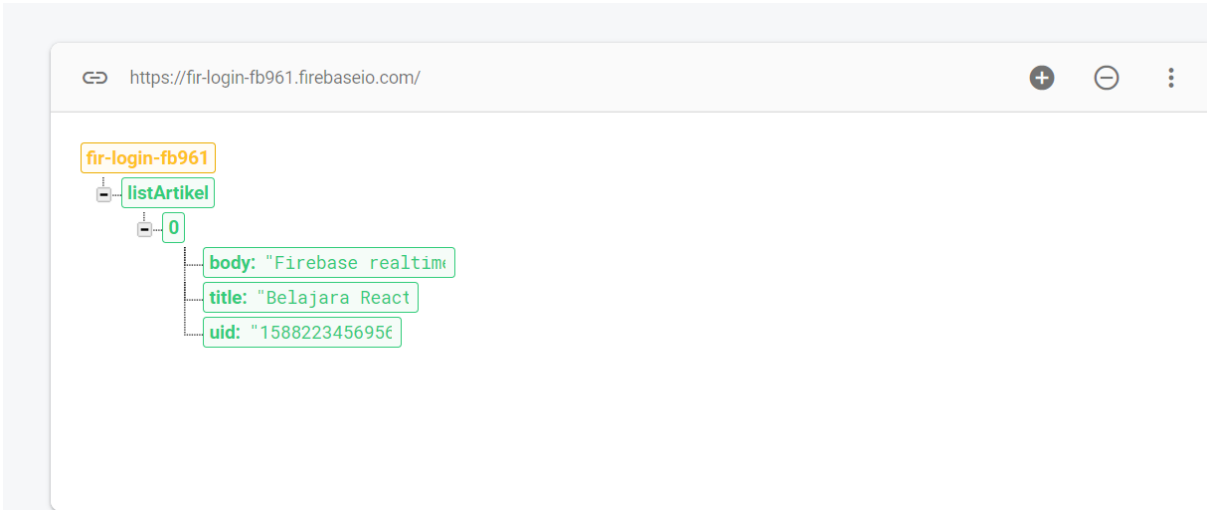
4. Lakukan proses insert data pada browser.







5. Tampilan di database.



C. 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?

   Jawab :

   Mengubah pada contructor, ambilData, simpanData, handelHapus, handleSimpan & menambahkan perintah componentDidMount & componenetDidUpdate.

   Penghapusan import API
   Import API dihapus karena pada praktikum sudah tidak menggunakan API, melainkan menggunakan firebase.

   Penambahan simpanDataKeServerAPI
   Fugsi ini ditujukan untuk mengirimkan data yang diinputkan ke database firebase.

   Penambahan componentDidUpdate
   Komponen ini dicek apakah data sama dengan data yang sudah ada pada database, jika data tidak sama maka akan diteruskan ke dalam fungsi simpadDataKeServerAPI.

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?

   Jawab:

   Pada file ini hanya menambahkan munculnya tampilan window baru, yaitu berupa konfirmasi ingin penghapusan.

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

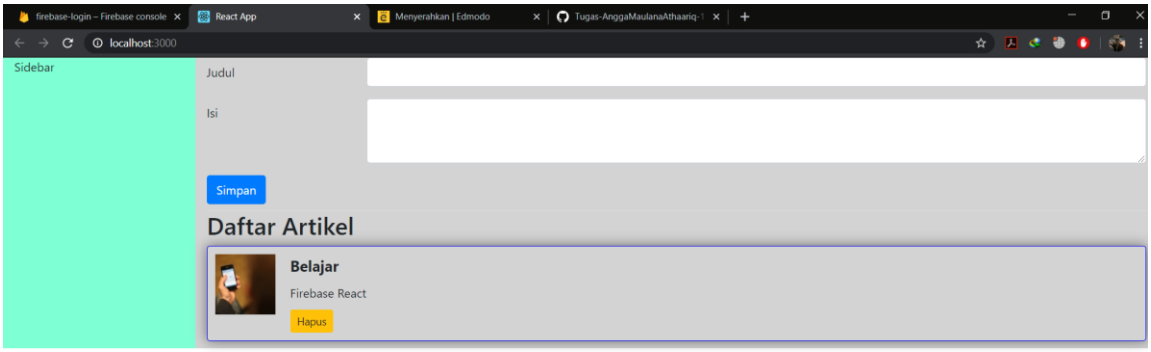
   Jawab:

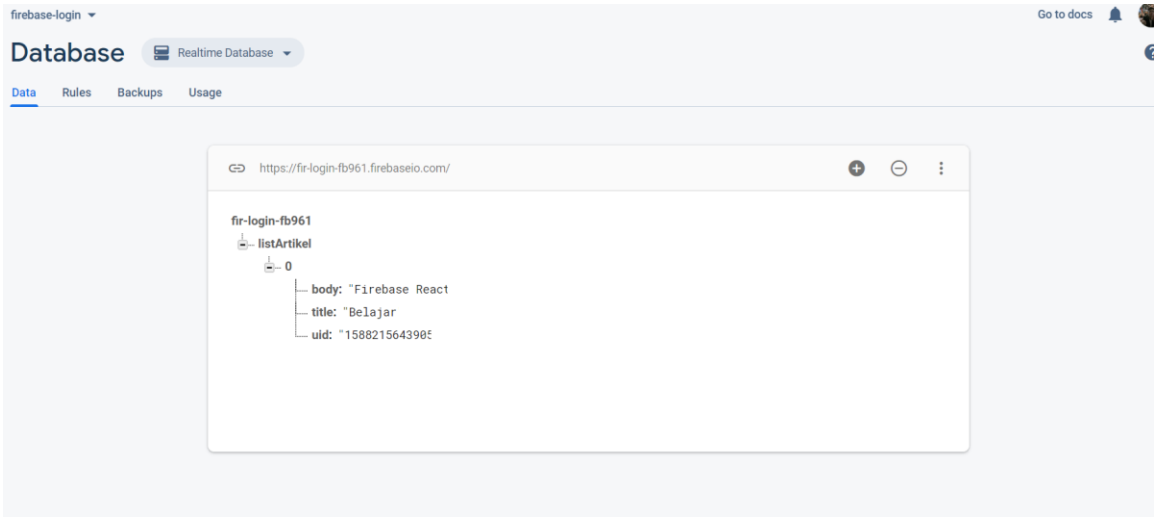
   Pada praktikum ini global API sudah tidak digunakan lagi, karena semua proses diarahkan pada Firebase. Dengan bukti bahwa pada file terdapat import firebase dan menghapus import API.

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.

   Jawab:

   Data yang kita inputkan di web tersebut akan masuk/disimpan di firebase pada realtime-database yang sudah dibuat sebelumnya.





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.

   Jawab:

   Lebih mudah menggunakan database pada firebase, karena jika menggunakan firebase ini kita tidak perlu lagi untuk membuka localhost baru untuk mengaksesnya.