

TUGAS PERORANGAN/INDIVIDU

Modul 8

Global API

Disusun sebagai

MATA KULIAH : Pemrograman Berbasis Framework

Oleh :

Mohammad Fauzan Arramadhan

TI-3A / 19

1741720212



PROGRAM STUDI D-IV T.INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

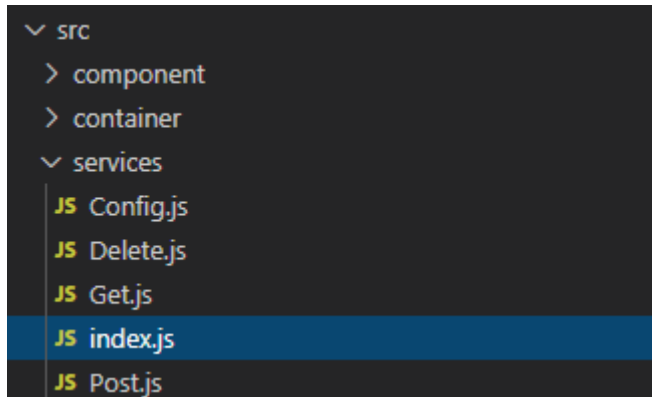
POLITEKNIK NEGERI MALANG

2020

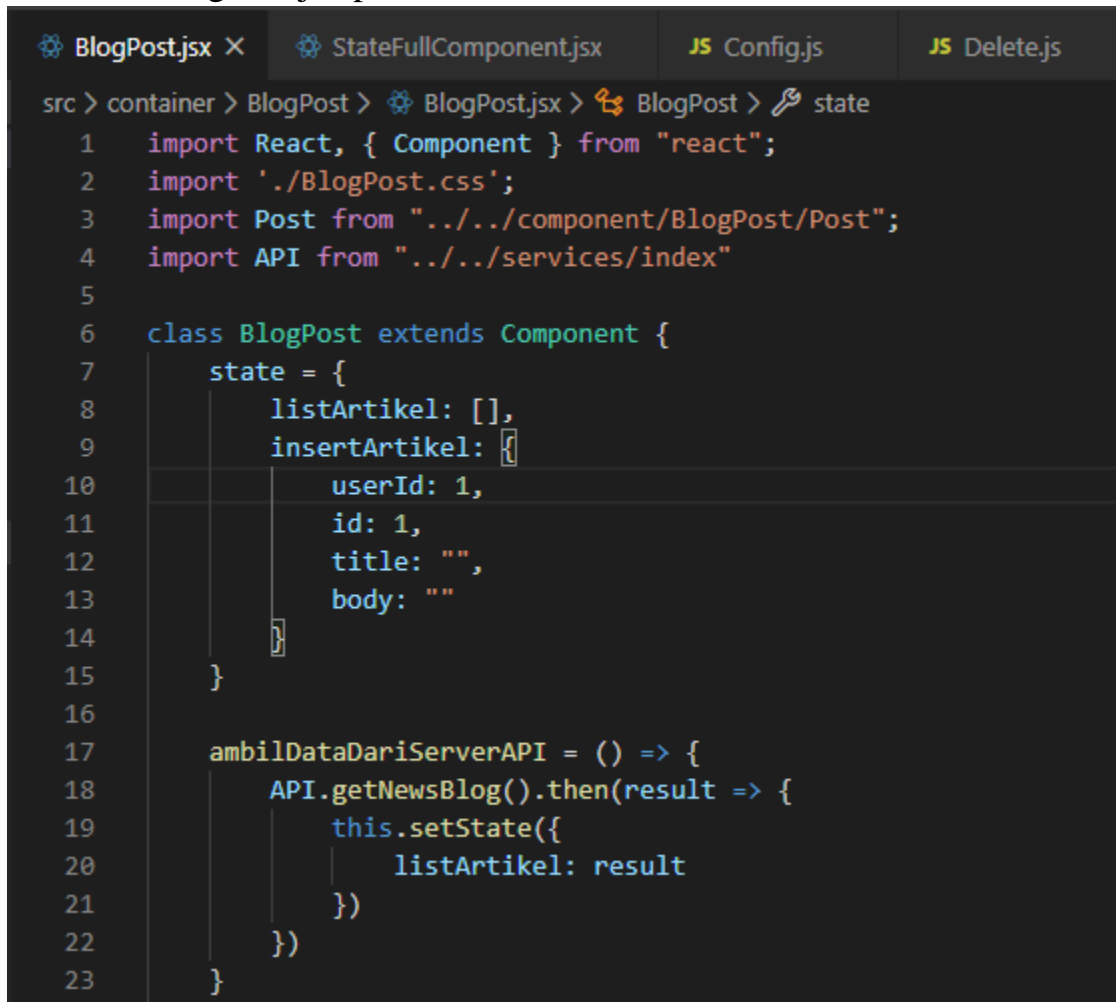
PRAKTIKUM

1. Langkah Praktikum

- 1) Membuat folder services dan menambahkan index.js pada folder tersebut



- 2) Buka file BlogPost.jsx pada folder container



- 3) Buka fungsi ambilDataDariServerAPI

```

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

```

4) Buka file index.js

```

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

```

5) Kembali pada file BlogPost.jsx

```

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

```

Pertanyaan

- Berfungsi untuk melakukan fetching pada data atau pada API kemudian di kembalikan untuk di olah pada sisi client side

2. Langkah Praktikum 2

- Fungsi handleTombolSimpan pada file BlogPost.js

```

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

```

- 2) Membuat fungsi untuk handle Post Action pada index.js di folder services

```
const postNewsBlog = (dataYangDikirim) => PostAPI("posts", dataYangDikirim)
```

- 3) Pada BlogPost.jsx ganti isi handleTombolSimpan

```

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

```

Pertanyaan

- a. Menjadi jembatan untuk memproses inputan user yang akan diteruskan pada API dan di simpan dalam database
- b. Sebagai variable yang menyimpan data yang di input user berbentuk object lalu di teruskan untuk di simpan dalam database

3. Langkah Praktikum 3

- 1) Mengelola handleHapusArtikel

```

    handleHapusArtikel = (data) => {
      API.deleteNewBlog(data)
        .then(response => {
          this.ambilDataDariServerAPI()
        })
    }

```

- 2) Fungsi untuk menampung action DELETE dari file BlogPost.jsx pada folder services/index.js

```
const deleteNewBlog = (dataYgDihapus) => DeleteAPI("posts", dataYgDihapus)
```

- 3) Selesaikan method handleHapusArtikel

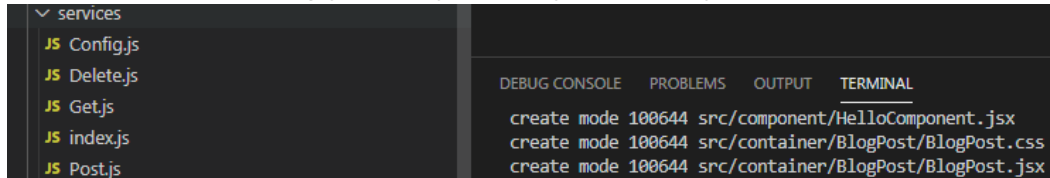
```
handleHapusArtikel = (data) => {  
  API.deleteNewBlog(data)  
    .then(response => {  
      this.ambilDataDariServerAPI()  
    })  
}
```

Pertanyaan Praktikum 3

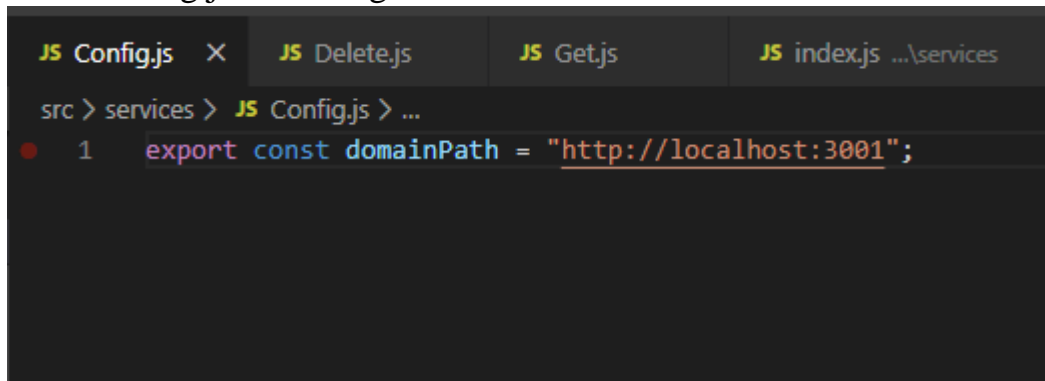
- Menghapus artikel yang dipilih
- Untuk mengirim data mana yang akan di hapus dan selanjutnya di proses di sisi API

4. Langkah Praktikum 4

- 1) Membuat file Config.js, Get.js, Post.js, Delete.js, dalam folder services



- 2) Buka Config.js dan mengisikan code



- 3) Buka Get.js dan isikan code

```

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  }
15
16  export default getAPI;

```

4) Buka Post.js dan isikan code

```

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;

```

5) Buka Delete.js dan isikan code

```

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;

```

6) Edit index.js

```

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

```

Pertanyaan Praktikum 4

- a. Tinggal membuat sebuah file yang bernama GET, POST, DELETE dan selanjutnya dikoneksikan dengan folder index.js