|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Rest API (LUMEN) Semester-4 Minggu ke-7 Video 12-20 | 95 |

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

1. Menggunakan validasi atau validate ketika menambahkan data ke dalam database untuk menyeleksi data yang dimasukkan sesuai ketentuan validate nya. Function yang digunakan untuk menambahkan data adalah function create, jadi validate nya diletakkan di dalam function ini. Penulisannya seperti ini

$this->validate($request, [

‘kategori’ => ‘required | unique:kategoris’,

‘keterangan’ => ‘required’

]);

Yang berarti data yang masuk dari request-an nya user akan diseleksi atau divalidasi apabila data yang dimasukkan untuk kolom kategori kosong maka dia akan menjalankan pesan required (tidak boleh kosong) sedangkan unique:kategoris berarti apabila data yang dimasukkan sama dengan data lain yang ada di kolom itu pada tabel kategoris maka dia akan menjalankan pesan unique nya (tidak boleh sama).

2. Mengupload image atau gambar ke dalam folder upload. Caranya yang pertama yaitu di dalam controller bagian function create (sesuai dengan function yang dituju untuk dipost pada routes nya) parameter nya diisi dengan Request $request agar terisi data dari request penggunanya, lalu isi functionnya sendiri yaitu yang pertama dibuatkan variable misal $gambar yang diisi dengan $request->file(‘gambar’)->getClientOriginalName(); yang berarti $gambar ini mengambil file dari key ‘gambar’ yang diisi sesuai request user beserta nama asli file nya. Apabila terjadi error pada bagian mengambil nilai ini maka coba dicheck menu headers yang ada di postman nya lalu unchecklist key Content-Type nya. Jika sudah berhasil mengambil nilai file nya, dibawah code $gambar dibuatkan $request->file(‘gambar’)->move(‘upload’, $gambar); yang berarti file gambar yang telah diambil tadi akan diupload ke dalam folder bernama upload. Untuk mengecek apakah upload ini berhasil menggunakan return response()->json($gambar); yang jika berhasil akan menampilkan nama file yang dipilih tadi.

3. Membuat nama gambar yang akan dimasukkan ke dalam database menjadi sebuah url sesuai dengan letak gambarnya. Caranya di dalam controller pada bagian function create yaitu array yang digunakan untuk menyimpan data sesuai dengan nama kolom dan akan dimasukkan ke database pada bagian kolom ‘gambar’ ini diisi dengan => url(‘upload/’.$gambar) yang nantinya nama gambar ini akan berubah menjadi sebuah url sesuai dengan letak gambar yang telah diupload. Untuk penulisannya upload/ harus berada didalam ‘’ karena dia string dan disambungkan dengan $gambar harus ditambahi . karena string dengan variable berbeda typedata.

4. Membuat status response agar dapat mengetahui apakah data yang akan dimasukkan ke dalam database berhasil masuk atau tidak, caranya dengan membuat if else untuk menentukan pada if nya diisi $menu dimana $menu ini adalah variable yang menyimpan Menu::create($data) untuk membuat $data masuk ke dalam database nya jadinya seperti ini

if ($menu) {

            $result = [

                'status' => 200,

                'pesan' => 'Data sudah ditambahkan',

                'data' => $data

            ];

        } else {

            $result = [

                'status' => 400,

                'pesan' => 'Data tidak bisa ditambahkan',

                'data' => ''

            ];

        }

Setelah dibuatkan if else lalu pada response nya diisi dengan $result agar response nya muncul sesuai dengan $result nya.

5. Middleware terletak di antara route dengan controller yang fungsinya sebagai tempat pertimbangan apakah diteruskan ke controller atau tidak. Contohnya seperti memberikan login terlebih dahulu sebelum masuk ke page dari group api nya. yang pertama yaitu membuat file UserMiddleware nya dulu di dalam folder middleware, lalu tambahkan code pada bagian function handle seperti ini

if ($request->user <> 'admin') {

            return redirect('login');

        }

Yang berarti jika user yang mencoba masuk ke page bukan admin maka dia akan redirect ke login atau menjalankan url /login. Setelah itu dibuat controller loginnya dengan membuatkan function index yang isinya untuk login. Lalu pada routes nya ditambahkan

$router->get('login', ['uses' => 'LoginController@index']); yang berfungsi apabila url nya login dia akan menjalankan LoginController bagian function index, lalu pada routes group nya ditambahkan 'middleware' => 'user' menjadi seperti ini $router->group(['prefix' => 'api', 'middleware' => 'user'], function () use ($router) yang berarti routes ini diawali api tetapi harus melewati middleware user dulu untuk masuk ke dalam page dari routes groupnya. Untuk mengetahui user ini yang mana, pada file app.php dibagian $app->routeMiddleware dibuka komenannya dan ditambahkan untuk user nya menjadi seperti ini

$app->routeMiddleware([

    'auth' => App\Http\Middleware\Authenticate::class,

    'user' => App\Http\Middleware\UserMiddleware::class,

]);

Dari $app->routeMiddleware bagian user nya ini dia akan menjalankan UserMiddleware yang telah dibuat tadi dimana file ini yang akan meredirect ke /login apabila bukan admin.

6. API Token berfungsi sebagai keamanan untuk routes nya, letak token sama seperti middleware. Untuk menggunakan token yang pertama adalah di dalam file app.php buka komenan dari $app->register(App\Providers\AuthServiceProvider::class) agar providers nya bisa digunakan. Setelah itu di dalam providers nya di file AuthServiceProvider nya pada bagian $this->app[‘auth’] bawah diisi dengan $token = diisi terserah misalkan “1234567890” nah $token ini sebagai tempat untuk menyimpan password tokennya lalu dibuatkan $header = $request->header(‘Api-Token’); yang dimana $header ini berfungsi untuk menyimpan data yang dimasukkan ke dalam value dari key Api-Token di dalam header postman. Setelah itu dibuatkan if ($header && $header == $token){ return new User();} yang berarti apabila ketika url yang dipilih dengan data yang dimasukkan ke dalam header sama dengan tokennya maka url tujuannya akan berjalan karena sudah melewati proses middleware.

7. Register User sebagai registrasi atau pendaftaran user baru, jadi routes untuk register nya bermethod post karena seperti daftar yaitu menambahkan sebuah data untuk dimasukkan ke dalam database. Models untuk user ini dari awal sudah disediakan otomatis dari lumennya, jadi tinggal membuat controller dan migrations nya. Jika sudah, selanjutnya mengatur migrations nya dulu untuk dibuatkan tabel seperti migration yang lain sebelumnya lalu mengatur LoginController untuk menyiapkan function register nya seperti biasa. Di dalam function register nya diisi dengan $data yang menyimpan array untuk mengisi kolom sesuai dengan yang ada di database nya.

8. Login User untuk proses login user nya, caranya pada function login yang ada di LoginController.php pertama dibuatkan $email yang berfungsi untuk menyimpan data dari data email yang dimasukkan atau direquest user lalu $password fungsinya juga sama seperti $email yaitu untuk menyimpan data password dari user. Jika sudah selanjutnya dibuatkan $user seperti ini $user = User::where('email', $email)->first(); nah $user ini fungsinya untuk mencari email di database yang sama dengan email yang dimasukkan oleh user nya, jadi dia memilah email atau mencocokkan email database dengan email request, jika sudah menemukan pasangan emailnya dia akan mengambil sebaris data ini karena ditandai dengan first() itu. Setelah itu dibuatkan if else untuk pertimbangan seperti ini

if ($user->password === $password) {

            $token = Str::random(40);

            $user->update([

                'api\_token' => $token

            ]);

            return response()->json([

                'pesan' => 'login berhasil',

                'token' => $token,

                'data' => $user

            ]);

        } else {

            return response()->json([

                'pesan' => 'login gagal',

                'data' => 'data kosong'

            ]);

        }

Jadi apabila akun yang direquest user passwordnya sesuai atau sama dengan yang telah terdaftar di database maka dia akan menjalankan yang pertama dibuatkan $token yang berfungsi untuk memberikan random secara random dengan 40digit setelah itu ada $user->update(['api\_token' => $token]); yang fungsinya untuk mengupdate kolom api\_token dengan token baru secara random. Setelah itu ada return response untuk memberikan response jika login dan update token berhasil maka akan muncul pesan login berhasil dengan token yang barunya apa dan menampilkan data dari akun yang digunakan untuk login. Else berfungsi untuk memberikan pesan apabila login gagal maka dia akan menjalankan pesan login gagal dengan data yang bertuliskan data kosong.

9. Autentikasi User berfungsi untuk menjalankan url agar tampil sesuai dengan url nya dengan cara memasukkan token dulu. Adanya update token pada login bagi user berfungsi untuk mendapatkan token baru yang digunakan untuk menjalankan url nya. jadi ketika user akan mengakses url misal localhost:8000/api/kategori dengan method get yang berarti tujuannya adalah untuk menampilkan semua data kategori nya. ketika url sudah disend tetapi hasilnya unauthorized karena belum memasukkan token, maka pada bagian header nya ditambahkan key api\_token lalu value nya diisi dengan token yang didapatkan ketika login. Dengan begitu url nya akan berjalan sesuai tujuannya, jadi token ini fungsinya sebagai autentikasi atau sandi yang dimasukkan ketika akan menampilkan data.

**Saya Belum Mengerti**

1.