INTERAKSI DENGAN API – Pertemuan 4

PEMOGRAMAN BERBASIS FRAMEWORK



Dosen Pengampu:

Arie Rahmat Syulistyo, S. Kom, M. Kom

Oleh:

Christian Daniel Prayogo

1941720181

TI 3F

Jurusan Teknologi Informasi

Prodi D4 Teknik Informatika

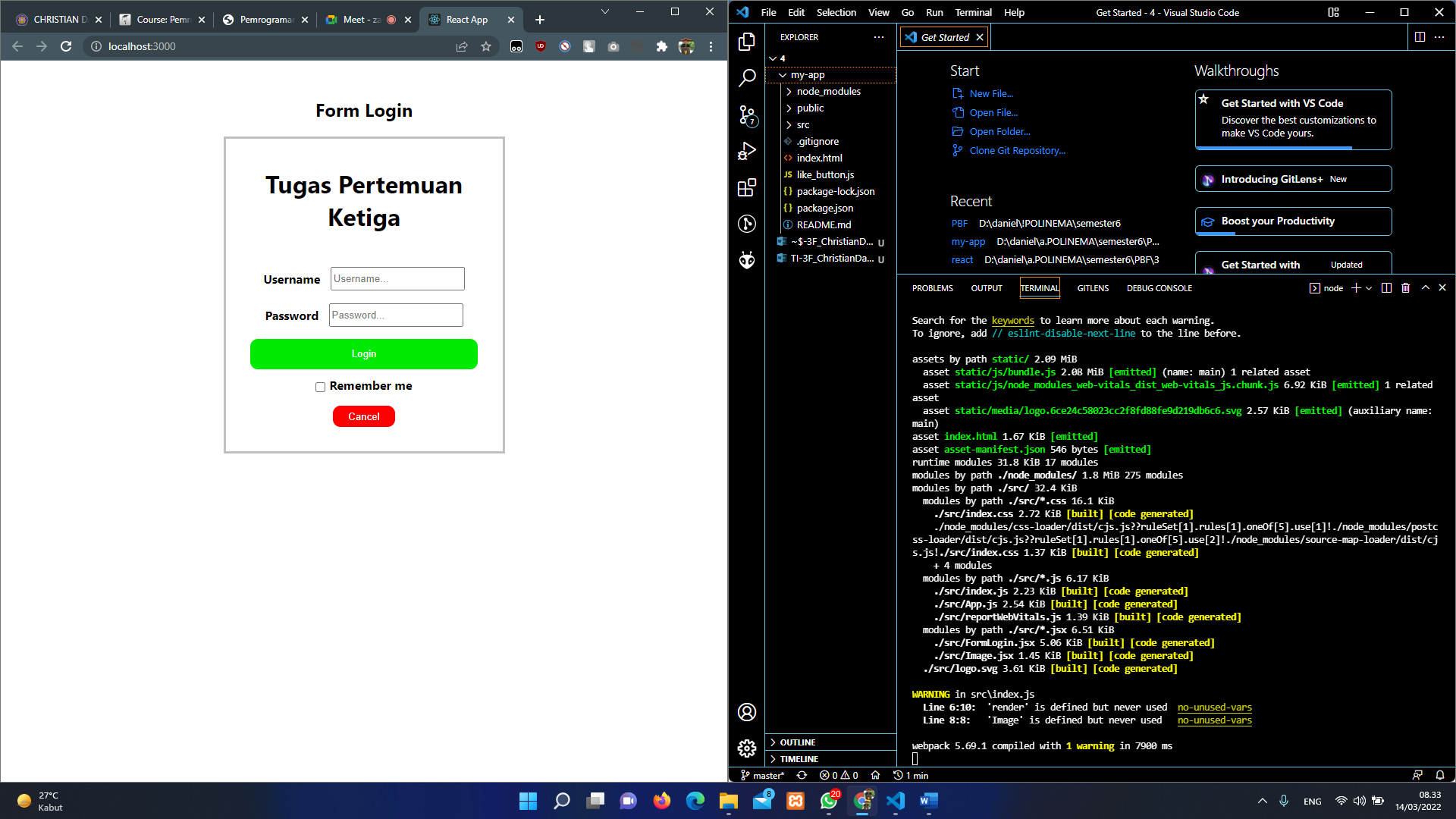
Politeknik Negeri Malang

2022

**Praktikum 1 Interaksi dengan API menggunakan method GET**

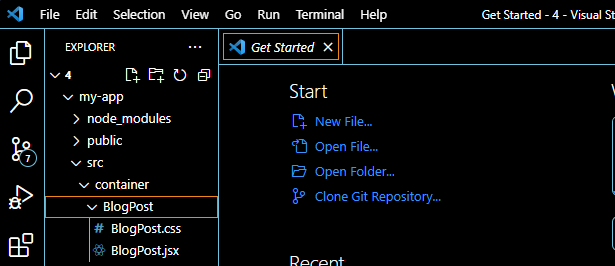
1.3 Langkah Praktikum

1. Buka Project React pada pertemuan sebelumnya dan jalankan “npm start” menggunakan cmd dalam direktori tersebut.

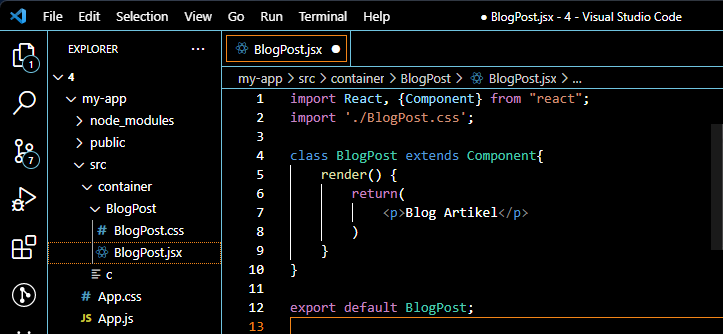


2. Buat folder baru bernama “BlogPost” pada folder container (statefull component).

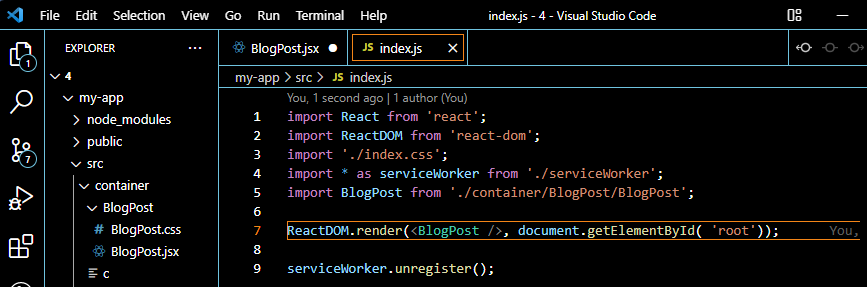
3. Buat file BlogPost.jsx dan BlogPost.css di dalam folder “BlogPost”, seperti pada Gambar 1.2.



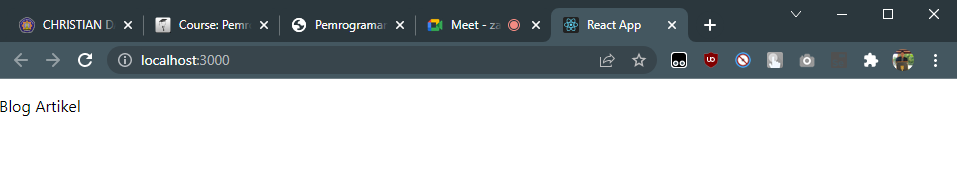
4. Buka file BlogPost.jsx dan ketikkan kode seperti Gambar 1.3.



5. Pada file index.js, lakukan import component BlogPost seperti Gambar 1.4.



6. Pada web browser akan tampil seperti pada Gambar 1.5.



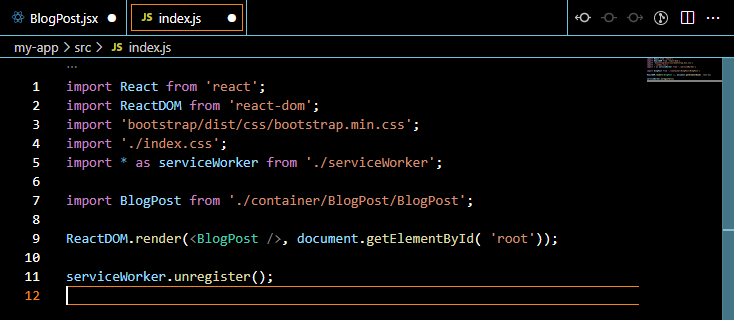
Tahapan selanjutnya adalah perbaikan tampilan sebuah website untuk mempercantik halaman

website tersebut dengan menggunakan Bootstrap yang umum digunakan.

7. Import css bootstrap.min.css (css bootstrap yang sudah dikompresi) ke dalam index.js (seperti

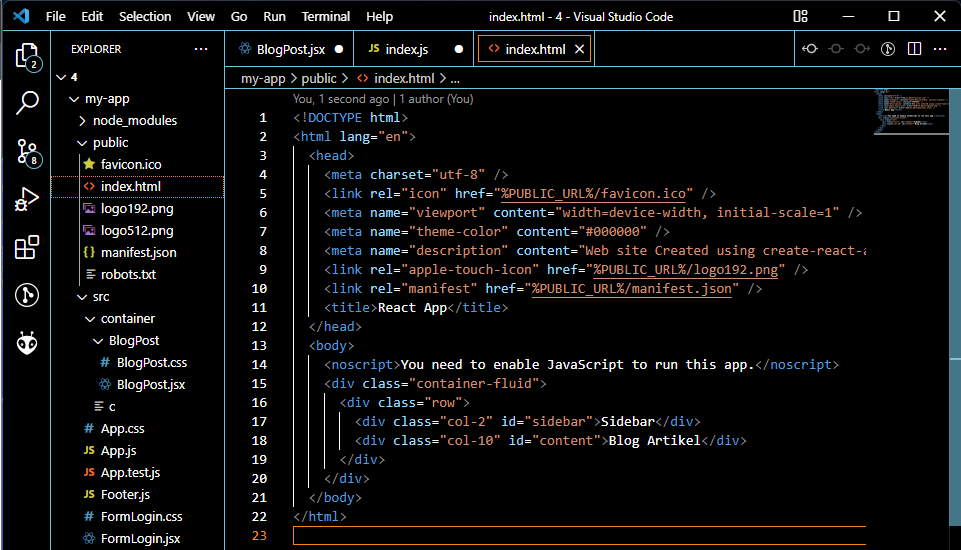
Gambar 1.6). Jika css tidak ditemukan, install lewat cmd dengan perintah “npm install

bootstrap”

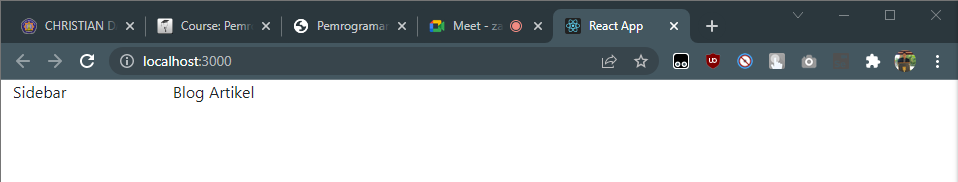


8. Modifikasi file index.html pada folder "public" seperti Gambar 1.7. Cermati code program

yang ada dalam gambar!.

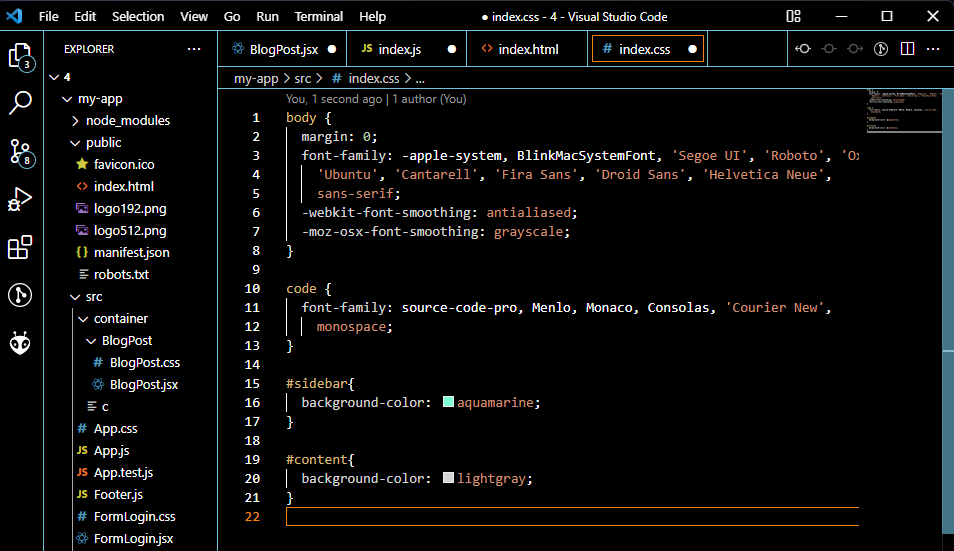


9. Amati tampilan yang ada pada browser (seperti Gambar 1.8)

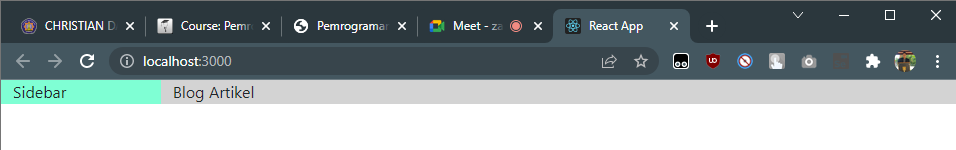


10. Buka file index.css dan tambahkan code css seperti Gambar 1.9, untuk menambah sedikit

style pada halaman web



11. Perhatikan kembali browser, dan lihat hasil tampilan seperti Gambar 1.10.



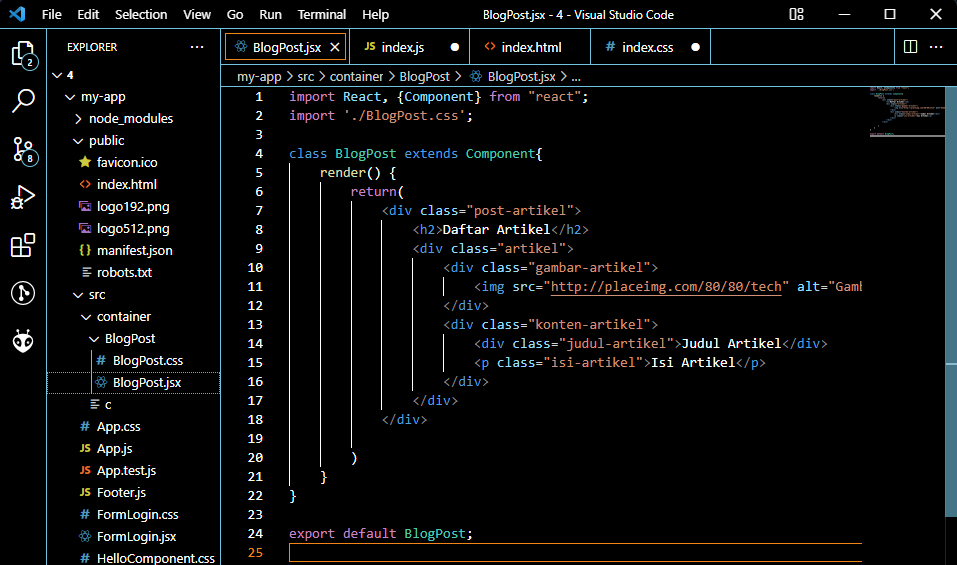
Kita ingin sebuah website memiliki tampilan seperti pada Gambar 1.1. Dengan minimal ada

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.

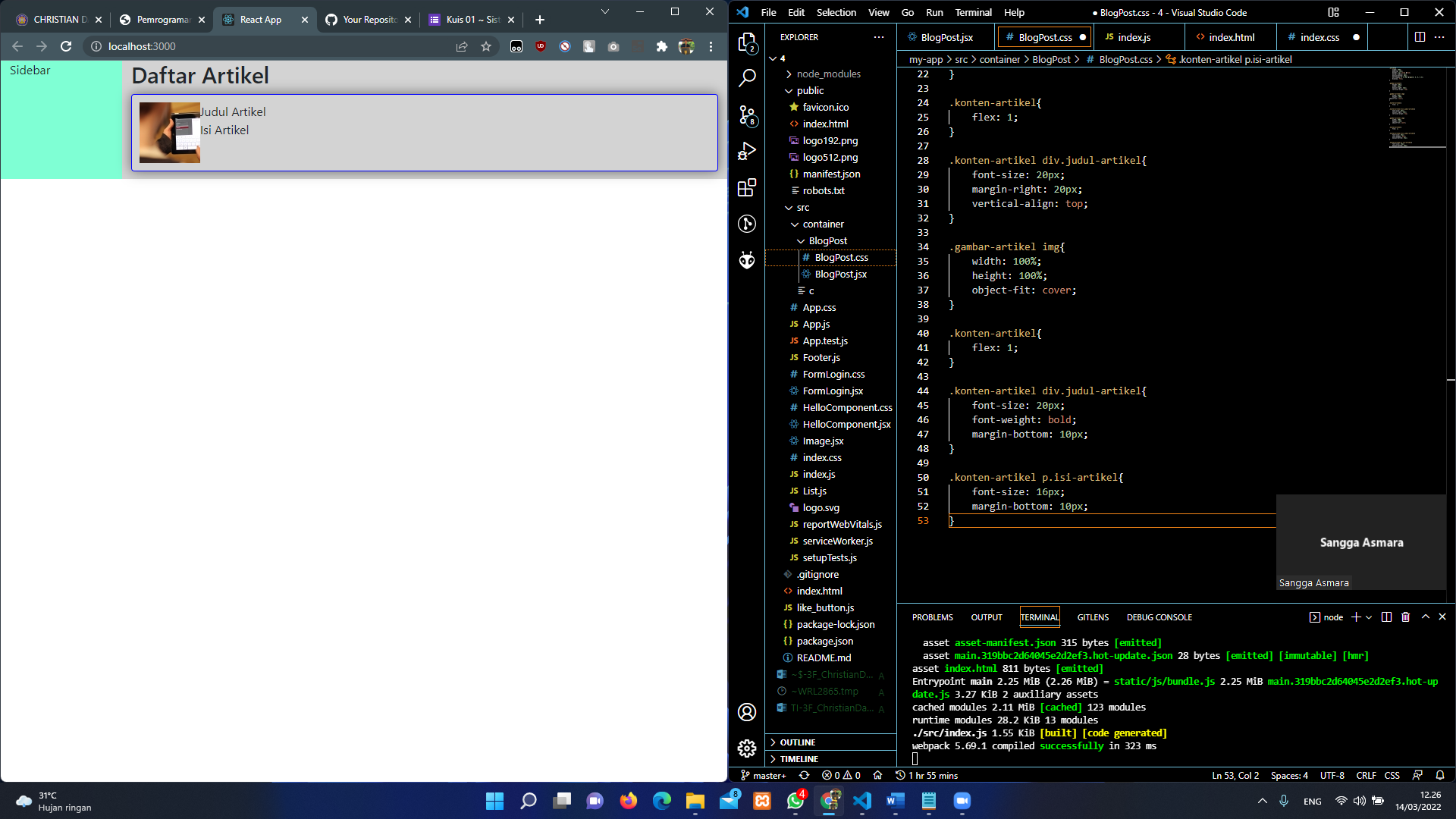
Tahapan edit tampilan post artikel:

12. Ubah kode program untuk statefull component BlogPost.jsx menjadi seperti Gambar 1.11



13. Tambahkan custom css ke BlogPost.css seperti Gambar 1.12

14. Perhatikan tampilan browser



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

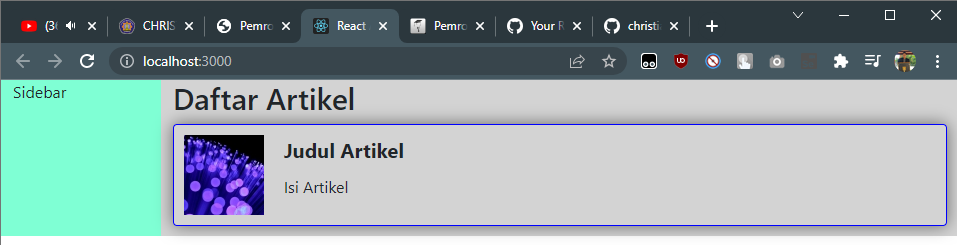
Post.jsx, dan modifikasi Post.jsx seperti Gambar 1.13.



17. Untuk statefull component BlogPost.jsx pada baris 10, panggil stateless component Post.jsx

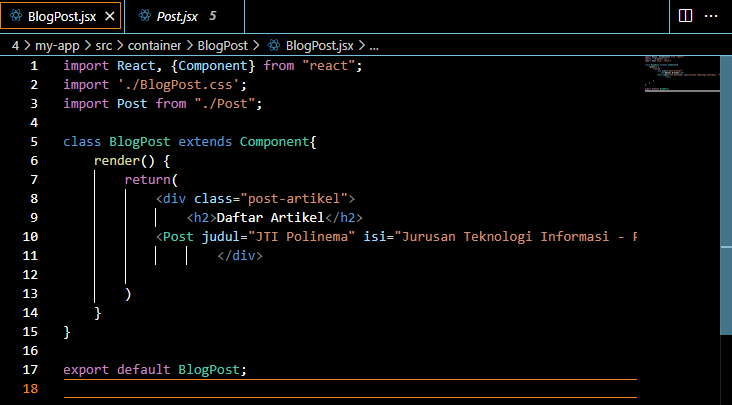
seperti Gambar 1.14.

18. Perhatikan hasil tampilan browser, apa yang terjadi?



Bagaimana caranya untuk dapat membuat data dinamis (lebih dari 1 artikel) dimana data Judul dan Deskripsi pada artikel didapat dari API?

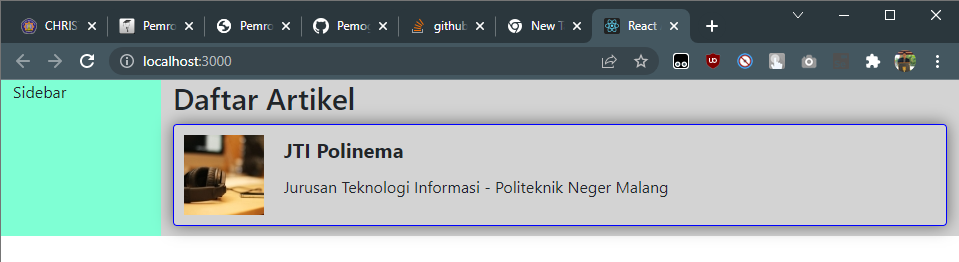
19. Pada statefull component BlogPost.jsx, tambahkan parameter yang ingin dilempar ke stateless component untuk ditampilkan. Kode program bisa dilihat pada Gambar 1.15



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!.



21. Simpan, dan amati apa yang terjadi pada browser kalian!.



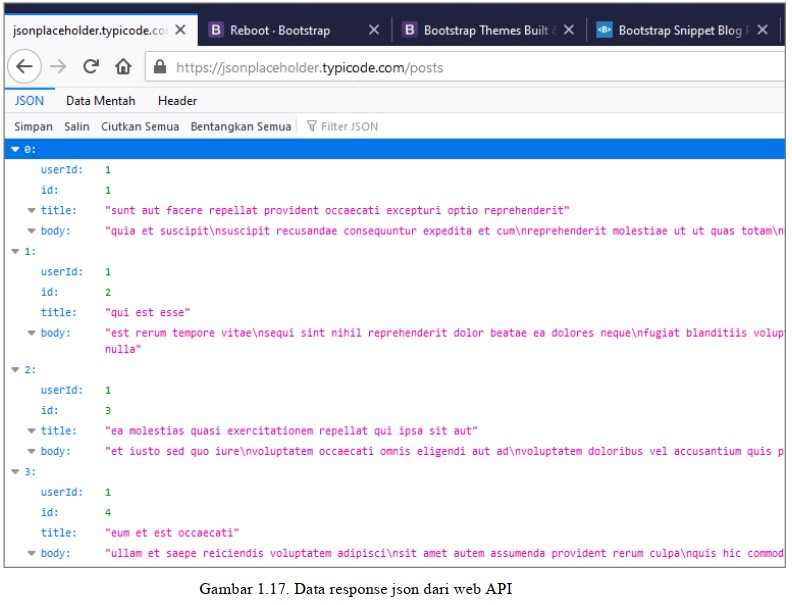
Mengambil data Post/Artikel dari API.

Bagaimana caranya untuk mendapatkan list artikel berdasarkan data json dari web API (contohnya: <https://jsonplaceholder.typicode.com/posts> ) ?

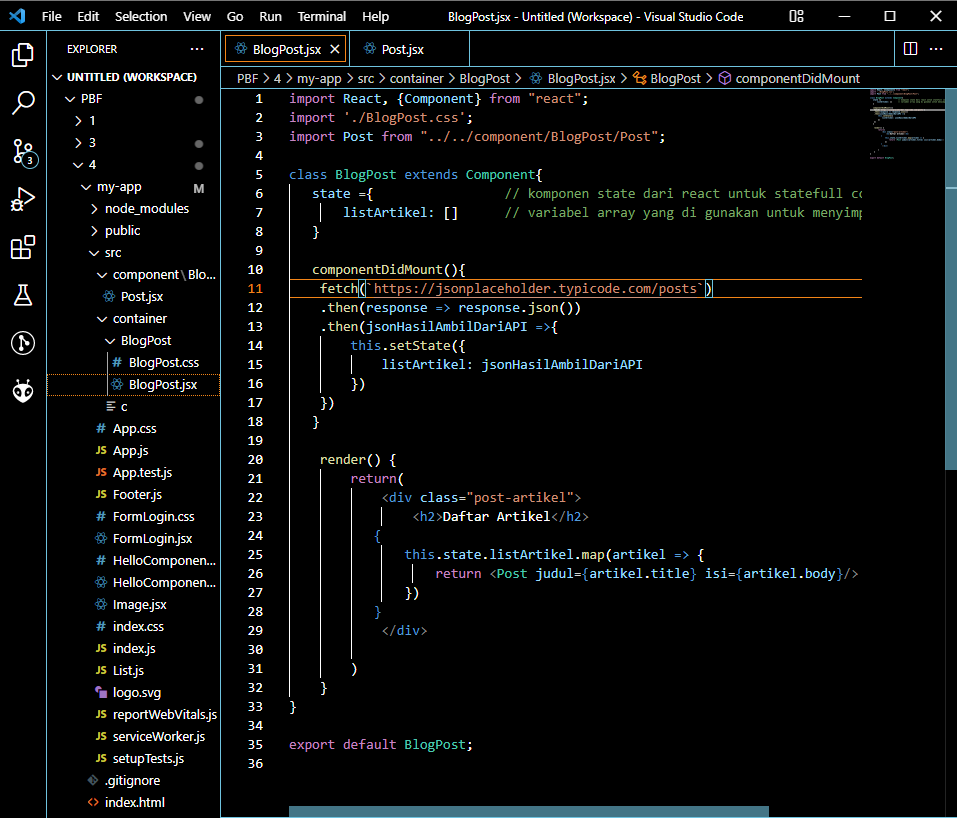
Kita gunakan life cycle component yaitu componentDidMount() dimana ketika komponen selesai dimount-ing, program akan memanggil API.

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)



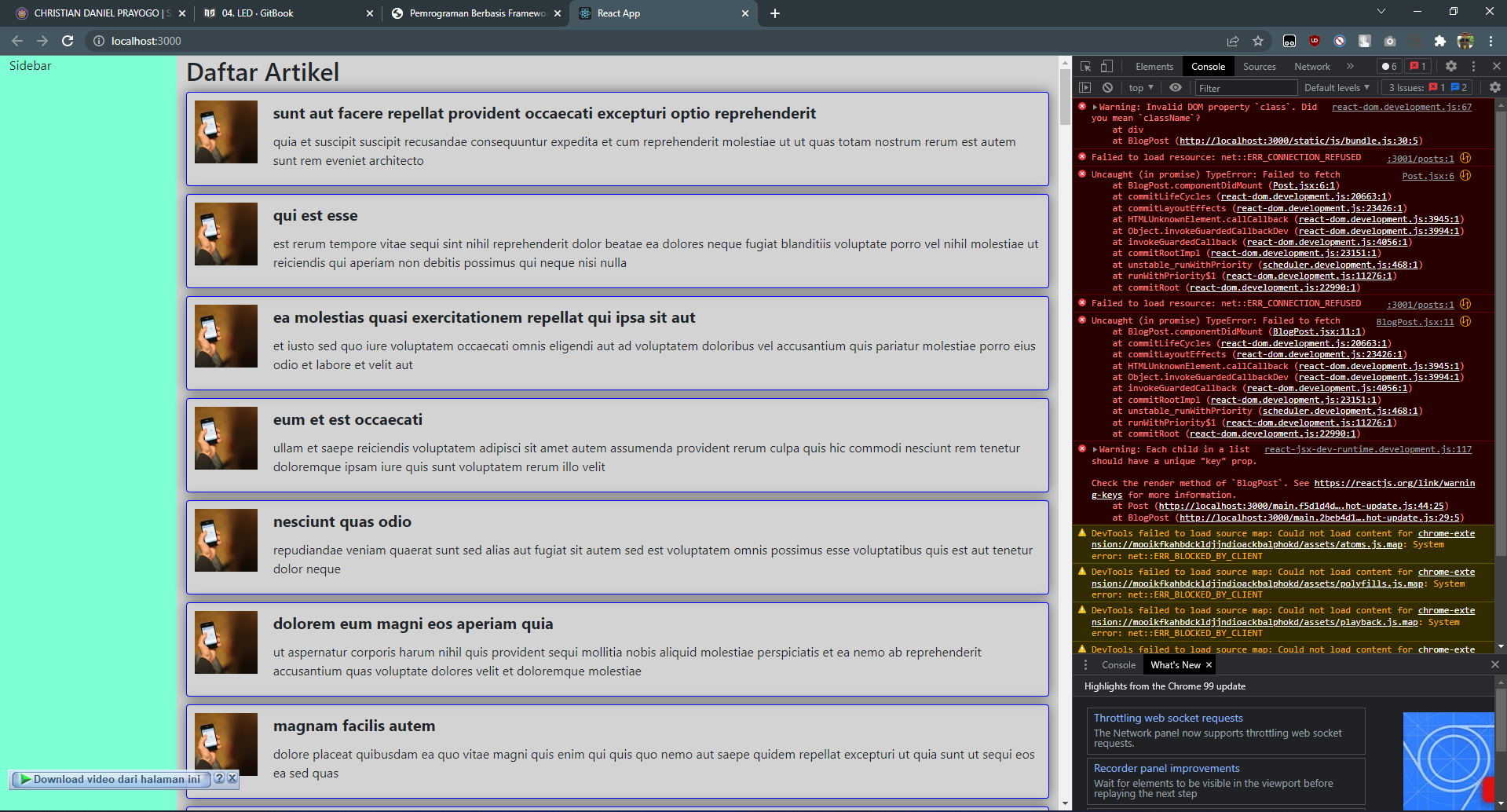
24. Edit pada statefull component BlogPost.jsx seperti pada Gambar dibawah ini dan perhatikan dengan seksama akan penjelasan dibeberapa baris kode program tersebut.



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 pada Gambar 1.19, maka terjadi kesalahan padaprogram 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

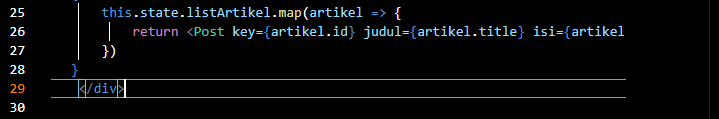
"UNIQUE KEY" pada tiap komponen yang diproses sehingga komponen perlu diberi UNIQUE

KEY.

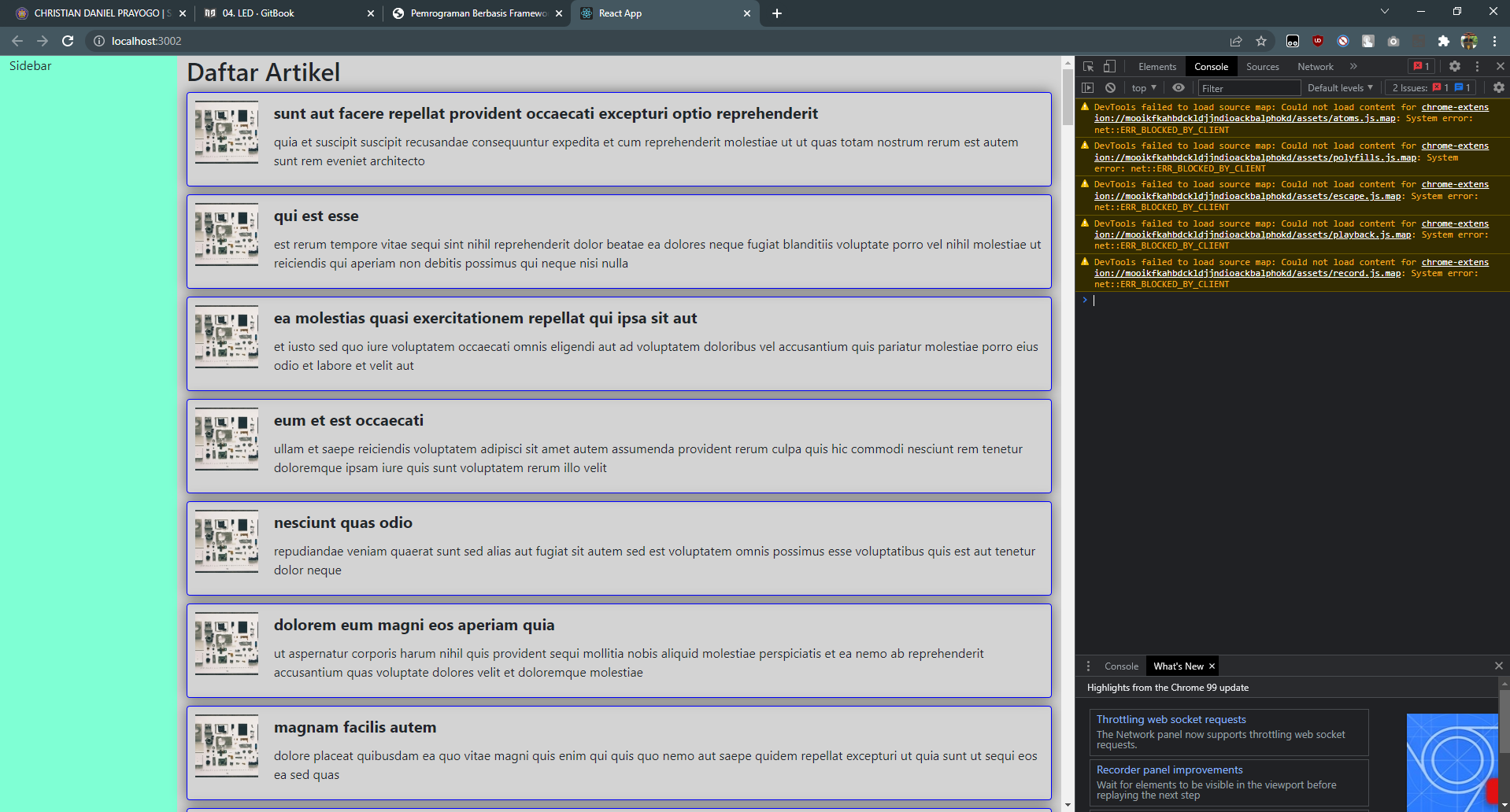
28. UNIQUE KEY dapat diambil dari element yang ada pada data API yang sudah kita ambil

(contoh saat ini adalah element id pada data API (userid, id, title, body) yang akan kita

gunakan untuk UNIQUE KEY. Lihat Gambar 1.20.



29. Simpan dan lihat apa yang terjadi pada console browser (Gambar 1.21).



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?

Praktikum 2

Interaksi dengan API menggunakan Fake API

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

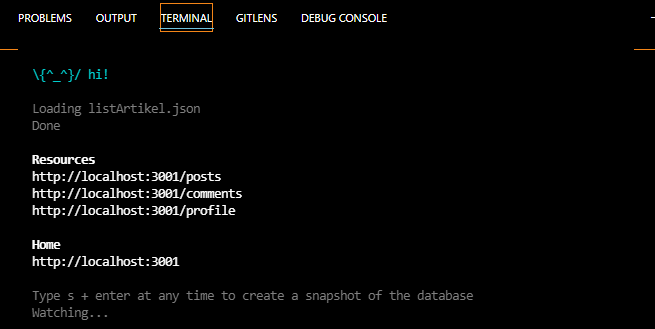
1. Install pada direktori project reactjs kita dengan perintah npm install -gjson-server

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

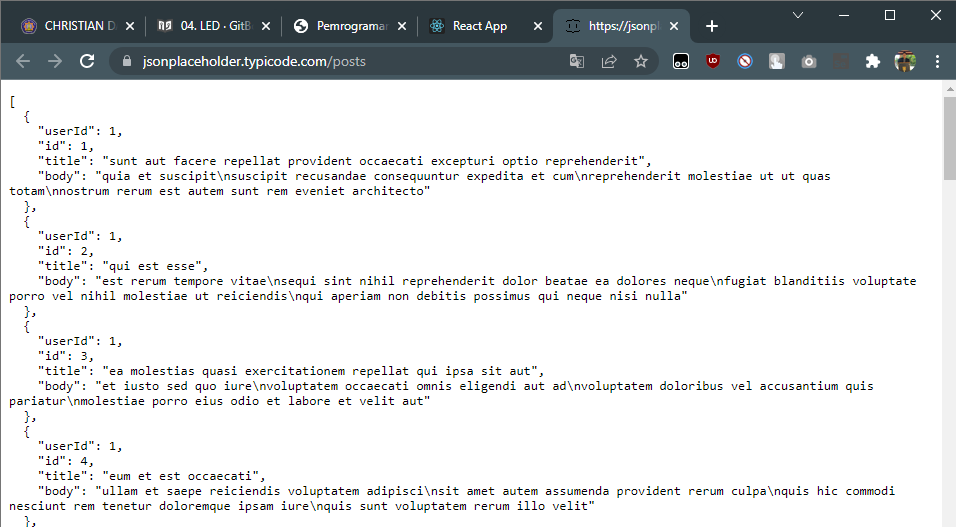
–-port 3001 .

4. Apabila pada cmd tampil seperti Gambar 2.1, maka server Fake API local kita telah siap



5. Kita cek url resource yang ada padaFake APIserver ke browser apakah bisa diakses. Ketik

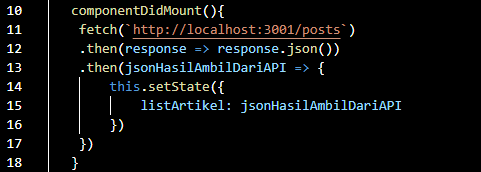
url <http://localhost:3001/posts> padabrowser.

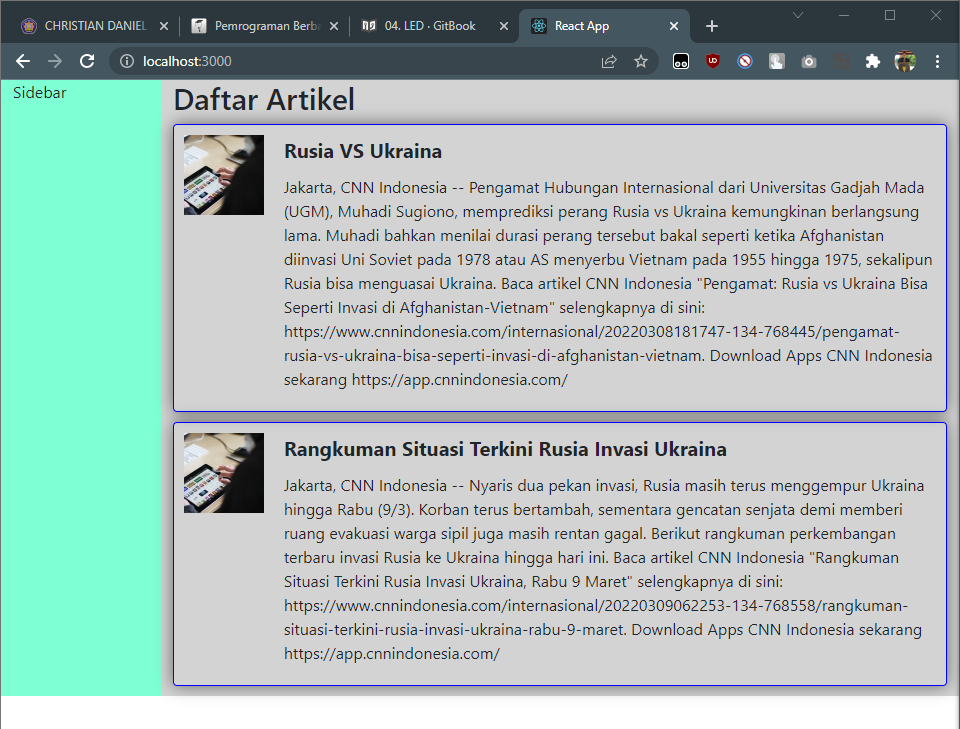


6. Untuk memastikan lagi, kita edit statefull component BlogPost (Gambar 1.18) pada baris

11. Kita ganti url API dari https://jsonplaceholder.typicode.com/posts menjadi

<http://localhost:3001/posts>



7. Simpan perubahan dan amati apa yang terjadi.

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?

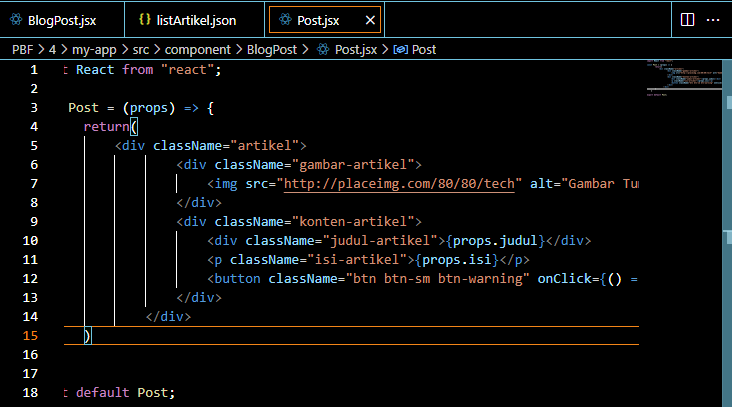
b. Bagaimana jadinya kalua kita ganti port json-server menjadi 3000?

Praktikum 3

Interaksi dengan API menggunakan method DELETE

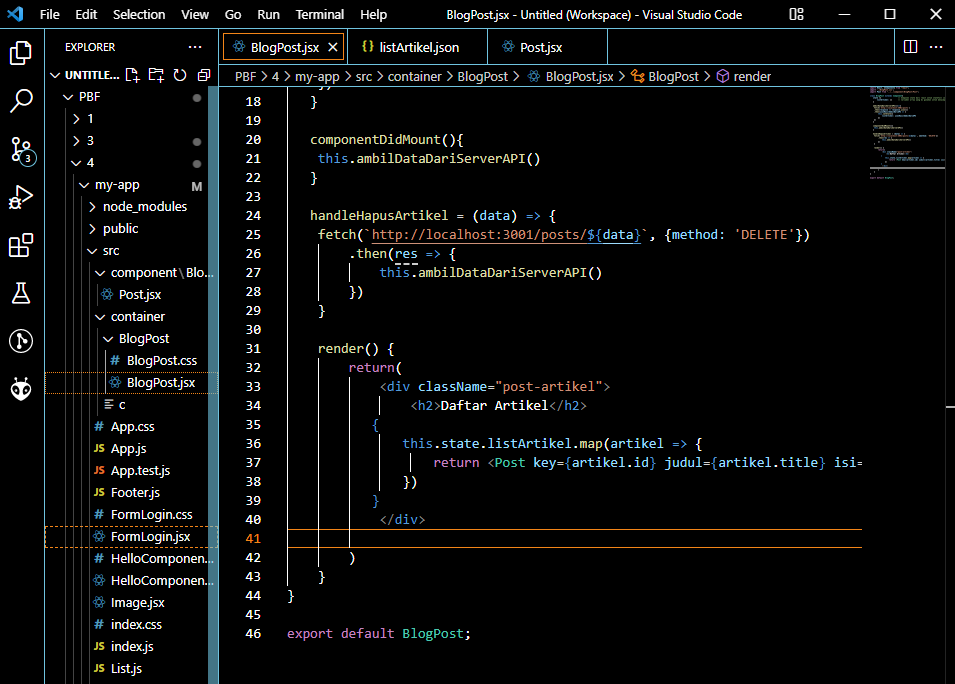
3.1 Langkah Praktikum 3

1. Buka stateless component Post. Tambahkan 1 baris kode program pada baris 10 seperti

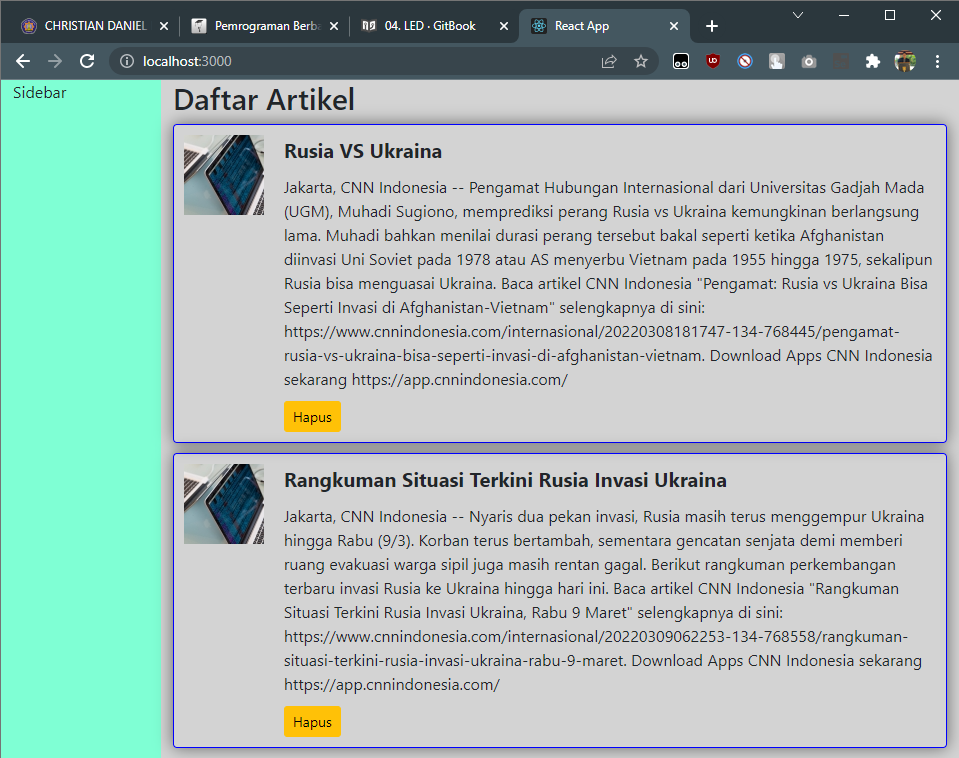
pada Gambar 3.1

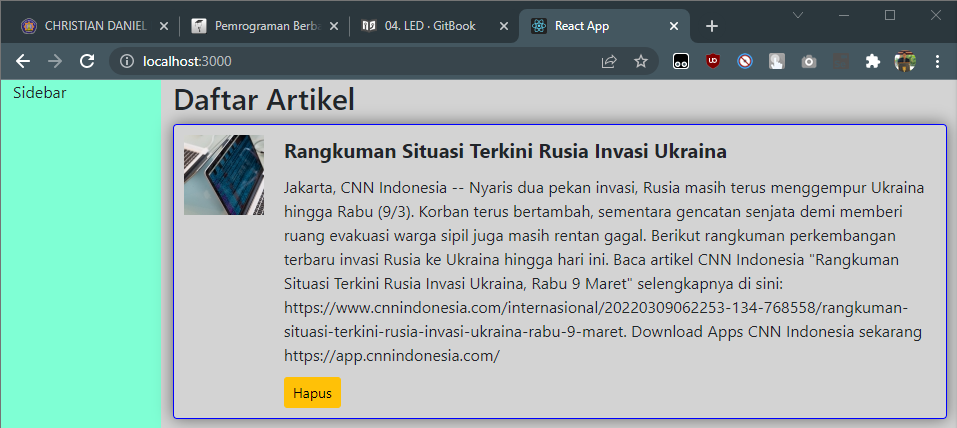
2. Kemudian pada statefull component BlogPost, modifikasi kode program sebelumnya sesuai

dengan Gambar 3.2



3. Klik tombol hapus pada list artikel di browser. Amati apa yang terjadi.





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

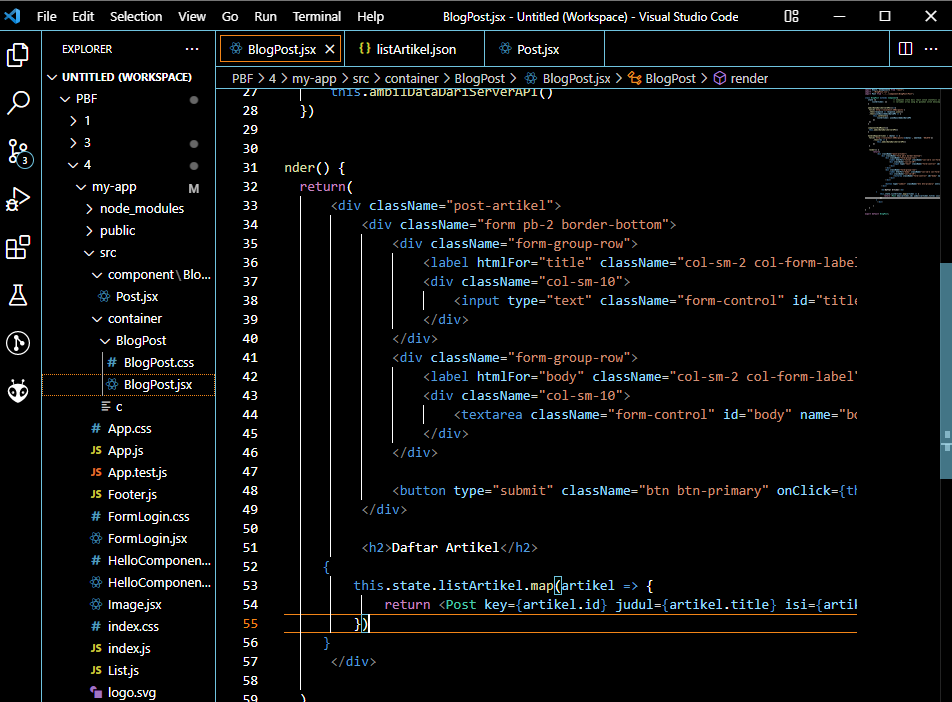
componentDidMount() pada Gambar 3.2 ?

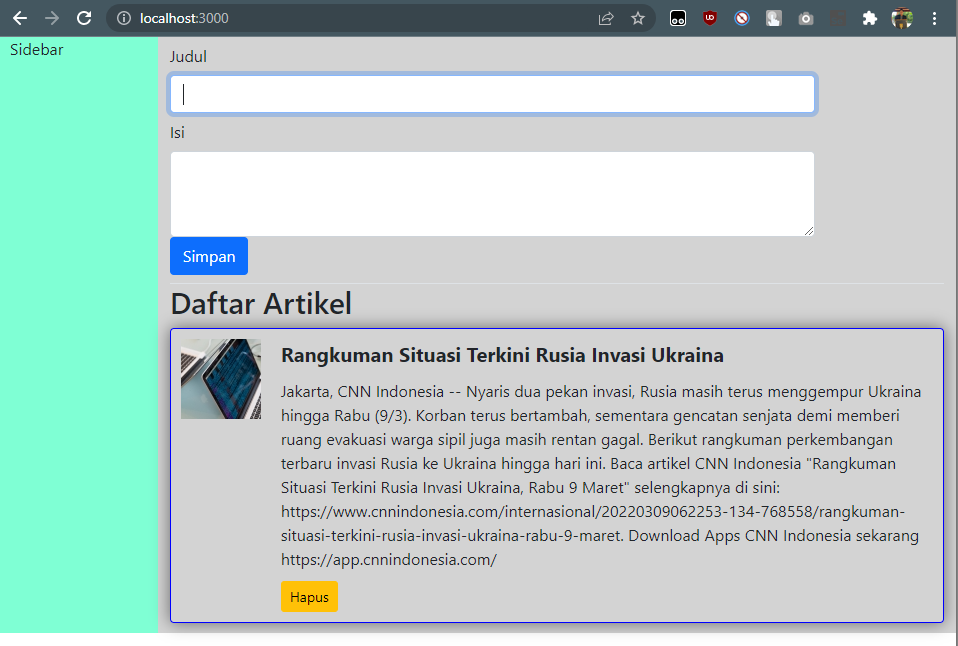
Praktikum 4

Interaksi dengan API menggunakan method POST

4.1 Langkah Praktikum 4

1. Buka statefull component BlogPost, dan modifikasi pada fungsi render() untuk

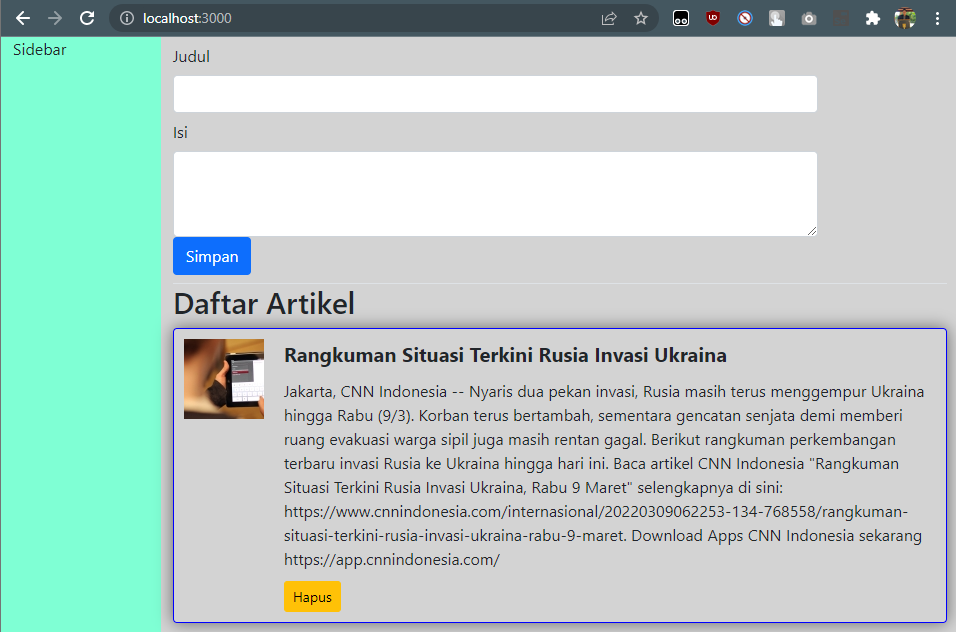
menampilkan form input artikel yang berisi judul dan isi berita. seperti pada Gambar 4.1



2. Kemudian modifikasi BlogPost untuk bagian state dan request API dari server, seperti

Gambar 4.2





3. Tambahkan untuk handle form tambah data artikel seperti Gambar 4.3

4. Langkah terakhir tambahkan fungsi untuk handle tombol simpan artikel, seperti pada

Gambar 4.4