|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Rest Api Lumen (Vid 12 – 20) | 90 |

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

1. Memahami fungsi validation pada lumen adalah untuk penyaringan data yang akan di inputkan dalam database sudah sesuai dengan kriteria yang ditentukan atau tidak

2. Memahami cara penulisan syntax untuk proses validasi dengan lumen yaitu $this->validate($request, [‘kolom tabel database’=>`rules validate’ ]);

3. Memahami bahwa setiap menuliskan validasi untuk unique: harus ada acuan data mana yang akan diberi kualifikasi unik (bisa tabel datanya)

4. Dapat menerapkan penggunaan validasi untuk proses input data pelanggan & kategori

5. memahami & mengetahui macam-macam rules untuk validation email,data,required,password,numeric dll.

6. Memahami & dapat meng-upload/insert gambar ke database :

* Membuat model dan controller menu menggunakan php artisan
* Pada migrations tambahkan kolom-kolom untuk tabel database menu juga tipe datanya
* Lalu migration untuk membuat database menu
* Pada controller menu di function insert tambahkan request pada parameter (Request $request) lalu membuat syntax untuk request upload gambar $gambar = $request->file('gambar')->getClientOriginalName();
* Kemudian menambahkan opsi agar gambar yang di upload akan di simpan pada sebuah folder $request->file('gambar')->move('upload', $gambar);
* Pada postman saat proses upload gambar menggunakan form-data tipe file lalu nama key nya harus sama dengan yg berada pada syntax yaitu ‘gambar’
* Gambar siap di upload dan di simpan pad folder

7. Memahami & dapat meng-upload/insert nama gambar ke database :

* Menambahkan validasi pada function create untuk upload gambar di controller menu yaitu required & numeric untuk idkategori dan harga
* Membuat array untuk request idkategori,menu,gambar & harga. $request->input('idkategori/menu/harga')
* Untuk gambar menggunakan function url(‘folder/’.$gambar) agar gambar berbentuk link dan dapat di buka
* Memanggil modal Menu untuk menjalankan array tersebut $menu = Menu::create($data);
* Menggunakan postman data gambar siap dikirim ke database

8. memahami bahwa status response berfungsi untuk mengatur message sesuai dengan request http nya

9. Memahami & dapat menerapkan penggunaan respon dalam program

10. Memahami penulisan syntax status response yaitu array assossiatif lalu diisi status menggunakan nomer status request http sesuai yg diinginkan,kemudian pesan dan terakhir menampilkan variable penyimpan datanya

11. Memahami bahwa middleware merupakan penengah antara request yang masuk dengan controller yg dituju pada request dan respon. Misal kita ingin memverivikasi/filter setiap request yg masuk seperti melakukan pengecekan status login dsb.

12. Memahami & dapat menerapkan penggunaan middleware pada program. Misalkan untuk memfilter apa saja yg boleh masuk ke dalam controller yg diinginkan dan apabila gagal maka akan di arahkan ke controller login, yaitu :

* Membuat middleware untuk user lalu mengisi function handle dengan if ($request->user <> 'admin') {return redirect('login');} yg artinya apabila request yg masuk tidak = admin maka page akan di arahkan ke controller Login
* Meregistrasi middleware melalui routeMiddleware pada file app.php di folder bootstrap yaitu dengan membuka comment $app->routeMiddleware kemudian menambahkan untuk user menggunakan UserMiddleware seperti pada auth
* Pada file web.php untuk routing tambahkan ruter untuk mengatur user yg tidak=admin agar diarahkan ke laman LoginController $router->get('login', ['uses' => 'LoginController@index']);
* Pada grup router untuk elemen lain tambahkan juga ‘middleware’=>’user’ untuk mencegah yang bukan admin mengakses laman tersebut
* Membuat LoginController lalu membuat function index untuk menampilkan form login. Menggunakan index dikarenakan pada router pengaturan login diarahkan menggunakan function index

13. Memahami bahwa token merupakan bagian middleware berfungsi sebagai perangkat security/keamanan dari aplikasi yang dibuat agar aplikasi tidak mudah di akses oleh oknum yang tidak ber hak

14. Memahami cara menambahkan token :

* Pada app.php di folder bootstrap buka comment untuk AuthServiceProvider::class
* Lalu pada file AuthServiceProvider di folder provider tambahkan $token = “isi bebas”
* Kemudian buat $header = $request->header(‘Api-Token/nama boleh custom’);
* Tambahkan if untuk menguji apakah header sama dengan token ($header && $header == $token) kalo benar maka akan menampilkan User atau isi dari yg di akses tersebut
* Pada web.php bagiaan router ‘middleware’ => ‘ini diubah menjadi auth’ karena mengikut dari AuthServiceProvider
* Dapat dicoba melalui postman dengan get key nya berasal dari nama request header kemudian value nya isi token

15. Memahami & dapat menerapkan pembuatan register pada program yaitu :

* Membuat migration untuk user terlebih dahulu menggunakan php artisan
* Menambahkan schema pada migration user dengan isi id,email,password,level,api\_token,status,relasi
* Pada LoginController buat 2 function untuk register dan login
* Sebelum class impor terlebih dahulu use App\Models\Users untuk menggunakan model user nantinya dan illuminate\http\Request untuk menggunakan request karena akan input data ke database
* Pada function register di paramater tambahkan (Request $request) lalu buat array untuk input data ke database sesuai jenisnya ‘email’ => $request->input(‘email’); {untuk password sementara tidak di enkripsi, juga level dan status dibiarkan default lalu relasi disamakan dengan email}
* Kemudian diluar array asossiatif panggil model User::create($array); untuk menggunakan function create untuk build data lalu di input ke database
* Pada model User kolom fillable isilah dengan kolom2 tabel user {sementara password di pindah ke sini karena belum di enkripsi}
* Pada router untuk register ubah login menjadi api/register dengan method post untuk login api/login
* Bisa dicoba menambahkan data lewat postman juga

16. Memahami bahwa function first() berfungsi mengambil satu data pertama jika ada 2 atau lebih data yg sama

17. Memahami penerapan pembuatan proses login dengan memanfaatkan token yaitu:

* Pada router ganti method untuk login menjadi post
* Pada controller login isi parameter function login dengan (Request $request)
* Buat variable untuk input email dan password $email = $request->input('email'); $password = $request->input('password');
* Buat variable untuk menampung modal untuk mengambil data email yg pertama $user = User::where('email', $email)->first();
* Membuat pengujian if untuk menguji apakah password inputan sudah benar dengan passoword user yg terdaftar di database ($user->password === $password)
* Jika if terpenuhi maka akan dibuatkan token jenis string random $token = Str::random(jumlah nya misal 40);
* Lalu user akan meng-update api\_token di database $user->update([‘api\_token’ => $token]); jadi tiap login token akan teregenerate dan ganti otomatis
* Lalu tinggal menampilkan response jika proses login telah berhasil dengan return response misal menampilkan pesan berhasil,token generate dan data user
* Tambahkan juga else apabila proses login tidak berhasil,misal menampilkan message gagal dsb.

18. Memahami bahwa proses autentikasi merupakan proses filter saat seorang user login, agar user yg tidak berhak tidak bisa menerobos masuk.

19. Dapat membuat proses autentikasi berdasarkan token user, misal :

* Pada AuthServiceProvider syntax untuk token yg sebelumnya di comment lalu buka comment syntax di bawahnya dan ubah input menjadi header karena token berada di dalam header
* Uji menggunakan postman dengan login menggunakan token maka permintaan akan terlaksana atau tidak

20. Memahami bahwa setiap permintaan/request harus menggunakan token dan token tersebut di pasang di dalam header.