

**TUGAS INDIVIDU**

**LAPORAN MINGGU 10**

**“API”**

Disusun sebagai:

**MATA KULIAH: PEMROGRAMAN BERBASIS FRAMEWORK**

oleh:

**Serilda Wahyu Nofita**

**1741720177**

**TI-3B**



**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2020**

## Praktikum 1

- a. Perhatikan file `index.js`, apa tujuan dibuatnya fungsi `GetAPI` pada baris 2 dan fungsi `getNewsBlog` pada baris 16?
- Fungsi `GetAPI` adalah digunakan untuk mendapatkan data dari file JSON
  - Fungsi `getNewsBlog` adalah digunakan untuk mendapatkan dan mengurutkan data dari file json berdasarkan *id* dan secara *descending*

## Praktikum 2

1. Perhatikan file `index.js`, apa tujuan dibuatnya fungsi `PostAPI` dan fungsi `postNewsBlog`?
2. Pada fungsi `postNewsBlog`, terdapat variable `dataYangDiKirim`. Apa tujuan dari dibuatnya variable tersebut?
  1. Fungsi `PostAPI` untuk memasukkan data ke file JSON  
Fungsi `postNewsBlog` untuk mengirimkan data inputan ke fungsi `PostAPI`
  2. Membungkus data yang akan ditambahkan (inputan) ke dalam variable `dataYangDiKirim`

## Praktikum 3

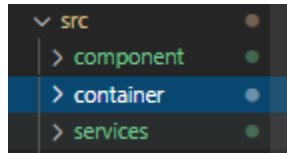
1. Perhatikan file `index.js`, apa tujuan dibuatnya fungsi `DeleteAPI` dan fungsi `deleteNewsBlog`?
2. Pada fungsi `deleteNewsBlog`, terdapat variable `dataYangDiHapus`. Apa tujuan dari dibuatnya variable tersebut?
  1. Fungsi `DeleteAPI` untuk menghapus data di JSON  
Fungsi `deleteNewsBlog` untuk menghapus data artikel yang dipilih ke fungsi `PostAPI`
  2. Untuk menghapus data yang akan dihapus dari artikel yang dipilih ke variable `dataYangDiHapus`

## Praktikum 4

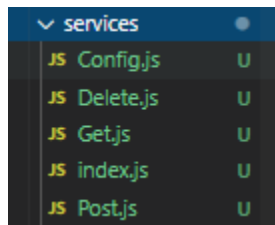
1. Bagaimana caranya untuk menambahkan daftar API baru baik untuk method GET, POST, DELETE pada Global API?
  - Membuat file baru kemudian meng-importnya pada `index.js`

## Tugas

1. Membuat folder **services** di dalam src



2. Membuat file **config.js**, **Delete.js**, **Get.js**, **index.js**, **Post.js** di dalam folder **services**



3. Config.js

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

4. Deleted.js

```
import { domainPath } from "../Config";

const DeleteAPI = (path, data) => {
  const promise = new Promise((resolve, reject) => {
    fetch(`${domainPath}/${path}/${data}`, {
      method: "DELETE",
      headers: {
        Accept: "application/json",
        "Content-Type": "application/json",
      },
    }).then(
      (result) => {
        resolve(result);
      },
      (err) => {
        reject(err);
      }
    );
  });
  return promise;
};
```

```
export default DeleteAPI;
```

## 5. 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;
```

## 6. Index.js

```
import GetAPI from "../Get";
import PostAPI from "../Post";
import DeleteAPI from "../Delete";

const getNewsBlog = () => GetAPI("posts?_sort=id&_order=desc");
const postNewsBlog = (dataYgDiKirim) => PostAPI("posts", dataYgDiKirim);
const deleteNewsBlog = (dataYgDiHapus) => DeleteAPI("posts", dataYgDiHapus);

const API = {
  getNewsBlog,
  postNewsBlog,
  deleteNewsBlog,
};

export default API;
```

## 7. 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;
```

## 8. Blogpost.jsx

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

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

```

ambilDataDariServerAPI = () => {
  API.getNewsBlog().then((result) => {
    this.setState({
      listArtikel: result,
    });
  });
};

componentDidMount() {
  this.ambilDataDariServerAPI();
}

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

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

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

render() {
  return (
    <div className="post-artikel">
      <div className="form pb-2 border-bottom">
        <div className="form-group row"></div>
        <label htmlFor="title" className="col-form-label">
          NIM
        </label>
        <div className="col-sm-10">
          <input

```

```
        type="text"
        className="form-nim"
        id="nim"
        name="nim"
        onChange={this.handleTambahArtikel}
    />
</div>
<div className="form-group row"></div>
<label htmlFor="title" className="col-form-label">
    Nama
</label>
<div className="col-sm-10">
    <input
        type="text"
        className="form-nama"
        id="nama"
        name="nama"
        onChange={this.handleTambahArtikel}
    />
</div>
<div className="form-group row"></div>
<label htmlFor="title" className="col-form-label">
    Alamat
</label>
<div className="col-sm-10">
    <input
        type="text"
        className="form-alamat"
        id="alamat"
        name="alamat"
        onChange={this.handleTambahArtikel}
    />
</div>
<div className="form-group row"></div>
<label htmlFor="title" className="col-form-label">
    Nomor HP:
</label>
<div className="col-sm-10">
    <input
        type="text"
        className="form-hp"
        id="hp"
        name="hp"
        onChange={this.handleTambahArtikel}
    />
</div>
```

```

    </div>
    <div className="form-group row"></div>
    <label htmlFor="title" className="col-form-label">
      Angkatan
    </label>
    <div className="col-sm-10">
      <input
        type="text"
        className="form-angkatan"
        id="angkatan"
        name="angkatan"
        onChange={this.handleTambahArtikel}
      />
    </div>
    <div className="form-group row"></div>
    <label htmlFor="title" className="col-form-label">
      Status
    </label>
    <div className="col-sm-10">
      <input
        type="text"
        className="form-status"
        id="status"
        name="status"
        onChange={this.handleTambahArtikel}
      />
    </div>
  </div>
  <button
    type="submit"
    className="btn btn-primary"
    onClick={this.handleTombolSimpan}
  >
    Simpan
  </button>
</h2>Daftar Mahasiswa</h2>
{this.state.listArtikel.map((artikel) => {
  return (
    <Post
      key={artikel.id}
      nim={artikel.id}
      nama={artikel.nama}
      alamat={artikel.alamat}
      hp={artikel.hp}
      angkatan={artikel.angkatan}
    >

```



```
        status={artikel.status}
        hapusArtikel={this.handleHapusArtikel}
      />
    );
  }}
</div>
);
}
}

export default BlogPost;
```

Sidebar

NIM

Nama

Alamat

Nomor HP:

Angkatan

Status

Simpan

Daftar Mahasiswa