

TUGAS PERORANGAN/INDIVIDU

Minggu 10

Modul 8

Disusun sebagai:

MATA KULIAH: Pemrograman Berbasis Framework

oleh:

Ahmad Musyadad Aminullah / 1741720141

3B / 06



PROGRAM STUDI D-IV TEKNIK INFORMATIKA

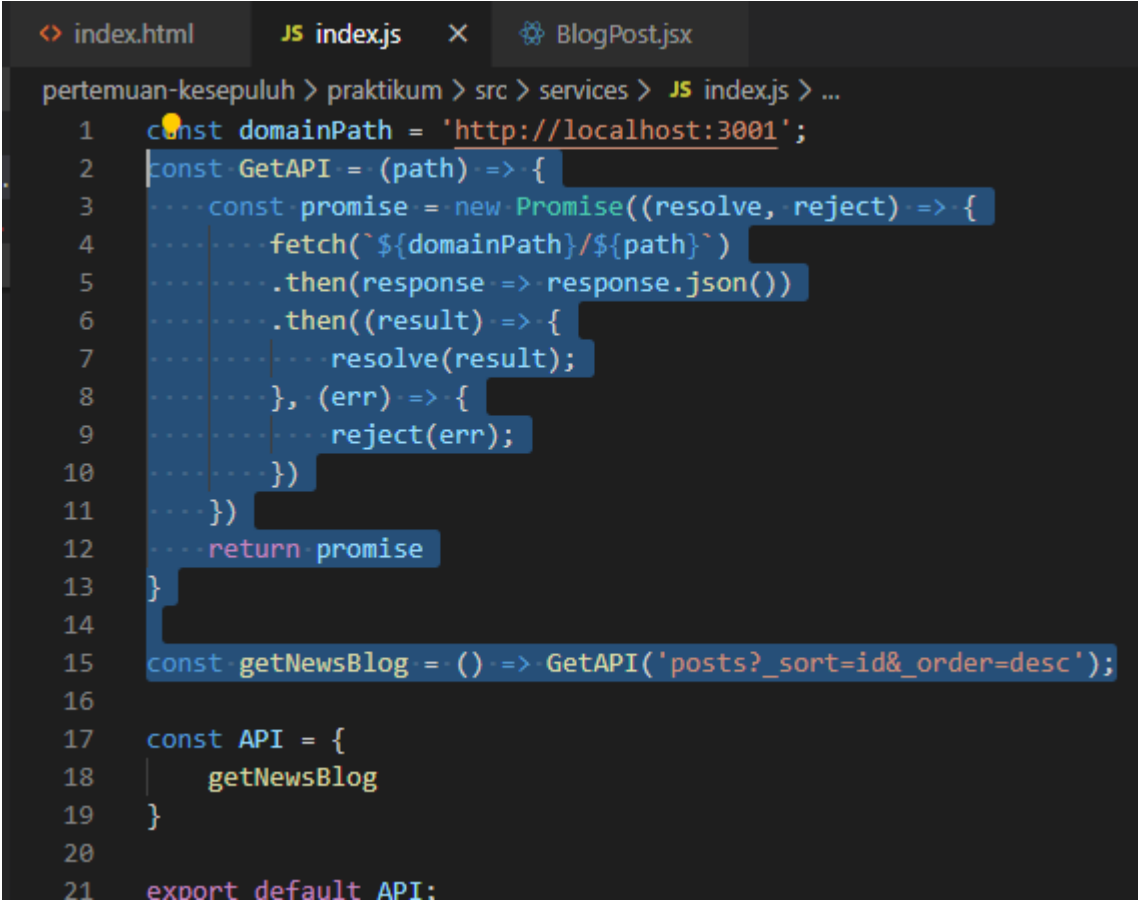
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

TAHUN 2020

1. Pertanyaan Praktikum 1

Di *index.js*, apa tujuan dibuatnya fungsi *GetAPI* pada baris 2 dan fungsi *getNewsBlog* pada baris 16?

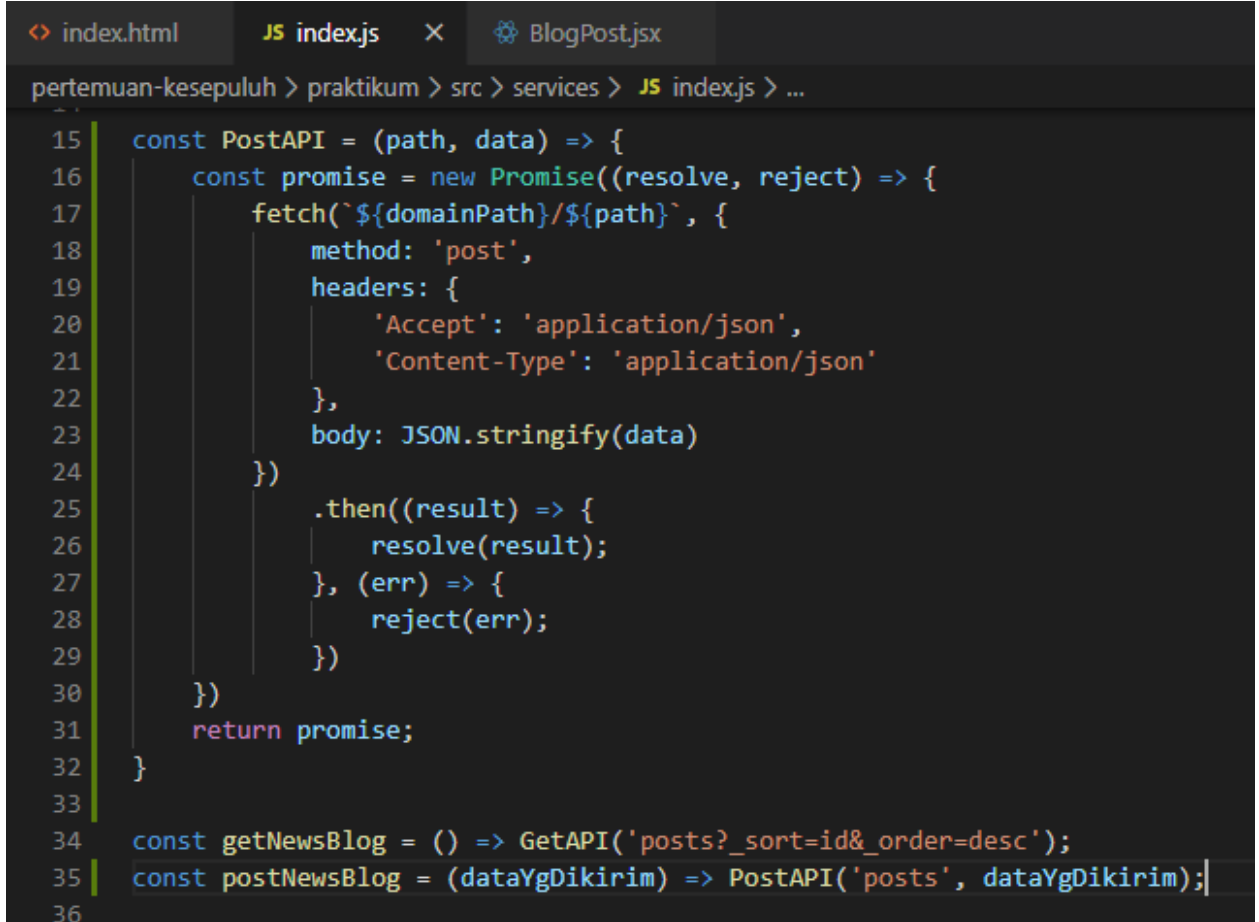


```
1  const domainPath = 'http://localhost:3001';
2  const GetAPI = (path) => {
3    const promise = new Promise((resolve, reject) => {
4      fetch(`${domainPath}/${path}`)
5        .then(response => response.json())
6        .then((result) => {
7          resolve(result);
8        }, (err) => {
9          reject(err);
10         })
11      })
12    return promise
13  }
14
15  const getNewsBlog = () => GetAPI('posts?_sort=id&_order=desc');
16
17  const API = {
18    getNewsBlog
19  }
20
21  export default API;
```

- Fungsi *GetAPI* digunakan untuk mendapatkan data dari file json
- Fungsi *getNewsBlog* digunakan untuk mendapatkan & mengurutkan data dari file json berdasarkan *id* dan secara *descending*

2. Pertanyaan Praktikum 2

1. Pada file *index.js*, apa tujuan dibuatnya fungsi *PostAPI* dan fungsi *postNewsBlog*?



```
index.html JS index.js X BlogPost.jsx
pertemuan-kese puluh > praktikum > src > services > JS index.js > ...

15 const PostAPI = (path, data) => {
16   const promise = new Promise((resolve, reject) => {
17     fetch(`${domainPath}/${path}`, {
18       method: 'post',
19       headers: {
20         'Accept': 'application/json',
21         'Content-Type': 'application/json'
22       },
23       body: JSON.stringify(data)
24     })
25     .then((result) => {
26       resolve(result);
27     }, (err) => {
28       reject(err);
29     })
30   })
31   return promise;
32 }
33
34 const getNewsBlog = () => GetAPI('posts?_sort=id&_order=desc');
35 const postNewsBlog = (dataYgDikirim) => PostAPI('posts', dataYgDikirim);
36
```

- Fungsi *PostAPI* digunakan untuk memasukkan data ke file json
 - Fungsi *postNewsBlog* digunakan untuk mengirimkan data inputan ke fungsi *PostAPI*
2. Pada fungsi *postNewsBlog*, terdapat variable *dataYangDiKirim*. Apa tujuan dari dibuatnya variable tersebut?
 - Membungkus data yang akan ditambahkan (data inputan) ke dalam variable *dataYangDiKirim*

3. Pertanyaan Praktikum 3

1. Di file *index.js*, apa tujuan dibuatnya fungsi *DeleteAPI* dan fungsi *deleteNewsBlog*?

```
34 | const DeleteAPI = (path, data) => {
35 |   const promise = new Promise((resolve, reject) => {
36 |     fetch(`${domainPath}/${path}/${data}`, {method: 'DELETE'})
37 |       .then((result) => {
38 |         resolve(result);
39 |       }, (err) => {
40 |         reject(err);
41 |       })
42 |   })
43 | }
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 |
```

- Fungsi *DeleteAPI* digunakan untuk menghapus data di file json
 - Fungsi *deleteNewsBlog* digunakan untuk menghapus data dari artikel yang dipilih ke fungsi *PostAPI*
2. Pada fungsi *deleteNewsBlog*, terdapat variable *dataYangDiHapus*. Apa tujuan dari dibuatnya variable tersebut?
 - Untuk membungkus data yang akan dihapus dari artikel yang dipilih ke variabel *dataYangDiHapus*,

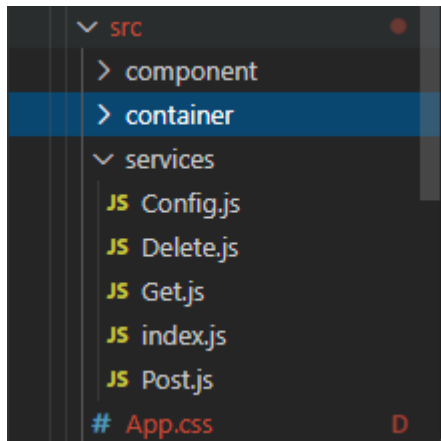
4. Pertanyaan Praktikum 4

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

- Menambahkan API baru di folder service, dan di import di file *index.js*

Tugas

- Membuat folder *services* dan membuat file *index.js*, *Config.js*, *Get.js*, *Post.js*, dan *Delete.js*



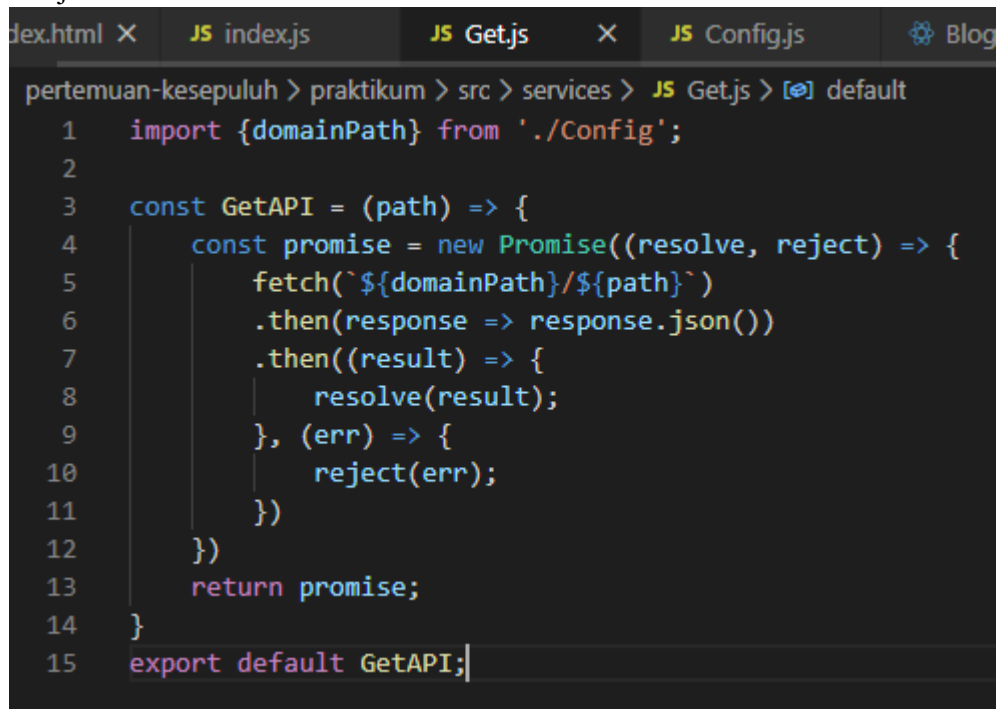
- index.js*

```
ex.js ...\praktikum\... JS index.js ...\tugas\... X BlogPost.jsx ...\praktikum\... BlogPost.jsx ...\tugas\...
pertemuan-keseperuluh > tugas > src > services > JS index.js > default
1  import GetAPI from "../Get";
2  import PostAPI from "../Post";
3  import DeleteAPI from "../Delete";
4
5  // Daftar API - Get
6  const getNewsBlog = () => GetAPI('mahasiswa?_sort=id&_order=desc');
7
8  // Daftar API - Post
9  const postNewsBlog = (dataYgDikirim) => PostAPI('mahasiswa', dataYgDikirim);
10
11 // Daftar API - Delete
12 const deleteNewsBlog = (dataYgDihapus) => DeleteAPI('mahasiswa', dataYgDihapus);
13
14 const API = {
15   getNewsBlog,
16   postNewsBlog,
17   deleteNewsBlog
18 }
19
20 export default API;
```

- Config.js*

```
index.html JS index.js JS Config.js X BlogPost.jsx ...\pra
pertemuan-keseperuluh > praktikum > src > services > JS Config.js > ...
1  export const domainPath = 'http://localhost:3001';
```

- Get.js



The screenshot shows a code editor with several tabs at the top: 'dex.html', 'JS index.js', 'JS Get.js', 'JS Config.js', and 'Blog'. The 'JS Get.js' tab is active. The editor displays the following JavaScript code:

```
pertemuan-kesepuluh > praktikum > 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  export default GetAPI;
```

- Post.js

```
JS Get.js JS Post.js X JS Config.js BlogPost.js
pertemuan-kesepuluh > praktikum > 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 export default PostAPI;
```

- Delete.js

```
JS Get.js JS Post.js JS Delete.js X JS Config.js BlogPost.js
pertemuan-kesepuluh > praktikum > 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}`, {method: 'DELETE'})
6       .then((result) => {
7         resolve(result);
8       }, (err) => {
9         reject(err);
10      })
11   })
12   return promise;
13 }
14 export default DeleteAPI;
```

- Mengubah script file *BlogPost.jsx* di folder *container*

```
BlogPost.jsx ...\tugas\... X JS index.js ...\tugas\...
bertemuan-kesepuluh > tugas > src > container > BlogPost > BlogPost.jsx >
1 import React, {Component} from "react";
2 import './BlogPost.css';
3 import Post from "../../component/BlogPost/Post";
4
5
```

Mengimport *API* dari *services*

```
BlogPost.jsx ...\tugas\... JS index.js ...\tugas\...
bertemuan-kesepuluh > tugas > 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";
5
```


Untuk fungsi *ambilDataDariServerAPI*

```
ex.js ...\praktikum\... BlogPost.jsx ...\tugas\... JS index.js ...\tugas\...
pertemuan-kese puluh > tugas > src > container > BlogPost > BlogPost.jsx

17     }
18   }
19
20   ambilDataDariServerAPI = () => {
21     fetch('http://localhost:3001/mahasiswa')
22     .then(response => response.json()) // U
23     .then(jsonHasilAmbilDariAPI => { // D
24       pada state
25       this.setState({
26         listArtikel: jsonHasilAmbilDariAPI
27       })
28     })
29   }
```

Dirubah menjadi

```
ex.js ...\praktikum\... BlogPost.jsx ...\tugas\... JS index.js
pertemuan-kese puluh > tugas > src > container > BlogPost > BlogPost.jsx

17     }
18   }
19
20   ambilDataDariServerAPI = () => {
21     API.getNewsBlog().then(result => {
22       this.setState({
23         listArtikel: result
24       })
25     })
26   }
27 }
```

Untuk fungsi *handleHapusArtikel*

```
BlogPost.jsx ...\tugas\... JS index.js ...\tugas\... JS Get.js BlogPos
pertemuan-keseperuluh > tugas > src > container > BlogPost > BlogPost.jsx > BlogPost > ambilDataDariServe
29     this.ambilDataDariServerAPI() // Ambil data dari server API lokal
30 }
31
32     handleHapusArtikel = (data) => {
33         fetch(`http://localhost:3001/mahasiswa/${data}`, {method: 'DELETE'})
34         .then(res => {
35             this.ambilDataDariServerAPI(console.log(this.ambilDataDariServerAPI))
36         })
37     }
38
```

Dirubah menjadi

```
BlogPost.jsx ...\tugas\... JS index.js ...
pertemuan-keseperuluh > tugas > src > container > BlogPost > BlogPost.jsx > BlogPost > ambilDataDariServe
29     this.ambilDataDariServerAPI() //
30 }
31
32     handleHapusArtikel = (data) => {
33         API.deleteNewsBlog(data)
34         .then((response) => {
35             this.ambilDataDariServerAPI();
36         })
37     }
38
```

Untuk fungsi *handleTombolSimpan*

```
BlogPost.jsx ...\tugas\... JS index.js ...\tugas\... JS
pertemuan-kesepuluh > tugas > src > container > BlogPost > BlogPost.jsx > BlogP

48
49     handleTombolSimpan = () => {
50         fetch('http://localhost:3001/mahasiswa', {
51             method: 'post',
52             headers: {
53                 'Accept': 'application/json',
54                 'Content-Type': 'application/json'
55             },
56             body: JSON.stringify(this.state.insertArtikel)
57         })
58         .then((Response) => {
59             this.ambilDataDariServerAPI();
60         })
61     }
62
```

Dirubah menjadi

```
BlogPost.jsx ...\tugas\... JS index.js ...\tugas\... JS
pertemuan-kesepuluh > tugas > src > container > BlogPost > BlogPost.jsx

48
49     handleTombolSimpan = () => {
50         API.postNewsBlog(this.state.insertArtikel)
51         .then((response) => {
52             this.ambilDataDariServerAPI();
53         })
54     }
55
```