

TUGAS INDIVIDU/PERORANGAN

Global API

Disusun sebagai

MATA KULIAH : PEMROGRAMAN BERBASIS FRAMEWORK



Oleh:

Arifanny Ramadhan Sukma

1741720217 / TI-3A

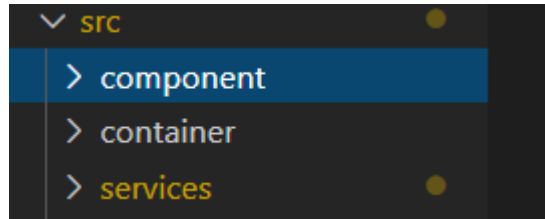
04

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

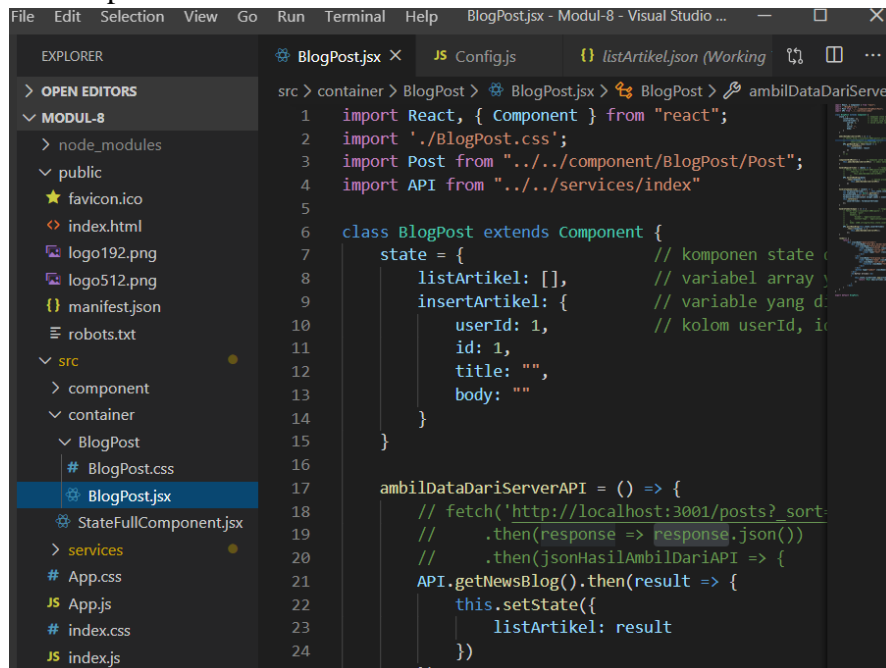
PRAKTIKUM

1. Praktikum 1 (Global API service GET)

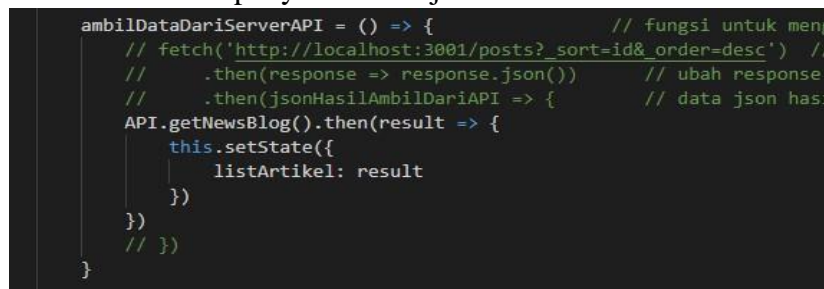
- 1) Buat folder baru bernama “Services” dalam folder src, kemudian buat file index.js.



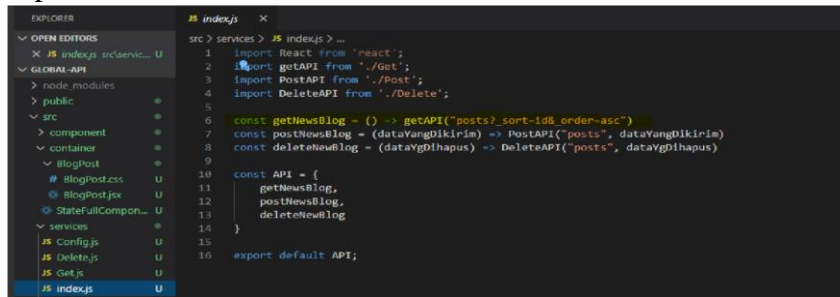
- 2) Buka file BlogPost.jsx pada folder container (*statefull component*) dan akan tampil kode seperti berikut

A screenshot of a code editor showing the 'BlogPost.jsx' file. The file is located in the 'src > container > BlogPost' directory. The code defines a 'BlogPost' class that extends 'Component'. It has a state with 'listArtikel' and 'insertArtikel' methods. The 'ambilDataDariServerAPI' function is defined, which uses 'fetch' to get data from 'http://localhost:3001/posts?_sort=id&_order=desc' and updates the state with 'listArtikel'.

- 3) Pada fungsi ambilDataDariServerAPI terdapat pemanggilan API GET untuk melakukan request data artikel. Proses dari fungsi inilah yang akan kita manage kedalam satu tempat yaitu index.js

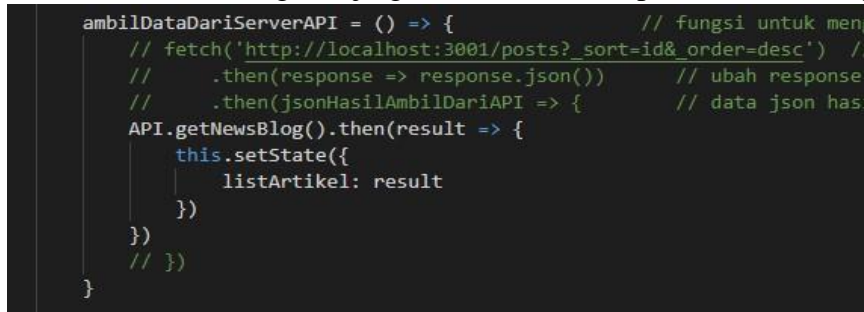
A screenshot of a code editor showing the 'index.js' file. The code defines the 'ambilDataDariServerAPI' function, which uses 'fetch' to get data from 'http://localhost:3001/posts?_sort=id&_order=desc' and updates the state with 'listArtikel'.

- 4) Buka file index.js pada folder service, yang telah kita buat tadi dan tuliskan baris kode seperti.



```
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 deleteNewsBlog = (dataYangDihapus) => DeleteAPI("posts", dataYangDihapus)
9
10 const API = {
11   getNewsBlog,
12   postNewsBlog,
13   deleteNewsBlog
14 }
15
16 export default API;
```

- 5) Kembali ke file BlogPost.js ganti baris 17 sampai baris 23 menjadi seperti berikut



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

- 6) Simpan file BlogPost.js dan index.js tersebut, dan amati apa yang terjadi pada browser.

Pertanyaan

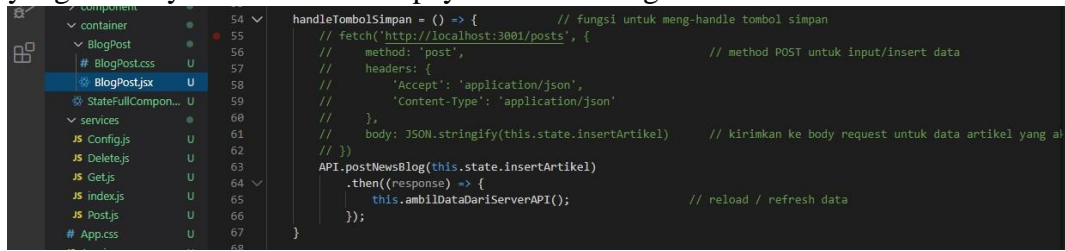
- a. Perhatikan file index.js, apa tujuan dibuatnya fungsi GetAPI pada baris 2 dan fungsi getNewsBlog pada baris 16?

Jawab :

berfungsi untuk merequest data artikel. Proses dari fungsi inilah yang akan kita manage kedalam satu tempat yaitu index.js

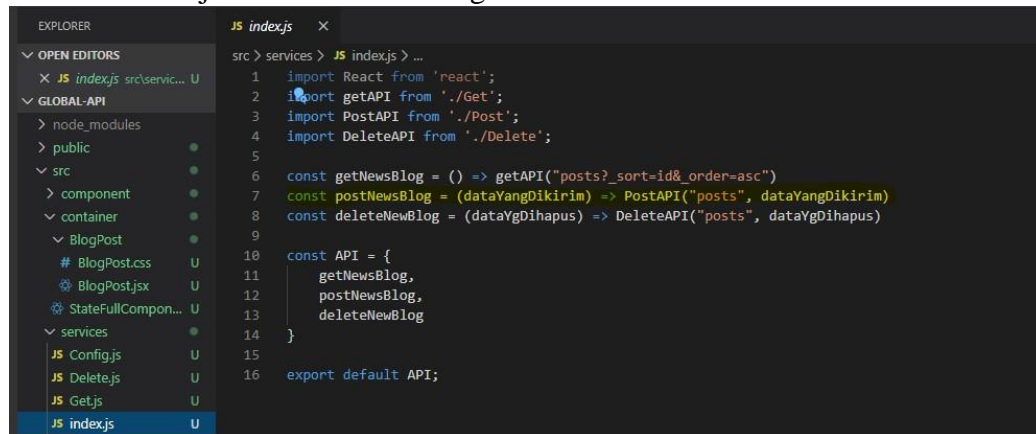
2. Praktikum 2 (Global API service POST)

- 1) Perhatikan fungsi `handleTombolSimpan` pada file `BlogPost.js`. kemudian bagian inilah yang nantinya akan dikelola supaya bisa di manage.



```
54 // fungsi untuk meng-handle tombol simpan
55 // fetch('http://localhost:3001/posts', {
56 //   method: 'post', // method POST untuk input/insert data
57 //   headers: {
58 //     'Accept': 'application/json',
59 //     'Content-Type': 'application/json'
60 //   },
61 //   body: JSON.stringify(this.state.insertArtikel) // kirimkan ke body request untuk data artikel yang al
62 // })
63 API.postNewsBlog(this.state.insertArtikel)
64 .then((response) => {
65   this.ambilDataDariServerAPI(); // reload / refresh data
66 });
67 }
68
```

- 2) Membuat fungsi untuk menampung action POST dari file BlogPost.js pada file services/index.js dan inialisasi fungsi tersebut.



```
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;
```

- 3) Selanjutnya pindah ke file BlogPost.js dan ganti isi dari fungsi handleTombolSimpan menjadi seperti berikut



```
54 handleTombolSimpan = () => { // fungsi untuk meng-handle tombol simpan
55   // fetch('http://localhost:3001/posts', {
56   //   method: 'post', // method POST untuk input/insert data
57   //   headers: {
58   //     'Accept': 'application/json',
59   //     'Content-Type': 'application/json'
60   //   },
61   //   body: JSON.stringify(this.state.insertArtikel) // kirimkan ke body request untuk data artikel yang a
62   // })
63   API.postNewsBlog(this.state.insertArtikel)
64   .then((response) => {
65     this.ambilDataDariServerAPI(); // reload / refresh data
66   });
67 }
68
```

- 4) Nantinya kode program menjadi sedikit lebih simple dari sebelumnya.

Pertanyaan

- a. Perhatikan file index.js, apa tujuan dibuatnya fungsi PostAPI dan fungsi postNewsBlog?

Jawab :

untuk mengirimkan data kepada server API.

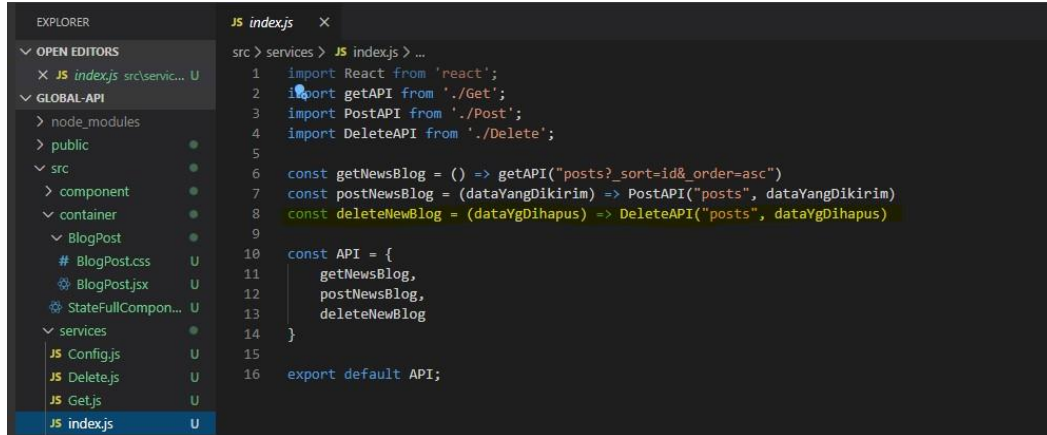
- b. Pada fungsi postNewsBlog, terdapat variable dataYangDiKirim. Apa tujuan dari dibuatnya variable tersebut?

Jawab :

untuk menampung data sementara yang akan dikirim, kemudian ditujukan ke postAPI untuk mengirimkan ke pada server API.

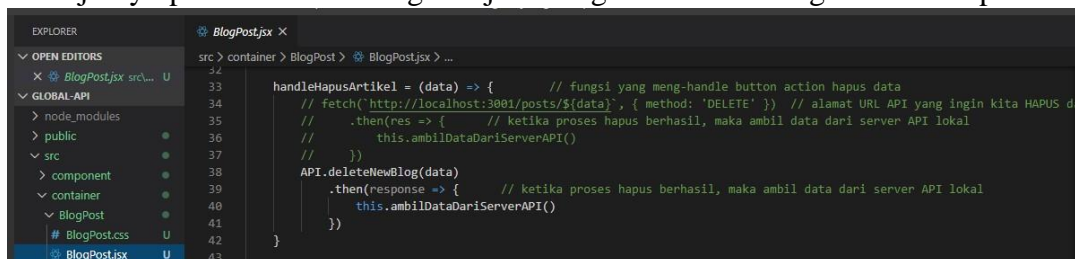
3. Praktikum 3 (Global API service DELETE)

1. Kita perhatikan pada fungsi handleHapusArtikel pada file BlogPost.jsx. Bagian inilah yang selanjutnya kita kelola agar bisa di-manage.
2. Buatlah fungsi untuk menampung action DELETE dari file BlogPost.jsx pada file services/index.js dan inialisasi fungsi tersebut



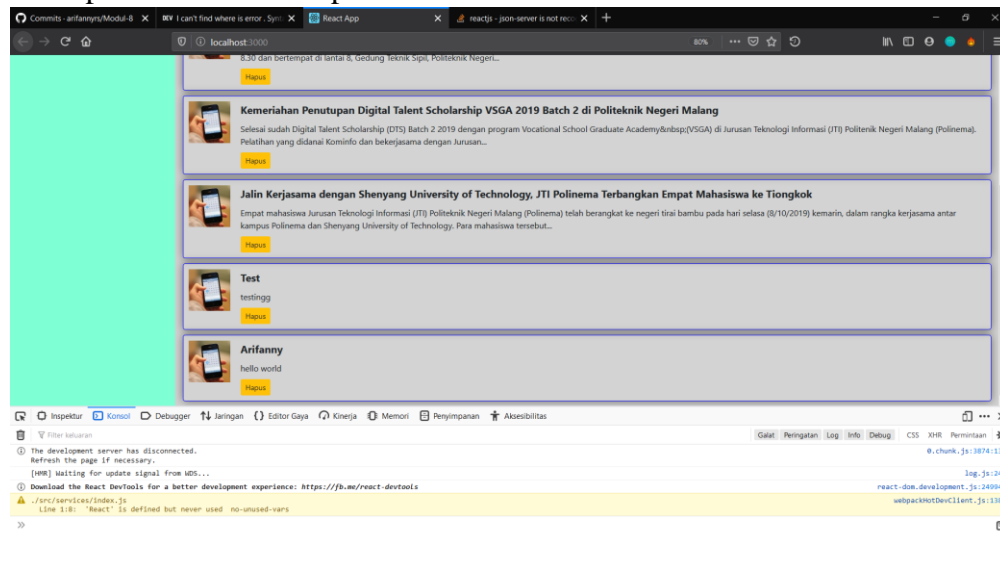
```
src > services > JS index.js > ...
1 import React from 'react';
2 import getAPI from './Get';
3 import PostAPI from './Post';
4 import DeleteAPI from './Delete';
5
6 const getNewsBlog = () => getAPI("posts?_sort=id&_order=asc")
7 const postNewsBlog = (dataYangDikirim) => PostAPI("posts", dataYangDikirim)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

3. Selanjutnya pindah ke file BlogPost.jsx dan ganti isi dari fungsi handleHapusArtikel

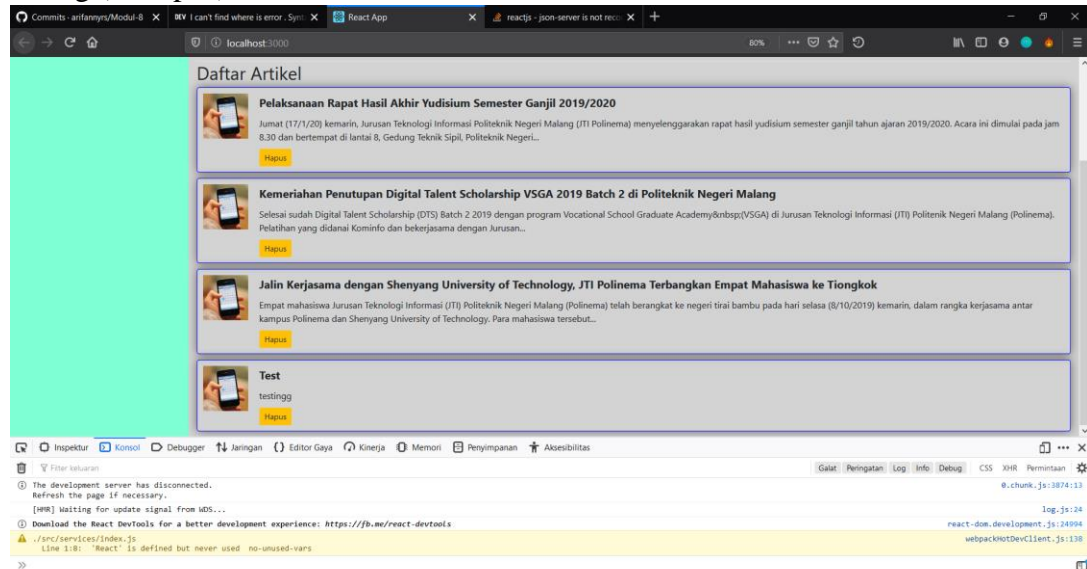


```
src > container > BlogPost > BlogPost.jsx > ...
33 handleHapusArtikel = (data) => { // fungsi yang meng-handle button action hapus data
34   // fetch('http://localhost:3001/posts/${data}', { method: 'DELETE' }) // alamat URL API yang ingin kita HAPUS d
35   // .then(res => { // ketika proses hapus berhasil, maka ambil data dari server API lokal
36   //   this.ambilDataDariServerAPI()
37   // })
38   API.deleteNewBlog(data)
39   .then(response => { // ketika proses hapus berhasil, maka ambil data dari server API lokal
40     this.ambilDataDariServerAPI()
41   })
42 }
43
```

4. Kode program pada BlogPost.jsx sedikit lebih simple dari seelumnya.
5. Silahkan kalian jalankan proses hapus artikel melalui browser dan amati apa yang terjadi.
6. Ketika program dijalankan.
 - Terdapat action HAPUS pada daftar artikel



- Saat tombol delete diklik, maka data artikel yang diklik tombol deletenya akan hilang (terhapus)



Menghapus Arifanny

Pertanyaan

- Perhatikan file index.js, apa tujuan dibuatnya fungsi DeleteAPI dan fungsi deleteNewsBlog?

Jawab :

untuk melakukan request hapus data pada server API.

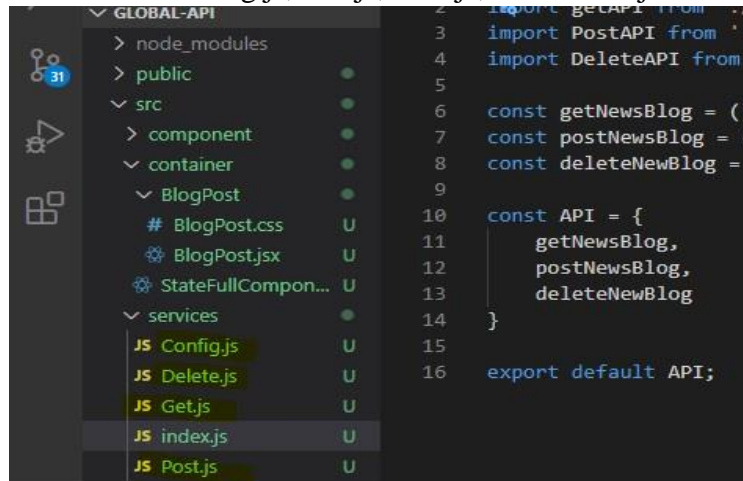
- Pada fungsi deleteNewsBlog, terdapat variable dataYangDiHapus. Apa tujuan dari dibuatnya variable tersebut?

Jawab :

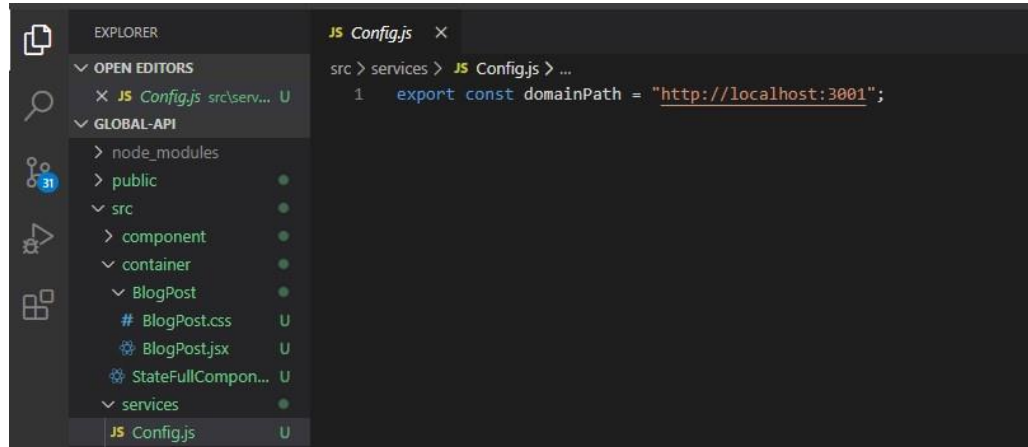
Untuk menyimpan sementara data yang akan dihapus, kemudian diteruskan kepada deleteAPI yang nantinya akan melakukan penghapusan pada data di server API.

4. Praktikum 4 (Manage Global API service)

1. Buatlah file Config.js, Get.js, Post.js, dan Delete.js dalam folder services



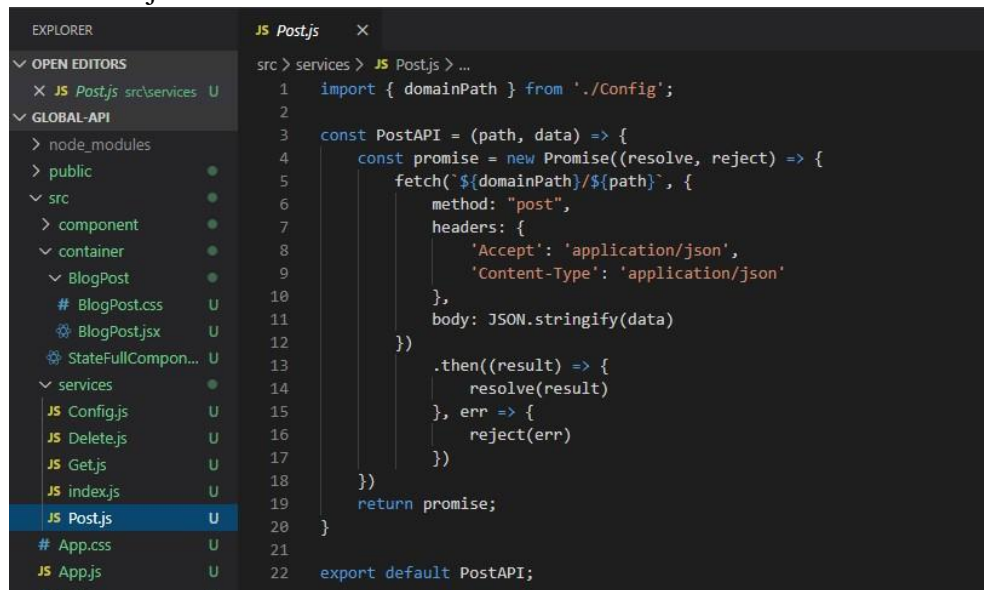
2. Buka Config.js dan isikan kode



3. Buka Get.js dan isikan kode

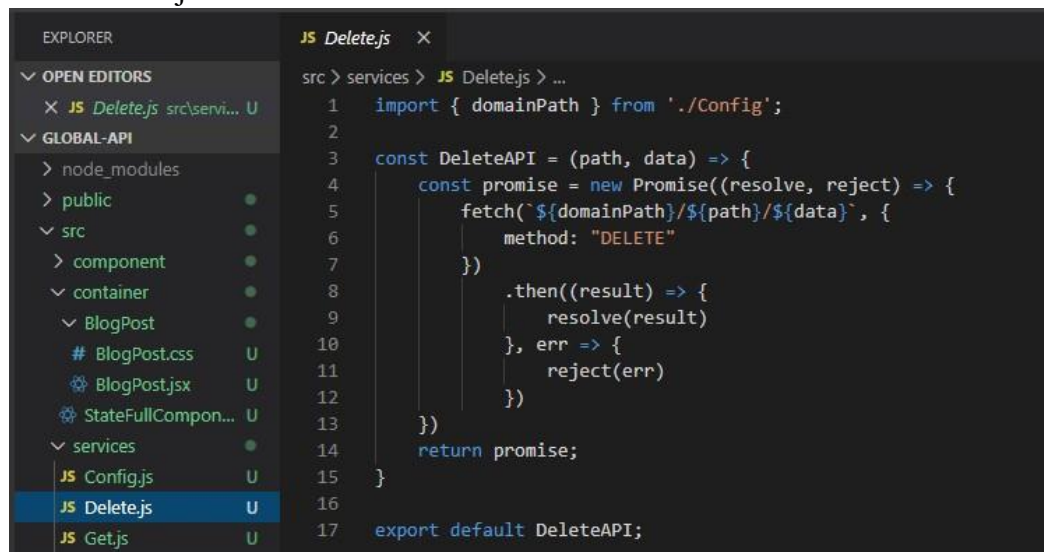


4. Buka Post.js dan isikan kode



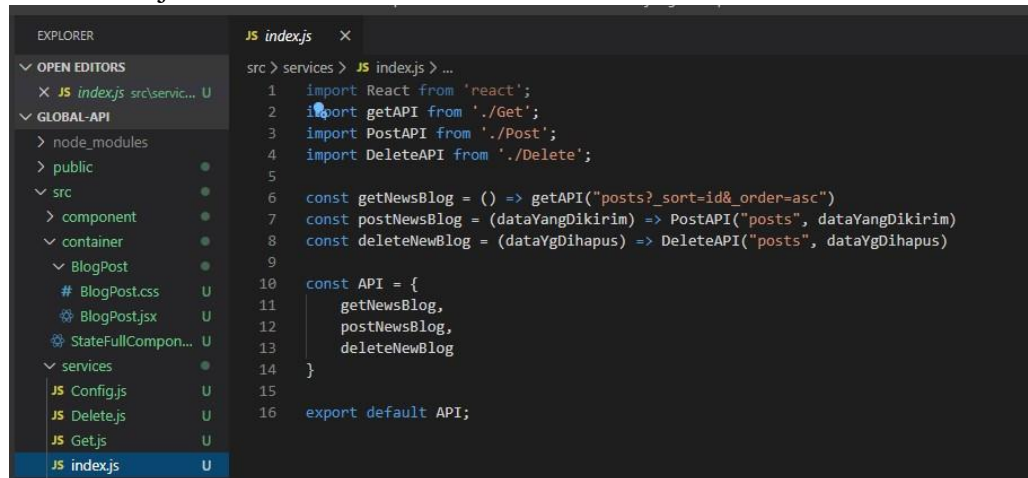
```
1 import { domainPath } from './Config';
2
3 const PostAPI = (path, data) => {
4   const promise = new Promise((resolve, reject) => {
5     fetch(`${domainPath}/${path}`, {
6       method: "post",
7       headers: {
8         'Accept': 'application/json',
9         'Content-Type': 'application/json'
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 kode

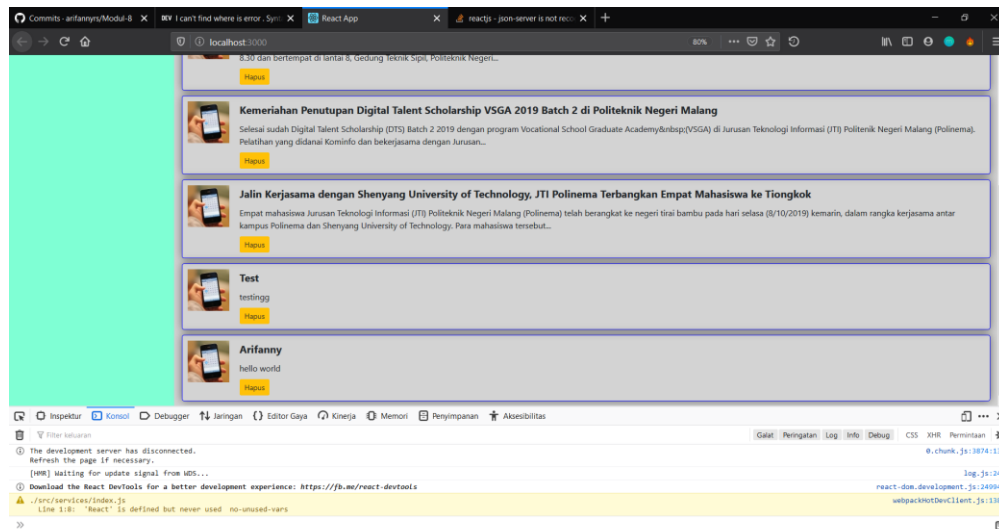


```
1 import { domainPath } from './Config';
2
3 const DeleteAPI = (path, data) => {
4   const promise = new Promise((resolve, reject) => {
5     fetch(`${domainPath}/${path}/${data}`, {
6       method: "DELETE"
7     })
8     .then((result) => {
9       resolve(result)
10     }, err => {
11       reject(err)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

6. Pada index.js kita edit



```
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;
```



Pertanyaan

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

Jawab :

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