

LAPORAN PRAKTIKUM MATA KULIAH
PEMROGRAMAN BERBASIS FRAMEWORK



Nama : Alenovan Wiradhika Putra
Kelas : TI-3C D-IV T.INFORMATIKA
NIM : 1741720065
No. Absen : 02

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2020

Praktikum 1 - 4

1. Kode pada file BlogPost.jsx

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

class BlogPost extends Component {
  state = {
    listArtikel: [],
    insertArtikel: {
      userId: 1,
      id: 1,
      title: "",
      body: ""
    }
  }

  ambilDataDariServerAPI() {
    fetch("http://localhost:3001/posts")
      .then(response => response.json())
      .then(jsonHasilAmbilDariAPI => {
        this.setState({
          listArtikel: jsonHasilAmbilDariAPI
        })
      })
  }

  componentDidMount() {
    this.ambilDataDariServerAPI()
  }

  handlerHapusArtikel = (data) => {
    fetch(`http://localhost:3001/posts/${data}`, { method: "DELETE" })
      .then(res => {
        this.ambilDataDariServerAPI()
      })
  }

  handlerTambahArtikel = (event) => {
    let formInsertArtikel = { ...this.state.insertArtikel };
    let timestamp = new Date().getTime();
```

```

    formInsertArtikel['id'] = timestamp;
    formInsertArtikel[event.target.name] = event.target.value;
    this.setState({
      insertArtikel: formInsertArtikel
    })
  }

  handlerTombolSimpan = () => {
    fetch(`http://localhost:3001/posts`, {
      method: "post",
      headers: {
        'accept': 'application/json',
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(this.state.insertArtikel)
    })
      .then(res => {
        this.ambilDataDariServerAPI()
      })
  }

  render() {
    return (
      <div className="post-artikel">
        <div className="form pb-2 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.handlerTambahArtikel}></input>
            </div>
          </div>
          <div className="form-group row">
            <label htmlFor="title" className="col-sm-2 col-form-label">Judul</label>
            <div className="col-sm-10">
              <textarea className="form-control" id="body" name="body" rows="3" onChange={this.handlerTambahArtikel}></textarea>
            </div>
          </div>
          <button type="submit" className="btn btn-primary" onClick={this.handlerTombolSimpan}>Simpan</button>
        </div>
      </div>
    )
  }

```

```

        <h2>Daftar Artikel</h2>
        {
          this.state.listArtikel.map(artikel => {
            return <Post key={artikel.id} judul={artikel.title} isi={
artikel.body} id={artikel.id} hapusArtikel={this.handlerHapusArtikel} />
          })
          // console.log(this.state.listArtikel)
        }
      </div>
    )
  }
}

export default BlogPost;

```

2. Kode pada file index.js

```

3. import React from 'react';
4. import ReactDOM from 'react-dom';
5. import './index.css';
6. import * as serviceWorker from './serviceWorker';
7. import 'bootstrap/dist/css/bootstrap.min.css';
8. import BlogPost from './container/BlogPost/BlogPost'
9.
10. ReactDOM.render(<BlogPost />, document.getElementById("content"));
11. serviceWorker.unregister();

```

12. Kode pada file index.html

```

13. <!DOCTYPE html>
14. <html lang="en">
15.
16. <head>
17.   <meta charset="utf-8" />
18.   <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
19.   <meta name="viewport" content="width=device-width, initial-scale=1" />
20.   <meta name="theme-color" content="#000000" />

```

```
21. <meta name="description" content="Web site created using create-react-
    app" />
22. <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
23. <!--
24.     manifest.json provides metadata used when your web app is installed
    on a
25.     user's mobile device or desktop. See https://developers.google.com/w
    eb/fundamentals/web-app-manifest/
26. -->
27. <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
28. <!--
29.     Notice the use of %PUBLIC_URL% in the tags above.
30.     It will be replaced with the URL of the `public` folder during the b
    uild.
31.     Only files inside the `public` folder can be referenced from the HTM
    L.
32.
33.     Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" w
    ill
34.     work correctly both with client-side routing and a non-
    root public URL.
35.     Learn how to configure a non-
    root public URL by running `npm run build`.
36. -->
37. <title>React App</title>
38.</head>
39.
40.<body>
41. <noscript>You need to enable JavaScript to run this app.</noscript>
42. <!-- <div id="root"></div> -->
43. <div class="">
44.   <div class="row">
45.     <div class="col-2" id="sidebar">Sidebar</div>
46.     <div class="col-10" id="content"></div>
47.   </div>
48.
49. </div>
50. <!--
51.     This HTML file is a template.
52.     If you open it directly in the browser, you will see an empty page.
53.
54.     You can add webfonts, meta tags, or analytics to this file.
55.     The build step will place the bundled scripts into the <body> tag.
56.
57.     To begin the development, run `npm start` or `yarn start`.
```

```
58.         To create a production bundle, use `npm run build` or `yarn build`.
59.     -->
60.</body>
61.
62.</html>
```

63. Kode pada file BlogPost.css

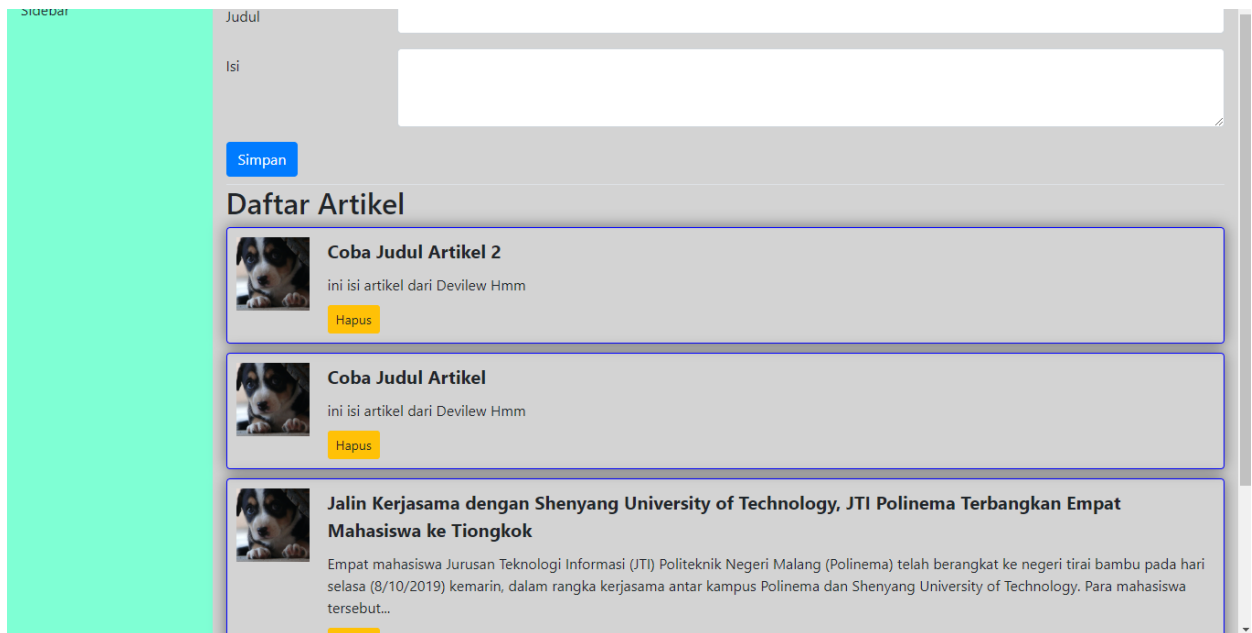
```
.artikel{
  width: 100%;
  padding: 10px;
  border: 1px solid blue;
  border-radius: 4px;
  margin-bottom: 10px;
  box-shadow: 0 0 16px rgba(0,0,0,0.5);
  display: flex;
}

.gambar-artikel{
  height: 80px;
  width: 80px;
  margin-right: 20px;
  vertical-align: top;
}

.gambar-artikel img{
  width: 100%;
  height: 100%;
  object-fit: cover;
}

.konten-artikel div.judul-artikel{
  font-size: 20px;
}
```

64. Hasil akhir p Praktikum:



Pertanyaan Praktikum 1

- a) Pada langkah 8, sekarang coba kalian ganclass 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?
 2. Apa perbedaan dari container dan container-fluid?

Perbedaan container dan container-fluid

Hasil :

```
15 <body>
16   <noscript>You need to enable JavaScript to run this app.</noscript>
17   <div class="container">
18     <div class="row">
19       <div class="col-2" id="sidebar">Sidebar</div>
20       <div class="col-10" id="content"></div>
21     </div>
22   </div>
23 </div>
```

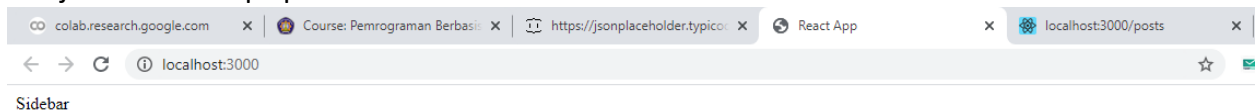
Perbedaan container dan container-fluid adalah jika menggunakan container-fluid maka tampilan pada website lebih penuh (sesuai layar) daripada menggunakan container

- 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? Pertama membuat file component, kemudian melakukan import sesuai lokasi component

```
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import 'bootstrap/dist/css/bootstrap.min.css';
5 import * as serviceWorker from './serviceWorker';
6 import BlogPost from './minggu4-5/container/BlogPost/BlogPost'
7
8 ReactDOM.render((BlogPost) => document.getElementById('content'),
9   function unregister(): void
```

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?
Karena jika sama-sama dijalankan pada port 3000, file json tidak dapat dijalankan
- b) Bagaimana jadinya kalau kita ganti port json-server menjadi 3000?
File json tidak tampil pada web



Pertanyaan Praktikum 3

- a) Apa yang terjadi setelah kalian klik tombol hapus?
Artikel pada web akan terhapus
- b) Perhatikan file listArtikel.json, apa yang terjadi pada file tersebut? Kenapa demikian?
Isi pada file listArtikel.json juga terhapus
- c) Fungsi handleHapusArtikel itu untuk apa?
Untuk menghapus data pada artikel
- d) Jelaskan perbedaan fungsi componentDidMount() pada Gambar 1.18 dengan fungsi componentDidMount() pada Gambar 3.2?
Pada Gambar 1.18 fungsi componentDidMount() langsung menampilkan data, pada Gambar 3.2 fungsi componentDidMount() menampilkan data dari fungsi ambilDataDariServerAPI

Pertanyaan Praktikum 4

- a) Jelaskan apa yang terjadi pada file listArtikel.json sebelum dan setelah melakukan penambahan data?
Isi pada file listArtikel.json juga bertambah setelah melakukan penambahan data
- b) Data yang ditampilkan di browser adalah data terbaru diposisi atas dan data lama berada dibawah, sedangkan pada file listArtikel.json data terbaru malah berada dibawah. Jelaskan mengapa demikian?
Karena pada fungsi ambilDataDariServerAPI di set desc