



KASUS 7: AUTHCONTROLLER PADA BACKEND

Catatan :

Pada kasus 7 ini kita akan membahas pembuatan Controller yang khusus menangani proses autentifikasi user dari aplikasi – aplikasi di luar Laravel. User – user ini harus melakukan proses register terlebih dahulu, kemudian setelah accountnya terdaftar maka user dapat melakukan login. Apabila proses loginnya sukses maka user tersebut akan memperoleh token dari Laravel Sanctum. Token ini akan memiliki masa berlaku sehingga apabila token telah expired maka user harus melakukan login ulang, atau user telah logout maka token akan dihapus dan user harus melakukan login ulang.

Proses login, register dan logout akan kita buat scriptnya di dalam sebuah controller khusus API. Berikut ini penjelasan mengenai pembuatan serta script dari authcontroller.

Untuk saat ini kita akan membuat RESTAPI dari Laravel hasil dari materi Laravel sebelumnya. Berikut ini Langkah – Langkah membuat restapi di dalam Laravel :

1. Buka kembali proyek Laravel kalian dengan Visual Studio Code.
2. Buka **terminal** (cmd) di dalam proyek Laravel kalian kemudian ketikkan perintah berikut ini :

php artisan make:controller API/AuthController

Keterangan :

Perintah di atas akan membuat sebuah file controller baru dengan nama **AuthController** di dalam subfolder API.

3. Buka file **AuthController** yang telah dibuat sebelumnya kemudian ketikkan perintah berikut ini:

```
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;

public function login(Request $request) {
    $validator = Validator::make($request->all(), [
        'email' => 'required',
        'password' => 'required',
    ]);
    if($validator->fails()) {
        return response()->json($validator->errors(), 422);
    }

    $credential = $request->only('email', 'password');
    if(!Auth::attempt($credential)) {
        return response()->json([
            'message' => 'Unauthorized',
            'status' => 'Failed'
        ]);
    }

    $user = User::where('email', $request->email)->first();
    if(! $user || ! Hash::check($request->password, $user->password)) {
        return response()->json([
            'message' => 'Unauthorized',
            'status' => 'Failed'
        ]);
    } else {
        $token = $user->createToken('auth_token')->plainTextToken;
        return response()->json([
            'message' => 'Login Sukses',
            'status' => 'Success',
            'user' => $user,
            'access_token' => $token,
            'type_token' => 'Bearer'
        ]);
    }
}
```

Keterangan:

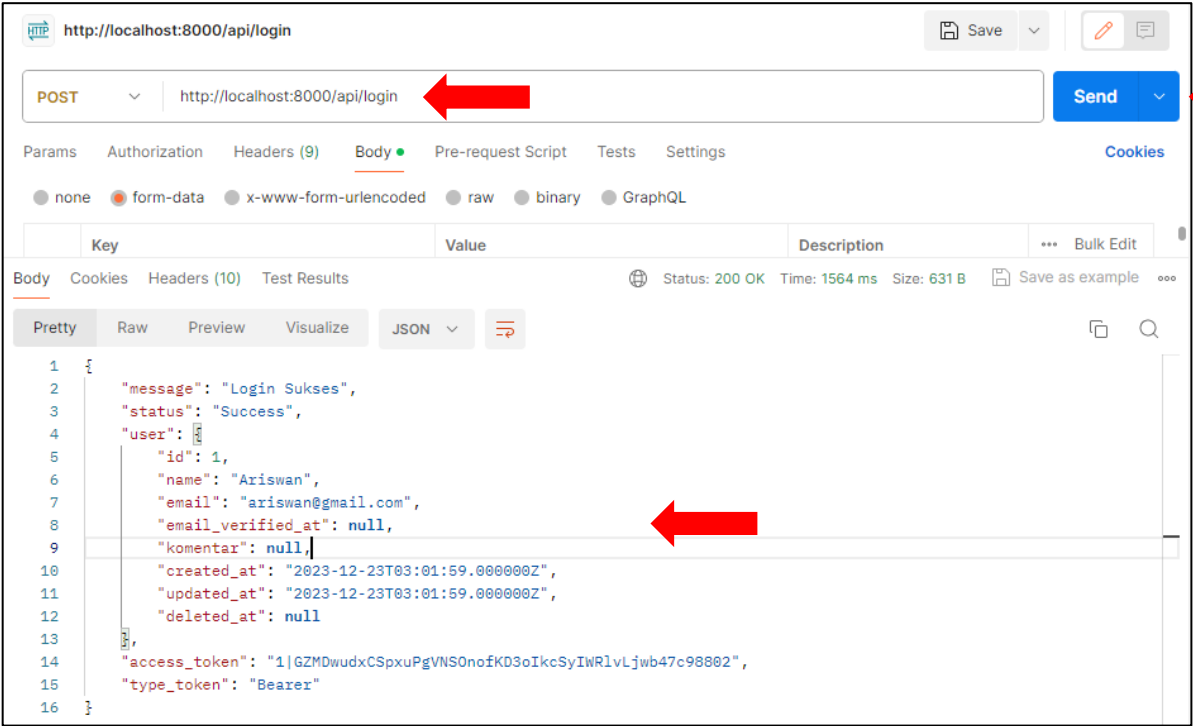
Function di atas akan kita gunakan utk melakukan proses Login via API yang akan membaca table users yang telah dicreate di dalam proyek laravel. Sedangkan untuk jendela tampilan loginnya kita buat di dalam IONIC.

4. Buka file route **api.php** yang ada di dalam folder **routes** proyek Laravel kalian, kemudian tambahkan route baru seperti di bawah ini:

```
Route::post('/login', [AuthController::class, 'login']);
Route::post('/register', [AuthController::class, 'login']);

Route::group(['middleware' => ['auth:sanctum']], function() {
    //membaca data user yang skg login
    Route::get('/user', function (Request $request) {
        return $request->user();
    });
    //Logout
    Route::post('/logout', [AuthController::class, 'logout']);
});
```

5. Untuk mencoba hasil dari script di atas maka kita install Aplikasi **POSTMAN** (silakan download master Aplikasi postman kemudian install ke dalam pc/laptop kalian).
6. Jalankan **Aplikasi Postman** kalian kemudian buat sebuah perintah untuk memanggil route api di atas :



7. Jika sudah bisa memunculkan hasil `access_token` seperti di atas maka artinya RESTAPI login kita sudah berhasil dan siap digunakan untuk berkomunikasi dengan halaman login di dalam proyek ionic.

----- SELAMAT MENCOBA -----