**LAPORAN PEMROGRAMAN FRAMEWORK**

**“SESSION 4 API”**



**Oleh:**

**Nama : Bagus Satria Putra**

**Kelas : 3F**

**Absen/NIM : 8 / 1841720146**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**4 Maret 2021**

**Untuk link youtube dan github ada di halaman terakhir.**

**API**

API adalah singkatan dari Application Programming Interface, dan memungkinkan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. API terdiri dari berbagai elemen seperti function, protocols, dan tools lainnya yang memungkinkan developers untuk membuat aplikasi. Tujuan penggunaan API adalah untuk mempercepat proses development dengan menyediakan function secara terpisah sehingga developer tidak perlu membuat fitur yang serupa. Penerapan API akan sangaat terasa jika fitur yang diinginkan sudah sangat kompleks. Gambar 1. API API dapat anda temui dalam kehidupan sehari-hari seperti saat anda memesan hotel, mengirimkan pesan, memesan makanan secara online maupun ketika mengunduh sebuah software. Kenapa menggunakan API? API membuat pemrograman menjadi lebih mudah. Kebutuhan kita sebagai pelanggan dan khususnya bagi developer sangat dimudahkan dengan adanya API. Dengan melihat hal tersebut, peran dari API sendiri sangat berat terlebih untuk membuat tampilan sebuah aplikasi menjadi interaktif, mudah untuk digunakan, dan bersahabat untuk pengguna. Tidak hanya itu, API juga digunakan untuk berkomunikasi antara layanan-layanan. API memiliki peran yang sangat penting dalam teknologi.

Keuntungan menggunakan API Bagi Para Developer Antara Lain: 1. Aplikasi. API membantu kinerja dari aplikasi lebih cepat dan fleksibel seperti layanan dan informasi yang diberikan karena API dapat memasuki komponen-komponen aplikasi. 2. Kustomisasi Dengan API, kustomisasi untuk konten dan layanan dapat dilakukan sesuai kebutuhan dan keinginan. 3. Fleksibel API membuat layanan menjadi lebih fleksibel. Hal tersebut karena API mendukung data migrasi lebih baik dan informasi yang didapat ditinjau lebih dekat. 4. Integrasi / integration API dapat menjamin pengiriman informasi lebih lancar dikarenakan API memungkinkan konten tertanam dari aplikasi maupun situs dengan mudah. Hal tersebut memberikan pengalaman yang terintegrasi bagi pengguna. 5. Lebih banyak data API memberikan banyak pilihan karena semua informasi yang dihasilkan di tingkat pemerintah tersedia untuk setiap warga negara.

|  |  |
| --- | --- |
| **No.** | **Keterangan** |
| 1. 1 | 1. Buka Project React pada pertemuan sebelumnya dan jalankan “npm start” menggunakan cmd dalam direktori tersebut.  2. Buat folder baru bernama “BlogPost” pada folder container (statefull component).  3. Buat file BlogPost.jsx dan BlogPost.css di dalam folder “BlogPost”, seperti pada Gambar 1.2.  **Jawaban.** |
|  | 4. Buka file BlogPost.jsx dan ketikkan kode seperti Gambar 1.3.  **Jawaban :** |
|  | 5. Pada file index.js, lakukan import component BlogPost seperti Gambar 1.4  6. Pada web browser akan tampil seperti pada Gambar 1.5  **Jawaban :** |
|  |  |
|  | 7. 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”  **Jawaban :** |
|  | 8. Modifikasi file index.html pada folder "public" seperti Gambar 1.7. Cermati code program yang ada dalam gambar!.  **Jawaban :** |
|  | 9. Amati tampilan yang ada pada browser (seperti Gambar 1.8)  **Jawaban :** |
|  | 10. Buka file index.css dan tambahkan code css seperti Gambar 1.9, untuk menambah sedikit style pada halaman web  **Jawaban :** |
|  | 11. Perhatikan kembali browser, dan lihat hasil tampilan seperti Gambar 1.10.  **Jawaban :** |
|  | 12. Ubah kode program untuk statefull component BlogPost.jsx menjadi seperti Gambar 1.11  **Jawaban :** |
|  | 13. Tambahkan custom css ke BlogPost.css seperti Gambar 1.12  **Jawaban :** |
|  | 14. Perhatikan tampilan browser.  **Jawaban ;** |
|  | 15. Buat folder BlogPost pada folder component (stateless component), lalu buat file Post.jsx  16. Potong (cut) baris 9-17 pada statefull component BlogPost.jsx ke stateless component Post.jsx, dan modifikasi Post.jsx seperti Gambar 1.13.  **Jawaban :** |
|  | 17. Untuk statefull component BlogPost.jsx pada baris 10, panggil stateless component Post.jsx seperti Gambar 1.14.  **Jawaban :** |
|  | 18. Perhatikan hasil tampilan browser, apa yang terjadi?  **Jawaban :** |
|  | 19. Pada statefull component BlogPost.jsx, tambahkan parameter yang ingin dilempar ke stateless component untuk ditampilkan. Kode program bisa dilihat pada Gambar 1.15.  **Jawaban :** |
|  | 20. 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!.  **Jawaban :** |
|  | 21. Simpan, dan amati apa yang terjadi pada browser kalian!.  **Jawaban ;** |
|  | 22. Gunakan state untuk menyimpan data hasil request dariAPI  23. 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)  24. Edit pada statefull component BlogPost.jsx seperti pada Gambar 1.18 dan perhatikan dengan seksama akan penjelasan dibeberapa baris kode program tersebut.  25. Lihat hasilnya pada browser. Kemudian klik kanan pada browser pilih "inspect element" kemudian pilih tab "console". Refresh browser dan amati apa yang terjadi.  26. Jika terlihat seperti padaGambar 1.19, maka terjadi kesalahan pada program yang kita buat.  27. 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.  28. 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.  **Jawaban :** |
|  | 29. Simpan dan lihat apa yang terjadi pada console browser (Gambar 1.21).  **Jawaban :** |
| **Praktikum 2**  **Interaksi dengan API menggunakan Fake API** | |
| 1 | 2.1 Install Fake API (JSON Server) Fake API/JSON Server bisa kita dapatkan di halaman https://github.com/typicode/jsonserver. Tahapan install dan membuat data json sendiri  1. Install pada direktori project reactjs kita dengan perintah npm install -gjson-server  **Jawaban :** |
| 2 | 2. Copy-kan file json listArtikel.json yang sudah ada pada direktori project reactjs kita.  3. Buka cmd baru pada direktori project, lalu ketik perintah json-server --watch listArtikel.json –-port 3001 .  **Jawaban :** |
| 3 | 4. Apabila pada cmd tampil seperti Gambar 2.1, maka server Fake API local kita telah siap  **Jawaban :** |
| 4 | 5. Kita cek url resource yang adapada Fake APIserver ke browser apakah bisa diakses. Ketik url http://localhost:3001/posts padabrowser  **Jawaban :** |
| **Praktikum 3**  **Interaksi dengan API menggunakan method DELETE** | |
| 1 | 3.1 Langkah Praktikum 3  1. Buka stateless component Post. Tambahkan 1 baris kode program pada baris 10 seperti pada Gambar 3.1  **Jawaban :** |
| 2 | 2. Kemudian pada statefull component BlogPost, modifikasi kode program sebelumnya sesuai dengan Gambar 3.2  **Jawaban :** |
| 3 | 3. Klik tombol hapus pada list artikel di browser. Amati apa yang terjadi  **Jawaban :** |
| **Praktikum 4**  **Interaksi dengan API menggunakan method POST** | |
| 1 | 4.1 Langkah Praktikum 4  1. 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  **Jawaban :** |
| 2 | 2. Kemudian modifikasi BlogPost untuk bagian state dan request API dari server, seperti Gambar 4.2  **Jawaban :** |
| 3 | 3. Tambahkan untuk handle form tambah data artikel seperti Gambar 4.3  **Jawaban :** |
| 4 | 4. Langkah terakhir tambahkan fungsi untuk handle tombol simpan artikel, seperti pada Gambar 4.4  **Jawaban :** |
| 5 | 5. Simpan, lakukan percobaan penambahan data, dan amati perubahannya  **Jawaban :** |

**1.4 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?

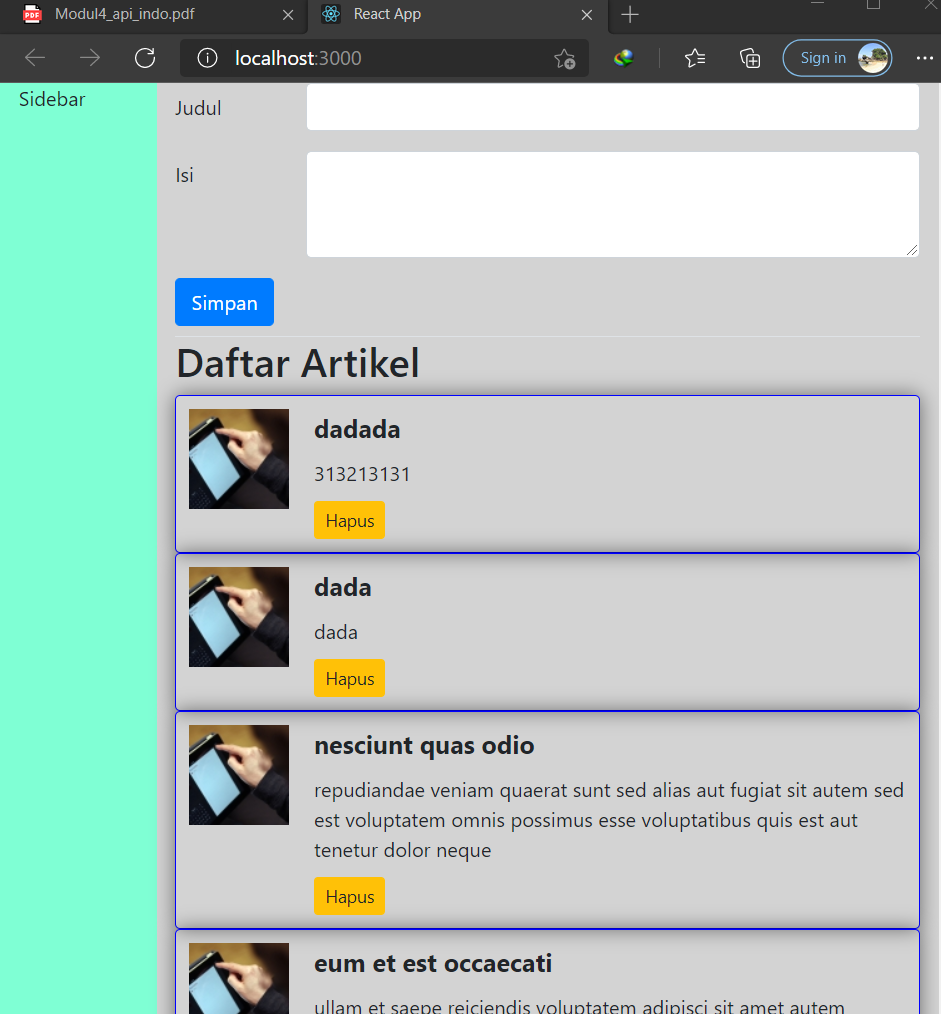
2. Apa perbedaan dari container dan container-fluid ?

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?

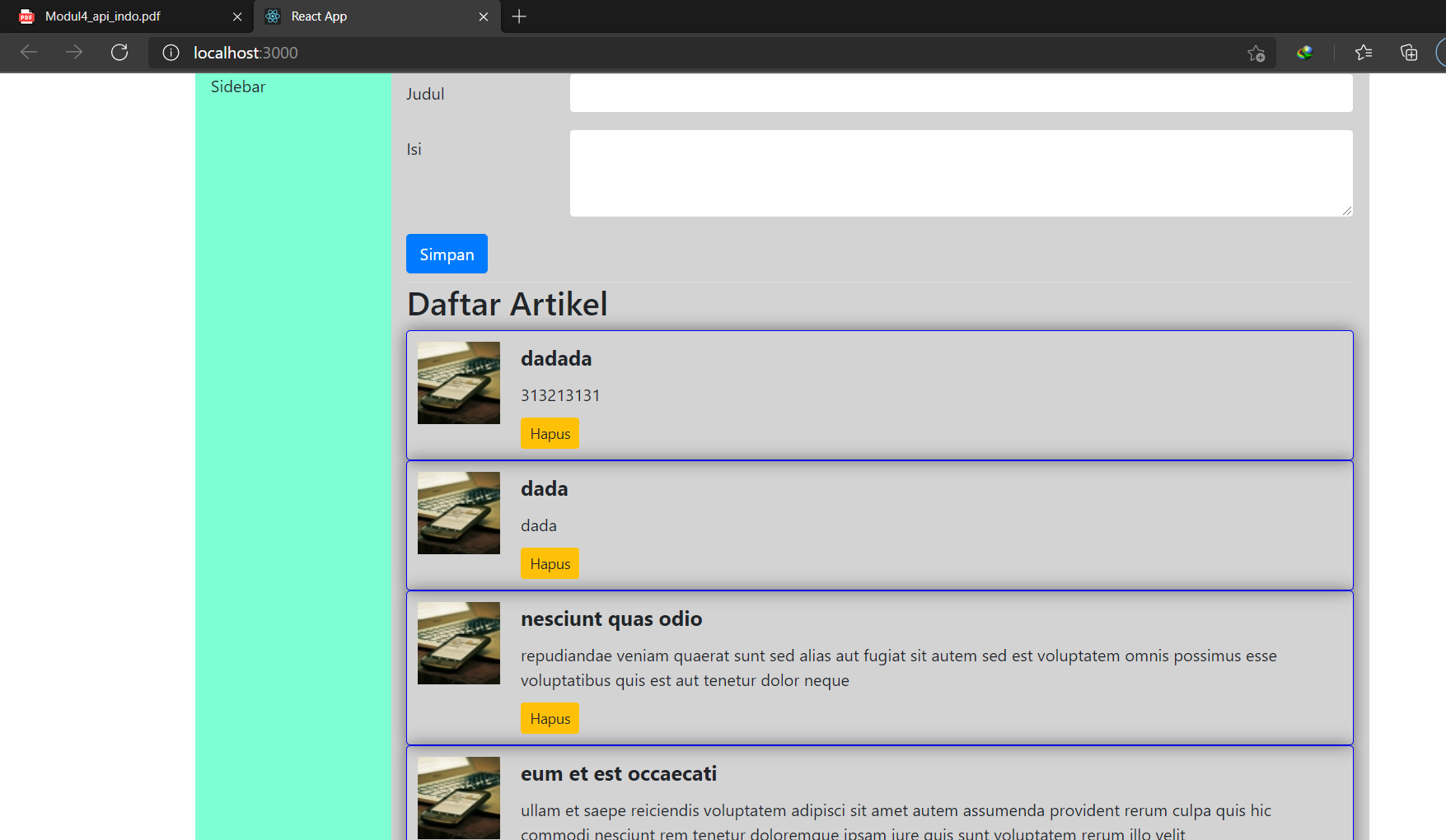
**Jawaban**

**a.**

**1.**

**Container-fluid**

**Container**



**2. container** berukuran tidak penuh, dan letaknya di tengah, maka **container**-**fluid** adalah kebalikannya, **container**-**fluid** memiliki lebar penuh.

b.

npm install bootstrap,atau download .zip bootstrap lalu copas di project kita.

**2.2 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?

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

**Jawaban**

1. Node-js dengan default maka port 3000 apabila json-server ingin menggunakan port 300 tidak bisa karena port 3000 sudah dipakai oleh node-js
2. fungsi untuk mendirikan sebuah server REST API tiruan dengan usaha minimal.maka kita harus matikan node-js dulu

**3.2 Pertanyaan Praktikum 3**

a. Apa yang terjadi setelah kalian klik tombol hapus?

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

c. Fungsi handleHapusArtikel itu untuk apa?

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

**Jawaban**

1. maka list terhapus
2. apabila klik hapus di server secara otomatis data json di listArtikel terhapus
3. melakukan panggilan data server lalu menggunakan method DELETE apabila salah satu data terhapus sesuai kita pilih maka langsung menjalankan fungsi ambilDataDariServer untuk melakukan update data json.
4. componentDidMount pada Gambar 1.18 mengecek Ketika component telah dimounting maka panggil API,dengan mengubah response data dari URL API menjadi sebuah data json,lalu pada akhirnya data json hasilnya di masukkan ke dalam list Artikel pada state. componentDidMount pada Gambar 3.2 untuk memudahkan dan membuat fungsi sendiri.

**4.2 Pertanyaan Praktikum 4**

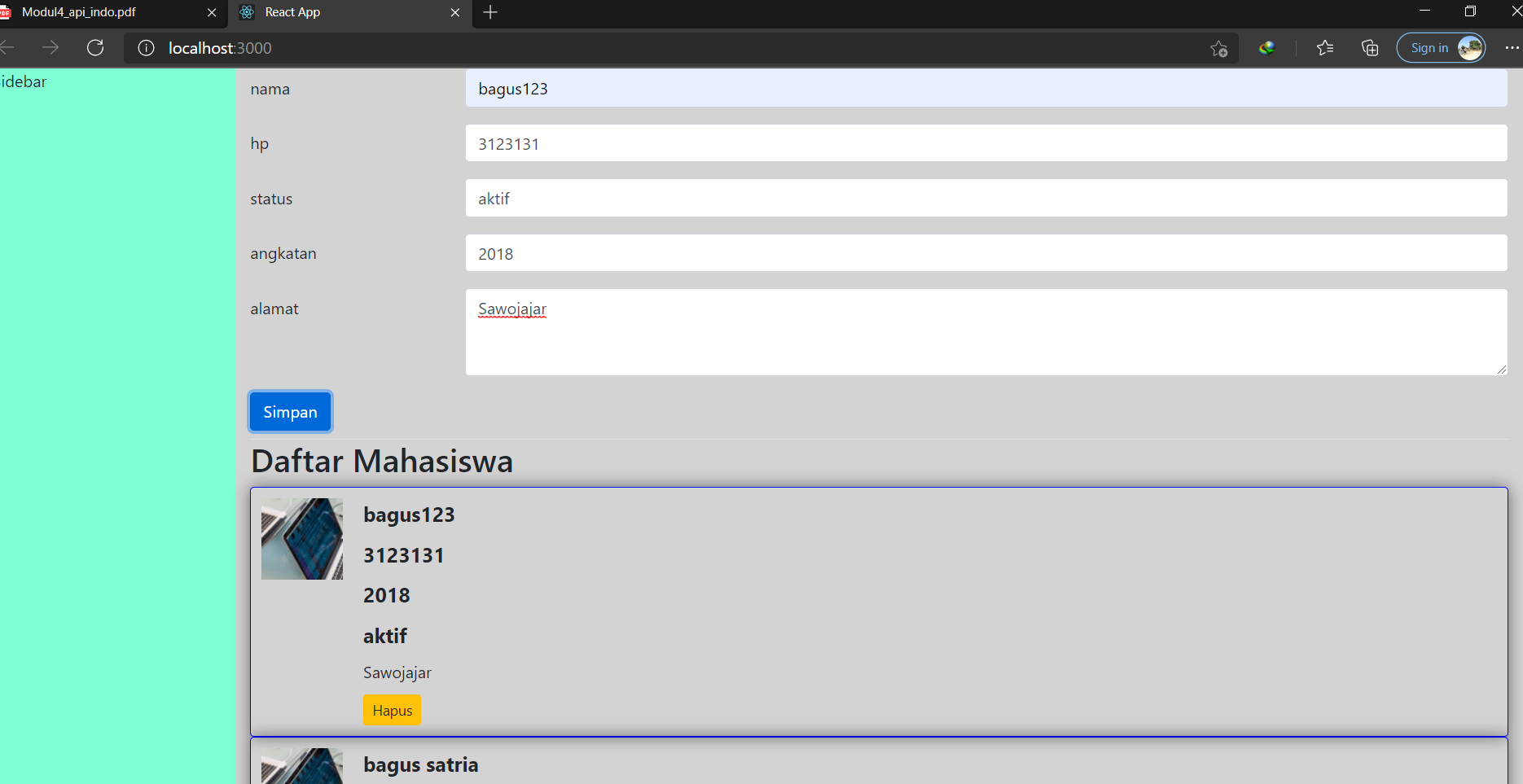
a. Jelaskan apa yang terjadi pada file listArtikel.json sebelum dan setelah melakukan penambahan data? b. Data yang ditampilkan di browser adalah data terbaru berada di posisi atas dan data lama berada di bawah, sedangkan pada file listArtikel.json data terbaru malah berada di bawah. Jelaskan mengapa demikian?

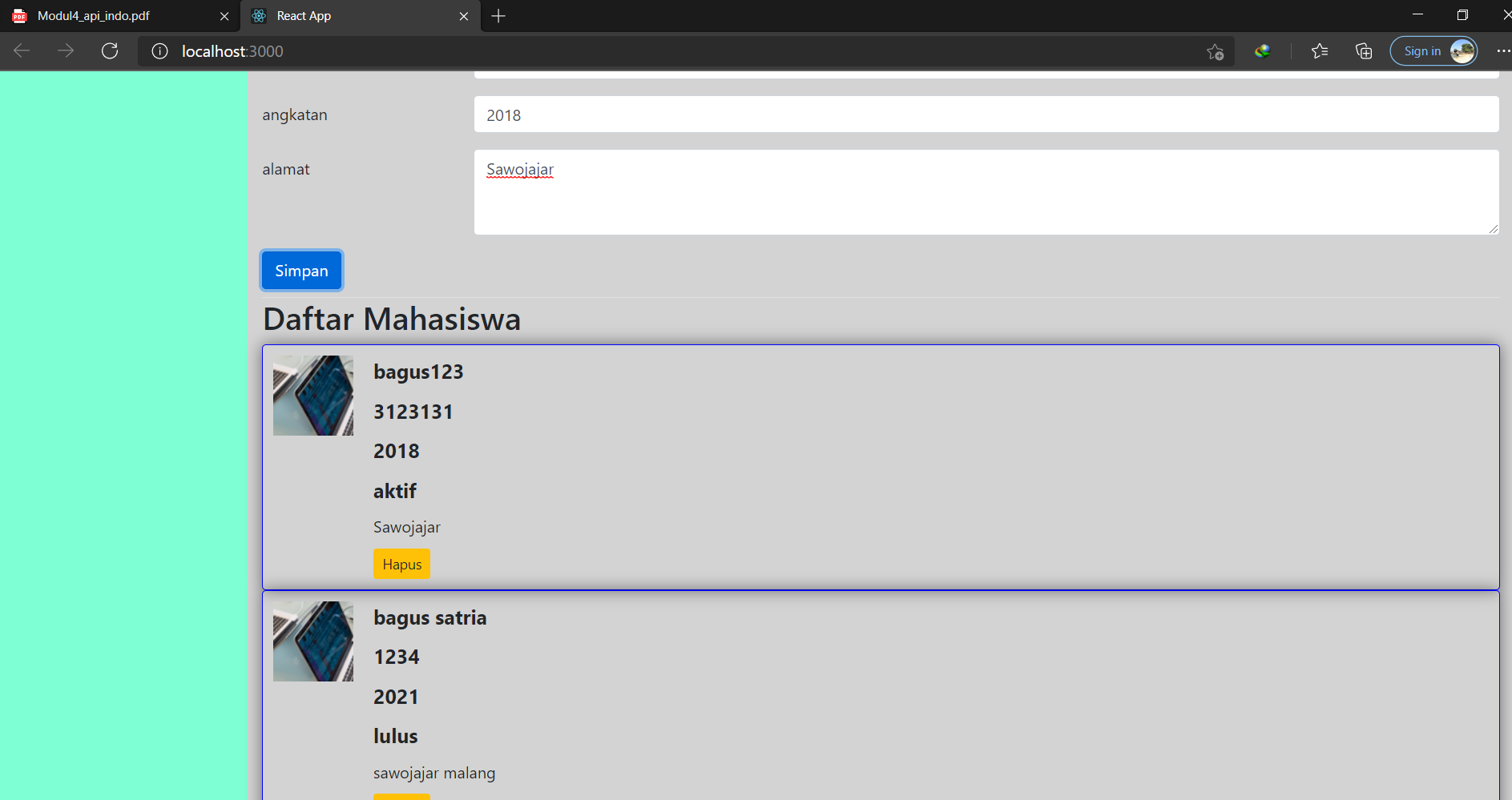
**Jawaban**

1. apabila melakukan penambahan data maka secara otomatis di file listAstikel.json tambah data sendiri, apabila melakukan pengurangan data maka secara otomatis di file listAstikel.json kurang data sendiri
2. dikarenakan kita menulis script “<http://localhost:3001/posts?_sort=id&_order=desc>” yang artinya data apabila kita insert maka secara otomatis di list artikel yang terbaru diatas

**TUGAS PRAKTIKUM Modul 4**

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 dan youtube**

1. Session 1 Pemrograman Berbasis Framework

<https://youtu.be/RlfXRslwKos>

<https://github.com/BagusSatria123/PemrogramanFrameworkReact/tree/master/Modul%201>

1. Session 2 Modern JavaScript Pemrograman Berbasis Framework

<https://youtu.be/POzDchFJAU4>

<https://github.com/BagusSatria123/PemrogramanFrameworkReact/tree/master/Pertemuan%202>

1. Session 3: React Component Pemrograman Berbasis Framework

<https://youtu.be/BlxF-eVq28I>

<https://github.com/BagusSatria123/PemrogramanFrameworkReact/tree/master/Pertemuan%203>

1. Session 4 API Pemrograman Berbasis Framework

<https://youtu.be/J-fZymTnQU4>

<https://github.com/BagusSatria123/PemrogramanFrameworkReact/tree/master/Pertemuan%204>

***--Alhamdulillah--***