|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Laravel Lumen vid 01 s/d 11 | 90 |

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

1. Saya bisa mendownload composer melalui getcomposer.org dan mengaktifkannya. Cara untuk mendownloadnya melalui ink di getcomposer lalu download. Lalu, untuk mengaktifkannya melalui control panel. Lalu, ketik cd lalu lokasi penyimpananan file untuk file instalasinya. lalu, ketik composer jika sudah muncul tulisan composer maka sudah bisa dijalankan untuk mendownload lumennya.

2. Lumen adalah mikro framework dari laravel.

3. Saya bisa mendownload lumen melalui composer tadi dengan cara didalam control panel dan ketik composer create-project --prefer-dist laravel/lumen lalu nama file untuk menyimpan file instalasinya.

4. Menyalakan lumennya, melalui terminal di vscode lalu ketik php -S localhost:8000 -t public. untuk melihat versi lumennya, bisa melalui link url http://localhost:8000/.

5. Menjalankan lumennya, didownload menggunakan composer require flipbox/lumen-generator melalui terminal yang di dalam vscode.

6. Konfigurasi lumennya, menambahkan $app->register(Flipbox\LumenGenerator\LumenGeneratorServiceProvider::class) di dalam folder bootstrap lalu file app.php bawaan dari lumennya.

7. Menset lumennya, menggunakan php artisan key:generate dalam terminal di vscode.

8. Membuat controller baru menggunakan artisan, melalui terminal lalu ketik php artisan make:controller kemudian nama controllernya. Maka, ototmatis file controllernya akan dibuatkan oleh lumen. Untuk lebih mudah, bisa menggunakan php artisan make:controller kemudian nama controllernya lalu --resource karena akan dibuatkan file cotroller tetapi sudah dibuatkan function-functionnya langsung oleh lumennnya.

9. Dapat membuat model di dalam lumen menggunakan artisan dengan cara mengetik php artisan make:model Kategori -mcfs --resource di dalam terminal. mcfs ini adalah m untuk migration, c untuk controller, f untuk function dan s untuk seeder.

10. Postman adalah platform kolaborasi untuk mengembangkan API.

11. Saya bisa mendownload dan menggunakan postman.

12. Menggunakan router dengan menambahkan $route->get('(parameter yang memanggil)', '(yang dipanggil)'). Untuk memanggil controllernya harus menuliskannya benar benar sama dengan filenya kemudian kita tambahkan @index. Bisa juga menampilkannya menggunakan postman dengan cara, pada functionnya menggunakan return response()->json("(isi yang ingin ditampilkan)"). Kemudian, salin url untuk menampilkannya yang di dalam browser. Lalu, urlnya di letakkan pada bagian get url di postman. Maka, otomatis akan tertampil hasilnya.

13. Menghapus elemen menggunaan router adalah dengan menambahkan $router->delete('(parameter yang memanggil)/{id}', 'parameter yang dipaggil kemudian @destroy(functionnya)'). Kemudian, pada functionnya tambahkan pada parameternya dengan Kategori $kategori. Lalu, isinya return response()->json("Ini delete $kategori"). Kemudian, untuk menampilkannya pada postman menggunakan metode destroy lalu klik send.

14. Mengubah elemen menggunakan router adalah dengan menggunakan $router->put('(parameter yang memanggil)/{id}', 'parameter yang dipaggil kemudian @update(functionnya)'). Kemudian, pada functionnya tambahkan pada parameternya dengan Request $request, Kategori $kategori. Lalu, isinyareturn response()->json("Ini Update"). Kemudian, untuk menampilkannya pada postman menggunakan metode put lalu klik send.

15. Menambahkan elemen denngan menggunakan route $router->put('(parameter yang memanggil)/{id}', 'parameter yang dipaggil kemudian @create(functionnya)'). Kemudian, pada functionnya tambahkan pada parameternya dengan Request $request. Lalu, isinya return response()->json($request). Kemudian, untuk menampilkannya pada postman, harus ditambahkan dulu key, dan valuenya pada kolom yang tersedia pada postman. Lalu, menggunakan metode post lalu klik send.

16. Menampilkan salah satu elemen dengan menggunakan $router->get('(parameter yang memanggil)/{id}', 'parameter yang dipaggil kemudian @show(functionnya)'). Kemudian, pada functionnya tambahkan pada parameternya dengan Kategori $kategori. Lalu, isinya return response()->json("Menampilkan satu data"). Kemudian, untuk menampilkannya pada postman, menggunakan metode get. Lalu, tambahkan idnya, Kemudian, klik send.

17. Mengelompokkan routes menjadi satu agar lebih mudah dengan cara menggunakan perintah $router->group(['prefix' => 'dasar pemanggilan dari setiap url'], function() use ($router) {Router yang ingin digabungkan} ). Pada parameter yang dipanggil menggunakan array yaitu seperti ['uses' => 'controller dan function yang digunakan'].

18. Memanggil salah satu anggota dari group routes dengan cara mengetik dasar pemanggilannya terlebih dahulu misal api/kategori.

19. Migrate digunakan untuk memindahkan tabel yang dibuat dalam file ke database. Caranya, membuat file baru di dalam folder migrations bawaan dari lumen. Kemudian, didalamnya terdapat function create gunakan perintah $table->(jenis data tabelnya)('nama tabel yang ingin ditambahkan'). Untuk memindahkannya menggunakan perintah php artisan migrate pada bagian terminal di dalam vscode. Maka, otomatis tabel beserta isi kolom yang ditambahkan tadi akan tertambah otomatis di dalam database. Jika terdapat kolom yang ingin ditambahkan bisa menggunakan perintah php artisan migrate:rollback pada bagian terminal di dalam vscode. Lalu, otomatis tabel tersebut akan otomatis ditarik dari database. Untuk menyimpan kolom yang barusan ditambahkan tadi bisa menggunakan lagi php artisan migrate. Otomatis kolom tabel yang di tambahkan tadi akan tersimpan dalam database.

20. Seeder digunakan untuk mengisi data tabel ke dalam database melalui file yang sudah disediakan oleh lumen. Caranya, menambahkan terlebih dahulu file yang digunakan untuk mengisi data tabel dengan menggunakan perintah $this->call(file yang digunakan). Untuk menambahkan datanya menggunakan array asosiatif dengan mengisikan sesuai dengan nama kolom yang ingin diisi dan data yang ingin diisikan. Misal, $data = ['(nama kolom tabelnya)', 'data yang ingin diisikan']. Lalu, untuk menampung data array yang diisikan dengan menggunakan perintah (Nama class)::create(nama functionnya)($data(nama arraynya)). Untuk menyimpan data tersebut ke dalam database menggunakan perintah php artisan db:seed di dalam terminal dalam database. Maka, otomatis data akan tersimpan kedalam database.

21. Faker adalah sebuah data palsu yang biasanya digunakan untuk melihat hasil dari web yang dibuat atau untuk sekedar coba-coba saja. Caranya di dalam file seeder tadi di deklarasikan erlebih dahulu data dari fakernya dan mengubahnya dengan menggunakan perintah use Faker\Factory as Faker. Untuk membuat data faker diawali dengan variable faker yang berisi classnya dan functionnya yaitu $faker=Faker::create(). Lalu, untuk mengisikan data fakernya menggunakan array dan pada bagian yang untuk isi tabelnya menggunakan $faker->name(nama provider fakernya). Untuk menyimpan data tersebut ke dalam database menggunakan perintah php artisan db:seed pada terminal di vscode. Maka, otomatis data akan tampil. Jika ingin menampilkan data dari faker banyak bisa menggunakan looping for.

22. Untuk melihat daftar url routenya bisa menggunakan perintah php artisan route:list dalam terminal dalam vscode. Maka, otomatis akan tampil semua daftar routes yang sudah dibuat.

23. Menampilkan semua data yang ada di dalam suatu tabel dalam database menggunakan variabel dalam controller yang digunakan dengan menggunakan model dan function misal $data(variabel penyimpannya) = Kategori(nama modelnya)::all()(functionnya). Untuk menyimpan datanya menggunakan return response()->json($data). Untuk memanggilnya di dalam postman menggunakan method get. Lalu, klik send. Maka, otomatis semua data akan tertampil.

24. Menampilkan salah satu data dalam tabel dalam database menggunakaan variabel dalam controller yang digunakan dengan menggunakan model dan function misal $data(variabel penyimpannya) = Kategori(nama modelnya)::where(function untuk mencari data)('(id datanya)', $id(variabel id))->get()(function untuk mengambil data). Untuk menyimpan datanya menggunakan return response()->json($data). Untuk memanggilnya di dalam postman menggunakan method get. Lalu, klik send. Maka, otomatis data dalam satu id tersebut akan tampil.

25. Memasukkan isi tabel/value tabel yang ada di dalam database dengan menggunaan controller pada function create. Menggunakan perintah Kategori(nama modelnya)::create(function yang digunakan untuk membut data baru)($request(variabel request untuk melakukan permintaan)->all(function yang digunakan untuk array asosiatif)). Kemudian, melakukan return response()->json($request(untuk menjalankan requestnya)). Pada bagian modelnya melakukan pengisian pada kolom tabel pada database dengan menggunakan protected $fillable = [(kolom tabel yang ingin diisi)]. Kemudian, pada postman untuk menambahkannya pada database tambahkan pada kolom key dan value yang tersedia pada postman kemudian isi key sesuai dengan nama kolom tabel yang ada di dalam database kemudian valuenya isi yang ingin di tambahkan ke dalam kolom tersebut. Kemudian, pilih method post. Lalu, klik send. Maka, ortomatis data akan tertambah pada database.

26. Menghapus salah satu isi tabel kolom di dalam tabel dengan menggunakan controller function destroy. Menggunakan perintah $data(nama variabel penyimpan)=Kategori(nama classnya)::where(function untuk menunjukkan idnya)('idkategori(kolom id tabelnya)', $id(variabel penyimpan idnya))->delete()(function untuk menghapus). Kemudian, untuk melakukan response atau tanda jika programnya sudah berhasil gunakan return response()->json("Data sudah dihapus"). Lalu, untuk melakukan proses hapusnya melalui postman dengan menggunakan method delete. Lalu, klik send.

27. Memperbarui isi data/Mengubah isi kolom tabel dalam database dengan cara menggunakan controller function update. Menggunakan perintah Kategori(nama classnya)::where(function yang menunnjukkan lokasi data)('idkategori(tabel yang berisikan id)', $id(variabel penampun id))->update(function untum mengubah/memperbarui data)($request(variabel untuk melakukan request)->all(function untuk mengambil data secara array)). Kemudian, untuk melakukan response atau tanda jika programnya sudah berhasil gunakan return response()->json("Data sudah diupdate"). Pada postman menggunakan x-www-form-urlencoded karena untuk mengirim teks pada database secara sederhana/data ASCII. Lalu, Lalu, untuk melakukan proses pengubahannya melalui postman dengan menggunakan method put. Lalu, klik send.

**Saya Belum Mengerti**

1.

2.

3.