

TUGAS PERORANGAN

PRAKTIKUM REACT API

Disusun sebagai

MATA KULIAH: PEMROGRAMAN BERBASIS FRAMEWORK

Oleh:

Rois Dwi Admaja/1741720193

TI – 3B / 23



PROGRAM STUDI: D-IV TEKNIK INFORMATIKA

JURUSAN: TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

Praktikum 1

Interaksi dengan API menggunakan method GET

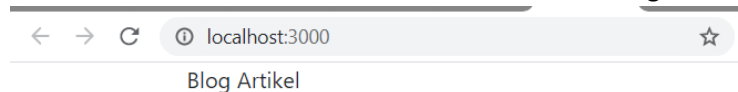
1. Membuat halaman BlogPost.jsx berfungsi sebagai halaman utama untuk menampilkan data

```
src > container > BlogPost > BlogPost.jsx > ...
1  import React, {Component} from 'react';
2  import './BlogPost.css';
3
4  class BlogPost extends Component {
5    render() {
6      return(
7        <p>Blog Artikel</p>
8      )
9    }
10 }
11
12 export default BlogPost;
```

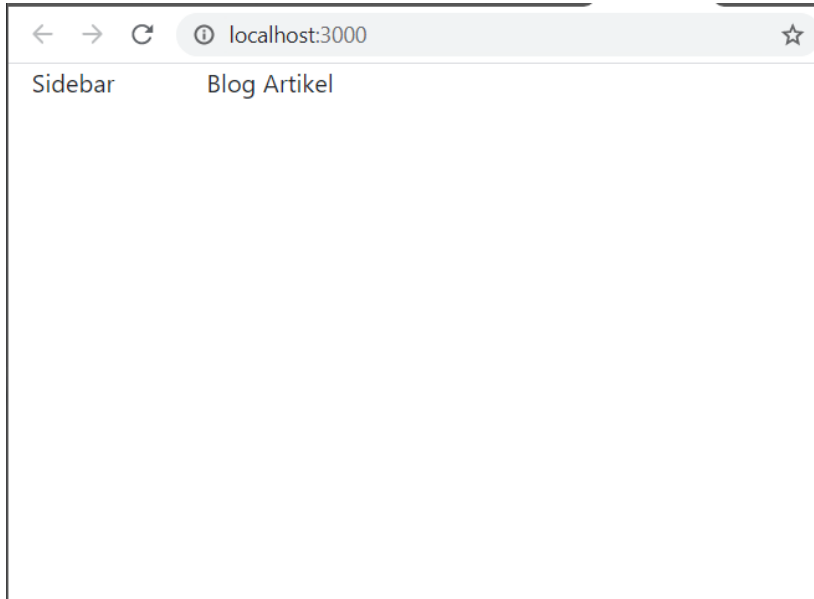
Setelah membuat halaman BlogPost dan mengisinya dengan class dan melakukan export class tersebut. Lalu import class tersebut kedalam index.js

```
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App';
5  import * as serviceWorker from './serviceWorker';
6
7  import BlogPost from './container/BlogPost/BlogPost';
8
9  ReactDOM.render(<BlogPost />, document.getElementById('root'));
10
11 // If you want your app to work offline and load faster, you can call
12 // unregister() to register() below. Note this comes with some pitfalls.
13 // Learn more about service workers: https://bit.ly/CRA-PWA
14 serviceWorker.unregister();
15
```

Lalu akan muncul di halaman browser isi dari class BlogPost



Setelah itu menambahkan component sidebar dan tampilan akan seperti ini



Setelah itu mengedit index.css agar tampilan lebih berwarna

```
src > index.css > #content
1 body {
2   margin: 0;
3   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
4     'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Ne
5     sans-serif;
6   -webkit-font-smoothing: antialiased;
7   -moz-osx-font-smoothing: grayscale;
8 }
9
10 code {
11   font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier N
12   monospace;
13 }
14
15 #sidebar {
16   background-color: aquamarine;
17 }
18
19 #content {
20   background-color: lightgray;
21 }
22
```

Lalu mengubah kode program pada component BlogPost agar menjadi statefull component, karena data akan berubah jadi harus dijadikan statefull component

```
src > container > BlogPost > BlogPost.js > BlogPost > render
1 import React, {Component} from 'react';
2 import './BlogPost.css';
3
4 class BlogPost extends Component {
5   render() {
6     return(
7       <div className="post-artikel">
8         <h2>Daftar Artikel</h2>
9         <div className="artikel">
10           <div className="gambar-artikel">
11             
12           </div>
13           <div className="konten-artikel">
14             <div className="judul-artikel">Judul Artikel</div>
15             <p className="isi-artikel">Isi Artikel</p>
16           </div>
17         </div>
18       </div>
19     );
20   }
21 }
22
23 export default BlogPost;
```

Lalu mengedit BlogPost.css agar tampilan pada list artikel lebih rapi

```

src > container > BlogPost > BlogPost.css > .konten-artikel p.isi-artikel
1  .artikel {
2      width: 100%;
3      padding: 10px;
4      border: 1px solid blue;
5      border-radius: 4px;
6      margin-bottom: 10px;
7      box-shadow: 0 0 16px rgba(0,0,0,0.5);
8      display: flex;
9  }
10
11  .gambar-artikel {
12      height: 80px;
13      width: 80px;
14      margin-right: 20px;
15      vertical-align: top;
16  }
17
18  .gambar-artikel img {
19      width: 100%;
20      height: 100%;
21      object-fit: cover;
22  }
23
24  .konten-artikel {
25      flex: 1;
26  }
27
28  .konten-artikel div.judul-artikel {
29      font-size: 20px;
30      font-weight: bold;
31      margin-bottom: 10px;
32  }
33
34  .konten-artikel p.isi-artikel {
35      font-size: 16px;
36      margin-bottom: 10px;
37  }

```

Berikut hasilnya



Memindahkan beberapa komponen div yang berisi judul artikel dan isi artikel ke Post.jsx yang merupakan stateless component, agar bisa digunakan ulang komponennya

```

src > component > BlogPost > Post.jsx > ...
1  import React from 'react';
2
3  const Post = (props) => {
4      return (
5          <div className="artikel">
6              <div className="gambar-artikel">
7                  
8              </div>
9              <div className="konten-artikel">
10                 <div className="judul-artikel">Judul Artikel</div>
11                 <p className="isi-artikel">Isi Artikel</p>
12             </div>
13         </div>
14     )
15 }
16
17 export default Post;

```

Lalu memanggil component Post ke komponen BlogPost

```

src > container > BlogPost > BlogPost.jsx > BlogPost > render
1 import React, {Component} from 'react';
2 import Post from '../../component/BlogPost/Post';
3 import './BlogPost.css';
4
5 class BlogPost extends Component {
6   render() {
7     return(
8       <div className="post-artikel">
9         <h2>Daftar Artikel</h2>
10        <Post/>
11      </div>
12    )
13  }
14 }
15
16 export default BlogPost;

```

Lalu menambahkan parameter di statefull component (BlogPost) yang ingin dilempar ke stateless component untuk ditampilkan

```

src > container > BlogPost > BlogPost.jsx > BlogPost
1 import React, {Component} from 'react';
2 import Post from '../../component/BlogPost/Post';
3 import './BlogPost.css';
4
5 class BlogPost extends Component {
6   render() {
7     return(
8       <div className="post-artikel">
9         <h2>Daftar Artikel</h2>
10        <Post judul="JTI Polinema" isi="Jurusan Teknologi Informasi-Polinema"/>
11      </div>
12    )
13  }
14 }
15
16 export default BlogPost;

```

Setelah itu stateless component Post.jsx menangkap parameter yang dilempar oleh statefull

```

src > component > BlogPost > Post.jsx > Post
1 import React from 'react';
2
3 const Post = (props) => {
4   return (
5     <div className="artikel">
6       <div className="gambar-artikel">
7         
8       </div>
9       <div className="konten-artikel">
10        <div className="judul-artikel">{props.judul}</div>
11        <p className="isi-artikel">{props.isi}</p>
12      </div>
13    </div>
14  )
15 }
16
17 export default Post;

```

Dan hasilnya akan seperti gambar berikut



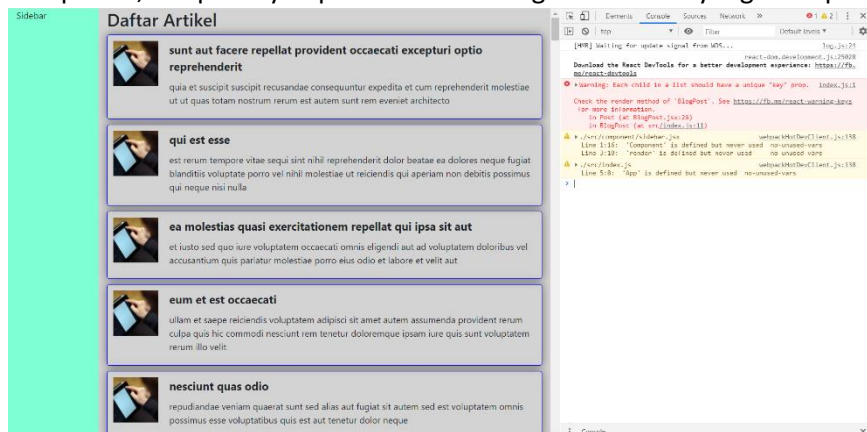
Mengambil data Post/artikel dari API

Mengedit BlogPost.jsx agar bisa mengambil data dari fake API yang sudah dibuat atau mengambil dari jsonplaceholder.typicode.com/posts.

Pada fungsi componentDidMount berfungsi untuk mengambil data dari json server

```
src > container > BlogPost > BlogPost.jsx > BlogPost > componentDidMount
1  import React, {Component} from 'react';
2  import Post from '../component/BlogPost/Post';
3  import './BlogPost.css';
4
5  class BlogPost extends Component {
6    state = {
7      listArtikel: []
8    }
9
10   componentDidMount() {
11     fetch('http://localhost:4000/posts')
12     .then(response => response.json())
13     .then(jsonHasilAmbilDariAPI => {
14       this.setState({
15         listArtikel: jsonHasilAmbilDariAPI
16       })
17     })
18   }
19
20   render() {
21     return(
22       <div className="post-artikel">
23         <h2>Daftar Artikel</h2>
24         {
25           this.state.listArtikel.map(artikel => {
26             return <Post key={artikel.id} judul={artikel.title} isi={artikel.body}/>
27           })
28         }
29       </div>
30     )
31   }
32 }
33
34 export default BlogPost;
```

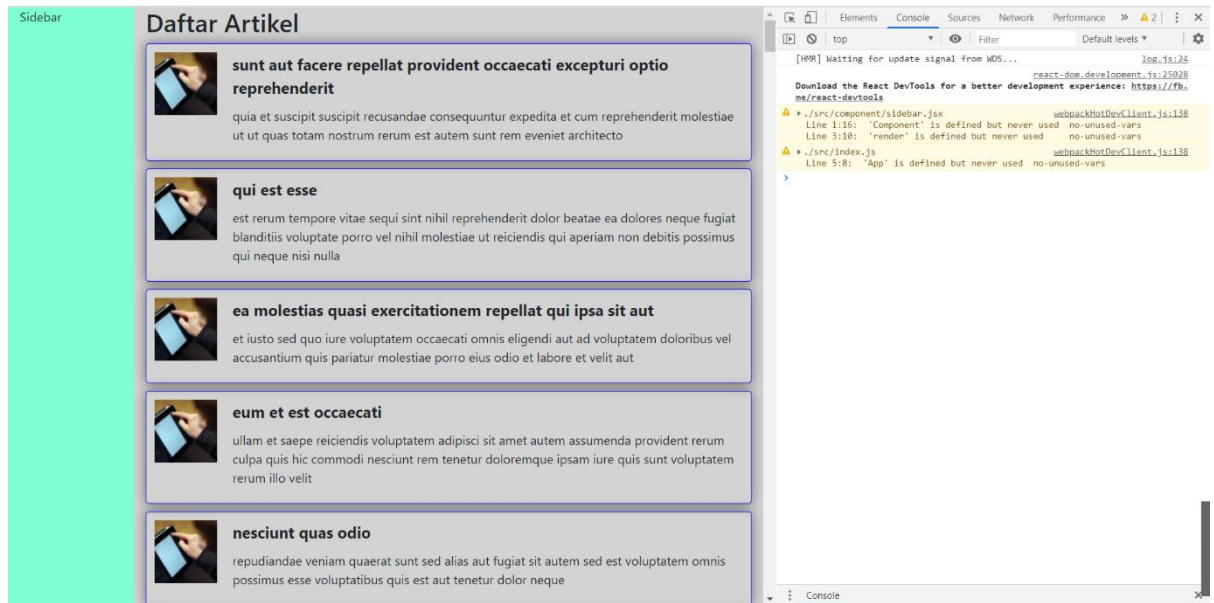
Tetapi masih terjadi error pada console log, karena harus ada unique key pada tiap komponen, unique key dapat diartikan sebagai id di data yang terdapat pada json



Maka ditambahkanlah id pada component

```
25   this.state.listArtikel.map(artikel => {
26     return <Post key={artikel.id} judul={artikel.title} isi={artikel.body}/>
```

Dan hasilnya sudah tidak ada error



2. Pertanyaan Praktikum 1

- a. Pada langkah 8, sekarang coba kalian ganti class container dengan container-fluid atau sebaliknya pada file "public/index.html" dan lihat apa perbedaannya.
 1. Tampilan seperti apa yang kalian temukan setelah mencoba mengganti nama class tersebut?
Komponen pada halaman web akan melebar mengikuti ukuran layar
 2. Apa perbedaan dari container dan container-fluid ?
Perbedaan container dan container-fluid terletak pada tampilan , dan jika halaman dizoom pada container component akan statis, jika container-fluid di zoom maka tampilan akan tetap mengikuti ukuran layar
- b. Jika kita ingin meng-import suatu component contoh component bootstrap, akan tetapi component dalam tersebut belum terdapat pada module ReactJS. Apa yang akan dilakukan untuk dapat menggunakan component tersebut? Bagaimana caranya?

Menginstal terlebih dahulu dengan perintah **npm install bootstrap --save** maka bootstrap akan terinstall dan tersimpan di node_module

Praktikum 2

Interaksi dengan API menggunakan Fake API

1. Praktikum 2

Untuk menggunakan fake API harus terlebih dahulu menginstal json-server dengan perintah

Npm install -g json-server , jika sudah terinstall lalu membuat file json yang digunakan untuk menyimpan data.

```
[
  {
    "userId": 1,
    "id": 1,
    "title": "Workshop Amazon Web Services (AWS) Untuk Para Dosen Jurusan Teknologi Informasi Politeknik Negeri Malang",
    "body": "Jumat (24/1/2020), Jurusan Teknologi Informasi Politeknik Negeri Malang (JTI Polinema) menyelenggarakan workshop tentang Amazon Web Services (AWS) yang bertempat di Lantai 7, Ruang LK 2, Gedung Teknik Sipil Polinema. Workshop ini diadakan untuk menunjang performa dan..."
  },
  {
    "userId": 1,
    "id": 2,
    "title": "Pelaksanaan Rapat Hasil Akhir Yudisium Semester Ganjil 2019/2020",
    "body": "Jumat (17/1/20) kemarin, Jurusan Teknologi Informasi Politeknik Negeri Malang (JTI Polinema) menyelenggarakan rapat hasil yudisium semester ganjil tahun ajaran 2019/2020. Acara ini dimulai pada jam 8.30 dan bertempat di lantai 8, Gedung Teknik Sipil, Politeknik Negeri..."
  },
  {
    "userId": 1,
    "id": 3,
    "title": "Kemeriahan Penutupan Digital Talent Scholarship VSGA 2019 Batch 2 di Politeknik Negeri Malang",
    "body": "Selesai sudah Digital Talent Scholarship (OTS) Batch 2 2019 dengan program Vocational School Graduate Academy(VSGA) di Jurusan Teknologi Informasi (JTI) Politeknik Negeri Malang (Polinema). Pelatihan yang didanai Kominfo dan bekerjasama dengan Jurusan..."
  },
  {
    "userId": 1,
    "id": 4,
    "title": "Jalin Kerjasama dengan Shenyang University of Technology, JTI Polinema Terbangkan Empat Mahasiswa ke Tiongkok",
    "body": "Empat mahasiswa Jurusan Teknologi Informasi (JTI) Politeknik Negeri Malang (Polinema) telah berangkat ke negeri tirai bambu pada hari Selasa (8/10/2019) kemarin, dalam rangka kerjasama antar kampus Polinema dan Shenyang University of Technology. Para mahasiswa tersebut..."
  },
  {
    "userId": 1,
    "id": 1583347697874,
    "title": "Ini adalah sebuah artikel ",
    "body": "isinya masih saya pikirkan, proses sambil jalan menulis sebuah artikel"
  }
]
```

Lalu menjalankan perintah **json-server -watch listArtikel.json -port 4000**

```
00
\n{^_^}/ hi!

Loading listArtikel.json
Done

Resources
http://localhost:4000/posts
http://localhost:4000/comments
http://localhost:4000/profile
http://localhost:4000/mahasiswa
```

Lalu melakukan penggantian API pada component BlogPost

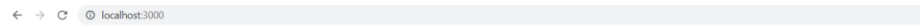
```
16  ambilDataDariServerAPI = () =>{
17    fetch('http://localhost:4000/posts?_sort=id&_order=desc')
18    .then(response => response.json())
19    .then(jsonHasilAmbilDariAPI => {
20      this.setState({
21        listArtikel: jsonHasilAmbilDariAPI
22      })
23    })
24  }
```


2. Pertanyaan Praktikum 2

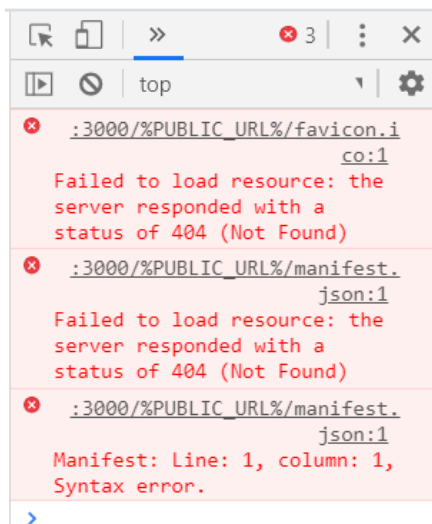
- a. Kenapa json-server dijalankan pada port 3001? Kenapa tidak sama-sama dijalankan pada port 3000 seperti project react yang sudah kita buat?

Jika json-server dan reactjs dijalankan pada port yang akan terjadi bentrok, yang menyebabkan halaman react maupun json-server tidak mau tampil

- b. Bagaimana jadinya kalau kita ganti port json-server menjadi 3000?



Jika dijalankan pada port yang sama maka pada browser tidak akan tampil apapun



Dan terjadi error pada console dimana server tidak dapat di load

Praktikum 3

Interaksi dengan API menggunakan method DELETE

1. Praktikum 3

Menambahkan button pada component stateless Post, dan menambahkan fungsi onClick(hapus) yang berfungsi untuk mengambil method hapus

```
12 <button className="btn btn-sm btn-warning" onClick={()=>props.hapusArtikel(props.idArtikel)}>Hapus</button>
```

Kemudian pada component statefull BlogPost, dimodifikasi dan ditambahkan fungsi hapus dengan memanggil method DELETE

```
54 handleHapusArtikel = (data) => {
55   fetch(`http://localhost:4000/posts/${data}`, {method: 'DELETE'})
56   .then(res => {
57     console.log(this.ambilDataDariServerAPI());
58   })
59 }
60
```

```
79 <h2>Daftar Artikel</h2>
80 {
81   this.state.listArtikel.map(artikel => {
82     return <Post key={artikel.id}
83       judul={artikel.title}
84       isi={artikel.body}
85       idArtikel = {artikel.id}
86       hapusArtikel={this.handleHapusArtikel}/>
87   })
88 }
```

2. Pertanyaan Praktikum 3

a. Apa yang terjadi setelah kalian klik tombol hapus?

Maka item yang dipilih akan terhapus

b. Perhatikan file listArtikel.json, apa yang terjadi pada file tersebut? Kenapa demikian?

Item yang ada di files listArtikel.json juga akan terhapus

c. Fungsi handleHapusArtikel itu untuk apa?

Untuk menghapus item yang ada pada json server

d. Jelaskan perbedaan fungsi componentDidMount() pada Gambar 1.18 dengan fungsi componentDidMount() pada Gambar 3.2 ?

perbedaan terletak pada pengambilan data dari json server, pada gambar 1.18 pengambilan data di masukkan langsung kedalam fungsi componentDidMount() sedangkan pada gambar 3.2 pengambilan data dari json server dilakukan pada fungsi yang berbeda dimana componentDidMount() hanya memanggil fungsi ambildata() saja

Praktikum 4

Interaksi dengan API menggunakan method POST

1. Praktikum 4

Method POST sering digunakan dalam mengirimkan form request ke server. Dalam API method POST biasa digunakan untuk melakukan insert/tambah data pada server.

Manambahkan form untuk menginputkan data pada json server

```
61 render() {
62   return(
63     <div className="post-artikel">
64       <div className="form pb-2 border-bottom">
65         <div className="form-group row">
66           <label htmlFor="title" className="col-sm-2 col-form-label">Judul</label>
67           <div className="col-sm-10">
68             <input type="text" className="form-control" id="title" name="title" onChange={this.handleTambahArtikel}/>
69           </div>
70         </div>
71         <div className="form-group row">
72           <label htmlFor="body" className="col-sm-2 col-form-label">Isi</label>
73           <div className="col-sm-10">
74             <textarea className="form-control" id="body" name="body" rows="3" onChange={this.handleTambahArtikel}></textarea>
75           </div>
76         </div>
77         <button type="submit" className="btn btn-primary" onClick={this.handleTombolSimpan}>Simpan</button>
78       </div>

```

Lalu menambahkan state dan request API

```
6   state = {
7     listArtikel: [],
8     insertArtikel: {
9       userId: 1,
10      id: 1,
11      title: "",
12      body: ""
13    }
14  }

```

Lalu menambahkan handle form , fungsi ini berguna untuk mengambil nilai inputan dari form

```
30 handleTambahArtikel = (event) =>{
31   let formInsertArtikel = {...this.state.insertArtikel};
32   let timestamp = new Date().getTime();
33   formInsertArtikel['id'] =timestamp;
34   formInsertArtikel[event.target.name] = event.target.value;
35   this.setState ({
36     insertArtikel: formInsertArtikel
37   })
38 }

```

Lalu menambahkan handle simpan , fungsi ini digunakan untuk menambahkan data json server dengan mengambil nilai dari fungsi handel form

```

30     handleTambahArtikel = (event) =>{
31         let formInsertArtikel = {...this.state.insertArtikel};
32         let timestamp = new Date().getTime();
33         formInsertArtikel['id'] =timestamp;
34         formInsertArtikel[event.target.name] = event.target.value;
35         this.setState ({
36             insertArtikel: formInsertArtikel
37         })
38     }

```

2. Pertanyaan Praktikum 4

a. Jelaskan apa yang terjadi pada file listArtikel.json sebelum dan setelah melakukan penambahan data?

Data akan bertambah dengan inputan yang dimasukkan

b. Data yang ditampilkan di browser adalah data terbaru berada di posisi atas dan data lama berada di bawah, sedangkan pada file listArtikel.json data terbaru malah berada di bawah. Jelaskan mengapa demikian?

Karena pada data yang ditampilkan pada browser di sorting secara descending, jadi data yang baru dimasukkan akan memiliki id yang lebih banyak dari yang lama maka dengan otomatis data yang masuk akan langsung ditampilkan pada bagian atas