

LAPORAN TUGAS AKHIR
PEMBUATAN WEBSITE “ANEKA RESEP”
Untuk Memenuhi Tugas Akhir Mata Kuliah : Pemrograman Web Lanjut



Oleh :
INTAN VIONASARI
1931710010/10
MI-2D

PROGRAM STUDI MANAJEMEN INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2020

PEMBUATAN WEBSITE “ANEKA RESEP”

Pada Tugas Besar Pemrograman Web Lanjut saya akan membuat web yang berisi tentang resep makanan yang bernama website aneka resep. Website aneka resep adalah website yang berisi kumpulan aneka resep baik resep makanan tradisional atau modern. Pada website ini akan ada beberapa halaman yaitu :

1. Halaman Utama yang memiliki menu login dan register,
2. Halaman Login, Halaman Register,
3. Halaman Beranda,
4. Halaman Detail Resep,
5. Halaman Daftar Resep,
6. Halaman Formulir Tambah Resep,
7. Halaman Formulir Edit Resep.

Database yang ada dalam website ini bernama laravel, dengan dua tabel yaitu user, kumpulan resep.

Untuk user dapat yang pertama kali akses harus register dahulu, jika sudah berhasil register maka user dapat login. Setelah itu akan diarahkan ke halaman beranda yang berisi resep-resep makanan juga ada menu daftar resep jika di klik maka akan mengarah ke halaman daftar resep yang berisi catatan resep kita, dalam halaman daftar resep user dapat melakukan edit data, tambah data serta hapus data.

LANGKAH – LANGKAH

1. Pertama kali yang dilakukan kita harus membuat project laravel dengan mengetikkan perintah pada cmd:

composer create-project --prefer-dist laravel/laravel=7.* cms

2. Setelah project terbentuk, pada terminal lakukan perintah berikut

php artisan key:generate

3. Untuk menjalankan Laravel ketik perintah berikut dalam terminal

php artisan serve

4. Karena saya ingin membuat menu login dan register maka harus membuat autentikasi terlebih dahulu.

- Autentikasi (authentication) berasal dari kata authentic, yang berarti asli, original. Secara istilah maksudnya adalah proses untuk memastikan keaslian atau kebenaran. Sehingga user authentication dapat diartikan sebagai proses untuk memastikan keaslian atau kebenaran pengguna yang mengakses aplikasi. Biasanya proses ini dijumpai dalam bentuk pengecekan menggunakan username dan password. Tujuannya yaitu memastikan keaslian atau kebenaran pengguna yang mengakses aplikasi. Dengan laravel, kita cukup jalankan beberapa command dan aplikasi Laravel siap untuk melakukan user authentication, lengkap dengan registrasi, login, forgot password.

5. Melakukan autentikasi pada cmd, masuk ke dalam direktori project kita lalu ketikkan perintah dan akan muncul perintah sukses:

```
C:\xampp\htdocs\cms>php artisan ui bootstrap --auth
Bootstrap scaffolding installed successfully.
Please run "npm install && npm run dev" to compile your fresh scaffolding.
```

6. Setelah itu compile file-file asset/javascript dengan perintah :

npm install dan npm run dev

```
C:\xampp\htdocs\cms>npm install
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules\watchpack\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":
"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch"
y"} (current: {"os":"win32","arch":"x64"})

added 1093 packages from 495 contributors and audited 1099 packages in 264.777s
```

```
C:\xampp\htdocs\cms>npm run dev

> @ dev C:\xampp\htdocs\cms
> npm run development

> @ development C:\xampp\htdocs\cms
> cross-env NODE_ENV=development node_modules/webpack/bin/we
avel-mix/setup/webpack.config.js

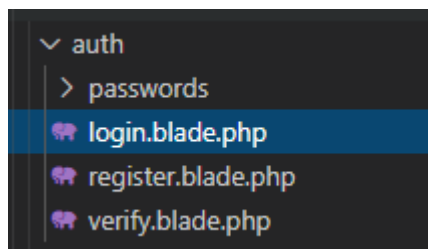
98% after emitting SizeLimitsPlugin
[DONE] Compiled successfully in 52784ms

  Asset      Size  Chunks             Chunk Names
  /css/app.css  178 KiB  /js/app  [emitted]  /js/app
  /js/app.js   1.07 MiB  /js/app  [emitted]  /js/app

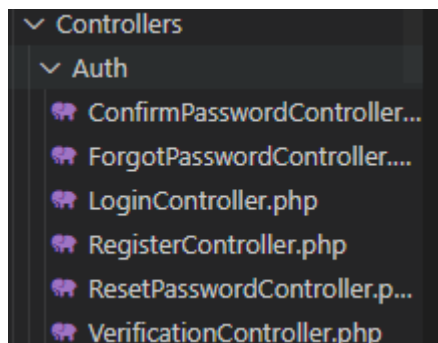
C:\xampp\htdocs\cms>
```

7. Setelah selesai maka akan terbentuk file-file berikut :

Pada view/auth



Pada controllers/auth



Kemudian kita dapat menikmati autentikasi dari laravel dengan cara buka server dari terminal yang sudah kita ketikkan php artisan serve .

8. Jangan lupa untuk membuat database, database ini akan berfungsi untuk menyimpan data yang ingin kamu gunakan juga menyimpan data user yang telah registrasi pada web. Disini saya membuat database bernama laravel.
9. Kemudian sambungkan database pada folder project anda dengan cara membuka file .env lalu mengisi dengan nama database laravel :

```
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=laravel
13 DB_USERNAME=root
14 DB_PASSWORD=
15
```

10. Setelah itu kita akan diarahkan pada halaman view welcome.blade.php

Proses menulis kode serta edit css project saya pada view welcome.blade.php :

```
resources > views > welcome.blade.php
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Home Website Aneka Resep
7 </title>
8 <!-- Fonts -->
9 <link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">
10 <!-- Styles -->
11 <style>
12 h1{
13 color: white;
14 font-family: Trebuchet MS;
15 text-align: center;
16 font-size: 50px;
17 font-style:italic;
18 }
19 .lead{
20 color: white;
21 font-family: Trebuchet MS;
22 text-align: center;
23 font-size: 20px;
24 font-style:italic;
25 }
26 .my-4{
```

```
resources > views > welcome.blade.php
80 <body>
81 <div class="flex-center position-ref full-height">
82     @if (Route::has('login'))
83         <div class="top-right links">
84             @auth
85                 <a href="{{ url('/home') }}">Home
86             </a>
87             @else
88                 <a href="{{ route('login') }}">Login
89             </a>
90             @if (Route::has('register'))
91                 <a href="{{ route('register') }}">Register
92             </a>
93             @endif
94             @endauth
95         </div>
96     @endif
97     <div class="jumbotron" >
98         <div class="container">
99             <br>
100            <br>
101            <h1 class="display-4">Selamat Datang Di Website
102            <br>
103            <span class="font-weight-bold">Aneka Resep Masakan
104            </span>
```

11. Jangan lupa untuk membuat Controller pada setiap file yang akan ditampilkan dengan perintah

Php artisan make:controller <namacontroller>

Jalankan server 127.0.0.1:8000

Dengan output halaman utama sebagai berikut :



12. Setelah itu pilih menu login atau registrasi yang ada pada halaman utama
Halaman login dan register ini termasuk dari bagian autentikasi yang telah dibuat tadi

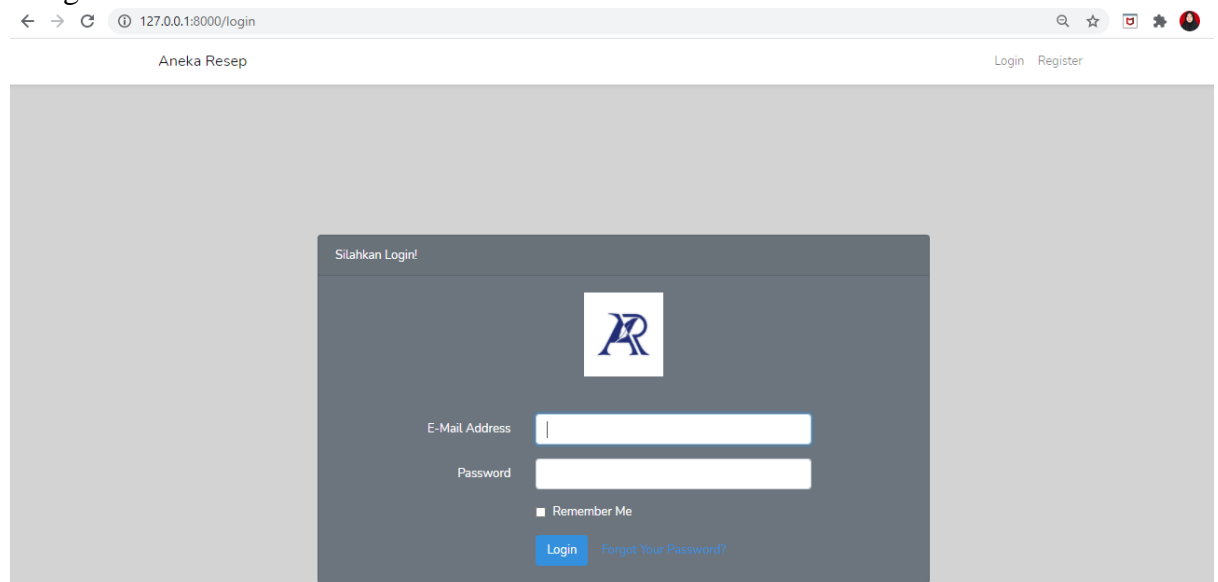
Kode :

login.blade.php

```
resources > views > auth > login.blade.php
14 <div class="row justify-content-center">
15 <div class="col-md-8">
16 <div class="card text-white bg-secondary mb-3" >
17 <div class="card-header">{{ __('Silahkan Login!') }}</div>
18 <div class="card-body">
19 <center></center>
20 <br><br>
21 <form method="POST" action="{{ route('login') }}">
22 @csrf
23
24 <div class="form-group row">
25 <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>
26
27 <div class="col-md-6">
28 <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" value="{{ old('email') }}">
29
30 @error('email')
31 <span class="invalid-feedback" role="alert">
32 <strong>{{ $message }}</strong>
33 </span>
34 @enderror
35 </div>
36 </div>
37
38 <div class="form-group row">
39 <label for="password" class="col-md-4 col-form-label text-md-right">{{ __('Password') }}</label>
```

Output view login.blade.php :

Dengan alamat 127.0.0.1:8000



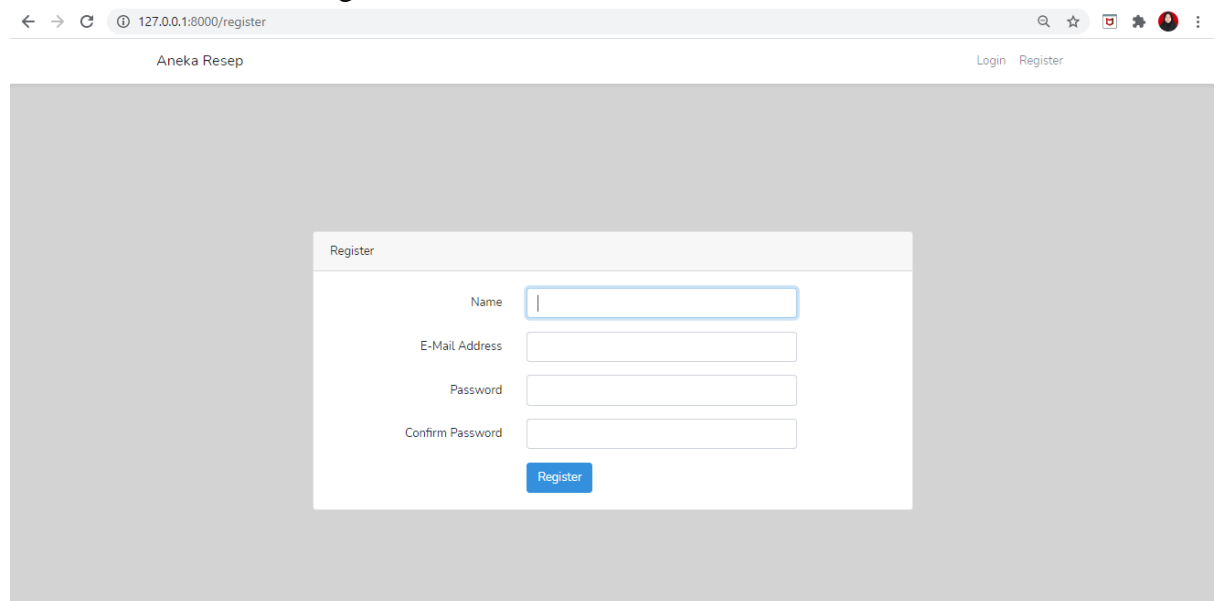
Kode register.blade.php :

```
resources > views > auth > register.blade.php
1  @extends('layouts.app')
2
3  @section('content')
4  <head>
5      <style>
6          body{
7              background-color: lightgrey;
8          }
9      </style>
10 </head>
11 <br><br><br><br><br><br>
12 <div class="container">
13     <div class="row justify-content-center">
14         <div class="col-md-8">
15             <div class="card">
16                 <div class="card-header">{{ __('Register') }}</div>
17
18                 <div class="card-body">
19                     <form method="POST" action="{{ route('register') }}">
20                         @csrf
21
22                         <div class="form-group row">
23                             <label for="name" class="col-md-4 col-form-label text-md-right">{{ __('Name') }}</label>
24
25                             <div class="col-md-6">
26                                 <input id="name" type="text" class="form-control @error('name') is-invalid @enderror">
```

```
resources > views > auth > register.blade.php
27
28         @error('name')
29             <span class="invalid-feedback" role="alert">
30                 <strong>{{ $message }}</strong>
31             </span>
32         @enderror
33     </div>
34 </div>
35
36     <div class="form-group row">
37         <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>
38
39         <div class="col-md-6">
40             <input id="email" type="email" class="form-control @error('email') is-invalid @enderror">
41
42             @error('email')
43                 <span class="invalid-feedback" role="alert">
44                     <strong>{{ $message }}</strong>
45                 </span>
46             @enderror
47         </div>
48     </div>
49 </div>
50 </div>
```

Dengan output :

Alamat 121.0.0.1:8000/register



13. Setelah itu kita ketikkan nama, email dll pada halaman register jika sudah selesai maka data kita akan masuk ke dalam database dengan table users :

Server: 127.0.0.1 Database: laravel Table: users

Showing rows 0 - 1 (2 total, Query took 0.0659 seconds.)

SELECT * FROM `users`

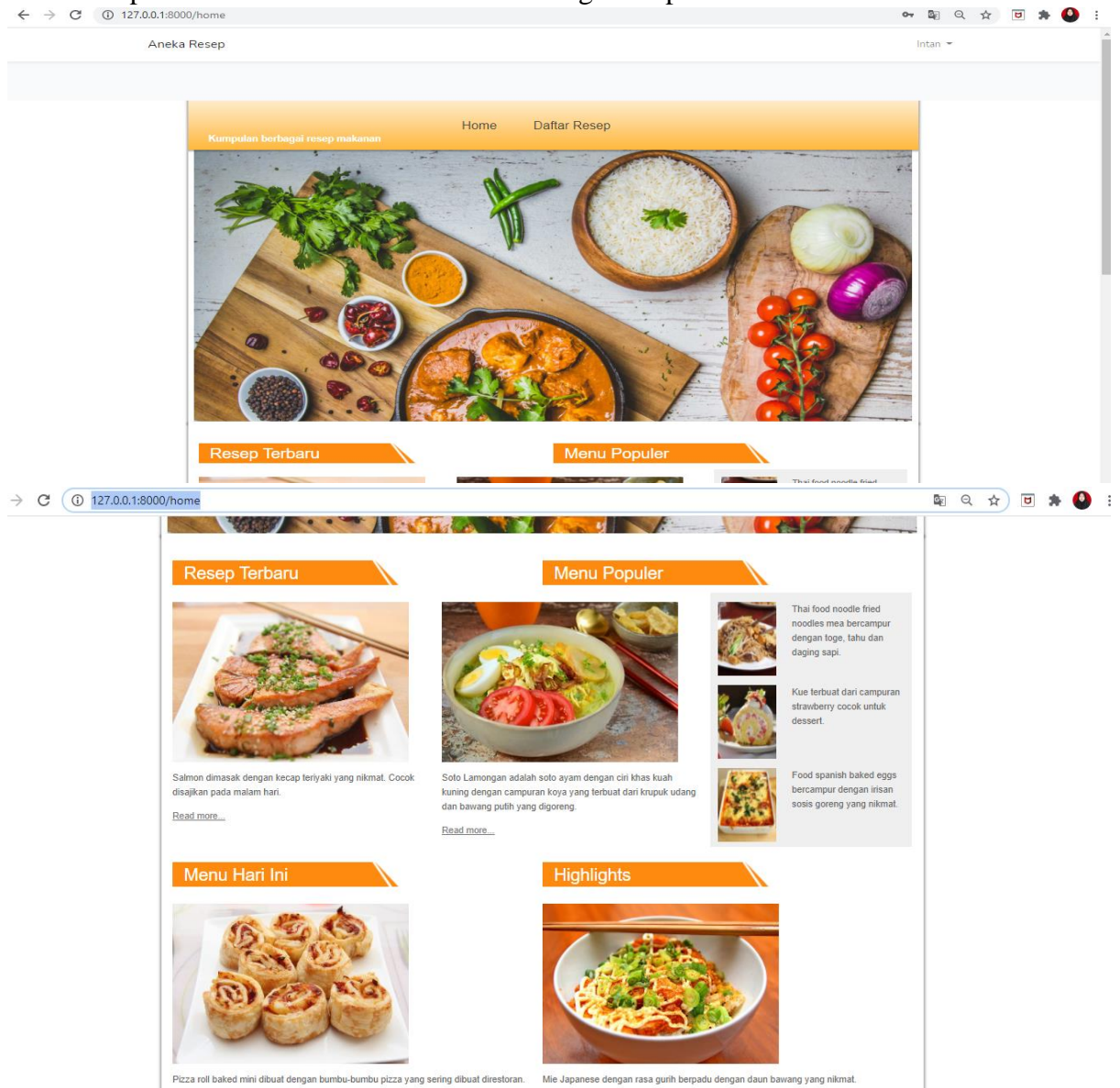
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

		id	name	email	email_verified_at	password	remember_token
<input type="checkbox"/>	Edit	1	Intan	intanviona@gmail.com	NULL	\$2y\$10\$OsJRfXmiexEmDJht2904O3EOJfRq/xOM7Uu7h64DDC...	NULL
<input type="checkbox"/>	Edit	2	Viona	viona888@gmail.com	NULL	\$2y\$10\$0e0JSY4ebA0bpuE.TvejWOvMVX8zNT0q9qGZRTIvbP...	NULL

14. Setelah terdata maka kita dapat login dengan email serta password kemudian akan diarahkan pada alamat 127.0.0.1:8000/home dengan output :



Controller home :

```
pp > Http > Controllers > 🐞 HomeController.php
6
7  class HomeController extends Controller
8  {
9      /**
10       * Create a new controller instance.
11       *
12       * @return void
13       */
14     public function __construct()
15     {
16         $this->middleware('auth');
17     }
18
19     /**
20      * Show the application dashboard.
21      *
22      * @return \Illuminate\Contracts\Support\Renderable
23      */
24     public function index()
25     {
26         return view('home');
27     }
28 }
29
```

Dengan kode pada view home.blade.php berikut :

```
sources > views > 🐞 home.blade.php
1  @extends('layouts.app')
2  @section('content')
3  <html>
4      <head>
5          <style>
6              #art-main
7              {
8                  background: #fff ;
9                  margin:0 auto;
10                 font-size: 12px;
11                 font-family: arial;
12                 font-weight: normal;
13                 font-style: normal;
14                 text-align: justify;
15                 position: relative;
16                 width: 100%;
17                 min-height: 100%;
18                 left: 0;
19                 top: 0;
20                 cursor:default;
21                 overflow:hidden;
22             }
23             table, ul.art-hmenu
24             {
25                 font-size: 12px;
26                 font-family: arial;
```

```

<div class="art-content-layout">
  <div class="art-content-layout-row">
    <div class="art-layout-cell layout-item-0" style="width: 50%; padding-right:380px" >
      <h1>Menu Hari Ini
    </h1>
    </div>
    <div class="art-layout-cell layout-item-0" style="width: 50%" >
      <h1>Highlights
    </h1>
    </div>
  </div>
</div>
<div class="art-content-layout-wrapper layout-item-1">
  <div class="art-content-layout layout-item-2">
    <div class="art-content-layout-row">
      <div class="art-layout-cell layout-item-5" style="width: 40%" >
        <p>
          
        </p>
        <p>Pizza roll baked mini dibuat dengan bumbu-bumbu pizza yang sering dibuat diresto
        </p>
        <p>
          <a href="#">Read more...
        </a>
        </p>
      </div>
    </div>
  </div>
</div>

```

15. Untuk setiap penjelasan makanan ada tulisan **read more**, jika diklik akan mengarah ke resep yang lebih detail contoh outputnya :

← → ↺ ⓘ 127.0.0.1:8000/sotolamongan

Soto Lamongan

WEBSITE ANEKA RESEP



Berbeda dengan soto ayam dengan kuah bening lainnya, resep masakan berkuah dari Lamongan, Jawa Timur ini menggunakan bumbu kemiri. Kombinasi dari kemiri dengan bumbu soto lainnya membuatnya memiliki citarasa yang berbeda dengan jenis lainnya. Sebut saja soto surabaya, soto madura, soto banjar ataupun soto ayam bening dengan kuah tanpa santan lainnya.

Bahan

- 1 ekor ayam kampung segar ukuran sedang
- 3 batang serai yang dimemarkan
- 5 lembar daun salam
- 3 batang daun bawang segar. Cuci bersih kemudian potong sepanjang kurang lebih 1 sampai 1,5 cm
- 1 ruas jari orang dewasa lengkuas yang dimemarkan
- 5 lembar daun jeruk
- 1 sdm gula pasir
- 2 1/2 sdm garam
- 3,5 liter air bersih secukupnya

Kode programnya pada file sotolamongan.html:

```
48 <div class="col-xs-12 nowidthlook">
49 <p>
50 <div style="text-align: center;">
51 
52 <br />
53 <br />
54 Berbeda dengan soto ayam dengan kuah bening lainnya, resep masakan berkuah dari Lamongan, Jawa Timur ini
55 </div>
56 <div>
57 <strong>Bahan
58 </strong>
59 </div>
60 <div>1 ekor ayam kampung segar ukuran sedang
61 </div>
62 <div>3 batang serai yang dimemarkan
63 </div>
64 <div>5 lembar daun salam
65 </div>
```

CRUD

16. Seperti pada awal laporan saya menjelaskan bahwa dalam web saya dapat melihat, menambah, mengedit, menghapus resep kita. Fitur ini terdapat pada pilihan menu daftar resep. Pasti jika terdapat fitur tersebut kita harus membuat table yang akan diisi dengan data-data kita. Pada Laravel untuk memudahkan mengisi database kita dapat menggunakan Model.

- Model adalah bagian dari MVC yang bertanggung jawab untuk menangani basis data (database). Model bertugas untuk query ke database, insert data baru, update, atau hapus record di database.

17. Pertama kita harus membuat model pada project dengan perintah :
Dan jangan lupa untuk menyambungkan database pada project kita

```
PS C:\xampp\htdocs\cms> php artisan make:model Kumpulanresep
Model created successfully.
PS C:\xampp\htdocs\cms>
```

```
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=laravel
13 DB_USERNAME=root
14 DB_PASSWORD=
15
```

18. Lalu melakukan migration dengan tujuan untuk membuat skema database secara langsung dengan mengeksekusi kode program, bukan melalui eksekusi SQL. Dengan migration, pembuatan skema database dilakukan menggunakan kode program PHP untuk menggantikan kode SQL.

Migration membuat table kumpulanresep dengan perintah :

```
PS C:\xampp\htdocs\cms> php artisan make:migration create_kumpulanresep_table
Created Migration: 2020_11_18_131657_create_kumpulanresep_table
```

Lalu kita isi atribut table seperti berikut :

```
database > migrations > 2020_11_18_131657_create_kumpulanresep_table.php
6
7 class CreateKumpulanresepTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('kumpulanresep', function (Blueprint $table) {
17             $table->bigIncrements('id');
18             $table->String('namamakanan',50);
19             $table->String('deskripsi',50);
20             $table->String('bahandancara',250);
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      *
28      * @return void
29      */
30     public function down()
31     {
32         Schema::dropIfExists('kumpulanresep');
33     }
34 }
```

19. Setelah mengedit isi table maka kita akan melakukan migrate, yang berfungsi untuk mengeksekusi file migration kita, lakukan migrate dengan perintah :

```
PS C:\xampp\htdocs\cms> php artisan migrate
Migrating: 2020_11_18_131657_create_kumpulanresep_table
Migrated: 2020_11_18_131657_create_kumpulanresep_table (0.48 seconds)
```

20. Saat ini kita sudah mempunyai table kumpulanresep dengan kolom-kolom yang telah kita lengkapi.

Showing rows 0 - 4 (5 total, Query took 0.0044 seconds.)

SELECT * FROM `kumpulanresep`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

id	namamakanan	deskripsi	bahandancara	created_at	updated_at
----	-------------	-----------	--------------	------------	------------

21. Mengisi data table menggunakan seeding, seeding adalah memberikan data awal ke database. Hal ini biasanya dilakukan saat pengembangan terutama jika kita ingin menguji apakah fitur tertentu telah berjalan sesuai ekspektasi

Jalankan perintah untuk membuat file seeder :

```
PS C:\xampp\htdocs\cms> php artisan make:seeder KumpulanresepTableSeeder
Seeder created successfully.
```

22. Maka akan terbuat file pada app/database/seeds, masukkan data dummy pada method run() dengan perintah :

```
database > seeds > KumpulanresepTableSeeder.php
1  <?php
2
3  use Illuminate\Database\Seeder;
4
5  class KumpulanresepTableSeeder extends Seeder
6  {
7      /**
8       * Run the database seeds.
9       *
10      * @return void
11      */
12      public function run()
13      {
14          DB::table('kumpulanresep')->insert([
15              'id' => 1,
16              'namamakanan' => 'Cumi Bakar',
17              'deskripsi' => 'Cumi Bakar Anti Ribet',
18              'bahandancara' => '1. Cumi Segar, 2. Bumbu Rempah, Caranya olesi cumi dengan bumbu rempah lalu bakar'
19          ]);
20      }
21  }
22
```

Setelah selesai memasukkan data dummy, kita harus menjalankan perintah seeder dengan perintah tersebut maka data dummy yang telah kita buat akan masuk kedalam table kumpulan resep :

```
PS C:\xampp\htdocs\cms> php artisan db:seed --class=KumpulanresepTableSeeder
Database seeding completed successfully.
```

23. Pada studi kasus operasi CRUD, agar memudahkan menggunakan CRUD maka kita menggunakan ORM (Object Relational Mapping), pada ORM kita bisa melakukan operasi CRUD ke database tanpa harus menulis SQL Command. ORM inilah yang bertugas untuk menulis SQL Command. Laravel menggunakan ORM yang bernama Eloquent.

24. Read data, untuk dapat menampilkan data maka kita harus membuat route pada routes/web.php

```
Route::get('/resep', 'ResepController@index')->name('resep');
```

25. Tambahkan fungsi pada resepController yang digunakan untuk menampilkan semua data.

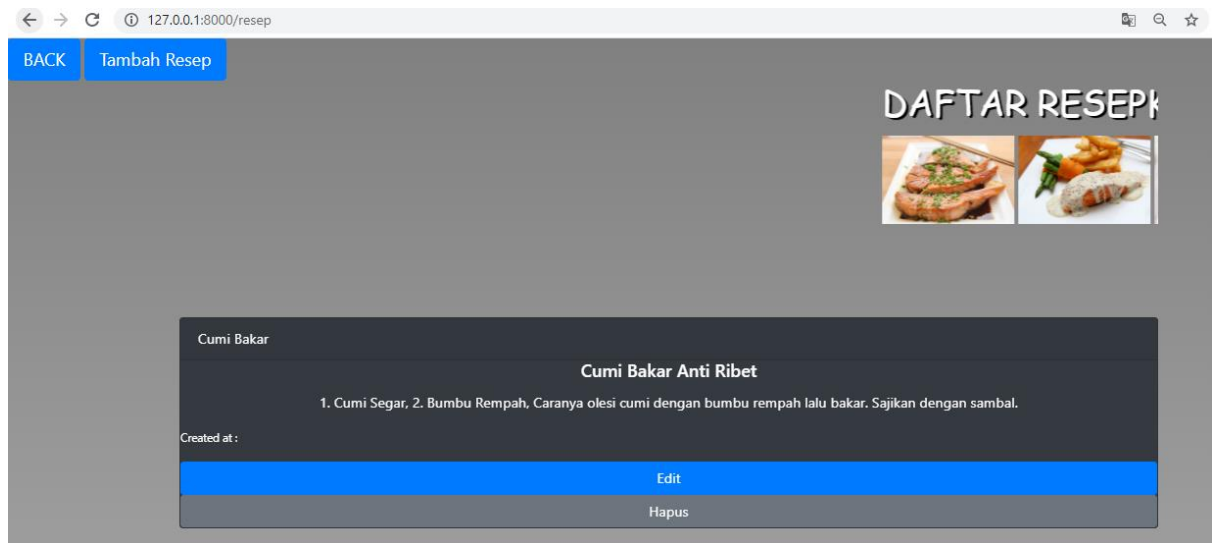
```
pp > Http > Controllers > ResepController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class ResepController extends Controller
8  {
9      public function index()
10     {
11         $data_resep= \App\kumpulanresep::all();
12         return view('resep.index',['data_resep' => $data_resep]);
13     }
14 }
```

26. Membuat file view dengan nama index.blade.php dalam folder resep berguna untuk menampilkan data- data yang ada dalam table.

```

7  
8  </marquee>
9  @foreach($data_resep as $rsp)
10  <br>
11  <div class="card-group">
12      <div class="card text-white bg-dark mb-3">
13          <div class="card-header">{{ $rsp->namamakanan }}
14          </div>
15          <center>
16              <h5 class="card-title">{{ $rsp->deskripsi }}
17              </h5>
18          </center>
19          <p class="card-text" style="text-align:center">{{ $rsp->bahandancara }}
20          </p>
21          <p class="card-text">
22              <small>Created at : {{ $rsp->created_at }}
23              </small>
24          </p>
25          <a href="resep/edit/{{ $rsp->id }}" class="btn btn-primary">Edit
26          </a>
27          <a href="resep/delete/{{ $rsp->id }}" class="btn btn-secondary">Hapus
28          </a>
29      </div>
30  </div>
31  @endforeach
32  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C
33  </script>
```

27. Maka akan tampil sebagai berikut :
Read Data :



28. Untuk mengedit/update data maka kita harus membuat route pada routes/web.php
Seperti berikut :

```
Route::get('/resep/edit/{id}', 'ResepController@edit');
```

29. Serta menambahkan kode pada ResepControllernya seperti berikut :

```
34 public function update($id, Request $request)
35 {
36     $data_resep = \App\kumpulanresep::find($id);
37     $data_resep->namamakanan = $request->namamakanan;
38     $data_resep->deskripsi = $request->deskripsi;
39     $data_resep->bahandancara = $request->bahandancara;
40     $data_resep->save();
41     return redirect('/resep');
42 }
```


30. Juga kita harus membuat Form Pengeditan seperti berikut :

← → ↻ ⓘ 127.0.0.1:8000/resep/edit/1 🔍 ☆

Edit Resepmu Disii

EDIT RESEP

NAMA MAKANAN

DESKRIPSI

Bahan dan Cara

[Back](#) [Ubah Data](#)

31. Dalam CRUD tentunya kita juga dapat menambahkan data, kita harus menambahkan kode pada ResepController :

```
14  
15     public function add()  
16     {  
17         return view ('resep.addresep');  
18     }
```

32. Serta menambahkan routes pada web.php

```
36     Route::post('/resep/update/{id}', 'ResepController@update');
```

33. Jangan lupa juga untuk membuat halaman Formnya :

Yuk Tambah

FORMULIR TAMBAH RESEP

Nama Makanan

Deskripsi Makanan

Bahan Dan Cara

Back

Submit

34. Terakhir dalam CRUD kita bisa mendelete/menghapus data yang kita miliki dengan menambah kode pada ResepController terlebih dahulu :

```
43  
44     public function delete($id)  
45     {  
46         $data_resep = \App\kumpulanresep::find($id);  
47         $data_resep->delete();  
48         return redirect('/resep');  
49     }  
50  
51 }  
52
```

35. Serta menambahkan route pada web.php

```
35 Route::get('/resep/delete/{id}', 'ResepController@delete');
```

36. Kita dapat meng klik tombol hapus maka data akan terhapus :

Cumi Bakar

Cumi Bakar Anti Ribet

1. Cumi Segar, 2. Bumbu Rempah, Caranya olesi cumi dengan bumbu rempah lalu bakar. Sajikan dengan sambal.

Created at :

Edit

Hapus