|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Project React Resto - 5 | 92 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Bisa membuat dan mengisikan menu order detail sesuai dengan yang ada di video.

2. Bisa menginsertkan data di dalam database (apirestoran) secara manual.

3. Mengopy code dari dalam show pada file OrderController yang lalu akan diletakkan di dalam file DetailController, yang kemudian juga mengubah beberapa isinya sesuai dengan kebutuhan seperti yang ada di video.

4. Mengopykan namespace (seperti use Illuminate\Support\facades\DB;) dari satu file yang sudah dibuatkan ini sebelumnya ke dalam file yang baru saja dibuat dan membutuhkannya.

5. Menambahkan routes (pada folder routes di dalam web.php bagian api lumen) untuk order detail sesuai dengan yang ada di dalam video.

6. Titik bintang (.\*), untuk memunculkan semua isi data dari data yang sudah dipilih sebelumnya. Bila tidak manggil semuanya dituliskan satu-satu mana yang ingin ditampilkan.

7. Bisa menampilkan rincian pembelian (order detail) seperti id detail, id order, nama menu, nama pelanggan, nama kategori dan lain sebagainya di dalam postman maupun web/browser yang telah dibuat sebelumnya (localhost:3000).

8. Di dalam order detail menjoinkan (menggabungkan) dan menampilkan isi dari banyak table untuk memperjelas pesanannya. Seperti orders, menus, pelanggans, dan kategoris.

9. Select, untuk memilih mana data yang akan ditampilkan nantinya.

10. Orderby, untuk mengurutkan isi dari dalam data sesuai dengan perintah yang dijalankan, seperti status berarti akan diurutkan sesuai dengan status dari orderannya. Sedangkan untuk tglorder, berarti akan diurutkan sesuai dengan tanggal dari orderan yang ada terlebih dahulu.

11. Membuat file baru untuk detail dan nantinya diisikan sesuai dengan kebutuhan seperti yang ada di dalam video.

12. Rfc, tag untuk membuat code lebih cepat (langsung terdapat import, function, dan export di dalamnya).

13. Mengatur content.js untuk menampilkan detail.js di dalam web/browser yang sudah dibuat sebelumnya.

14. Bisa mengimportkan beberapa hal seperti yang ada di dalam video (salah satunya seperti import {link} from “..Axios/link”, dan masih banyak lagi lainnya disesuaikan dengan kebutuhan penggunannya).

15. H1, untuk menjadikan suatu tulisan menjadi berbeda ukuran dan ketebalannya dari tulisan biasanya. Untuk ukuran dan ketebalannya menyesuaikan angka dari headernya (terdapat h1 hingga h6).

16. Mengopykan const yang berisikan useState awal dan akhir dari dalam file Order.js ke dalam Detail.js.

17. Table, untuk menjadikan tulisan yang dipilih menjadi berbentuk tabel. Terdapat thead, tbody, th, td, dan tr di dalam bagiannya.

18. Map, untuk mengambil hasil dari array yang telah dibuat sebelumnya dan hasil saat ditampilkannya membuat baris baru yang isinya juga berupa array (hasil yang diambil tadi). Namun pada react-js, penggunaan dari mapnya harus diberikan key (bagian indeksnya), agar tidak error pada consolenya (inspect).

19. Bisa membenarkan error bila ada yang error sesuai dengan yang ada di dalam video.

20. Div (div saja yang tidak bawaan, seperti .row), sebuah tag yang dapat diisikan berbagai macam hal (seperti form, table, dan lain sebagainya). Digunakan agar code yang dibuat lebih rapi dan agar pada saat memberikan css (boottstrap) lebih mudah.

21. Faktur, digunakan untuk mengerti nomer orderan setiap pelanggan yang nantinya pada saat di order detail, setiap menu yang dibeli oleh pelanggan tersebut tidak terpisah dan bergabung menjadi satu. Misalkan seperti pelanggan a membeli 2 menu, maka id order/faktur kedua menu tersebut sama, tidak seperti menu yang satu bernomorkan 1 dan satunya lagi bernomorkan 2, padahal kedua menu tersebut dalam satu orderan (satu pelanggan yang membeli 2 menu).

22. Bisa mencari order detail dari tanggal pemesanannya mirip seperti yang ada pada file order (seperti yang ada di dalam video).

23. Get, untuk mendapatkan sesuatu (seperti data) dari yang sudah dipilihkan sebelumnya.

24. Mengopy form yang ada di dalam file order, yang kemudian diletakkan di dalam file order detail (detail) dan mengubahnya sesuai dengan kebutuhan.

25. Mengubah nama link admin di Side.js dengan user (Nama buttonnya juga diubah menjadi User Admin).

26. Menambahkan if di dalam Cotent.js untuk user admin (if (isi === ‘user){tampil=<User><User/>}).

27. Mengimportkan User di dalam Content.js sesuai dengan yang ada di dalam video.

28. Membuat function index di dalam LoginController.php (api lumen) sesuai dengan yang ada di dalam video.

29. Bisa membuat function register di dalam LoginController.php (api lumen) sesuai dengan yang ada di dalam video.

30. Menambahkan user admin dengan postman (admin, koki, dan kasir) yang otomatis akan tertambahkan di dalam databasenya (apirestoran).

31. Post, untuk mengirimkan suatu data (menggunakan x-www-form-urlenconded di dalam postamnnya).

32. Bisa memfilter pencarian (baik di dalam postman, maupun langsung di web/browsernya) pada saat mencari data admin (tidak tercampur dengan data pelanggan).

33. Bisa menampilkan isi data dari dalam database (apirestoran) bagian user admin (admin, koki, dan kasir) dengan format tabel.

34. Bisa membuat modal untuk menambahkan user sesuai dengan yang ada di dalam video.

35. Bisa membuat button untuk menambahkan user beserta memberikannya css (bootstrap) dengan btn btn-primary agar buttonnya menjadi berwarna biru dan di setiap ujungnya menjadi oval (tidak kotak seperti button pada awalnya).

36. Bisa membuat status dari user admin menjadi aktif (bila bernomor 1) dan menjadi banned (bila bernomor 0). Nomor dari status tersebut akan otomatis berubah databasenya (apirestoran) bila diubah di dalam web/browsernya.

37. Sama dengan tiga kali (===), untuk menjadikan suatu data menjadi sama persis dengan data yang disamakan (lebih akurat daripada sama dengan dua kali).

38. Memberikan button di dalam status user admin. Bila statusnya aktif, maka diberikan button berwarna hijau (dengan bootstrap btn btn-success), dan bila statusnya banned, maka akan diberikan button berwarna merah (dengan bootstrap btn btn-danger). Bentuk dari buttonnya juga berubah menjadi lebih oval setelah diberikan bootstrap tersebut (tidak kotak seperti button pada awalnya).

39. No++, digunakan untuk menjadikan tabel nomor berisikan nomor sesuai dengan urutan nomornya seiring ada isinya (1, 2, 3, 4, dan seterusnya).

40. OnClick, akan menjalankan/memunculkan sesuatu dari yang sudah dibuat sebelumnya (dengan code) dan akan berjalan pada saat di klik (bila tidak di klik terlebih dahulu, maka tidak akan terjalankan maupun muncul).

41. Alert, digunakan untuk memunculkan kotakan form kecil (localhost:3000 says) untuk menunjukkan sesuatu hal, misalkan seperti memunculkan tambah pada saat menekan button tambah user di bagian admin.

42. Menambahkan (mengopy) useState yang pernah digunakan di dalam file Order.js (yang ada mopennya) dan meletakkannya di dalam file User.js, serta menambahkan dan mengganti beberapa hal sesuai dengan kebutuhan (serta sesuai dengan yang ada di dalam video).

43. Mengopy form dari dalam file Kategori.js untuk digunakan di dalam file User.js, serta menghapus dan menambahkan beberapa hal sesuai dengan yang dibutuhkan untuk menambahkan user admin dengan modal.

44. Option, digunakan pada saat ada beberapa pilihan (letaknya di dalam form) yang pada saat membuat sesuatu pertama kali harus diisikan tergolong ke dalam yang mana (seperti pada saat membuat (menambahkan) user admin, yang dimana harus memilih terlebih dahulu (memilih termasuk ke dalam admin, koki, maupun kasir)). Di dalam code pilihannya dituliskan dengan select. Letaknya di bawah password pada saat ada di dalam form kotakan modalnya.

45. Format Document, untuk merapikan coding yang telah dibuat sebelumnya. Dengan cara klik kanan, kemudian pilih Format Document (pastikan prettiernya sudah berjalan terlebih dahulu).

46. ClassName, untuk membuat css. Menuliskannya ini sebelum memberikan css dari suatu tag div polos (div yang sendiri).

47. Margin left (ml), untuk memberikan jarak di sebelah kiri.

48. Col, untuk menjadikan form menjadi penuh (memenuhi form kotakan modalnya, namun penuhnya tidak keseluruhan, hanya memanjang ke samping kanan dan tetap masih ada jarak sedikit, namun jaraknya antara kotakan form email dengan password sama (rapi)).

49. Bisa menambahkan user admin di dalam web/browsernya langsung (dengan modal kotakan form), yang nantinya otomatis akan ditambahkan di dalam databasenya (apirestoran). Dan pada saat pembuatan user bisa memilih bagian menjadi yang mana (admin, koki, dan kasir). Setelah dibuat juga langsung terbentuk status aktifnya, dan bisa langsung diubah menjadi banned juga bila sudah tidak aktif. Dengan cara mengeklik buttonnya (bila pada saat berwarna hijau di klik, akan berubah menjadi berwarna merah, begitu pula sebaliknya).

50. Console.log (clg), untuk memunculkan hasil dari klik saat mencoba dan akan muncul di console bagian inspect.

51. Async, untuk menjalankan program dengan bertahap dan tidak perlu menunggu proses yang lain terlebih dahulu (tidak secara langsung).

52. Await, menunggu perintah dari sesuatu yang sudah dibuat sebelumnya (jadi akan jalan setelah dijalankan terlebih dahulu).

53. Update, untuk mengubah sesuatu dari yang awalnya bagaimana menjadi bagaimana.

54. Bisa mengupdate button status yang awalnya aktif kemudian diubah menjadi banned, dan banned kemudian diubah menjadi aktif. Otomatis juga akan langsung terupdate di dalam databasenya (apirestoran).

55. Menyalakan xampp (bagian Apache dan MySQL) untuk menyalakan api lumennya (localhost:8000).

56. Menyalakan yarn di dalam terminal visual studio code dengan mengetikkan yarn start. Bila tidak bisa berjalan, bisa juga menyalakannya lewat command prompt (cmd) dengan mengetikkan hal yang sama (yarn start). Namun sebelumnya harus diletakkan di dalam folder project react resto dibagian react-resto nya pada saat di dalam cmd nya (menyambungkan foldernya dengan menuliskan cd di depan url folder yang akan disambungkan nantinya). Setelah menyala akan menuju localhost:3000.

57. Bisa membuat data user admin pada saat dimunculkan bila sudah tidak aktif (banned) maka hasilnya akan memunculkan tulisan login gagal. Dan bila berhasil (bila user adminnya masih aktif) maka akan memunculkan tulisan login berhasil beserta data dari user admin yang dipilihnya. Menampilkannya bisa lewat postman.

58. Bisa membuat login admin sesuai dengan yang ada di video.

59. If, digunakan untuk membuat dua kondisi, bila kondisi yang pertama sudah benar, maka akan langsung dijalankan isi dari kondisi pertamanya. Namun bila salah, maka akan menjalankan isi dari kondisi keduanya.

60. Else, merupakan kondisi kedua dari if.

61. Menambahkan route untuk login di dalam App.js.

62. Bisa membuat form untuk halaman login pada saat pertama kali masuk ke dalam web/browsernya.

63. Bisa menjadikan data user admin yang statusnya aktif dapat masuk dan pada saat dicek di dalam console bagian inspectnya, akan memunculkan tulisan login berhasil beserta data dari user admin yang akan login tersebut. Dan bila statusnya banned, maka akan memunculkan tulisan login gagal saja (tidak ada data-datanya seperti pada saat login berhasil). (sama seperti pada saat mencoba di postman). Memunculkan login gagal tersebut juga akan muncul bila email dan password yang di masukkan (login) berbeda dan tidak ada/terdaftar dalam database (apirestoran) maupun web/browsernya.

64. Pada saat memberikan await di dalam suatu function, maka function tersebut harus diberikan async agar dapat berjalan.

65. Bisa mengganti menambahkan kategori dengan modal (dengan kotakan form modal seperti user admin). Tidak seperti awalnya yang ada form langsung di halaman tampilan depannya.

66. Margin bottom (mb), untuk memberikan jaraka pada bagian bawah.

**Saya Belum Mengerti**

1. Belum bisa menjadikan ubah dibagian kategori bila di klik langsung masuk ke dalam kotakan form modalnya (seperti form awal yang sebelum diubah menggunakan modal). Bisa tetap terubah nama kategorinya, namun tidak otomatis (nama kategorinya tidak terdapat dalam formnya pada saat di klik tombol ubahnya), dengan cara mengklik tombol ubahnya, kemudian mengklik tombol tambah data kategori yang kemudian akan memunculkan kotakan form modal, namun kotakan formnya kosong dan tidak terisi apa-apa. Kemudian dituliskan nama kategori baru yang akan menggantikan nama sebelumnya disitu (kotakan form setelah di klik tadi) dan kemudian klik submit. Nama kategorinya akan berubah sesuai dengan apa yang telah diubah setelah mengisikan kotakan form modal kedua kalinya (hasilnya akan mengubah nama kategorinya, bukan menambahkan kategori baru. Bila setelah mengubah dan kemudian menambahkan kategori lagi di dalam kotakan form modalnya, maka hasilnya akan menambahkan kategori baru seperti fungsi awalnya).

2.

3.