

TUGAS PERORANGAN

GLOBAL API

Disusun sebagai
MATA KULIAH: PBF

Oleh:

Rois Dwi Admaja/1741720193

TI – 3B / 23



PROGRAM STUDI: D-IV TEKNIK INFORMATIKA

JURUSAN: TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

Praktikum 1

```
rois-minggu5-api > src > services > JS index.js > [⌕] default
1  const domainPath = 'http://localhost:4000';
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;
```

Pada praktikum 1 dibuat index.js yang berfungsi untuk melakukan pengampilan data dari fake api yang telah dibuat menggunakan metode promise.

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

Lalu pada file BlogPost.jsx mengubah ambilDataDariServerAPI dengan memanggil component API dan memanggil fungsi getNewsBlog.

Jawaban pertanyaan praktikum 1

GetAPI merupakan fungsi yang digunakan untuk memodelkan data yang di ambil dari server API sedangkan getNewsBlog digunakan untuk menampilkan data dan mengurutkannya secara descending sehingga data yang baru dimasukkan akan berada diatas.

Praktikum 2

```
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     }).then((result) =>{
25       resolve(result);
26     }, (err) => {
27       reject(err);
28     })
29   })
30   return promise;
31 }
32
33 const getNewsBlog = () => GetAPI('posts?_sort=id&_order=desc');
34 const postNewsBlog = (dataygdir kirim) =>PostAPI('posts', dataygdir kirim)
35
36 const API = {
37   getNewsBlog,
38   postNewsBlog
39 }
40
```

Fungsi PostAPI digunakan untuk menyimpan data dari inputan ke API.

```
37 handleTombolSimpan = () =>{
38   API.postNewsBlog(this.state.insertArtikel)
39   .then((response) => {
40     this.ambilDataDariServerAPI();
41   })
42 }
```

Lalu pada handle simpan memanggil API dan fungsi postNewsBlog jadi Ketika tombol diklik maka akan menjalankan fungsi post API

Jawaban pertanyaan praktikum 2

1. postAPI digunakan untuk mengirim data dari inputan form ke API sedangkan postnewsblog digunakan untuk menyimpan data ke server API
2. variable tersebut menyimpan data yang akan disimpan.

Praktikum 3

Jawaban Praktikum 3

```
33 const DeleteAPI = (path, data) =>{
34   const promise = new Promise((resolve, reject)=>{
35     fetch(`${domainPath}/${path}`, {method: 'DELETE'
36   }).then((result) =>{
37     resolve(result);
38   }, (err) => {
39     reject(err);
40   })
41 })
42 return promise;
43 }
44
45 const getNewsBlog = () => GetAPI('posts?_sort=id&_order=desc');
46 const postNewsBlog = (dataygdiririm) =>PostAPI('posts', dataygdiririm)
47 const deleteNewsBlog = (dataygdihapus) =>DeleteAPI('posts', dataygdihapus)
48
49 const API = {
50   getNewsBlog,
51   postNewsBlog,
52   deleteNewsBlog
53 }
```

pada delete API terdapat method delete yang digunakan untuk menghapus data dan pada deleteNewBlog merupakan fungsi untuk mengambil data yang akan dihapus.

```
44   handleHapusArtikel = (data) => {
45     API.postNewsBlog(data)
46     .then((response) => {
47       this.ambilDataDariServerAPI();
48     })
49   }
```

handleHapus merupakan sebuah fungsi untuk menyambungkan antara button hapus dan fungsi untuk menghapus data.

Praktikum 4

```
1 export const domainPath = 'http://localhost:4000';|
```

Pada file config terdapat domainPath yang berfungsi untuk menyimpan domain dari fake API yang telah dijalankan.

```
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;
```

Lalu pada file Get.js terdapat fungsi untuk mengambil data dari API dan menampilkannya

```

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         }).then((result)=>{
13             resolve(result);
14         }, (err) =>{
15             reject(err);
16         })
17     })
18
19     return promise;
20 }
21
22 export default PostAPI;

```

File post.js terdapat fungsi untuk melakukan pengiriman data ke API

```

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              resolve(err);
10         })
11     })
12     return promise;
13 }
14
15 export default DeleteAPI;

```

Delete.js digunakan untuk menghapus data dari API

```

1 import GetAPI from './Get';
2 import PostAPI from './Posts';
3 import DeleteAPI from './Delete'
4
5 const getNewsBlog = () => GetAPI('posts?_sort=id&_order=desc');
6 const postNewsBlog = (dataygdir kirim) => PostAPI('posts', dataygdir kirim)
7 const deleteNewsBlog = (dataygdihapus) => DeleteAPI('posts', dataygdihapus)
8
9 const API = {
10   getNewsBlog,
11   postNewsBlog,
12   deleteNewsBlog
13 }
14
15 export default API;

```

Pada file index.js ini hanya melakukan pemanggilan dari fungsi GET, POST dan DELETE.

Pertanyaan praktikum 4

1. untuk menambahkan API baru cukup mengganti API pada file config.js

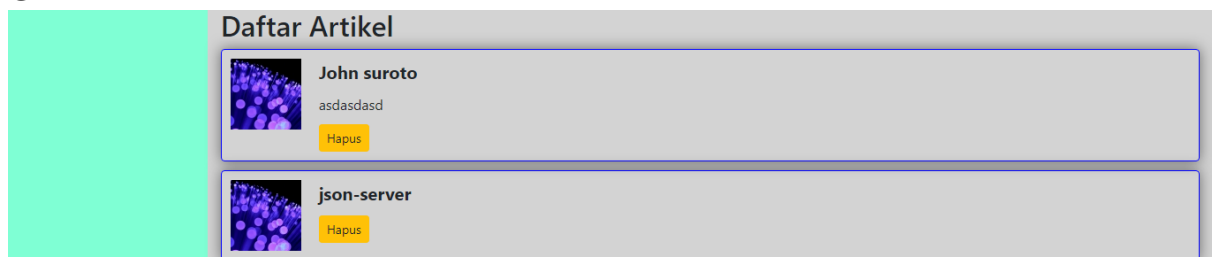
```

1 export const domainPath = 'http://localhost:4000';

```

HASIL PRAKTIKUM

1. GET



berikut hasil yang diambil dari server API

2. POST

Sidebar

Judul

artikel baru

Isi

ini artikel baru

Simpan

Daftar Artikel

 John suroto
asdasd
Hapus

 json-server
Hapus

Sesudah dissave

Sidebar

Judul


artikel baru


Isi


ini artikel baru

Simpan

Daftar Artikel

 artikel baru
ini artikel baru
Hapus

 John suroto
asdasd
Hapus

 json-server
Hapus

Artikel baru akan muncul diatas karena pada fungsi get terdapat sorting nilai tertinggi ditampilkan diatas

3. DELETE

Sidebar

Judul


artikel baru


Isi


ini artikel baru

Simpan

Daftar Artikel

 artikel baru
ini artikel baru
Hapus

 John suroto
asdasd
Hapus

 json-server
Hapus

Sebelum dihapus


Sidebar

Judul


Isi

Simpan

Daftar Artikel

 **artikel baru**
ini artikel baru

Hapus

 **John suroto**
asdasdasd

Hapus

Setelah dihapus