

MODUL MINGGU 5 – SETUP API LARAVEL DAN API RESPONSE GET

1.1 Tujuan Instruksi Khusus

Setelah mengikuti modul ini, mahasiswa diharapkan mampu:

- Mampu membuat project Laravel baru dan mengkonfigurasinya
- Mampu menerapkan layouting menggunakan Laravel
- Mampu membuat API pemanggilan data menggunakan Laravel
- Mampu membuat fitur Read dan Pencarian Data dengan metode AJAX


1.2 Instruksi Modul

1.2.1 Instalasi, Konfigurasi dan Layouting Project Laravel

- Pastikan sudah menginstall composer dan setting path PHP pada environment variable
- Buka **Command Prompt (CMD)**, lalu arahkan ke folder dimana project Laravel akan dibuat. Gunakan perintah **cd** pada CMD untuk pindah folder.
- Jika folder sudah aktif pada CMD, jalankan perintah :

```
composer create-project laravel/laravel:^9.0 cart_laravel_xxxx
```

untuk mengunduh dan membuat project Laravel (versi 9) dengan nama **cart_laravel_xxxx** pada folder tersebut.



```
C:\Windows\System32\cmd.e x + v
E:\demo materi\cart_online>composer create-project laravel/laravel:^9.0 cart_laravel
Creating a "laravel/laravel" project at "./cart_laravel"
Installing laravel/laravel (v9.0.0)
- Downloading laravel/laravel (v9.0.0)
- Installing laravel/laravel (v9.0.0): Extracting archive
Created project in E:\demo materi\cart_online\cart_laravel
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
```

- Pastikan internet aktif, dan tunggu hingga semua proses download dan instalasi project selesai.
- Jika sudah selesai, akan terdapat folder baru dengan nama **cart_laravel_xxxx** sesuai dengan nama project yang dibuat tadi. Jalankan perintah untuk pindah folder pada CMD :

```
cd cart_laravel_xxxx
```

- 6) JANGAN TUTUP APLIKASI CMD-nya
- 7) Nyalakan **XAMPP** dan start **Apache** dan **Mysql**. Lalu akses PHPMyAdmin menggunakan browser. Buat database baru dengan nama **cart_online_xxxx**. Import file .sql yang sudah di share ke dalam database tersebut melalui menu import.
- 8) Buka **file manager**, lalu carilah folder project Laravel yang baru saja dibuat
- 9) Buka file **.env** yang ada pada project tersebut dan edit konfigurasi databasenya seperti berikut (atau menyesuaikan nama dan konfigurasi database kalian masing-masing)

```
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=cart_online_xxxx
15 DB_USERNAME=root
16 DB_PASSWORD=
17
```

- 10) Ekstrak file **adminLTE.zip** yang sudah dibagikan dan pindahkan isinya (folder adminLTE) ke dalam folder **public** pada project Laravel.
- 11) Ekstrak file **view-template.zip** yang sudah dibagikan dan pindahkan isinya (folder template) ke dalam folder **resources/view** pada project Laravel.
- 12) Kembali ke CMD, kita akan membuat controller **Produk.php** dan model **Produkmodel.php**. Gunakan perintah berikut pada CMD :

```
php artisan make:controller Produk -m Produkmodel
```

Inputkan **yes** jika diminta untuk men-generate file. Jika sukses akan ada info berhasil membuat 2 file (controller dan model)

```
C:\Windows\System32\cmd.exe x Command Prompt x + v
E:\demo materi\cart_online\cart_laravel>php artisan make:controller Produk -m Produkmodel
A App\Models\Produkmodel model does not exist. Do you want to generate it? (yes/no) [yes]
> yes
[INFO] Model [E:\demo materi\cart_online\cart_laravel\app\Models\Produkmodel.php] created successfully.
[INFO] Controller [E:\demo materi\cart_online\cart_laravel\app\Http\Controllers\Produk.php] created successfully.
E:\demo materi\cart_online\cart_laravel>
```

- 13) Kemudian, buka file **web.php** pada folder **routes**. Tambahkan pemanggilan class controller Produk dan konfigurasi url routingnya seperti kode berikut (baris 4 dan 17)

```
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Controllers\Produk;
5
6  /*
7  |-----
8  | Web Routes
9  |-----
10 |
11 | Here is where you can register web routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | contains the "web" middleware group. Now create something great!
14 |
15 */
16
17 Route::get('/', [Produk::class, 'index']);
```

- 14) Buka folder **resources/views** dan buat folder baru dengan nama produk, kemudian dalam folder **produk** buat file Views baru dengan nama **list.blade.php**. Masukkan kode berikut pada file tersebut :

```
@extends('template/layout')

@section('judul')
    List Produk
@endsection

@section('konten')
<form>
    <div class="row">
        <div class="col">
            <label>Masukkan Kode</label>
            <input class="form-control" type="text" name="cari" id="cari">
        </div>
        <div class="col">
            <input type="submit" value="Cari ID" class="btn btn-primary" style="margin-top:33px">
        </div>
    </div>
</form>

<table border="1" id="data-list" class="table">
    <tr>
        <th>No.</th>
        <th>Kode</th>
        <th>Nama</th>
        <th>Stok</th>
        <th>Harga</th>
        <th>Deskripsi</th>
        <th>Gambar</th>
        <th>OPSI</th>
    </tr>
</table>
@endsection
```

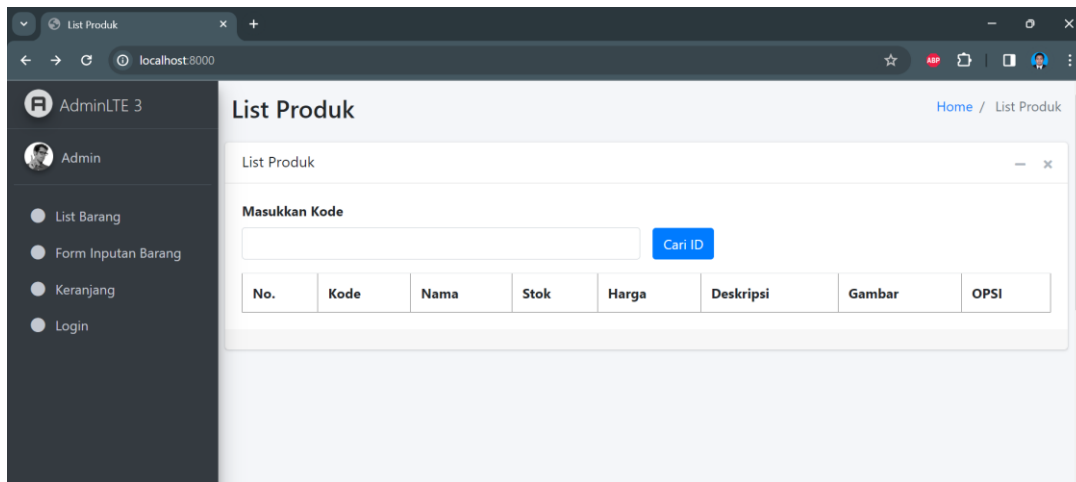
- 15) Kemudian, buka file Controller **Produk.php** dalam folder **app/Http/Controllers**. Ubah fungsi **index()** dan tambahkan kode untuk memanggil file view **list.blade.php** yang baru saja dibuat. Kodenya :

```
public function index()
{
    return view('produk/list');
}
```

- 16) Buka kembali aplikasi CMD, dan jalankan server Laravel dengan perintah :

```
php artisan serve
```

- 17) Akses <http://localhost:8000/> melalui browser, maka akan tampil seperti gambar berikut



1.2.2 Implementasi API dan AJAX Get All Produk

- 1) Buka file Model **Produkmodel.php** yang ada dalam folder **app/Models**. Isikan dengan kode konfigurasi tabel produk yang ada pada database. Kodenya :

```
protected $table = 'produk';
protected $primaryKey = 'id_produk';
```

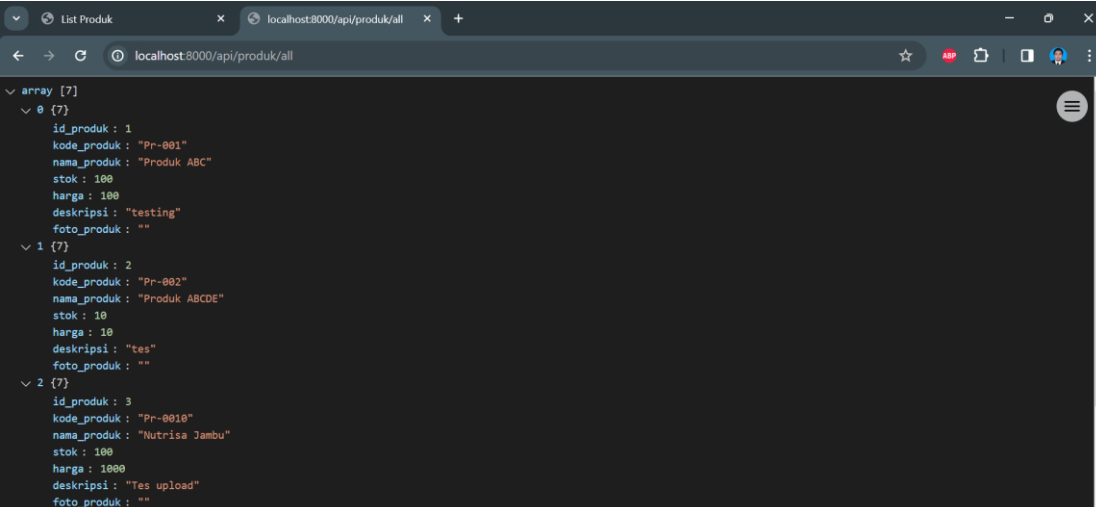
- 2) Buka kembali file Controller **Produk.php**, kemudian buat fungsi baru dengan nama **getListProduk()**. Kodenya :

```
public function getListProduk(Produkmodel $produkmodel)
{
    $data = $produkmodel::all();
    return json_encode($data);
}
```

- 3) Kemudian, buka file **api.php** pada folder **routes**. Tambahkan pemanggilan class controller Produk dan konfigurasi url routingnya seperti kode berikut (baris 5 dan 22)

```
4 use Illuminate\Support\Facades\Route;
5 use App\Http\Controllers\Produk;
6
7 /*
8 |-----
9 | API Routes
10 |-----
11 |
12 | Here is where you can register API routes for your application. These
13 | routes are loaded by the RouteServiceProvider within a group which
14 | is assigned the "api" middleware group. Enjoy building your API!
15 |
16 */
17
18 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
19     return $request->user();
20 });
21
22 Route::get('/produk/all', [Produk::class, 'getListProduk']);
```

- 4) Ujicoba API dengan mengakses url <http://localhost:8000/api/produk/all> menggunakan browser



The screenshot shows a web browser window with the address bar displaying `localhost:8000/api/produk/all`. The page content shows a JSON array of 3 product objects. The first object has `id_produk: 1`, `kode_produk: "Pr-001"`, `nama_produk: "Produk ABC"`, `stok: 100`, `harga: 100`, `deskripsi: "testing"`, and `foto_produk: ""`. The second object has `id_produk: 2`, `kode_produk: "Pr-002"`, `nama_produk: "Produk ABCDE"`, `stok: 10`, `harga: 10`, `deskripsi: "tes"`, and `foto_produk: ""`. The third object has `id_produk: 3`, `kode_produk: "Pr-0010"`, `nama_produk: "Nutrisi Jambu"`, `stok: 100`, `harga: 1000`, `deskripsi: "Tes upload"`, and `foto_produk: ""`.

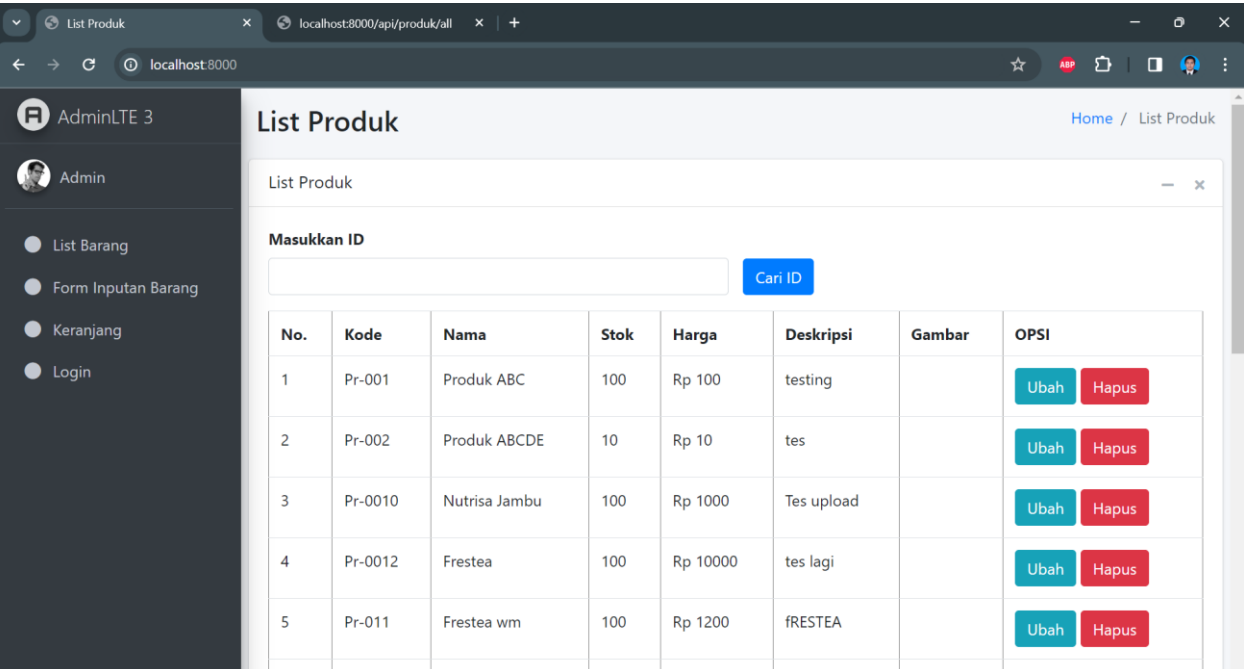
- 5) Buka file View list.blade.php dalam folder resources/views. Tambahkan section `js_custom` dan buat kode pemanggilan API menggunakan AJAX sebagai berikut :

```

34 @section('script_custom')
35 <script>
36     $(document).ready(function(){
37         var link = '{ url("api/produk/all") }';
38         $.ajax(link, {
39             type: 'GET',
40             success : function (data, status, xhr) {
41                 var objData = JSON.parse(data);
42                 var dataTable = '';
43                 $.each( objData, function( key, value ) {
44                     dataTable += '<tr>';
45                     dataTable += '<td>'+(key+1)+'</td>';
46                     dataTable += '<td>'+value.kode_produk+'</td>';
47                     dataTable += '<td>'+value.nama_produk+'</td>';
48                     dataTable += '<td>'+value.stok+'</td>';
49                     dataTable += '<td>Rp '+value.harga+'</td>';
50                     dataTable += '<td>'+value.deskripsi+'</td>';
51                     dataTable += '<td></td>';
52                     dataTable += '<td><a href="#" class="btn btn-info">Ubah</a> <a href="#" class="btn btn-danger">Hapus</a></td>';
53                     dataTable += '</tr>';
54                 });
55                 $( "#data-list" ).append( dataTable );
56             },
57             error : function (jqXHR, textStatus, errorMsg) {
58                 alert('Error Pengambilan Data : ' + errorMsg);
59             }
60         })
61     })
62 </script>
63 @endsection

```

- 6) Akses <http://localhost:8000/> menggunakan browser, jika sukses maka data produk akan muncul di dalam tabel.



The screenshot shows a web browser window with the URL <http://localhost:8000/>. The page displays a sidebar with the user 'Admin' and a main content area titled 'List Produk'. Below the title, there is a search bar labeled 'Masukkan ID' and a 'Cari ID' button. The main content area contains a table with the following data:

No.	Kode	Nama	Stok	Harga	Deskripsi	Gambar	OPSI
1	Pr-001	Produk ABC	100	Rp 100	testing		Ubah Hapus
2	Pr-002	Produk ABCDE	10	Rp 10	tes		Ubah Hapus
3	Pr-0010	Nutrisa Jambu	100	Rp 1000	Tes upload		Ubah Hapus
4	Pr-0012	Frestea	100	Rp 10000	tes lagi		Ubah Hapus
5	Pr-011	Frestea wm	100	Rp 1200	fRESTEA		Ubah Hapus

1.2.3 Implementasi API dan AJAX Pencarian Produk

- 1) buka file **api.php** pada folder **routes**. Tambahkan konfigurasi url routing untuk pemanggilan fungsi pencarian produk. Kodenya :

```
Route::get('/produk/kode/{kode}', [Produk::class, 'getByKode']);
```

- 2) Buka file Controller **Produk.php** dan buat fungsi baru dengan nama **getByKode()**. Kodenya

```
93     public function getByKode(Produkmodel $produkmodel, $kode)
94     {
95         $data = $produkmodel::where('kode_produk', 'LIKE', "%{$kode}%")->get();
96
97         return json_encode($data);
98     }
```

- 3) Buka file view **produk/list.blade.php** dan ubah section **js_custom** dengan memasukkan ajax engine dalam fungsi **ambil_data()** dan menambahkan pemanggilan fungsi **ambil_data()**, kodenya :

```
--
34 @section('script_custom')
35 <script>
36     $(document).ready(function(){
37
38         function ambil_data(kode = '') {
39             if (kode == '') {
40                 var link = '{{ url("api/produk/all") }}';
41             } else {
42                 var link = '{{ url("api/produk/kode/") }}' + '/' + kode;
43             }
44
45             $.ajax(link, {
46                 type: 'GET',
47                 success : function (data, status, xhr) {
48                     resetTable();
49                     var objData = JSON.parse(data);
50                     var dataTable = '';
51                     $.each( objData, function( key, value ) {
52                         dataTable += '<tr>';
53                         dataTable += '<td>'+(key+1)+'</td>';
54                         dataTable += '<td>'+value.kode_produk+'</td>';
55                         dataTable += '<td>'+value.nama_produk+'</td>';
56                         dataTable += '<td>'+value.stok+'</td>';
57                         dataTable += '<td>Rp '+value.harga+'</td>';
58                         dataTable += '<td>'+value.deskripsi+'</td>';
59                         dataTable += '<td></td>';
60                         dataTable += '<td><a href="#" class="btn btn-info">Ubah</a> <a href="#" class="btn btn-danger">Hapus</a></td>';
61                         dataTable += '</tr>';
62                     });
63
64                     $('#data-list').append( dataTable );
65                 },
66                 error : function (jqXHR, textStatus, errorMsg) {
67                     alert('Error Pengambilan Data : ' + errorMsg);
68                 }
69             })
70         }
71
72         ambil_data();
73     })
74 </script>
75 @endsection
--
```

- 4) Buat event ketika onsubmit form maka dilakukan pemanggilan API dengan data kode produk yang dicari. Kodenya

```

74      $(".form").on('submit', function(e){
75          e.preventDefault();
76          var kode = $("input[name=cari]").val();
77          ambil_data(kode);
78      })
79

```

- 5) Agar data pada tabel tidak berulang-ulang, buat fungsi resetTable() pada javascript. Kodenya :

```

function resetTable() {
    $("#data-list").html( "<tr> <th>No.</th> <th>Kode</th> <th>Nama</th> <th>Stok</th> <th>Harga</th> <th>Deskripsi</th> <th>Gambar</th> <th>OPSI</th> </tr>" );
}

```

- 6) Ubah fungsi ambil_data() agar sebelum data dimasukkan ke tabel tabel harus di-reset dahulu (baris ke 49)

```

46      $.ajax(link, {
47          type: 'GET',
48          success : function (data, status, xhr) {
49              resetTable();
50              var objData = JSON.parse(data);
51              var dataTable = '';
52              $.each( objData, function( key, value ) {
53                  dataTable += '<tr>';
54                  dataTable += '<td>'+(key+1)+'</td>';
55                  dataTable += '<td>'+value.kode_produk+'</td>';
56                  dataTable += '<td>'+value.nama_produk+'</td>';
57                  dataTable += '<td>'+value.stok+'</td>';
58                  dataTable += '<td>Rp '+value.harga+'</td>';

```

- 7) Akses <http://localhost:8000/> dan lakukan pencarian data.

