

**TUGAS PERORANGAN/INDIVIDU**

**LAPORAN JOBSHEET MINGGU 10**

**GLOBAL API SERVICE**

**Disusun Sebagai:**

**MATA KULIAH: PEMROGRAMAN BERBASIS FRAMEWORK**

**Oleh:**

**Moh. Khoirul Arifin/1741720051**

**TI-3B**



**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2020**

## 1. Praktikum

### a. Global API Service GET

#### 1) Pertanyaan

- a) Perhatikan file index.js, apa tujuan dibuatnya fungsi GetAPI pada baris 2 dan fungsi getNewsBlog pada baris 16?
  - Fungsi dari GetAPI pada baris kedua adalah untuk pemrosesan Pengambilan API. Pada GetAPI juga terlihat bahwa dia mnginput domainPath yang merupakan alamat API.
  - Fungsi getNewsBlog adalah untuk menampilkan yang telah diproses oleh GetAPI. Terlihat bahwa getNewsBlog memanggil GetAPI dan mengarahkan ke post dan diurutkan secara descending.

### b. Global API Service POST

#### 1) Pertanyaan

- a) Perhatikan file index.js, apa tujuan dibuatnya fungsi PostAPI dan fungsi postNewsBlog?
  - PostAPI ditujukan untuk memproses Post, disini juga terdapat menginputkan disini terdapat perbedaan dari Get tadi, disini terdapat method post untuk post data yang diisikan.
  - postNewsBlog merupakan tempat yang disediakan untuk BlogPost.jsx mengisikan inputan post dan diproses oleh PostAPI.
- b) Pada fungsi postNewsBlog, terdapat variable dataYangDiKirim. Apa tujuan dari dibuatnya variable tersebut?
  - Dibuatnya variabel tersebut untuk memasukkan data yang diinputkan kedalam PostAPI. Karena pada PostAPI terdapat aramete dengan nama data.

c. Global API Service DELETE

1) Pertanyaan

- a) Perhatikan file index.js, apa tujuan dibuatnya fungsi DeleteAPI dan fungsi deleteNewsBlog?
  - DeleteAPI berfungsi untuk menjalankan proses penghapusan. Dan deleteNewsBlog merupakan perintah delete yang terdapat pada BlogPost.jsx dan perintah itu akan dijalankan oleh DeleteAPI.
- b) Pada fungsi deleteNewsBlog, terdapat variable dataYangDiHapus. Apa tujuan dari dibuatnya variable tersebut?
  - Variabel tersebut bertujuan untuk tempat data yang akan dihapus di dalam PostAPI. Karena pada PostAPI terdapat parameter dengan nama data.

d. Manage Global API Service

1) Pertanyaan

- a) Bagaimana caranya untuk menambahkan daftar API baru baik untuk method GET, POST, DELETE pada Global API?
  - Untuk menambahkan daftar api baru, maka kita perlu untuk menambahkannya pada config.js dan diimportkan ke dalam masing-masing method.

## 2. Tugas

### a. Pembuatan React

```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu10> npx create-react-app tugas

Creating a new React app in G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu10\tugas.

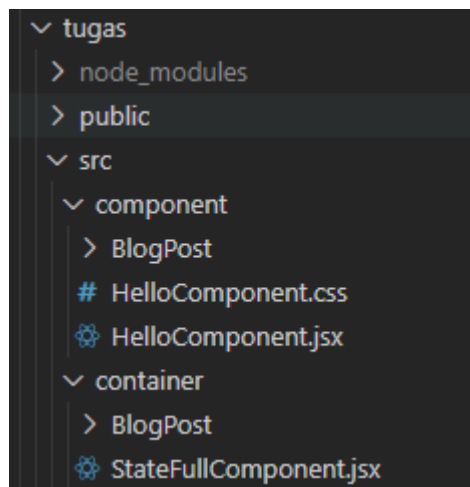
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

> core-js@2.6.11 postinstall G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu10\tugas\node_modules\babel-runtime\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"
```

### b. Install Bootstrap

```
PS G:\Kuliah\Semester6\PemrogramanBerbasisFramework\mohkhoirularifin-minggu10\tugas> npm install react-bootstrap bootstrap
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-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 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"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\watchpack\node_modules\fsevents):
```

### c. Copy component dan container dari tugas minggu 5



### d. Membuat file-file dalam folder services

#### 1) config.js

```
export const domainPath = 'http://localhost:3001';
```

#### 2) get.js

```
import {domainPath} from './config';

const GetAPI = (path) => {
  const promise = new Promise((resolve, reject) => {
    fetch(`${domainPath}/${path}`)
      .then(response => response.json())
      .then((result) => {
        resolve(result);
      });
  });
}
```

```

        }, (err) => {
            reject(err);
        })
    })
    return promise;
}

export default GetAPI;

```

### 3) post.js

```

import {domainPath} from './config'

const PostAPI = (path, data) => {
    const promise = new Promise((resolve, reject) => {
        fetch(`${domainPath}/${path}`, {
            method: 'post',
            headers: {
                'Accept': 'application/json',
                'Content-Type': 'application/json'
            },
            body: JSON.stringify(data)
        })
        .then((result) => {
            resolve(result);
        }, (err) => {
            reject(err);
        })
    })
    return promise;
}

export default PostAPI;

```

### 4) delete.js

```

import {domainPath} from './config'

const DeleteAPI = (path, data) => {
    const promise = new Promise((resolve, reject) => {
        fetch(`${domainPath}/${path}/${data}`, {
            method: 'DELETE'
        })
        .then((result) => {
            resolve(result);
        }, (err) => {
            reject(err);
        })
    })
    return promise;
}

export default DeleteAPI;

```

### 5) index.js

```

import GetAPI from './get'
import PostAPI from './post'
import DeleteAPI from './delete'

const getNewsBlog = () =>
    GetAPI('mahasiswa?_sort=id&_order=desc');
const postNewsBlog = (dataYangDikirim) => PostAPI('mahasiswa',
    dataYangDikirim);

```

```

const deleteNewsBlog = (dataYangDihapus) =>
DeleteAPI('mahasiswa', dataYangDihapus);

const API = {
  getNewsBlog,
  postNewsBlog,
  deleteNewsBlog
}

export default API;

```

#### e. Edit BlogPost.jsx

```

import React, {Component} from "react";
import './BlogPost.css';
import Post from "../../component/BlogPost/Post";
import API from "../../services";

class BlogPost extends Component {
  state = {
    listArtikel: [],
    insertArtikel: {
      id: 1,
      NIM: 1,
      nama: "",
      alamat: "",
      hp: "",
      angkatan: "",
      status: "",
    }
  }

  ambilDataDariServerApi = () => {
    API.getNewsBlog().then(result =>{
      this.setState({
        listArtikel: result
      })
    })
  }

  componentDidMount() {
    this.ambilDataDariServerApi()
  }

  handleHapusArtikel = (data) => {
    API.deleteNewsBlog(data)
      .then((response) => {
        this.ambilDataDariServerApi();
      })
  }

  handleTambahArtikel = (event) => {
    let formInsertArtikel = {...this.state.insertArtikel};
    let timestamp = new Date().getTime();
    formInsertArtikel['id'] = timestamp;
    formInsertArtikel[event.target.name] = event.target.value;
    this.setState({
      insertArtikel: formInsertArtikel
    })
  }
}

```

```

    handleTombolSimpan = () => {
      API.postNewsBlog(this.state.insertArtikel)
        .then((response) => {
          this.ambilDataDariServerApi();
        });
    }

    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">Nama</label>
              <div className="col-sm-10">
                <input type="text" className="form-control" id="nama" name="nama"
                  onChange={this.handleTambahArtikel}/>
              </div>
            </div>
            <div className="form-group row">
              <label htmlFor="body" className="col-sm-2 col-form-label">NIM</label>
              <div className="col-sm-10">
                <input type="text" className="form-control" id="NIM" name="NIM" rows="3"
                  onChange={this.handleTambahArtikel}/>
              </div>
            </div>
            <div className="form-group row">
              <label htmlFor="body" className="col-sm-2 col-form-label">Alamat</label>
              <div className="col-sm-10">
                <input type="text" className="form-control" id="alamat" name="alamat" rows="3"
                  onChange={this.handleTambahArtikel}/>
              </div>
            </div>
            <div className="form-group row">
              <label htmlFor="body" className="col-sm-2 col-form-label">No. Hp</label>
              <div className="col-sm-10">
                <input type="text" className="form-control" id="hp" name="hp" rows="3"
                  onChange={this.handleTambahArtikel}/>
              </div>
            </div>
            <div className="form-group row">
              <label htmlFor="body" className="col-sm-2 col-form-label">Angkatan</label>
              <div className="col-sm-10">
                <input type="text" className="form-control" id="angkatan" name="angkatan" rows="3"
                  onChange={this.handleTambahArtikel}/>
              </div>
            </div>
            <div className="form-group row">
              <label htmlFor="body" className="col-sm-2 col-form-label">Status</label>
              <div className="col-sm-10">

```

```

        <input type="text" className="form-
control" id="status" name="status" rows="3"
onChange={this.handleTambahArtikel}></input>
      </div>
    </div>
    <button type="submit" className="btn btn-
primary" onClick={this.handleTombolSimpan}>Simpan</button>
  </div>
  <h2>Daftar Mahasiswa</h2>
  {
    this.state.listArtikel.map(artikel => {
      return(
        <Post
          key={artikel.id}
          judul={artikel.nama}
          NIM={artikel.NIM}
          alamat={artikel.alamat}
          hp={artikel.hp}
          angkatan={artikel.angkatan}
          status={artikel.status}
          idArtikel={artikel.id}
          hapusArtikel={this.handleHapusArtikel} />
        )
      )
    })
  }
</div>
)
}
}
export default BlogPost;

```

f. Mengubah index.js pada src

```

import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import 'bootstrap/dist/css/bootstrap.min.css';
// import App from './App';
import BlogPost from './container/BlogPost/BlogPost';
import * as serviceWorker from './serviceWorker';

ReactDOM.render(
  <BlogPost />,
  document.getElementById('root')
);

// If you want your app to work offline and load faster, you can
// change
// unregister() to register() below. Note this comes with some
// pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();

```



## g. Output

### 1) Tampilan Awal

The screenshot shows a web application running on localhost:3000. It features a registration form with the following fields: Nama, NIM, Alamat, No. Hp, Angkatan, and Status. Below the form is a blue 'Simpan' button. Underneath the button is a section titled 'Daftar Mahasiswa' which displays a single student entry for Ryan Reinold. The entry includes a placeholder profile picture, the student's name, NIM (1765456678), address (Jl. Semanggi Utara No. 98), phone number (087767886544), graduation year (2016), and status (Anakatan).

Nama

NIM

Alamat

No. Hp

Angkatan

Status

Simpan

Daftar Mahasiswa

**Ryan Reinold**  
NIM : 1765456678  
alamat : Jl. Semanggi Utara No. 98  
No. Hp : 087767886544  
Anakatan : 2016

### 2) Setelah Mengalami Perubahan

#### a) Penambahan

This screenshot shows the same web application after a new student has been added. The registration form is now filled out with the following data: Nama (Putra Ramadhan), NIM (1741720876), Alamat (Ds. Pandean, Kecamatan Gondang Kabupaten Nganjuk), No. Hp (087865743221), Angkatan (2016), and Status (Lulus). The 'Simpan' button is still present. The 'Daftar Mahasiswa' section now lists two students: Ryan Reinold and the newly added Putra Ramadhan. The new entry includes a placeholder profile picture, name, NIM (1741720876), address (Ds. Pandean, Kecamatan Gondang Kabupaten Nganjuk), phone number (087865743221), graduation year (2016), status (Lulus), and a red 'Hapus' button.

Nama

NIM

Alamat

No. Hp

Angkatan

Status

Simpan

Daftar Mahasiswa

**Ryan Reinold**  
NIM : 1765456678  
alamat : Jl. Semanggi Utara No. 98  
No. Hp : 087767886544  
Angkatan : 2016  
Status : Lulus

**Putra Ramadhan**  
NIM : 1741720876  
alamat : Ds. Pandean, Kecamatan Gondang Kabupaten Nganjuk  
No. Hp : 087865743221  
Angkatan : 2016  
Status : Lulus  
Hapus

#### b) Penghapusan

This screenshot shows the web application after deleting a student. The registration form remains the same as in the previous state. The 'Daftar Mahasiswa' section now only displays the entry for Ryan Reinold, as the previously added student (Putra Ramadhan) has been removed from the list.

Nama

NIM

Alamat

No. Hp

Angkatan

Status

Simpan

Daftar Mahasiswa

**Ryan Reinold**  
NIM : 1765456678  
alamat : Jl. Semanggi Utara No. 98  
No. Hp : 087767886544  
Angkatan : 2016  
Status : Lulus  
Hapus