

Laporan Praktikum
Mata Kuliah Pemrograman Berbasis Framework
“Week 9”



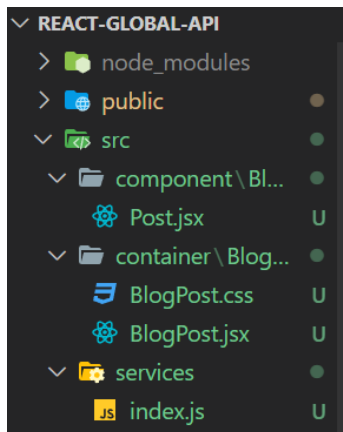
Nama Penyusun:

Khosy Robbin Hood (1941720067) / TI3D

JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D-IV TEKNIK INFORMATIKA
APRIL 2022

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)

```
import React, { Component } from "react";
import "./BlogPost.css";
import Post from "../../component/BlogPost/Post";
import API from "../../services";

class BlogPost extends Component {
  state = {
    // komponen state dari React untuk statefull component
    listArtikel: [], // variabel array yang digunakan untuk
    menyimpan data API
    insertArtikel: {
      // variabel yang digunakan untuk menampung sementara
      data yang akan diimport
      userId: 1, // kolom userId, id, title, dan body sama,
      mengikuti kolom yang ada pada listArtikel.json
      id: 1,
      title: "",
      body: "",
    },
  };

  ambilDataDariSeverAPI = () => {
    // fungsi untuk mengambil data dari API dengan penambahan
    sort dan order
    API.getNewsBlog().then((result) => {
      this.setState({
        listArtikel: result,
      });
    });
  };
}
```

```

componentDidMount() {
    // komponen untuk mengecek ketika component telah
    dimounting, maka panggil API
    this.ambilDataDariSeverAPI(); // ambil data dari server
    API lokal
}

handleHapusArtikel = (data) => {
    API.deleteNewsBlog(data).then((response) => {
        this.ambilDataDariSeverAPI();
    });
};

handleTambahArtikel = (event) => {
    // fungsi untuk meng-handle form tambah data artikel
    let formInsertArtikel = { ...this.state.insertArtikel };
    // cloning data state insertArtikel ke dalam variabel
    formInsertArtikel
    let timestamp = new Date().getTime(); // digunakan untuk
    menyimpan waktu (sebagai id artikel)
    formInsertArtikel["id"] = timestamp;
    formInsertArtikel[event.target.name] =
    event.target.value; // menyimpan data onChange ke
    formInsertArtikel sesuai dengan target yang diisi
    this.setState({
        insertArtikel: formInsertArtikel,
    });
};

handleTombolSimpan = () => {
    // fungsi untuk meng-handle tombol simpan
    API.postNewsBlog(this.state.insertArtikel).then((response) => {
        this.ambilDataDariSeverAPI(); // reload / refresh
        data
    });
};

render() {
    return (
        <div className="post-artikel">
            <div className="form pb-3 border-bottom">

```

```

        <div className="form-group row">
            <label htmlFor="title"
className="col-sm-2 col-form-label">
                Judul
            </label>
            <div className="col-sm-10">
                <input
                    type="text"
                    className="form-control"
                    id="title"
                    name="title"

onChange={this.handleTambahArtikel}
                />
            </div>
        </div>
        <div className="form-group row my-3">
            <label htmlFor="body" className="col-sm-2
col-form-label">
                Isi
            </label>
            <div className="col-sm-10">
                <textarea
                    name="body"
                    id="body"
                    rows="3"
                    className="form-control"

onChange={this.handleTambahArtikel}
                ></textarea>
            </div>
        </div>
        <button
            type="submit"
            className="btn btn-primary"
            onClick={this.handleTombolSimpan}
        >
            Simpan
        </button>
    </div>
    <h2 className="mt-2">Daftar Artikel</h2>
    {this.state.listArtikel.map((artikel) => {

```

```

        // looping dan masukkan untuk setiap data
yang ada di listArtikel ke variabel artikel
        return (
            <Post
                key={artikel.id}
                judul={artikel.title}
                isi={artikel.body}
                idArtikel={artikel.id}

hapusArtikel={this.handleHapusArtikel}
            />
        ); // mappingkan data json dari API sesuai
dengan kategorinya
    ))}
</div>
    );
}
}

export default BlogPost;

```

3. Pada fungsi ambilDataDariServerAPI (baris16) terdapat pemanggilan API GET untuk merequest 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 berikut.

```

const domainPath = 'http://localhost:3001';
const GetAPI = (path) => {
    const promise = new Promise ((resolve, reject) => {
        fetch(`${domainPath}/${path}`)
            .then(response => response.json())
            .then((result) => {
                resolve(result);
            }, (err) => {
                reject(err);
            })
    })
    return promise;
}

const getNewsBlog = () => GetAPI('posts?_sort=id&_order=desc');

const API = {
    getNewsBlog

```

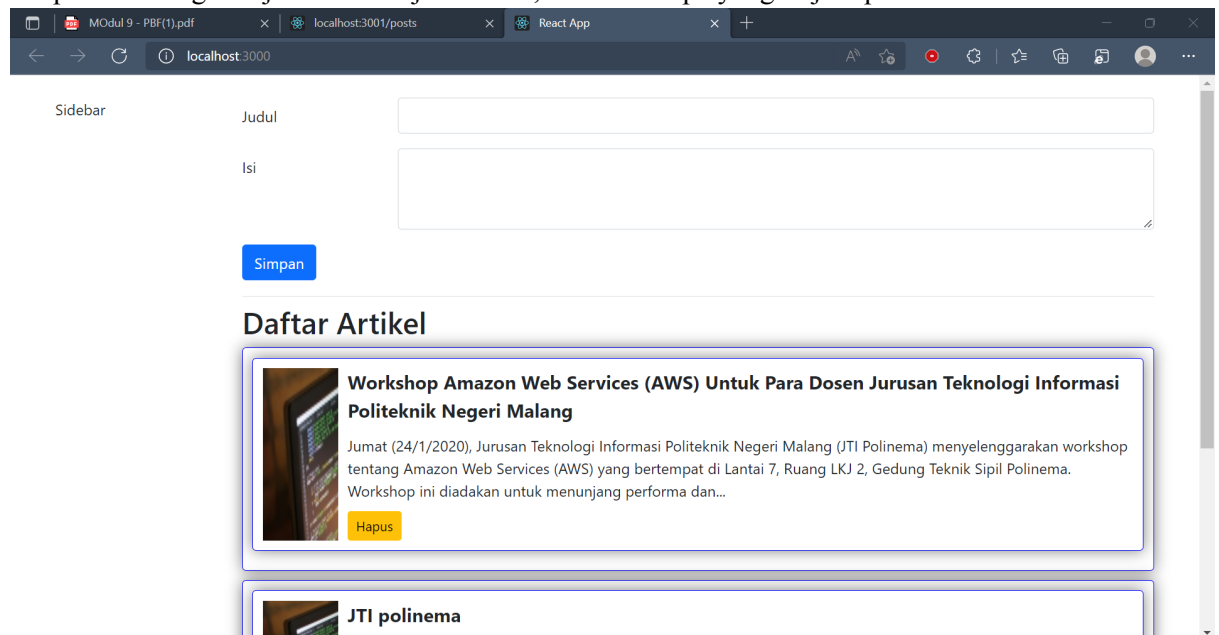
```
}

export default API;
```

- Kembali ke file BlogPost.js. Ganti baris 17 sampai baris 23

```
ambilDataDariSeverAPI = () => {
    // fungsi untuk mengambil data dari API dengan penambahan
    sort dan order
    API.getNewsBlog().then((result) => {
        this.setState({
            listArtikel: result,
        });
    });
};
```

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



Pertanyaan Praktikum 1

- Perhatikan file `index.js`, apa tujuan dibuatnya fungsi `GetAPI` pada baris 2 dan fungsi `getNewsBlog` pada baris 16?
 - Untuk memanggil API `lisArtikel.json` yang telah diinisialisasi di file `index.js` pada folder `services`.

Praktikum 2 Global API service POST

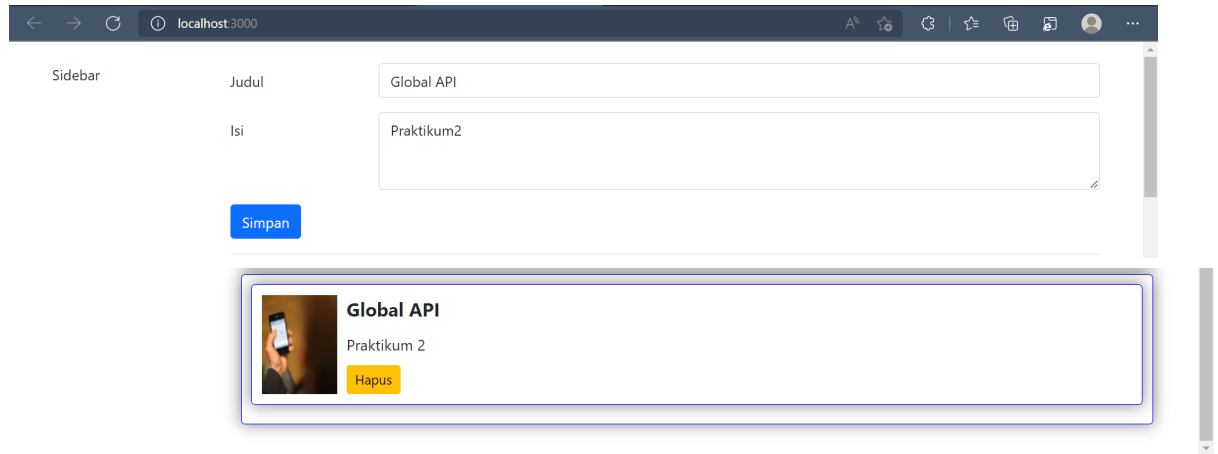
1. Kita perhatikan pada fungsi `handleTombolSimpan` pada file `BlogPost.js`. Bagian inilah yang selanjutnya kita kelola agar bisa di-manage.
2. Buatlah fungsi untuk menampung action POST dari file `BlogPost.js` pada file `services/index.js` dan inialisasi fungsi tersebut

```
const PostAPI = (path, data) => {  
  const promise = new promise ((resolve, reject) => {  
    fetch(`${domainPath}/${path}`, {  
      method: 'post',  
      headers: {  
        'Accept': 'application/json',  
        'Content-Type': 'application/json'  
      },  
      body: JSON.stringify(data)  
    })  
    .then((result) => {  
      resolve(result);  
    }, (err) => {  
      reject(err);  
    })  
  })  
  return promise;  
}  
  
const getNewsBlog = () => GetAPI('posts?_sort=id8_order=desc');  
const postNewsBlog = (dataYgDiKirim) => PostAPI('post',  
dataYgDiKirim);  
  
const API = {  
  getNewsBlog,  
  postNewsBlog  
}  
  
export default API;
```

3. Selanjutnya pindah ke file `BlogPost.js` dan ganti isi dari fungsi `handleTombolSimpan`

```
handleTombolSimpan = () => {  
  // fungsi untuk meng-handle tombol simpan  
  API.postNewsBlog(this.state.insertArtikel).then((response) => {  
    this.ambilDataDariSeverAPI(); // reload / refresh  
    data  
  });  
};
```

4. Selesai. Kode program pada `BlogPost.js` sedikit lebih simple dari seelumnya.
5. Silahkan kalian jalankan proses input artikel melalui browser dan amati apa yang terjadi.



Pertanyaan Praktikum 2

1. Perhatikan file index.js, apa tujuan dibuatnya fungsi PostAPI dan fungsi postNewsBlog?
 - untuk melakukan insert/input data ke API listArtikel.
2. Pada fungsi postNewsBlog, terdapat variable dataYangDiKirim. Apa tujuan dari dibuatnya variabel tersebut?
 - inialisasi untuk method PostAPI

Praktikum 3 Global API service DELETE

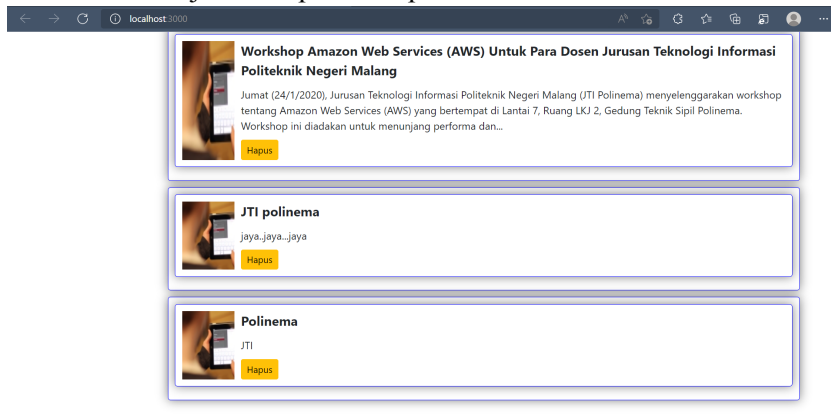
1. Kita perhatikan pada fungsi handleHapusArtikel pada file BlogPost.jsx. Bagian inilah yang selanjutnya kita kelola agar bisa di-manage.
2. Selanjutnya pindah ke file BlogPost.jsx dan ganti isi dari fungsi handleHapusArtikel

```
handleTombolSimpan = () => {  
    // fungsi untuk meng-handle tombol simpan  
    API.postNewsBlog(this.state.insertArtikel).then((response) => {  
        this.ambilDataDariSeverAPI(); // reload / refreshdata  
    });  
};
```

3. Buatlah fungsi untuk menampung action DELETE dari file BlogPost.jsx pada file services/index.js dan inialisasi fungsi tersebut

```
const DeleteAPI = (path, data) => {  
    const promise = new Promise ((resolve, reject) => {  
        fetch(`${domainPath}/${path}/${data}`, {method: 'DELETE'})  
        .then((result) => {  
            resolve(result);  
        }, (err) => {  
            reject(err);  
        })  
    })  
}
```

4. Selesai. Kode program pada BlogPost.jsx sedikit lebih simple dari seelumnya.
5. Silahkan kalian jalankan proses hapus artikel melalui browser dan amati apa yang terjadi.



Pertanyaan Praktikum 3

1. Perhatikan file index.js, apa tujuan dibuatnya fungsi DeleteAPI dan fungsi deleteNewsBlog?
 - untuk melakukan method hapus dengan menggunakan inialisasi deleteNewsBlog.
2. Pada fungsi deleteNewsBlog, terdapat variable dataYangDiHapus. Apa tujuan dari dibuatnya variable tersebut?
 - inialisasi untuk method DeleteAPI

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

```
export const domainPath = "http://localhost:3001";
```

3. Buka Get.js dan isikan kode

```
import { domainPath } from "../Config";

const GetAPI = (path) => {
  // path digunakan untuk menunjuk alamat API mana yang akan
  di-request
  const promise = new Promise((resolve, reject) => {
    fetch(`${domainPath}/${path}`) // alamat url domain + path
    untuk mengakses full alamat API yang di-request
    .then((response) => response.json()) // response dari
    server harus dijadikan json
    .then(
      (result) => {
        resolve(result); // jika sukses menerima response dari
        server, maka resolve response ke user
      },
      (err) => {
        // jika terjadi error dari server (server down, dll),
        reject
        reject(err); // maka kirim pesan error ke user melalui
        reject
      }
    );
  });
  return promise;
};

export default GetAPI;
```

4. Buka Post.js dan isikan kode

```
import { domainPath } from "../Config";

const PostAPI = (path, data) => {
  const promise = new Promise((resolve, reject) => {
    fetch(`${domainPath}/${path}`, {
      method: "POST", // method POST untuk input atau insert data
      headers: {
        Accept: "application/json",
        "Content-Type": "application/json",
      },
    },
  );
  return promise;
};
```

```

        body: JSON.stringify(data), // kirimkan ke body request
        untuk data artikel yang akan ditambahkan (insert)
    }).then(
        (result) => {
            resolve(result); // jika sukses menerima response dari
server maka resolve response ke user
        },
        (err) => {
            // jika terjadi error dari server (server down, dll),
            reject(err); // maka kirim pesan error ke user melalui
reject
        }
    );
});
return promise;
};

export default PostAPI;

```

5. Buka Delete.js dan isikan kode

```

import { domainPath } from "../Config";

const DeleteAPI = (path, data) => {
    const promise = new Promise((resolve, reject) => {
        fetch(`${domainPath}/${path}/${data}`, {
            method: "DELETE",
        }).then(
            (result) => {
                resolve(result); // jika sukses menerima response dari
server maka resolve response ke user
            },
            (err) => {
                // jika terjadi error dari server (server down, dll),
                reject(err); // maka kirim pesan error ke user melalui
reject
            }
        );
    });
    return promise;
};

export default DeleteAPI;

```

6. Pada index.js kita edit

```
import DeleteAPI from "../Delete";
import GetAPI from "../Get";
import PostAPI from "../Post";

const getNewsBlog = () => GetAPI("posts?_sort=id&_order=desc");
const postNewsBlog = (dataYgDiKirim) => PostAPI("posts",
dataYgDiKirim);
const deleteNewsBlog = (dataYgDiHapus) => DeleteAPI("posts",
dataYgDiHapus);

const API = {
  // inisialisasi function" yang akan disediakan global API
  getNewsBlog,
  postNewsBlog,
  deleteNewsBlog,
};

export default API;
```

7. Amati apa yang terjadi

Pertanyaan Praktikum 4

1. Bagaimana caranya untuk menambahkan daftar API baru baik untuk method GET, POST, DELETE pada Global API?
 - dengan lakukan import seperti ini

```
import DeleteAPI from "../Delete";
import GetAPI from "../Get";
import PostAPI from "../Post";
```

TUGAS PRAKTIKUM

Ubahlah Tugas pada Modul 4 yang sudah kalian buat dengan memanfaatkan Global API seperti praktikum yang kita lakukan pada Modul 8 ini.

- container/blogmahasiswa

```
import React, { Component } from "react";
import "./BlogMahasiswa.css";
import Mahasiswa from "../../component/BlogMahasiswa/Mahasiswa";
import API_MHS from "../../services-mhs";

class BlogMahasiswa extends Component {
  state = {
    listMahasiswa: [],
    insertMahasiswa: {
      id: 1,
      nama: "",
      nim: "",
      alamat: "",
      hp: "",
      angkatan: "",
      status: "",
    },
  };

  ambilDataDariSeverAPI = () => {
    API_MHS.getMahasiswa().then((result) => {
      this.setState({
        listMahasiswa: result,
      });
    });
  };

  componentDidMount() {
    this.ambilDataDariSeverAPI();
  }

  handleHapusMahasiswa = (data) => {
    API_MHS.deleteMahasiswa(data).then((response) => {
      this.ambilDataDariSeverAPI();
    });
  };

  handleTambahMahasiswa = (event) => {
    let formInsertMahasiswa = { ...this.state.insertMahasiswa };
```

```

    let timestamp = new Date().getTime();
    formInsertMahasiswa["id"] = timestamp;
    formInsertMahasiswa[event.target.name] = event.target.value;
    this.setState({
      insertMahasiswa: formInsertMahasiswa,
    });
  };

  handleTombolSimpan = () => {

API_MHS.postMahasiswa(this.state.insertMahasiswa).then((response)
=> {
    this.ambilDataDariSeverAPI();
  });
};

```

- Pada services-mahasiswa
- post.js

```

import { domainPath } from "../Config";

const PostAPI = (path, data) => {
  const promise = new Promise((resolve, reject) => {
    fetch(`${domainPath}/${path}`, {
      method: "POST", // method POST untuk input atau insert data
      headers: {
        Accept: "application/json",
        "Content-Type": "application/json",
      },
      body: JSON.stringify(data), // kirimkan ke body request
      untuk data artikel yang akan ditambahkan (insert)
    }).then(
      (result) => {
        resolve(result); // jika sukses menerima response dari
server maka resolve response ke user
      },
      (err) => {
        // jika terjadi error dari server (server down, dll),
        reject(err); // maka kirim pesan error ke user melalui
reject
      }
    );
  });
  return promise;
};

```

```
};
export default PostAPI;
```

- **get.js**

```
import { domainPath } from "../Config";

const GetAPI = (path) => {
  // path digunakan untuk menunjuk alamat API mana yang akan
  di-request
  const promise = new Promise((resolve, reject) => {
    fetch(`${domainPath}/${path}`) // alamat url domain + path
    untuk mengakses full alamat API yang di-request
    .then((response) => response.json()) // response dari
    server harus dijadikan json
    .then(
      (result) => {
        resolve(result); // jika sukses menerima response dari
        server, maka resolve response ke user
      },
      (err) => {
        // jika terjadi error dari server (server down, dll),
        reject
        reject(err); // maka kirim pesan error ke user melalui
        reject
      }
    );
  });
  return promise;
};
export default GetAPI;
```

- **delete.js**

```
import { domainPath } from "../Config";

const DeleteAPI = (path, data) => {
  const promise = new Promise((resolve, reject) => {
    fetch(`${domainPath}/${path}/${data}`, {
      method: "DELETE",
    }).then(
      (result) => {
        resolve(result); // jika sukses menerima response dari
        server maka resolve response ke user
      },
      (err) => {
        // jika terjadi error dari server (server down, dll),

```

```

        reject(err); // maka kirim pesan error ke user melalui
reject
    }
    );
});
return promise;
};

export default DeleteAPI;

```

- config.js

```

export const domainPath = "http://localhost:3002";

```

- index.js

```

import DeleteAPI from "../Delete";
import GetAPI from "../Get";
import PostAPI from "../Post";

const getMahasiswa = () =>
  GetAPI("mahasiswa?_sort=id&_order=desc");
const postMahasiswa = (dataYgDiKirim) => PostAPI("mahasiswa",
  dataYgDiKirim);
const deleteMahasiswa = (dataYgDiHapus) =>
  DeleteAPI("mahasiswa", dataYgDiHapus);

const API_MHS = {
  // inisialisasi function" yang akan disediakan global API
  getMahasiswa,
  postMahasiswa,
  deleteMahasiswa,
};

export default API_MHS;

```

Hasil

React App localhost:3002/mahasiswa localhost:3001/posts Modul 9 - PBF(1).pdf localhost:3000

Navbar

SideBar

Nama KHOSY ROBBIN HOOD

Nim 1941720067

Alamat bwi

Hp 123

Angkatan 2019

Status aktif

Simpan

Daftar Mahasiswa

Windows Taskbar: 15:38 23/04/2022

React App localhost:3002/mahasiswa localhost:3001/posts Modul 9 - PBF(1).pdf localhost:3000


Hp 123

Angkatan 2019

Status aktif

Simpan

Daftar Mahasiswa



KHOSY ROBBIN HOOD
Nim: 1941720067
Alamat: bwi
Hp: 123
Angkatan: 2019
Status: aktif
Hapus

Windows Taskbar: 15:38 23/04/2022

Link github : [Global-API](#)