

TUGAS INDIVIDU

MODUL 8

Disusun sebagai

MATA KULIAH : PEMROGRAMAN BERBASIS FRAMEWORK

Oleh :

MUHAMMAD IQBAL KHARIS FIRISMANDA

1741720121

TI 3A/21



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

A. Praktikum 1

1. Run project dan server fake API
2. Membuat index.js pada folder services
3. Membuka file BlogPost.jsx

```
package.json BlogPost.jsx X
src > container > BlogPost > @ BlogPost.jsx > ...
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: { // variabel yang digunakan untuk menampung sementara data yang akan di insert
10       userId: 1, // kolom userid, id, title, dan body sama, mengikuti kolom yang ada pada listArtikel.json
11       id: 1,
12       title: "",
13       body: ""
14     }
15   }
16
17   ambilDataDariServerAPI = () => { // fungsi untuk mengambil data dari API dengan penambahan sort dan order
18     // fetch('http://localhost:3001/posts?_sort=id&_order=desc') // penambahan sort dan order berdasarkan parameter
19     // .then(response => response.json()) // ubah response data dari URL API menjadi sebuah data json
20     // .then(jsonHasilAmbilDariAPI => { // data json hasil ambil dari API kita masukkan ke dalam listArtikel pada state
21     API.getNewsBlog().then(result => {
22       this.setState({
23         listArtikel: result
24       })
25     })
26     // })
27   }
28
29   componentDidMount() { // komponen untuk mengecek ketika component telah di-mount-ing, maka panggil API
30     this.ambilDataDariServerAPI() // ambil data dari server API lokal
31   }
32
33   handleInsertArtikel = (data) => { // fungsi untuk menambahkan data ke dalam listArtikel
34     // ...
35   }
36 }
```

4. Membuat kode pada index.js

```
JS Config.js BlogPost.jsx JS index.js X
src > services > JS index.js > ...
1 import React from 'react';
2 import getAPI from './Get';
3 import PostAPI from './Post';
4 import DeleteAPI from './Delete';
5
6 const getNewsBlog = () => getAPI("posts?_sort=id&_order=asc")
7 const postNewsBlog = (dataYangDikirim) => PostAPI("posts", dataYangDikirim)
8 const deleteNewBlog = (dataYgDihapus) => DeleteAPI("posts", dataYgDihapus)
9
10 const API = {
11   getNewsBlog,
12   postNewsBlog,
13   deleteNewBlog
14 }
15
16 export default API;
```

5. Mengubah baris 17 pada BlogPost.jsx

```
ambilDataDariServerAPI = () => { // fungsi untuk mengambil
  // fetch('http://localhost:3001/posts?_sort=id&_order=desc') // penam
  // .then(response => response.json()) // ubah response data d
  // .then(jsonHasilAmbilDariAPI => { // data json hasil ambi
  API.getNewsBlog().then(result => {
    this.setState({
      listArtikel: result
    })
  })
  // })
}
```

Pertanyaan Praktikum 1

1. Perhatikan file 'index.js', apa tujuan dibuatnya fungsi 'GetAPI' pada baris 2 dan fungsi 'getNewsBlog' pada baris 16?
 - Digunakan untuk melakukan request data dari artikel, fungsi inilah yang digunakan untuk memanage kedalam satu tempat yaitu index.js

B. Praktikum 2

1. Buka fungsi handleTombolSimpan

```
handleTombolSimpan = () => { // fungsi untuk meng-handle tombol simpan
  fetch('http://localhost:3001/posts', {
    method: 'post', // method POST untuk input/insert data
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(this.state.insertArtikel) // kirimkan ke body request untuk data artikel y
  })
  // API.postNewsBlog(this.state.insertArtikel)
  // .then((response) => {
  //   this.ambilDataDariServerAPI(); // reload / refresh data
  // });
}
```

2. Mengedit fungsi handleTombolSimpan dan membuat post

```
handleTombolSimpan = () => { // fungsi untuk meng-handle tombol simpan
  // fetch('http://localhost:3001/posts', {
  //   method: 'post', // method POST unt
  //   headers: {
  //     'Accept': 'application/json',
  //     'Content-Type': 'application/json'
  //   },
  //   body: JSON.stringify(this.state.insertArtikel) // kirimkan ke body
  // })
  API.postNewsBlog(this.state.insertArtikel)
    .then((response) => {
      this.ambilDataDariServerAPI(); // reload / refresh da
    });
}
```

Pertanyaan Praktikum 2

1. Perhatikan file 'index.js', apa tujuan dibuatnya fungsi PostAPI dan fungsi 'postNewsBlog'?
 - Untuk mengirimkan data kepada server API
2. Pada fungsi 'postNewsBlog', terdapat variable 'dataYangDiKirim'. Apa tujuan dari dibuatnya variable tersebut?
 - Untuk menampung data sementara yang akan dikirim lalu dilanjutkan ke postAPI untuk meneruskannya kepada server API

C. Praktikum 3

1. Buka fungsi handleHapusArtikel

```
handleHapusArtikel = (data) => { // fungsi yang meng-handle button action hapus data
  fetch(`http://localhost:3001/posts/${data}`, { method: 'DELETE' }) // alamat URL API
    .then(res => { // ketika proses hapus berhasil, maka ambil data dari server API
      this.ambilDataDariServerAPI()
    })
  // API.deleteNewBlog(data)
  // .then(response => { // ketika proses hapus berhasil, maka ambil data dari
  //   this.ambilDataDariServerAPI()
  // })
}
```

2. Membuat fungsi untuk menampung delete pada index.js

```
JS Config.js X BlogPost.jsx JS Delete.js X
src > services > JS Delete.js > ...
1 import { domainPath } from './Config';
2
3 const DeleteAPI = (path, data) => {
4   const promise = new Promise((resolve, reject) => {
5     fetch(`${domainPath}/${path}/${data}`, {
6       method: "DELETE"
7     })
8     .then((result) => {
9       resolve(result)
10      }, err => {
11        reject(err)
12      })
13   })
14   return promise;
15 }
16
17 export default DeleteAPI;
```

3. Mengedit isi fungsi handleHapusArtikel

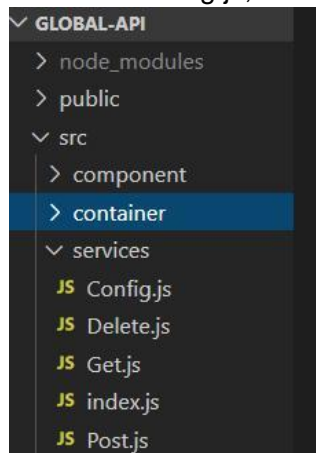
```
handleHapusArtikel = (data) => { // fu
  // fetch(`http://localhost:3001/posts/${data}`
  //   .then(res => { // ketika proses
  //     this.ambilDataDariServerAPI()
  //   })
  API.deleteNewBlog(data)
  .then(response => { // ketika pr
    this.ambilDataDariServerAPI()
  })
}
```

Pertanyaan Praktikum 3

1. Perhatikan file 'index.js', apa tujuan dibuatnya fungsi 'DeleteAPI' dan fungsi 'deleteNewsBlog'?
 - ☐ Untuk melakukan request hapus data pada server API
2. Pada fungsi 'deleteNewsBlog', terdapat variable 'dataYangDiHapus'. Apa tujuan dari dibuatnya variable tersebut?
 - ☐ Untuk menyimpan sementara data yang akan dihapus lalu diteruskan kepada deleteAPI yang nantinya akan melakukan proses penghapusan pada data di server API

D. Praktikum 4

1. Membuat Config.js, Delete.js, Get.js, Post.js pada folder services



2. Mengisikan kode pada Config.js

```
src > services > JS Config.js > ...  
1 | export const domainPath = "http://localhost:3001";
```

3. Mengisikan kode pada Delete.js

```
src > services > JS Delete.js > [⌕] default  
1 | import { domainPath } from './Config';  
2 |  
3 | const DeleteAPI = (path, data) => {  
4 |   const promise = new Promise((resolve, reject) => {  
5 |     fetch(`${domainPath}/${path}/${data}`, {  
6 |       method: "DELETE"  
7 |     })  
8 |     .then((result) => {  
9 |       resolve(result)  
10 |     }, err => {  
11 |       reject(err)  
12 |     })  
13 |   })  
14 |   return promise;  
15 | }  
16 |  
17 | export default DeleteAPI;
```

4. Mengisikan kode pada Get.js

```
src > services > JS Get.js > [⌕] default
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 | }
15 |
16 | export default getAPI;
```

5. Mengisikan kode pada Post.js

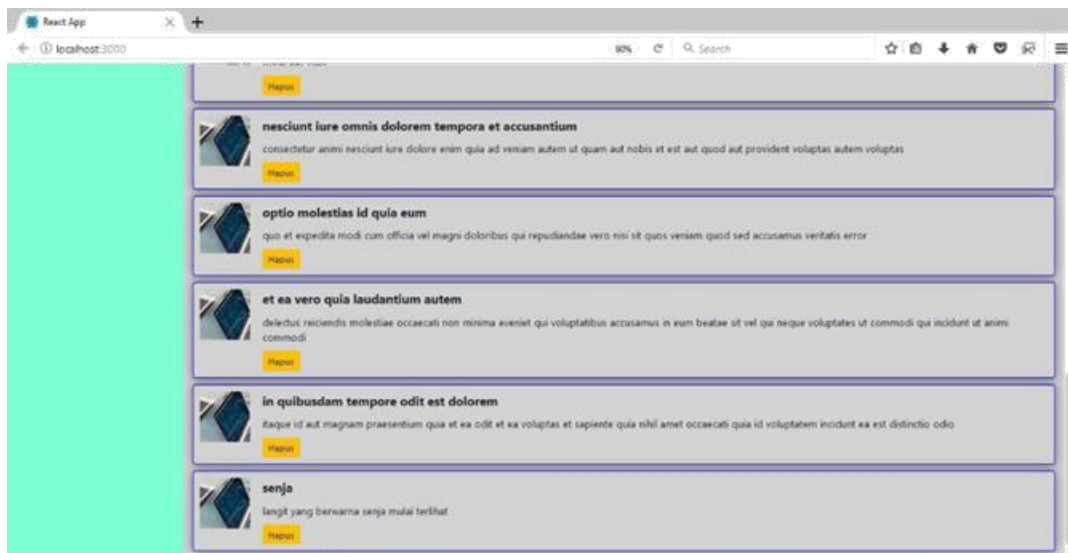
```
src > services > JS Post.js > [⌕] default
1 | import { domainPath } from './Config';
2 |
3 | const PostAPI = (path, data) => {
4 |   const promise = new Promise((resolve, reject) => {
5 |     fetch(`${domainPath}/${path}`, {
6 |       method: "post",
7 |       headers: {
8 |         'Accept': 'application/json',
9 |         'Content-Type': 'application/json'
10 |       },
11 |       body: JSON.stringify(data)
12 |     })
13 |       .then((result) => {
14 |         resolve(result)
15 |       }, err => {
16 |         reject(err)
17 |       })
18 |   })
19 |   return promise;
20 | }
21 |
22 | export default PostAPI;
```


6. Mengedit index.js

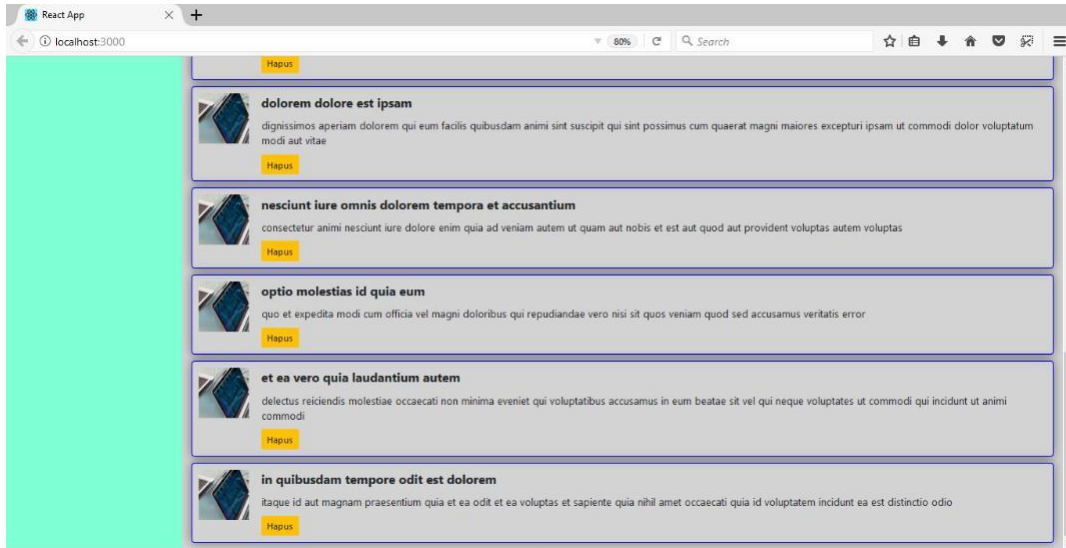
```
src > services > JS index.js > default
1  import React from 'react';
2  import getAPI from './Get';
3  import PostAPI from './Post';
4  import DeleteAPI from './Delete';
5
6  const getNewsBlog = () => getAPI("posts?_sort=id&_order=asc")
7  const postNewsBlog = (dataYangDikirim) => PostAPI("posts", dataYangDikirim)
8  const deleteNewBlog = (dataYgDihapus) => DeleteAPI("posts", dataYgDihapus)
9
10 const API = {
11   getNewsBlog,
12   postNewsBlog,
13   deleteNewBlog
14 }
15
16 export default API;
```

7. Hasil setelah dijalankan

- Setelah ditambahkan



- Setelah dihapus



Pertanyaan Praktikum 4

1. Bagaimana caranya untuk menambahkan daftar API baru untuk method GET,POST,DELETE pada Global API?

- Dengan cara membuat file baru yaitu GET, POST, DELETE dengan ekstensi .js