

TUGAS INDIVIDU

MODUL 8

Disusun sebagai

MATA KULIAH : PEMROGRAMAN BERBASIS FRAMEWORK

Oleh :

M Jovanka Artha Ramandika / 1741720218

TI 3-A



**PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020**

A. Praktikum 1

1. Run project dan server fake API

```
Compiled successfully!

You can now view modul-8 in the browser.

Local:      http://localhost:3000
On Your Network:  http://10.209.232.211:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```

2. Membuat index.js pada folder services

3. Membuka File BlogPost.jsx

```
JS index.js  BlogPost.jsx x
src > container > BlogPost > BlogPost.jsx > BlogPost
1  import React, { Component } from "react";
2  import './BlogPost.css';
3  import Post from "../../component/BlogPost/Post";
4  import API from "../../services/index"
5
6  class BlogPost extends Component {
7    state = {                                // komponen state dari React untuk statefull component
8      listArtikel: [],                       // variabel array yang digunakan untuk menyimpan data API
9      insertArtikel: {                      // variable yang digunakan untuk menampung sementara data yang akan di
10        userId: 1,                          // kolom userId, id, title, dan body sama, mengikuti kolom yang ada ps
11        id: 1,
12        title: "",
13        body: ""
14      }
15    }
16
17    ambilDataDariServerAPI = () => {          // fungsi untuk mengambil data dari API dengan pena
18      API.getNewsBlog().then(result => {
19        this.setState({
```

4. Membuat Kode pada Index.js

```
src > services > JS index.js > ...
1  const domainPath = 'http://localhost:3001';
2  const GetAPI = (path) => {
3    const promise = new Promise((resolve, reject) => {
4      fetch(`${domainPath}/${path}`)
5        .then(response => response.json())
6        .then((result) => {
7          resolve(result);
8        }, (err) => {
9          reject(err);
10        })
11    })
12    return promise;
13  }
```

5. Mengubah baris 17 pada BlogPost.jsx



```
JS index.js • BlogPost.jsx ×
src > container > BlogPost > BlogPost.jsx > BlogPost
14      }
15    }
16
17    ambilDataDariServerAPI = () => {
18      API.getNewsBlog().then(result => {
19        this.setState({
20          listArtikel: result
21        })
22      })
23      // })
24    }
25
```

B. Praktikum 2

1. Buka fungsi handleTombolSimpan



```
JS index.js • BlogPost.jsx •
src > container > BlogPost > BlogPost.jsx > BlogPost > handleTombolSimpan
50
51    handleTombolSimpan = () => { // fungsi untuk meng-handle tombol simpan
52      fetch('http://localhost:3001/posts', {
53        method: 'post', // method POST untuk input/1
54        headers: {
55          'Accept': 'application/json',
56          'Content-Type': 'application/json'
57        },
58        body: JSON.stringify(this.state.insertArtikel) // kirimkan ke body request
59      })
60    }
61
```

2. Edit Fungsi tombolsimpan



```
JS index.js • BlogPost.jsx ×
src > container > BlogPost > BlogPost.jsx > BlogPost > handleTombolSimpan
51    handleTombolSimpan = () => { // fungsi untuk meng-
52      // fetch('http://localhost:3001/posts', {
53      //   method: 'post',
54      //   headers: {
55      //     'Accept': 'application/json',
56      //     'Content-Type': 'application/json'
57      //   },
58      //   body: JSON.stringify(this.state.insertArtikel)
59      // })
60      API.postNewsBlog(this.state.insertArtikel)
61      .then((response) => {
62        this.ambilDataDariServerAPI();
63      });
64
65
```

C. Praktikum 3

1. Buka Fungsi handleHapusArtikel

```
JS index.js • BlogPost.jsx •
src > container > BlogPost > BlogPost.jsx > BlogPost > handleHapusArtikel
29
30 handleHapusArtikel = (data) => { // fungsi yang meng-handle button
31   fetch(`http://localhost:3001/posts/${data}`, { method: 'DELETE' })
32   .then(res => { // ketika proses hapus berhasil, maka ambil
33     this.ambilDataDariServerAPI()
34   })
}
```

2. Membuat fungsi untuk menampung delete pada index.js

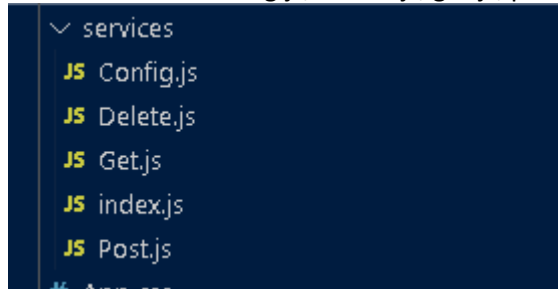
```
JS index.js • BlogPost.jsx •
src > services > JS index.js > DeleteAPI
31
32 const DeleteAPI = (path, data) => {
33   const promise = new Promise((resolve, reject) => {
34     fetch(`${domainPath}/${path}`)
35     .then(response => response.json())
36     .then((result) => {
37       resolve(result);
38     }, (err) => {
39       reject(err);
40     })
41   })
42 }
```

3. Edit Fungsi Handlehapusartikel

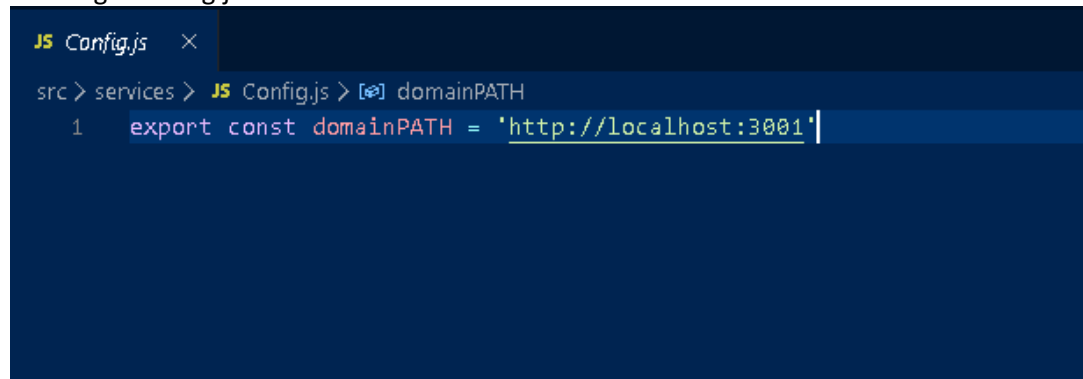
```
JS index.js • BlogPost.jsx ×
src > container > BlogPost > BlogPost.jsx > BlogPost > handleHapusArtikel
29
30 handleHapusArtikel = (data) => { // fungsi yang
31   // fetch(`http://localhost:3001/posts/${data}`, {
32   //   .then(res => { // ketika proses hapus
33   //     this.ambilDataDariServerAPI()
34   //   })
35   API.deleteNewBlog(data)
36   .then(response => { // ketika proses hapus
37     this.ambilDataDariServerAPI()
38   })
39 }
```

D. Praktikum 4

1. Membuat file config.js, delete.js, get.js, post.js dalam folder service



2. Mengisi Config.js



3. Mengisi Delete.js



4. Mengisi Get.js

```
JS Get.js  ×
src > services > JS Get.js > ...
1  import { domainPATH } from "../Config";
2
3  const GetAPI = (path) => {
4      const promise = new Promise((resolve, reject) => {
5          fetch(`${domainPath}/${path}`)
6              .then(response => response.json())
7              .then((result) => {
8                  resolve(result);
9              }, (err) => {
10                 reject(err);
11             })
12      })
13      return promise;
14  }
```

5. mengisi Post.js

```
JS Post.js  ×
src > services > JS Post.js > ...
1  import { domainPATH } from "../Config";
2
3  const PostAPI = (path) => {
4      const promise = new Promise((resolve, reject) => {
5          fetch(`${domainPath}/${path}`)
6              .method('post',
7              headers: {
8                  'accept' : 'application/json',
9                  'Content-Type'; 'application/json'
10             }
11             body; JSON.stringify(data)
12             .then((result) => {
13                 resolve(result);
14             }, (err) => {
```

6. Mengedit index.js

```
JS index.js ×
src > services > JS index.js > ...
44
45   const getNewsBlog = () => GetAPI('posts?_sort=id&_order=desc');
46   const postNewsBlog = (dataYgDiKirim) => PostAPI('posts',dataYgDiKirim);
47   const deleteNewsBlog = (dataYgDiHapus) => DeleteAPI('posts',dataYgDiHapus);
48
49   const API = {
50     getNewsBlog,
51     postNewsBlog,
52     deleteNewsBlog
53   }
54
55   export default API;
```