**Auntentikasi API**

Langkah awal yaitu membuat middleware

* Membuat file baru di HTTP->Middleware->namafile buat nama file dengan nama UserMiddleware dan isi nya sebagai berikut:

|  |
| --- |
| <?php  namespace App\Http\Middleware;  use Closure;  class UserMiddleware  {  /\*\*  \* Handle an incoming request.  \*  \* @param \Illuminate\Http\Request $request  \* @param \Closure $next  \* @return mixed  \*/  public function handle($request, Closure $next)  {  if ($request->user <> 'admin') {  return redirect('login');  }  return $next($request);  }  } |

* Langkah berikutnya meregistrasikan middleware, berada di dalam folder

bootstrap->app.php di line 82 sampai 84 buka hidupkan sintaknya dan sesuaikan pada berikut ini:

|  |
| --- |
| $app->routeMiddleware([  'auth' => App\Http\Middleware\Authenticate::class,  'user' => App\Http\Middleware\UserMiddleware::class,  ]); |

* Menambahkan sintaks pada routes->web.php, menambahkan routes login controller di atas group router dan menambahkan middleware user didalam router group

Berikut sintaknya dengan font warna merah:

|  |
| --- |
| // router login (untuk mengakses Api akan diarahkan ke router ini terlebih dahulu)  $router->get('login', ['uses' => 'LoginController@index']);  // router grop  $router->group(['prefix' => 'api', 'middleware' => 'user'], function () use ($router) {  $router->get('mahasiswa', ['uses' => 'MahasiswaController@index']);  $router->get('mahasiswa/{id}', ['uses' => 'MahasiswaController@show']);  $router->delete('mahasiswa/{id}', ['uses' => 'MahasiswaController@destroy']);  $router->put('mahasiswa/{id}', ['uses' => 'MahasiswaController@update']);  $router->post('mahasiswa', ['uses' => 'MahasiswaController@create']);  $router->get('matkul', ['uses' => 'TblMatkulController@index']);  $router->get('dosen', ['uses' => 'TblDosenController@index']);  $router->get('nilai', ['uses' => 'TblNilaiController@index']);  $router->get('api', ['uses' => 'ApiController@getDataBymatkul']);  }); |

* Membuat file controller login di dalam HTTP->Controllers buat dengan nama LoginController.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  use Illuminate\Support\Facades\Hash;  use App\Models\User;  class LoginController extends Controller  {  public function index()  {  echo "<h1>Ini adalah halaman login</h1>";  }  } |

Langkah berikutnya membuat token

* Pada folder bootstrap->app.php cari sintak di line 99 lalu hilangkan komentarnya

Berikut sintaknya:

|  |
| --- |
| $app->register(App\Providers\AuthServiceProvider::class); |

* Di folder providers->AuthServicesProvider.php pada function boot ubah sintaknya dengan sintak dibawah berikut

|  |
| --- |
| public function boot()  {  // Here you may define how you wish users to be authenticated for your Lumen  // application. The callback which receives the incoming request instance  // should return either a User instance or null. You're free to obtain  // the User instance via an API token or any other method necessary.  $this->app['auth']->viaRequest('api', function ($request) {  $token = "Pesonahub1234";  $header = $request->header('Api-Token');  if ($header && $header == $token) {  return new User();  }  return null;  // if ($request->input('api\_token')) {  // return User::where('api\_token', $request->input('api\_token'))->first();  // }  });  } |

* Para folder routes->web.php Mengganti isi dari middleware user dengan auth. Berikut sintaknya

|  |
| --- |
| // router grop  $router->group(['prefix' => 'api', 'middleware' => 'auth'], function () use ($router) { |

Membuat login

* Langkah awal membuat tabel pada database,silahkan ke disini menggunakan migration untuk tabel user\_api, berikut perintah diterminalnya:

|  |
| --- |
| Php artisan make:migration user\_api |

* Kemudian kita akan memasukan field apa saja yang ada di tabel user\_api dengan cara menambahkan sintak berikut ini pada function up yang ada pad file migration yang telah dibuat tadi:

|  |
| --- |
| public function up()  {  Schema::create('tbl\_matkuls', function (Blueprint $table) {  $table->increments('id');  $table->string('username');  $table->string('password');  $table->integer('level');  $table->string('api\_token');  $table->integer('status');  $table->timestamps();  });  } |

* Menambahkan use model User pada LoginController.php berikut sintaknya

|  |
| --- |
| Use App\Models\User; |

* Langkah selanjutnya pergi ke models->User.php lalu sesuaikan protected fillable dengan file yang ada di tabel user\_api yang telah kita buat. Untuk sintaknya sebagai berikut:

|  |
| --- |
| protected $table = "user\_api";  protected $fillable = [  'username', 'level', 'api\_token', 'status', 'password',  ]; |

* Selanjutnya isi tabel user\_api sesuai dengan data yang diinginkan pada database
* Merubah route pada controller login. Silahkan buka folder routes->web.php lalu ubah router untuk login. Dari get ubah menjadi post:

|  |
| --- |
| $router->post('api/login', ['uses' => 'LoginController@index']); |

* Langkah berikutnya merubah isi LoginController.php pada function index pada sintak berikut ini :

|  |
| --- |
| public function index(Request $request)  {  $this->validate($request, [  'username' => 'required',  'password' => 'required|min:6'  ]);  $username = $request->input('username');  $password = $request->input('password');  $user = User::where('username', $username)->first();  if (!$user) {  return response()->json(['message' => 'Login gagal'], 401);  } else if ($user->password === $password) {  $generateTokenupdate = bin2hex(random\_bytes(40));  $user->update([  'api\_token' => $generateTokenupdate  ]);  $generateToken = User::find(1);  $token = $generateToken->api\_token;  return response()->json([  'message' => 'login berhasil',  'api\_token' => $token,  'data' => $user  ]);  } else {  return response()->json([  'message' => 'login gagal',  ]);  }  } |

Catatan: pada tulisan yang berwarna merah itu untuk generate token jika user melakukan login, “*setiap user melakukan login maka otomatis token akan terubah*” jika tidak ingin ada seperti itu maka kita bisa komentar pada font yang berwarna merah tersebut.

Autentikasi

* Merubah isi file dari folder providers->AuthServicesProvider.php ubah pada function boot:

|  |
| --- |
| public function boot()  {  // Here you may define how you wish users to be authenticated for your Lumen  // application. The callback which receives the incoming request instance  // should return either a User instance or null. You're free to obtain  // the User instance via an API token or any other method necessary.  $this->app['auth']->viaRequest('api', function ($request) {  if ($request->header('api\_token')) {  return User::where('api\_token', $request->header('api\_token'))->first();  }  return null;  });  } |

**Penjelasan untuk keamanan api ini:**

|  |
| --- |
| Untuk keamanan api ini menggunakan middleware yang dimana jika user ingin mengakses data api yang disediakan harus login terlebih pada postmant dengan cara memasukan url dengan method post lalu memasukan “*username dan password*” pada body yang sudah terdaftar pada database tabel user\_api. Jika user sudah melakukan login maka user akan mendapatkan token yang nantinya akan digunakan untuk melihat data api yang telah disediakan untuk token harus dimasukan di header pada postmant. Ketika user sudah memasukan token pada header dengan “*key=api-token & value=isitoken*” jika benar maka akan berhasil menampilkan data yang telah disediakan. |