

TUGAS PERORANGAN/INDIVIDU

MODUL 8

GLOBAL API

Disusun sebagai

MATA KULIAH : Pemrograman Berbasis Framework

Oleh :

Nurcahya Nania Anabela

TI-3A / 24

1741720156



PROGRAM STUDI D-IV T. INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

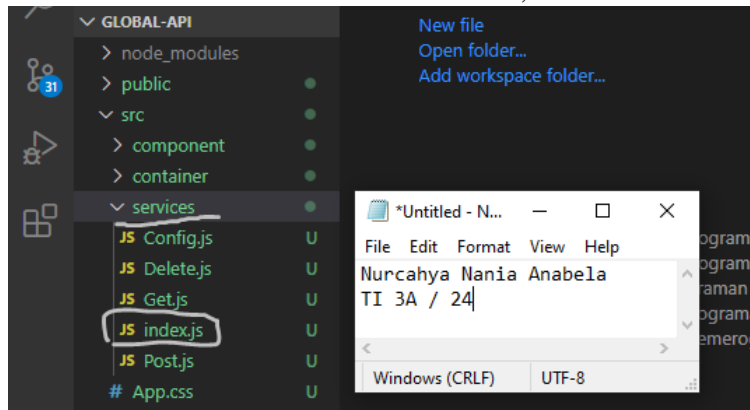
POLITEKNIK NEGERI MALANG

2020

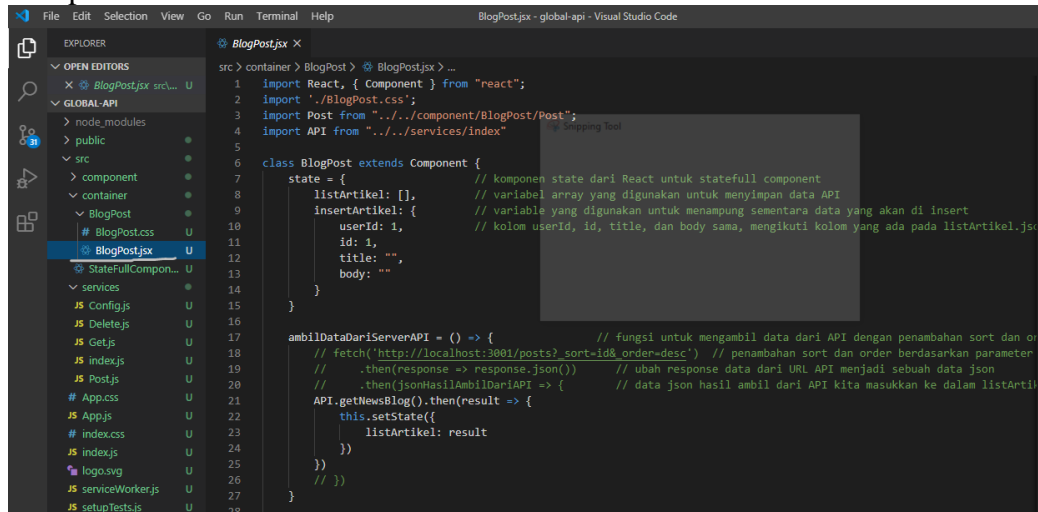
PRAKTIKUM

1. Praktikum 1 (Global API service GET)

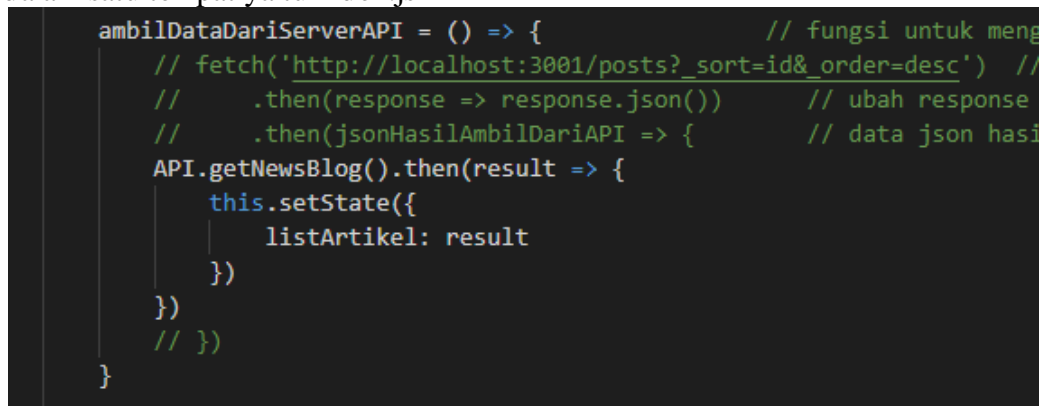
- 1) Buat folder baru bernama “Services” dalam folder src, kemudian buat file index.js.



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



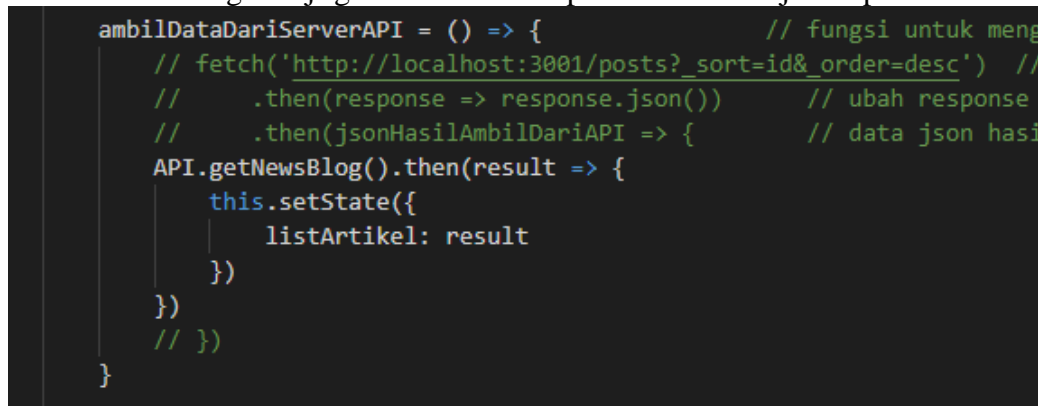
- 3) Pada fungsi ambilDataDariServerAPI terdapat pemanggilan API GET untuk melakukan request data artikel. Proses dari fungsi inilah yang akan kita manage kedalam satu tempat yaitu index.js



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



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



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

Pertanyaan Praktikum 1

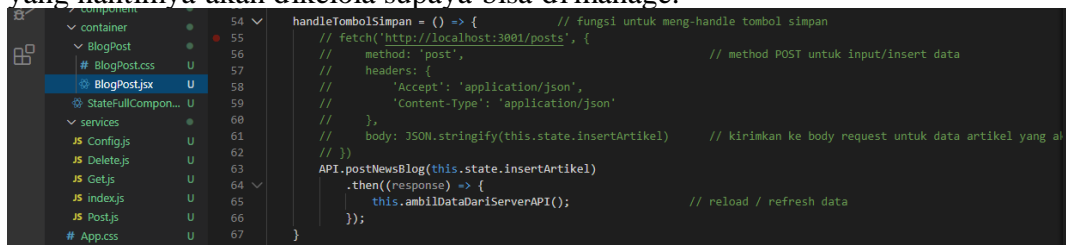
- a. Perhatikan file index.js, apa tujuan dibuatnya fungsi GetAPI pada baris 2 dan fungsi getNewsBlog pada baris 16?

Jawab :

GET API 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.



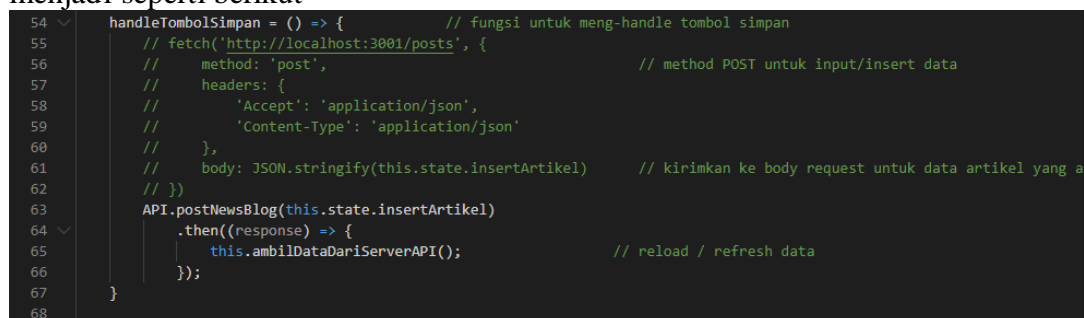
- 2) Membuat fungsi untuk menampung action POST dari file BlogPost.js pada file services/index.js dan inialisasi fungsi tersebut.



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a file tree with folders like 'src', 'services', and 'BlogPost'. The 'index.js' file in the 'services' folder is selected. The main editor area shows the content of 'index.js', which includes imports for React, getAPI, PostAPI, and DeleteAPI, and defines functions for getNewsBlog, postNewsBlog, and deleteNewBlog, along with an API object and an export 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;
```

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



The screenshot shows the VS Code editor with the 'handleTombolSimpan' function in 'BlogPost.js'. The function uses the 'API.postNewsBlog' method to send a POST request to the server with the state's 'insertArtikel' data. It also includes a comment about reloading the data.

```
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 Praktikum 2

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

Jawab :

Tujuan POST API adalah untuk mengirimkan data kepada server API.

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

Jawab :

Tujuannya adalah 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



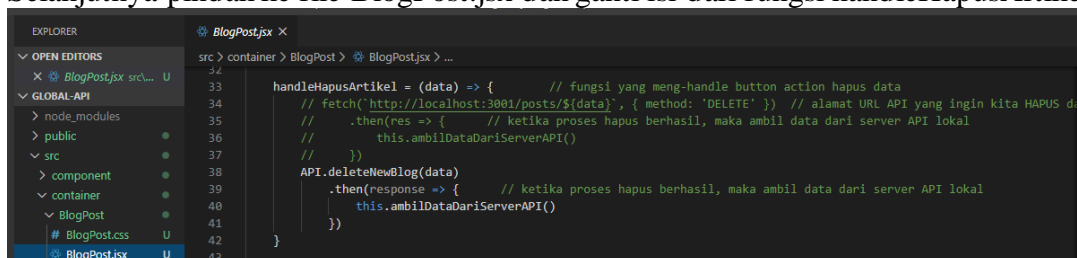
The screenshot shows the VS Code Explorer on the left with the file structure expanded to 'src > services > index.js'. The main editor displays the content of 'index.js', which is a service module for a React application. It imports 'React' from 'react', and 'getAPI', 'PostAPI', and 'DeleteAPI' from './Get', './Post', and './Delete' respectively. It defines three functions: 'getNewsBlog' (a GET request to '/posts?_sort=id&_order=asc'), 'postNewsBlog' (a POST request to '/posts' with 'dataYangDikirim'), and 'deleteNewBlog' (a DELETE request to '/posts' with 'dataYgDihapus'). These functions are then assigned to properties of an 'API' object, which is exported as the default export.

```

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;

```

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



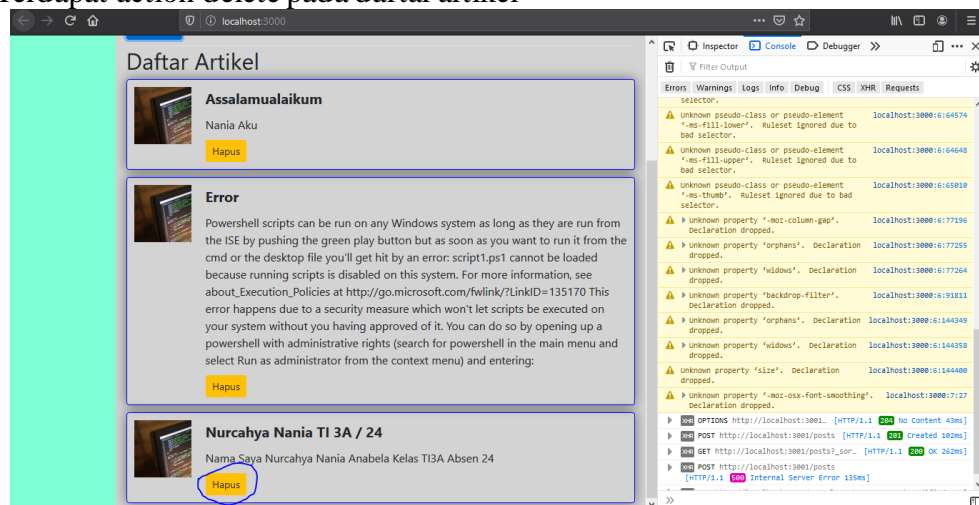
The screenshot shows the VS Code Explorer on the left with the file structure expanded to 'src > container > BlogPost > BlogPost.jsx'. The main editor displays the content of 'BlogPost.jsx', showing the 'handleHapusArtikel' function. This function is a callback for a button click, which calls 'API.deleteNewBlog(data)' and then 'this.ambilDataDariServerAPI()' to refresh the data from the server.

```

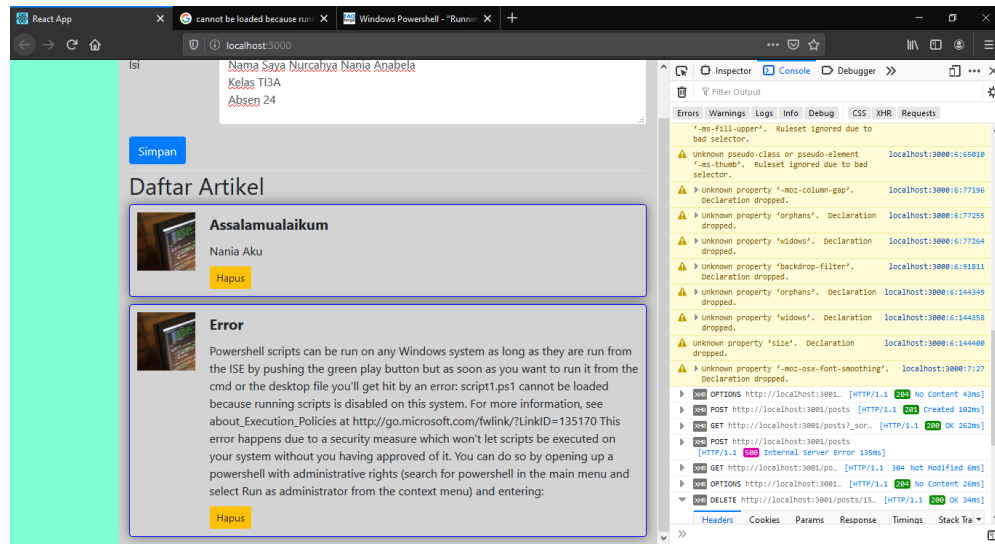
src > container > BlogPost > BlogPost.jsx > ...
34
35 // fungsi yang meng-handle button action hapus data
36 // fetch('http://localhost:3001/posts/${data}', { method: 'DELETE' }) // alamat URL API yang ingin kita HAPUS d
37 // .then(res => { // ketika proses hapus berhasil, maka ambil data dari server API lokal
38 //   this.ambilDataDariServerAPI()
39 // })
40 API.deleteNewBlog(data)
41 // ketika proses hapus berhasil, maka ambil data dari server API lokal
42 // .then(response => {
43 //   this.ambilDataDariServerAPI()
44 // })
45 }

```

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 delete pada daftar artikel



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



Keterangan : Saya menghapus artikel dengan judul Nurcahya Nania Anabela / TI 3A dan sudah terhapus.

Pertanyaan Praktikum 3

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

Jawab :

Tujuannya adalah untuk melakukan request hapus data pada server API.

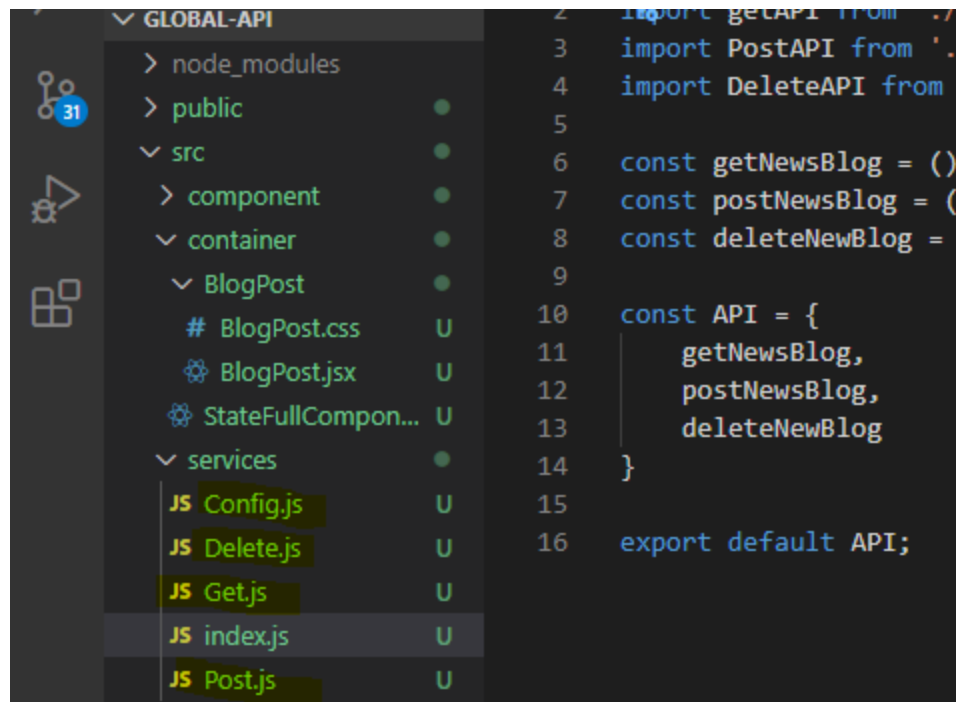
- Pada fungsi deleteNewsBlog, terdapat variable dataYangDiHapus. Apa tujuan dari dibuatnya variable tersebut?

Jawab :

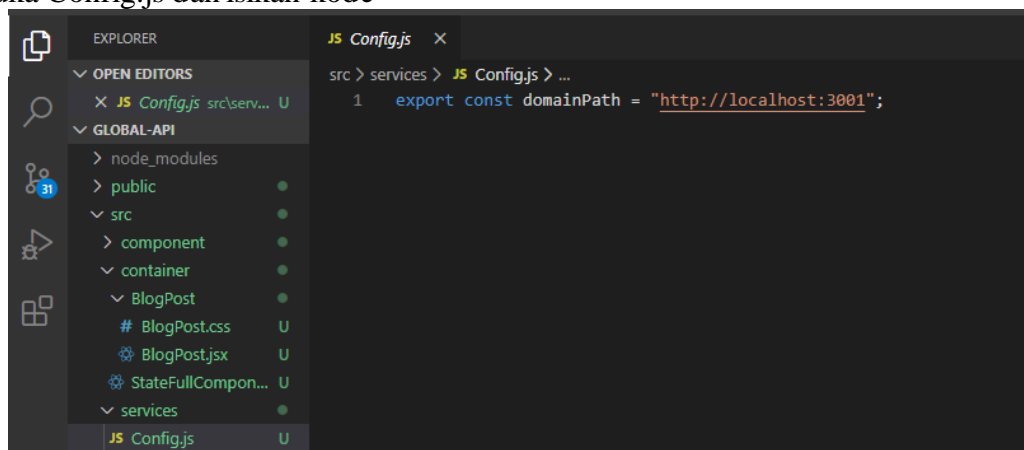
Untuk menyimpan sementara data yang akan dihapus, kemudian diteruskan kepada deleteAPI yang nantinya akan melakukan penghapusan pada data di server API.

4. Praktikum 4 (Manage Global API service)

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



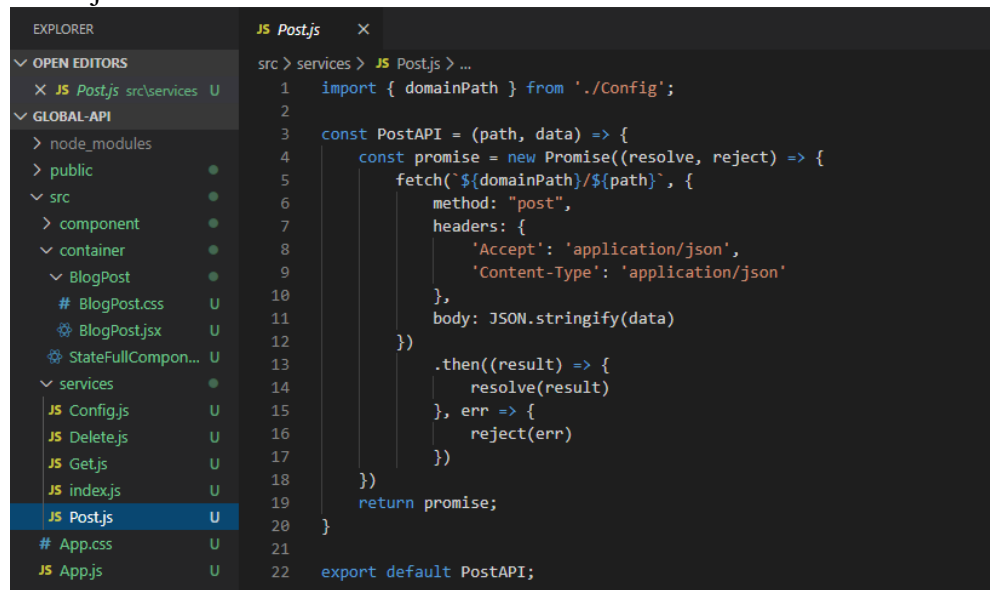
2. Buka Config.js dan isikan kode



3. Buka Get.js dan isikan kode



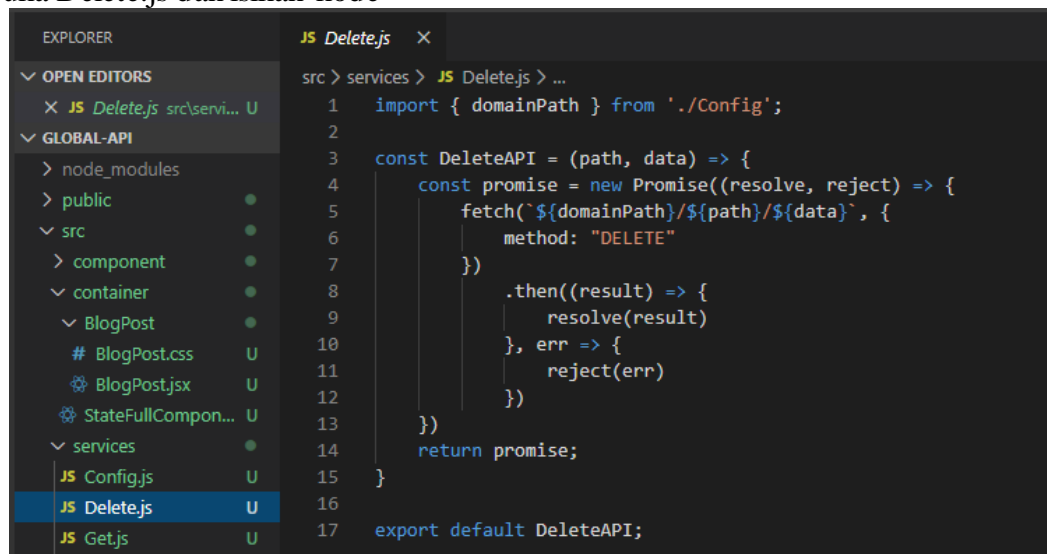
4. Buka Post.js dan isikan kode



```
EXPLORER
  OPEN EDITORS
    JS Post.js src/services U
  GLOBAL-API
    node_modules
    public
    src
      component
      container
      BlogPost
        BlogPost.css U
        BlogPost.jsx U
        StateFullCompon... U
      services
        JS Config.js U
        JS Delete.js U
        JS Get.js U
        JS index.js U
        JS Post.js U
        App.css U
        App.js U

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

5. Buka Delete.js dan isikan kode



```
EXPLORER
  OPEN EDITORS
    JS Delete.js src/sevi... U
  GLOBAL-API
    node_modules
    public
    src
      component
      container
      BlogPost
        BlogPost.css U
        BlogPost.jsx U
        StateFullCompon... U
      services
        JS Config.js U
        JS Delete.js U
        JS Get.js U

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    })
9    .then((result) => {
10      resolve(result)
11    }, err => {
12      reject(err)
13    })
14    return promise;
15  }
16
17  export default DeleteAPI;
```

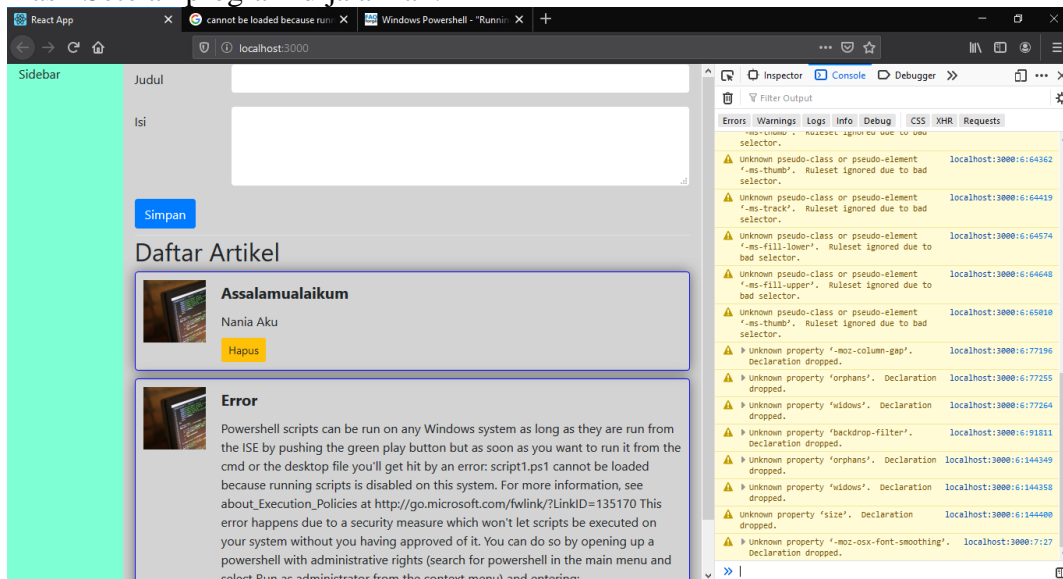
6. Pada index.js kita edit


```

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;

```

7. Hasil Setelah program dijalankan:



Keterangan :

Saat Judul dimasukkan dan isi dimasukkan, kemudian diklik tombol simpan. Maka mereka akan tersimpan dalam artikel dan muncul pada bagian Daftar Artikel.

Pertanyaan Praktikum 4

- 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.