|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran Lumen Video ke 1 -10  (Tugas semester 4 #6) | 95 |

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

1. Melakukan instalasi composer di dalam website composer.org, setelah melakukan proses instalasi, kita cek composernya ke dalam cmd, dengan mengetikkan cd C:\xampp\htdocs, lalu enter, setelah itu ketik composer dan enter. Apabila terdapat tulisan composer berukuran besar maka composer sudah berjalan dengan baik.
2. Lumen adalah micro framework PHP dari laravel, yang digunakan untuk membuat projek yang skala nya lebih kecil agar lebih ringan.
3. Lumen juga digunakan untuk pembuatan API (Application Programming Interface).
4. Kecepatan yang dimiliki oleh lumen lebih cepat dibandingkan dengan framework slim dan silex. Lumen dapat menangani 1900 permintaan setiap detik dibandingkan slim yang mampu menangani 1800 permintaan dan silex yang mampu menangani 1000 permintaan.
5. Melakukan instalasi lumen dengan mengetikkan code dibawah ini di dalam composer, setelah mengetikkan kode tersebut, kita harus menunggu beberapa waktu untuk melakukan instalasi composer create-project --prefer-dist laravel/lumen api-lumen.
6. Karena saya gagal menggunakan instalasi lumen menggunakan cmd, maka untuk instalasi lumen, saya menggunakan instalasi manual, yaitu dengan mengcopy file lumen yang sudah didownload ke dalam file htdocs dan menyalakannya dengan membuka terminal pada vscode, setelah itu, menyalakan lumen pada browser
7. Mengecek hal – hal yang dapat dilakukan oleh artisan dengan menuliskan, code : php artisan list. Setelah itu akan muncul list – list yang dapat dilakukan oleh php artisan.
8. Karena yang dapat dilakukan oleh artisan di dalam lumen adalah terbatas, maka untuk menambahkan lebih banyak hal yang dapat dilakukan kita menggunakan lumen generator. Yaitu dengan cara, search lumen generator > web flipbox github > composer require flipbox/lumen-generator.
9. Pada waktu instalasi terdapat error, akhirnya untuk instalasi lumen generator menggunakan instalasi manual, yaitu dengan mengcopy file lumen yang sudah didownload ke dalam file htdocs dan menyalakannya dengan membuka terminal pada vscode, setelah itu, menyalakan lumen pada browser.
10. Menambahkan $app->register(Flipbox\LumenGenerator\LumenGeneratorServiceProvider::class); ke dalam file app
11. Menambahkan key dengan mengetikkan, key:generate Set the application key, di dalam terminal.
12. Php artisan adalah perintah command line yang digunakan untuk melakukan aktivitas – aktivitas pada pengembangan web. Salah satu kemudahan yang diberikan oleh laravel dan lumen adalah kemudahan dengan menggunakan Php artisan.
13. Make:controller adalah perintah artisan yang digunakan untuk membuat controller, dimana ketika perintah diketikkan secara otomatis akan membuat controller baru di dalam folder controllers.

Cth :

// make:controller

PS C:\xampp\htdocs\api\_lumen> php artisan make:controller EditController

Controller created successfully.

Artinya, terdapat perintah make:controller yang digunakan untuk membuat controller dengan nama EditController. Controller berhasil dibuat ketika terdapat tulisan Controller created successfully.

//make:controller --resource

PS C:\xampp\htdocs\api\_lumen> php artisan make:controller EditController –resource

Controller created successfully.

Artinya, terdapat perintah make:controller yang digunakan untuk membuat controller dengan nama EditController, karena terdapat –resource, maka di dalam controller yang dibuat secara otomatis berisi function function yang sudah disediakan oleh artisan. Controller berhasil dibuat ketika terdapat tulisan Controller created successfully.

1. Make:model adalah perintah artisan yang digunakan untuk membuat model, dimana ketika perintah diketikkan, maka secara otomatis akan menambahkan model baru di dalam folder Models.

Cth :

//make:model

PS C:\xampp\htdocs\api\_lumen> php artisan make:model Kategori

Model created successfully.

Artinya, terdapat perintah make:model yang digunakan untuk membuat model dengan nama Kategori. Model berhasil dibuat ketika terdapat tulisam Model created successfully.

//make:model –mcfs (model, controller, factory, seeder)

PS C:\xampp\htdocs\api\_lumen> php artisan make:model Kategori -mcfs --resource

Model created successfully.

Factory created successfully.

Created Migration: 2021\_02\_13\_095108\_create\_kategoris\_table

Seeder created successfully.

Controller created successfully.

Artinya, terdapat perintah make:model yang digunakan untuk membuat model dengan nama Kategori, tetapi dengan sekali perintah, kita juga bisa membuat file model, file controller, file factory, dan file seeder dengan menggunakan -mcfs.

1. Melakukan instalasi ekstensi postman di dalam chrome.
2. Get() adalah method yang digunakan untuk menampilkan data pada route. Parameter pada get berisi 2 parameter, yaitu bagian kiri adalah yang memanggil dan bagian kanan adalah yang dipanggil. Dan untuk memanggil function harus menggunakan tanda @.
3. Melakukan pengaturan route di dalam folder routes file web.php

Cth :

$router->get('kategori', 'KategoriController@index');

Artinya, terdapat fungsi get untuk pengaturan route dengan memanggil kategori dan file yang dipanggil adalah function index yang ada di dalam file KategoriController. Jadi ketika url kategori diketikkan, maka akan menampilkan isi dari file index pada file KategoriController.

1. Menggunakan postman untuk melihat hasil dari controller. Untuk menggunakan postman, maka nilai di dalam function harus dikeluarkan dan berbentuk json.

Cth :

  public function index()

    {

*return* response()->json("ini adalah Index");

    }

Artinya, di dalam function index terdapat return yang digunakan untuk mengeluarkan nilai pada function tersebut, lalu function response digunakan untuk mengirim data yang berbentuk json yang di dalamnya berisi tulisan ini adalah index. Setelah itu, copy url yang menampilkan function index pada file KategoriController dan klik send, maka postman akan menampilkan hasil dari function index yaitu teks yang dikirimkan dalam bentuk json.

1. Delete() adalah perintah yang digunakan untuk delete pada route, tetapi untuk menggunakannya harus diberi /variabel {id} agar function delete dapat berjalan.

Cth :

$router->delete('kategori/{id}', 'KategoriController@destroy');

Artinya, fungsi delete yang digunakan untuk menghapus data sesuai dengan id dengan memanggil kategori dan {id}, file yang dipanggil adalah function destroy yang ada di dalam file KategoriController. Jadi ketika url kategori diketikkan, maka akan menampilkan isi dari file destroy pada file KategoriController.

Apabila melakukan pemanggilan function ini, maka url harus diberi tambahan id.

1. Put() adalah perintah yang digunakan untuk update data pada route, tetapi untuk menggunakannya harus diberi /variabel {id} agar function put dapat berjalan.

Cth :

$router->put('kategori/{id}', 'KategoriController@update');

Artinya, fungsi put yang digunakan untuk update data sesuai dengan id dengan memanggil kategori dan {id}, file yang dipanggil adalah function update yang ada di dalam file KategoriController. Jadi ketika url kategori diketikkan, maka akan menampilkan isi dari file update pada file KategoriController.

Apabila melakukan pemanggilan function ini, maka url harus diberi tambahan id.

1. Post() adalah perintah yang digunakan untuk mengirimkan nilai data pada route.

Cth :

$router->post('kategori', 'KategoriController@create');

Artinya, fungsi post adalah untuk mengirimkan nilai data ketika kategori dituliskan ke dalam url dan file yang dipanggil adalah function create pada file KategoriController.

1. Mengisi request di dalam function create :

public function create(Request $request)

    {

*return* response()->json($request);

    }

Artinya, parameter di dalam function create berisi perintah request yaitu perintah untuk mengirim $request, jadi sebanyak apapun data yang dikirim, maka data akan disimpan ke dalam $request. Setelah itu, nilai akan dikeluarkan menggunakan return, dan response() akan mengirim data pada $request dengan bentuk json.

1. Menampilkan satu data menggunakan get(), tetapi di dalam bagian yang memanggil diberi /variabel {id} sesuai dengan data yang ingin ditampilkan.
2. Membuat group router yang berisi 2 parameter, sebelah kiri yaitu dengan menentukan apa nama group yang akan dituliskan di url dan sebelah kanan dengan meletakkan function yang berisi $router - $router yang dipanggil. Setelah itu, setiap bagian yang dipanggil dari $router harus diberi kurung staples dan diberi [‘uses’ => ‘nama file dan function’].

Cth :

$router->group(['prefix' => 'api'], function () use ($router) {

    $router->get('kategori', ['uses' => 'KategoriController@index']);

});

Artinya, group ini bernama api, dan apabila ingin melakukan pemanggilan terhadap salah satu file di dalamnya, maka harus dengan menambahkan localhost/8000/api/namapemanggil.

1. Migrations adalah fitur yang digunakan untuk mempermudah pembuatan dan pelacakan database schema.
2. Menambahkan tabel menggunakan file migrations dan menentukan tabel apa saja yang dibuat.
3. Php artisan migrate digunakan untuk migrate tabel apa saja yang akan dibuat di dalam file migrate. Setelah migrate maka tabel akan ditambahkan ke dalam database.
4. Php artisan migrate:rollback digunakan untuk menghilangkan / menarik kembali semua tabel yang sudah dibuat karena ada kolom dari salah satu tabel yang akan ditambahkan. Apabila ingin menambahkan tabel ke dalam database kembali, maka menggunakan perintah php artisan migrate.
5. Php artisan db:seed digunakan untuk mengirimkan data yang sudah dimasukkan pada file Seedeer ke dalam database. Tetapi, sebelum mengirimkan data, maka file seeder harus ditambahkan terlebih dahulu ke dalam file DatabaseSeeder.
6. Faker adalah data palsu yang digunakan. Caranya adalah dengan mendeklarasikan Faker terlebih dahulu, setelah itu menjalankan $faker dan memasukkan $faker ke dalam data.

Cth :

use Faker\Factory as Faker; // deklarasikan Faker

        $faker = Faker::create(); // $ faker berisi objek yang menjalankan Faker

        $data = [

     'kategori' => $faker->name, // memasukkan faker ke dalam kolom kategori

     'keterangan' => $faker->text //memasukkan faker ke dalam kolom keterangan

        ];

1. Untuk memasukkan banyak data menggunakan faker, kita hanya perlu menggunakan for, setelah itu memasukkan $data yang akan dikirimkan.

Cth :

*for* ($i = 0; $i < 100; $i++) {

            $data = [

                'pelanggan' => $faker->name,

                'alamat' => $faker->address,

                'telp' => $faker->phoneNumber

            ];

            Pelanggan::create($data);

        }

Artinya, setiap data dari kolom pelanggan, alamat, dan telp akan dikirimkan menggunakan $faker. Dan untuk mengirimkan faker, maka di dalam terminal dituliskan php artisan db:seed.

1. Php artisan route:list adalah perintah yang digunakan untuk menampilkan route apa saja yang sudah diatur.
2. Memanggil semua isi dari kategori pada database dengan menggunakan all().

Cth :

  $data = Kategori::all();

*return* response()->json($data);

Artinya, $data mengambil nilai dari Models Kategori lalu semua data yang ada di dalam tabel kategori akan ditampilkan menggunakan all(). Setelah itu, nilai dari $data akan dikeluarkan dan dikirimkan dalam bentuk json.

1. Menampilkan kolom yang dipilih menggunakan perintah where().

Cth :

public function show($id)

    {

        $data = Kategori::where('idkategori', $id)->get();

*return* response()->json($data);

    }

Artinya, pada function show terdapat parameter yang diisi dengan $id, lalu di dalam function terdapat $data yang mengambil nilai dari Models Kategori dan menjalankan function where dengan memilih kolom idkategori yang sesuai dengan $id yang dipilih, setelah itu data dengan nilai id yang sama akan ditampilkan menggunakan get(). Lalu, nilai dari $data akan dikeluarkan dan dikirimkan ke dalam bentuk json.

1. Insert data menggunakan function create dengan mengatur $fillable / kolom yang akan dimasukkan data terlebih dulu.

Cth :

//Di dalam KategoriController.php

public function create(Request $request)

    {

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

*return* response()->json('Data berhasil dimasukkan');

    }

Artinya, pada function create berisi parameter yang di dalamnya terdapat Request untuk menerima semua nilai yang dimasukkan dan data disimpan ke dalam $request, yang berarti semua data yang akan dimasukkan akan ditampung ke dalam $request terlebih dahulu. Setelah itu, Models Kategori menjalankan function create dengan mengirimkan semua data yang ada pada $request menggunakan all(). Nilai akan dikeluarkan dan apabila data berhasil dikirim, maka akan menampilkan tulisan Data berhasil dikirimkan.

//Di dalam file Kategori.php pada Models

protected $fillable = ['kategori', 'keterangan'];

Untuk dapat memasukkan data, maka kita harus menentukan kolom mana yang dapat diisi terlebih dahulu dengan menambahkan $fillable di dalam file Model. Kolom yang dapat dimasuki data pada tabel kategori di dalam database adalah kolom kategori dan kolom keterangan.

1. Delete data sesuai dengan id yang dimasukkan menggunakan function destroy.

Cth :

public function destroy($id)

    {

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

*return* response()->json("Data berhasil dihapus");

    }

Artinya, pada function destroy terdapat parameter yang berisi $id, dan di dalam function berisi Model Kategori menjalankan function where yang memilih idkategori yang sama dengan

$id yang dimasukkan, setelah itu menjalankan function delete. Apabila berhasil menghapus data, maka nilai akan dikeluarkan dan menjalankan function response yang mengirimkan data berbentuk json dengan tulisan data berhasil dihapus.

**Saya Belum Mengerti**