|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Rest API (LUMEN) Semester-4 Minggu ke-6 Video 01-11 | 95 |

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

1. Melakukan Download dan Instalasi composer dari website getcomposor.org dengan memilih menu download lalu mengikuti langkah-langkah juga menyesuaikan/mencocokkan php yang sudah diinstall di dalam penyimpanan computer.

2. Mengecek composer apakah sudah terinstall dan bisa dijalankan menggunakan cmd, caranya copy letak htdocs mulai dari folder c nya lalu di dalam cmd nya ketik cd dan paste letak htdocs nya lalu enter. Jika sudah muncul letak htdocs di bawahnya ketik composer, apabila composer bisa dijalankan maka dia akan muncul tulisan COMPOSER yang besar dan keterangan yang sangat banyak dibawahnya.

3. Lumen adalah micro-framework dari laravel yang digunakan untuk pembuatan API (Application Programming Interface).

4. Karena composor sudah diinstall, selanjutnya menginstall lumen dari website lumen.laravel.com dengan mengcopy composer create-project --prefer-dist laravel/lumen dan diletakkan di dalam cmd nya lalu memberi nama untuk folder nya misalkan api-lumen. Jika sudah klik enter dan tunggu hingga proses downloading dan installing selesai sampai muncul tempat diletakkannya folder api-lumen yang sudah dipasang tadi.

5. Untuk menjalankan lumen, caranya seperti menjalankan react-js yaitu melalui terminal yang ada di vs code, terminal->new terminal-> lalu copy php -S localhost:8000 -t public dari website nya lumen.laravel.com dan diletakkan di new terminal tadi, setelah enter akan muncul keterangan yang ada tulisan localhost:8000 berarti lumen sudah berjalan. Untuk itu bisa dicek di browser nya dengan mengetikkan localhost:8000 yang apabila berhasil dia akan muncul tulisan Lumen(versinya)(Laravel Components^8.0) karena laptop saya versi 32 bit dan hanya bisa diinstall php 7.2 maka versi Lumen saya (6.3.3) dan Laravel nya 6.0

6. Karena di dalam php artisan list bagian make nya hanya ada 2 untuk itu harus melakukan installasi lumen generator dengan cara mengcopy composer require flipbox/lumen-generator dan dipaste ke dalam new terminal lalu tekan enter, selanjutnya dia akan menjalankan installasi untuk lumen generatornya. Jika sudah, selanjutnya copas $app->register(Flipbox\LumenGenerator\LumenGeneratorServiceProvider::class); ke dalam file app.php yang ada di dalam folder bootstrap, dengan begitu ketika dicek di php artisan list nya bagian menunya sudah ada banyak yang sebelumnya hanya dua, setelah install lumen generator jadi banyak.

7. Di dalam APP\_KEY yang ada di .env masih kosong tidak memiliki isi, untuk mengisi key itu caranya dengan mengetikkan php artisan key:generate ke dalam new terminal, jika sudah enter maka APP\_KEY yang ada di .env sudah terisi dengan code random.

8. Membuat sebuah controller melalui new terminal dengan cara mengetikkan php artisan make:controller lalu diberi nama controller yang akan dibuat misalkan KategoriController lalu enter apabila berhasil maka akan keluar tulisan Controller created successfully. Jika berhasil maka controller akan diletakan di dalam folder Controller, letak folder Controller ini ada di dalam folder App->Http->Controller.

9. Memasang paket lengkap dari php artisan dengan cara mengetikkan di dalam new terminalnya seperti ini php artisan make:model namamodel misal Kategori lalu –mcfs –resource dan enter, jika sudah enter maka dia akan membuat sebuah model, factory, migration, seeder, dan controller.

10. Memasang Postman ke dalam chrome sebagai tempat untuk menampilkan percobaan dari routes nya nanti, jadi url yang akan ditampilkan bisa diletakkan di postman dan postman akan menampilkan hasilnya dibawah.

10. Routing ini digunakan untuk memanggil controller dan function yang ada di dalam controller. Misalkan $routes->get(‘kategori’,’KategoriController@index’); berarti routes ini menggunakan perintah get apabila di dalam url localhost:8000 ditambahi /kategori maka yang tampil adalah isi dari function index (ditandai dengan @ yang berarti itu functionnya) yang ada di dalam KategoriController. Jadi parameter yang ada di dalam routes selalu berisi 2 dan ditandai dengan adanya koma (,) untuk parameter yang pertama atau depan sebagai pemanggil yang diletakkan di url dan parameter yang kedua sebagai yang dipanggil oleh parameter pertama.

11. Routes get digunakan untuk mengambil atau menampilkan controller, routes delete digunakan untuk menghapus dengan menggunakan id dan menjalankan function destroy, routes put digunakan untuk update dan memerlukan id seperti delete, dan routes post digunakan untuk membuat sebuah data yang bisa diatur dari postman nya.

12. Routes Group adalah routes yang digunakan untuk menyimpan beberapa routes menjadi satu dengan mengatur awalan yang akan digunakan untuk routes nya, misalkan routes groupnya $router->group(['prefix' => 'api'], function () use ($router) berarti routes ini adalah routes group yang url nya diawali dengan api maka dia akan menjalankan isi dari routes group ini yaitu routes biasa dan yang memiliki url seperti yang diketikkan user. Untuk routes biasa yang ada di dalam routes group parameter yang keduanya ditambahkan [] dan ‘user’ => sebagai awalan di dalam [] nya.

13. Melakukan installasi Extension Laravel Snippets untuk memudahkan saat menjalankan laravelnya.

14. Menambahkan sebuah tabel yang ada di dalam database menggunakan migrate, caranya di dalam .env diketikkan dulu nama database nya, usernamnya, dan password nya. jika sudah, selanjutnya masuk ke dalam file yang ada di folder migrations untuk membuat tabel seperti ini

public function up()

    {

        Schema::create('kategoris', function (Blueprint $table) {

            $table->increments('idkategori');

            $table->string('kategori');

            $table->timestamps();

        });

    }

Yang berarti tabelnya akan diisi dengan kolom yang pertama yaitu increments yang bernama idkategori, jadi idkategori ini nanti akan auto increments biasanya sebagai id, lalu ada kolom yang kedua diisi dengan string bernama kategori. Jika sudah, untuk menjalankan agar masuk ke dalam database nya dengan cara mengetikkan php artisan migrate di dalam new terminalnya. Jika berhasil dia akan memberikan keterangan succesfuly migrating dan migrated nya.

15. Menambahkan sebuah kolom ke dalam tabel yang telah di migrate, misalkan tabel kategori sudah di migrate dan kolom sebelumnya sudah masuk ke dalam database sedangkan masih ada kolom yang ternyata tertinggal dan belum dimasukkan misalkan ini $table->string(‘keterangan’); untuk itu tabel yang dimigrate tadi dikembalikan ke dalam kondisi belum dimigrate dengan cara mengetikkan php artisan migrate:rollback ke dalam new terminalnya, setelah itu mengisi file migrations nya dengan kolom yang akan ditambahkan. Jika sudah baru mengetikkan lagi php artisan migrate untuk tabel yang telah diperbarui tadi.

16. Seeder berfungsi untuk mengisi kolom yang sudah dibuat di tabel database nya, jika migrate tadi berfungsi untuk menambah kolom maka seeder berfunsgi untuk menambah isi dari kolomnya atau barisnya. Caranya dengan mengatur seeder apa saja yang akan digunakan ke dalam DatabaseSeeder nya yaitu dengan mengetikkan $this->call(KategoriSeeder::class); di dalam function run nya yang berarti databaseseeder ini akan memanggil kategoriseeder yang merupakan seeder dari tabel Kategori. Jika sudah dipanggil selanjutnya di dalam file KategoriSeeder nya sendiri diisi dengan

$data = [

            'kategori' => 'Gorengan',

            'keterangan' => 'Goreng Pisang'

        ];

        Kategori::create($data);

Di dalam function run nya, $data ini merupakan sebuah array yang menyimpan data untuk ditambahkan ke dalam kolom sesuai namanya, misalkan ‘kategori’ => ‘Gorengan’ berarti kolom yang bernama kategori diisi dengan ‘Gorengan’ begitu juga yang keterangan. Setelah itu $data ini akan dimasukkan ke dalam function create untuk ditambahkan ke dalam tabel Kategori yang telah disambungkan dengan models nya.

17. Selain Seeder, ada juga Faker. Faker ini adalah data palsu yang diberikan secara random oleh lumennya. Untuk menggunakan Faker di dalam file seeder nya harus ditambahkan use Faker\Factory dulu jika terlalu panjang bisa ditambahkan as Faker jadi yang digunakan nanti cukup kata Faker nya saja. Setelah itu didalam function run nya ditambahkan $faker = Faker::create() lalu yang $data tadi isinya diubah menjadi ‘kategori’ => $faker->name yang berarti kolom kategori akan diisi dengan data nama palsu yang random atau jika $faker->text akan diisi dengan data teks palsu yang random, begitu juga dengan faker lainnya. Bedanya dengan seeder biasa adalah jika seeder biasa untuk data yang akan diisi ke dalam kolom sesuai dengan data yang dimasukkan user, sedangkan faker adalah data yang diisi ke dalam kolom secara random sesuai dengan kategori yang dipilih (name, text, address, dll)

18. Jika data yang digunakan untuk dimasukkan ke dalam kolom banyak, maka $data beserta Kategori::create($data); nya bisa dimasukkan ke dalam for sebagai looping $data nya sebanyak yang dimau. Misalkan datanya menggunakan Faker dengan jumlah yang banyak, maka tinggal dimasukkan ke dalam for (looping) dan hasilnya data faker menjadi banyak dan isiannya berbeda-beda karena random.

19. Menampilkan data dari tabel yang ada di database ke dalam postman melalui controllernya. Karena routes memanggil controllernya, jadi code untuk menampilkan data database nya ada di controller. Di dalam controller pastikan sudah menggunakan models dari tabel database yang akan dipilih, dengan adanya use App\Models dst. Lalu pada bagian function index diisi dengan $data = Kategori::all(); yang berarti $data ini menyimpan semua data yang diambil dari model Kategori. Lalu direturn ke dalam json yang nantinya bisa dijalankan dipostman. Jika itu tadi cara untuk menampilkan semua, maka apabila hanya menampilkan 1 data saja caranya dengan mengetikkan $data = Kategori::where(‘idkategori’ , $id)->get yang berarti di dalam functionnya sudah mendapat kiriman $id dari yang dipilih user, apabila $id sama dengan idkategori yang ada di Kategori sama, maka data itu yang akan ditampilkan.

20. Menginsert data ke dalam kolom tabel database, yang pertama yaitu memberi protected $fillable = [‘kolom’,’kolom’,’kolom’]; di dalam class models nya. $fillable ini berfungsi untuk memilih kolom apa saja yang dapat diisi atau diinsert dengan data baru. Setelah itu di dalam function create yang ada di controller nya parameternya diisi dengan Request $request yang berarti nanti datanya akan diisi sesuai dengan data yang dimasukkan oleh requestan user nya. Di dalam function create nya diisi dengan

Kategori::create($request->all());

return response()->json('Data sudah dimasukkan');

yang berarti models kategori akan dibuatkan data baru dari semua request user nya, lalu apabila insert data nya berhasil akan muncul tulisan Data sudah dimasukkan. Letak untuk mengisi data nya ada di postman, caranya dengan mengetikkan url nya dulu sesuai dengan yang ada di routes lalu pilih menu body untuk diisi key dan value sesuai dengan nama kolom dari tabel database dan data apa yang akan dimasukkan ke dalamnya, setelah itu send dan data akan terkirim ke dalam database nya.

21. Menghapus data yang ada di dalam tabel database dengan mengambil id nya. Yang pertama yaitu mengatur routes yang ada di dalam web.php nya agar url yang digunakan untuk mendelete berjalan. Karena routes nya diarahkan ke dalam controller bagian function destroy, parameter function destroy nya diisi dengan $id yang berfungsi untuk menyimpan data id yang mana yang dipilih user untuk dihapus. Setelah itu isi dari functionnya seperti ini

Kategori::where('idkategori', $id)->delete();

return response()->json("Data sudah Dihapus");

Kategori adalah nama models milik tabel database yang digunakan lalu where idkategori dengan $id yang berarti $id akan dicocokkan dengan idkategori yang ada di dalam tabel Kategori, apabila ada yang sama idkategori nya dengtan $id nya maka data itu yang akan dihapus. Jika proses hapus berhasil maka dia akan menampilkan respon bertuliskan “Data sudah Dihapus”.

22. Mengupdate data yang ada di dalam tabel database dengan mengambil id nya. Yang pertama yaitu mengatur routes yang ada di dalam web.php nya agar url yang digunakan untuk mengupdate data berjalan. Karena routes nya diarahkan ke dalam controller bagian function update, parameter function update nya diisi dengan (Request $request, $id) yang berfungsi untuk menyimpan data yang dimasukkan oleh user di $request nya dan $id nya untuk mengambil id dari data yang dipilih user untuk diupdate. Setelah itu isi functionnya seperti ini

Kategori::where('idkategori', $id)->update($request->all());

return response()->json("Data sudah Di-Update");

Kategori adalah nama models milik tabel database yang digunakan lalu where idkategori dengan $id yang berarti $id akan dicocokkan dengan idkategori yang ada di dalam tabel Kategori, apabila ada yang sama idkategori nya dengan $id nya maka data itu yang akan diupdate. Setelah itu, cara memasukkan data yang akan diupdate dengan memasukkan url nya lalu memilih menu Body dan opsi x-www-form-urlencoded baru setelah itu mengisi key dan value nya, key adalah tempat mengisi nama kolom yang sesuai dengan tabel yang akan diupdate sedangkan value adalah tempat mengisi data yang akan dimasukkan ke dalam kolomnya, jika sudah klik send dan proses sending akan berjalan. Apabila proses update berhasil, maka akan muncul tulisan “Data sudah Di-Update”.

**Saya Belum Mengerti**

1.