LAPORAN TUGAS PEMROGRAMAN BERBASIS FRAMEWORK

Modul 4

Interaksi dengan API

****

Disusun Oleh :

Nama : Hunayn Risatayn

NIM : 1841720148

Kelas : TI-3E

Program Studi D-IV Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

Maret 2021

# Praktikum – Bagian 1: Interaksi dengan API menggunakan method GET

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| **1** | Buka Project React pada pertemuan sebelumnya dan jalankan “npm start” menggunakan cmd dalam direktori tersebut.      Buat folder baru bernama “BlogPost” pada folder container (statefull component).    Buat file BlogPost.jsx dan BlogPost.css di dalam folder “BlogPost”, seperti pada Gambar 1.2.    Buka file BlogPost.jsx dan ketikkan kode seperti Gambar 1.3.    Pada file index.js, lakukan import component BlogPost seperti Gambar 1.4    Pada web browser akan tampil seperti pada Gambar 1.5.    Tahapan selanjutnya adalah perbaikan tampilan sebuah website untuk mempercantik halaman website tersebut dengan menggunakan Bootstrap yang umum digunakan.  Import css bootstrap.min.css (css bootstrap yang sudah dikompresi) ke dalam index.js (seperti Gambar 1.6). Jika css tidak ditemukan, install lewat cmd dengan perintah “npm install bootstrap”      Modifikasi file index.html pada folder "public" seperti Gambar 1.7. Cermati code program yang ada dalam gambar!.      Amati tampilan yang ada pada browser (seperti Gambar 1.8)    Buka file index.css dan tambahkan code css seperti Gambar 1.9, untuk menambah sedikit style pada halaman web    Perhatikan kembali browser, dan lihat hasil tampilan seperti Gambar 1.10.    Kita ingin sebuah website memiliki tampilan seperti pada Gambar 1.1. Dengan minimal ada gambar artikel, judul, dan deskripsi artikel. Maka contoh data dummy yang akan kita pakai bisa menggunakan data dari http://placeimg.com contoh http://placeimg.com/120/120/any. Tahapan edit tampilan post artikel:  Ubah kode program untuk statefull component BlogPost.jsx menjadi seperti Gambar 1.11      Tambahkan custom css ke BlogPost.css seperti Gambar 1.12    Perhatikan tampilan browser.    **Pemindahan dari statefull component ke stateless component**  Buat folder BlogPost pada folder component (stateless component), lalu buat file Post.jsx    Potong (cut) baris 9-17 pada statefull component BlogPost.jsx ke stateless component Post.jsx, dan modifikasi Post.jsx seperti Gambar 1.13.    Untuk statefull component BlogPost.jsx pada baris 10, panggil stateless component Post.jsx seperti Gambar 1.14.    Perhatikan hasil tampilan browser, apa yang terjadi?    **Muat Data Dinamis.**  Pada statefull component BlogPost.jsx, tambahkan parameter yang ingin dilempar ke stateless component untuk ditampilkan. Kode program bisa dilihat pada Gambar 1.15.    Setelah itu pada stateless component Post.jsx tangkap parameter yang dilempar oleh statefull component seperti pada Gambar 1.16 dan lihat pada browser apa yang terjadi!.    Simpan, dan amati apa yang terjadi pada browser kalian!.    **Mengambil data Post/Artikel dari API.**  Gunakan state untuk menyimpan data hasil request dariAPI  data API yang akan kita gunakan adalah data dummy dari https://jsonplaceholder.typicode.com/posts, dimana memiliki 4 element data yaitu userid, id, title, body (seperti pada Gambar 1.17)    Edit pada statefull component BlogPost.jsx seperti pada Gambar 1.18 dan perhatikan dengan seksama akan penjelasan dibeberapa baris kode program tersebut.        Lihat hasilnya pada browser. Kemudian klik kanan pada browser pilih "inspect element" kemudian pilih tab "console". Refresh browser dan amati apa yang terjadi  Jika terlihat seperti padaGambar 1.19, maka terjadi kesalahan pada program yang kita buat    Jika terjadi hal demikian, hal ini terjadi karena dalam react "class" dalam tag html harus ditulis menjadi "className". selain itu, pada statefull component yang dinamis, harus ada "UNIQUE KEY" pada tiap komponen yang diproses sehingga komponen perlu diberi UNIQUE KEY.  UNIQUE KEY dapat diambil dari element yang ada pada data API yang sudah kita ambil (contoh saat ini adalah element id pada data API (userid, id, title, body) yang akan kita gunakan untuk UNIQUE KEY. Lihat Gambar 1.20.    Simpan dan lihat apa yang terjadi pada console browser (Gambar 1.21).    **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 apaperbedaannya.  1. Tampilan seperti apa yang kalian temukan setelah mencoba mengganti nama class tersebut?  Apabila class container maka content akan berada di tengah tidak memenuhi ruangan, sedangkan jika class container-fluid content akan memenuhi ruangan hingga ujung kanan kiri full  2. Apa perbedaan dari container dan container-fluid ?  Container untuk mendefinisikan kotak terluar yang menyelimuti row tetapi tidak memenuhi ruangan full kanan kiri sedangkan container-fluid memenuhi ruangan  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?  Pada root app kita buka melalui terminal kemudian ketikkan “npm install bootstrap” kemudian import dengan syntax “import "bootstrap/dist/css/bootstrap.min.css";”  **Praktikum 2**  **Interaksi dengan API menggunakan Fake API**  Install Fake API (JSON Server)    Copy-kan file json listArtikel.json yang sudah ada pada direktori project reactjs kita.    Buka cmd baru pada direktori project, lalu ketik perintah json-server --watch listArtikel.json –-port 3001    Kita cek url resource yang ada padaFake APIserver ke browser apakah bisa diakses. Ketik url http://localhost:3001/posts padabrowser    Untuk memastikan lagi, kita edit statefull component BlogPost (Gambar 1.18) pada baris 11. Kita ganti url API dari https://jsonplaceholder.typicode.com/posts menjadi <http://localhost:3001/posts>    Simpan perubahan dan amati apa yang terjadi.    **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 port 3000 sudah digunakan untuk operasi GET pada <https://jsonplaceholder.typicode.com/posts> sedangkan untuk mengambil data di local project react-web-hunayn harus menggunakan port yang berbeda  b. Bagaimana jadinya kalua kita ganti port json-server menjadi 3000?  Bisa diganti ke port 3000 untuk mengambil data local yang menjadi server tetapi job yang sebelumnya harus dilakukan terminate kemudian run pada port 3000  **Praktikum 3**  **Interaksi dengan API menggunakan method DELETE**  Buka stateless component Post. Tambahkan 1 baris kode program pada baris 10 seperti pada Gambar 3.1    Kemudian pada statefull component BlogPost, modifikasi kode program sebelumnya sesuai dengan Gambar 3.2    Klik tombol hapus pada list artikel di browser. Amati apa yang terjadi    Ketika tombol dihapus maka index tersebut akan terhapus sehingga hanya menampilkan index yang tersisa    **Pertanyaan Praktikum 3**   1. Apa yang terjadi setelah kalian klik tombol hapus?   Maka data yang diklik hapus akan hilang, sehingga hanya menampilkan data yang tersisa   1. Perhatikan file listArtikel.json, apa yang terjadi pada file tersebut? Kenapa demikian?   Data pada id = 1 terhapus karena telah di klik button hapus pada id tersebut.   1. Fungsi handleHapusArtikel itu untuk apa?   Function yang digunakan untuk menghandle dalam penghapusan data, dimulai dengan fetch data pada localhost:3001/posts kemudian menjalankan method ‘DELETE’   1. Jelaskan perbedaan fungsi componentDidMount() pada Gambar 1.18 dengan fungsi componentDidMount() pada Gambar 3.2 ?   componentDidMount() pada gambar 1.18 digunakan untuk mengambil data pada <https://jsonplaceholder.typicode.com/posts> sedangkan pada gambar 3.2 digunakan untuk mengambil data pada server local  **Praktikum 4**  **Interaksi dengan API menggunakan method POST**  Buka statefull component BlogPost, dan modifikasi pada fungsi render() untuk menampilkan form input artikel yang berisi judul dan isi berita. seperti pada Gambar 4.1    Kemudian modifikasi BlogPost untuk bagian state dan request API dari server, seperti Gambar 4.2    Tambahkan untuk handle form tambah data artikel seperti Gambar 4.3    Langkah terakhir tambahkan fungsi untuk handle tombol simpan artikel, seperti pada Gambar 4.4    Simpan, lakukan percobaan penambahan data, dan amati perubahannya        **Pertanyaan Praktikum 4**   1. Jelaskan apa yang terjadi pada file listArtikel.json sebelum dan setelah melakukan penambahan data?   Sebelum dilakukan penambahan data, file listArtikel.json dilakukan pengambilan data terlebih dahulu kemudian dilakukan sorting dan order secara desc lalu data diubah menjadi JSON kemudian data JSON tersebut dilakukan insert ke listArtikel pada state. Setelah penambahan data, data akan ditampung sementara dalam insertArtikel kemudian dilakukan eksekusi function handleTambahArtikel, dan terakhir eksekusi function handleTombolSimpan   1. 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 saat function ambilDataDariServerAPI() dieksekusi dilakukan order berdasarkan id secara desc sehingga data dengan id yang lebih besar (terakhir dilakukan input) akan berada pada posisi paling atas diikuti dengan id yang lebih kecil berada di bawahnya  **TUGAS PRAKTIKUM**  Buatlah program menggunakan Fake API (JSON Server) tentang pendataan Mahasiswa aktif/cuti/lulus di Jurusan Teknologi Informasi. Atribut-atribut yang ada dari mahasiswa adalah NIM, nama, alamat, no hp, tahun Angkatan, dan status. Buatlah aplikasi yang menggunakan API dengan method GET, DELETE, dan POST.      Link GitHub : <https://github.com/hunaynr/framework_prog/tree/main/4thWeek>  Link Youtube : <https://youtu.be/Dq8-0gz1W1o> |