**TUGAS INDIVIDU**

**REALTIME DATABASE FIREBASE**

**Disusun sebagai**

**Pemerograman Berbasis Framework**

**Oleh :**

**Ahmaid Syifaur Romli**

**TI 3A**

**1741720156/01**



**PROGRAM STUDI D-IV T.INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI MALANG**

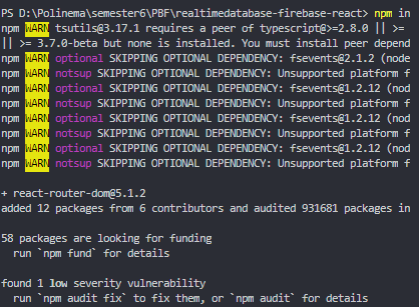
**2020**

**PRAKTIKUM**

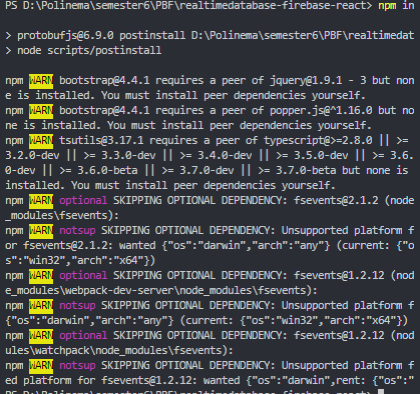
Langkah-langkah praktikum :

1. Buat project **reactjs** baru, atau pakai project **reactjs** yang sudah ada.
2. Backup folder **public** dan **src**, kemudian hapus semua file dan folder yang ada di dalam folder **public** dan **src.**
3. Ekstrak file **Sorce Modul 8.rar** ke dalam project reactjs**.**
4. Silahkan install beberapa *package* yang dibutuhkan melalui CMD, seperti

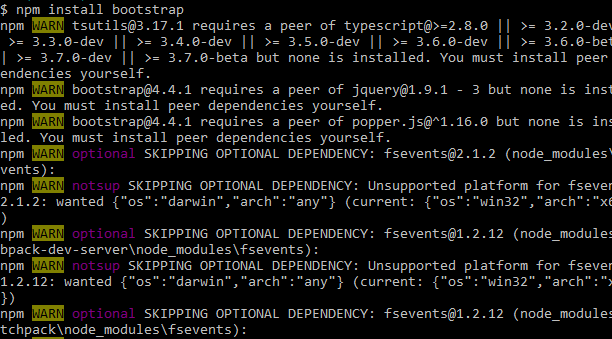
#### npm install react-router-dom



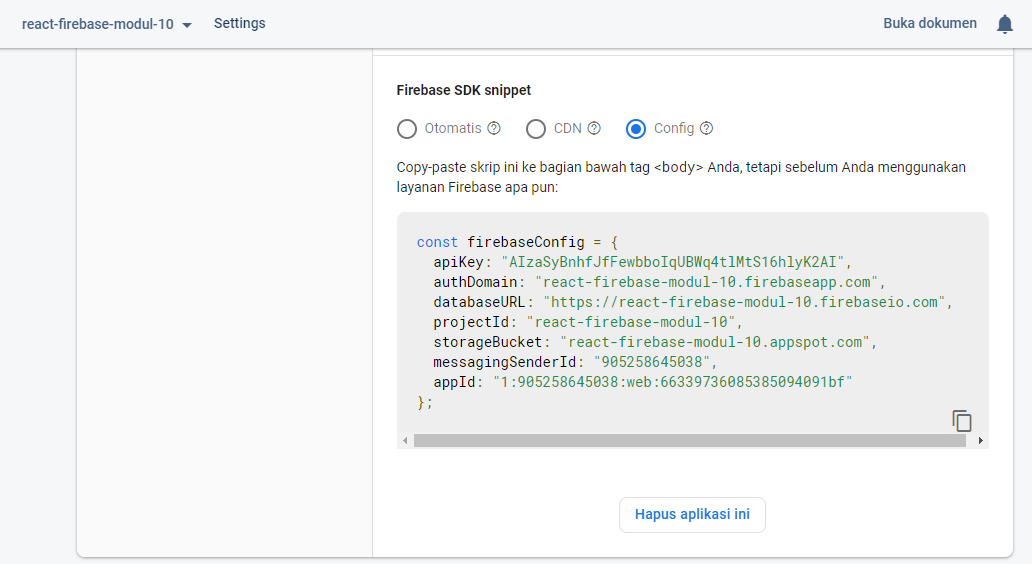
#### npm install --save firebase



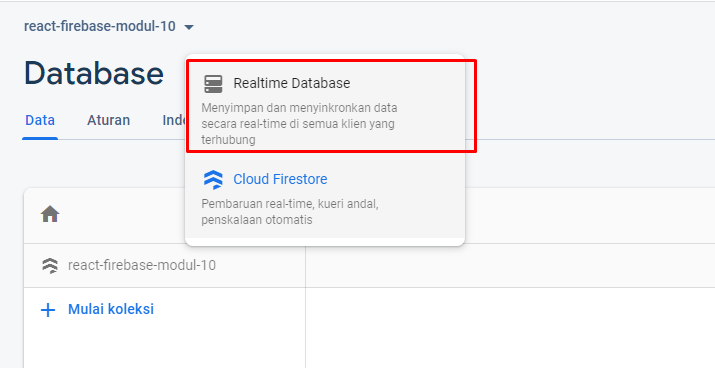
#### npm install bootstrap



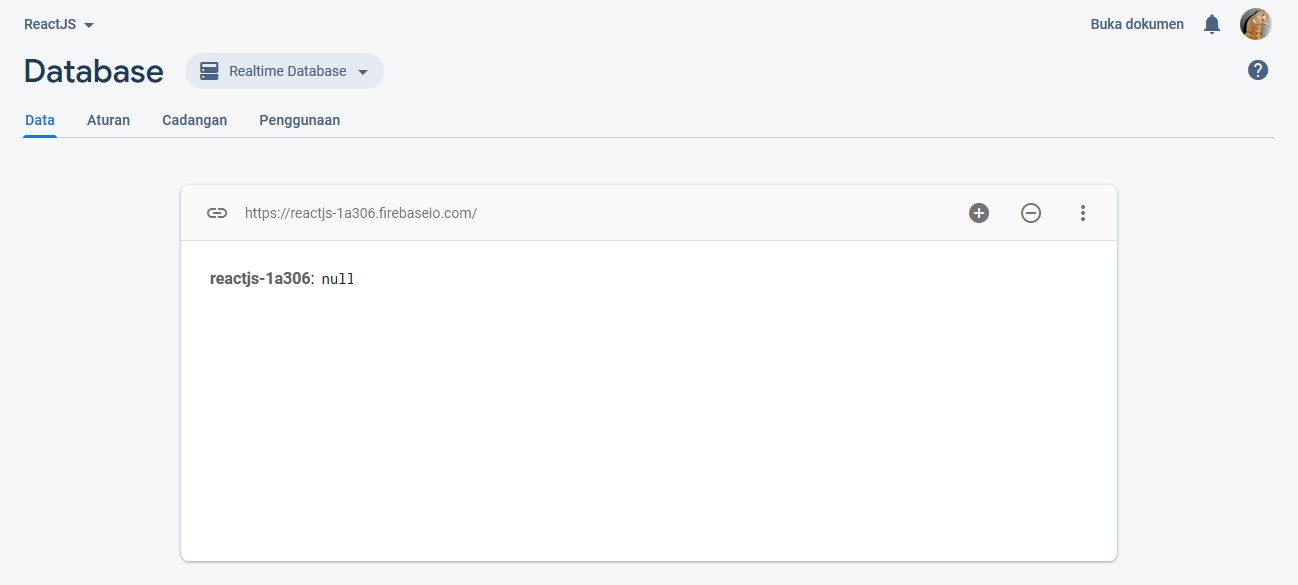
1. Pastikan anda tidak lupa konfigurasi dari aplikasi firebase yang telah dibuat. Jika lupa bisa masuk project firebase konsol d[i https://console.firebase.google.com](https://console.firebase.google.com/)
2. Kemudian klik **“ringkasan project”**, dan pilih app yang sudah dibuat. Sesuai dengan materi sebelumnya adalah app **“firebase-login-app”**, kemudian klik setting.
3. Pada tab **umum/common** di menu **Setting**, scroll ke bawah, dan pilih opsi **config** pada **Firebase**
4. Konfigurasi inilah yang akan terus kita gunakan saat berinteraksi dengan API Firebase yang telah kita buat.



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



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

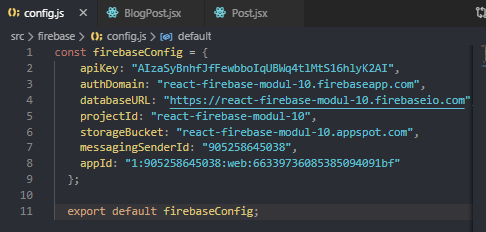


1. Langkah Persiapan telah selesai. Kita mulai praktikumnya.

### 1.2 Langkah Praktikum

Praktikum ini sedikit membahas antara praktikum **Modul 8 (Global API)** dengan **Modul 10 dan 11** tentang **Firebase**. Langkah-langkah praktikum yang dikerjakan adalah

1. 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.**



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

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);

        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 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 = (data) => {        // fungsi yang meng-handle button action hapus data

        API.deleteNewsBlog(data)

            .then(res => {      // ketika proses hapus berhasil, maka ambil data dari server API lokal

                this.ambilDataDariServerAPI()

            })

    }

    handleTambahArtikel = (idArtikel) => {      // fungsi untuk meng-hadle form tambah data artikel

        const{listArtikel}=this.state;

        const newState = listArtikel.filter(data=>{

            return data.uid !== idArtikel;

        });

        this.setState({listArtikel:newState});

    }

    handleTombolSimpan = (event) => {            // fungsi untuk 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"></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>

                {

                    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;

1. Edit pula *stateless component* **Post.jsx** menjadi seperti gambar berikut

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 Anda yakin menghapus artikel ini ?'))props.hapusArtikel(props.idArtikel)}}>Hapus</button>

            </div>

        </div>

    )

}

export default Post;

1. Jalankan pada browser dan coba fitur add data , maka akan tersimpan pada firebase realtime database

