**Tugas Jobsheet 5 Pemrograman Web Lanjut (Revisi)**



**Muhammad Farrel Caesarian**

**TI-2D/16**

**2341720053**

**POLITEKNIK NEGERI MALANG**

|  |  |
| --- | --- |
| 1. | **Praktikum 1 - Layouting AdminLTE**  Kita download **AdminLTE v3.2.0** yang rilis pada 8 Feb 2022    Seleteh kita berhasil download, kita ekstrak file yang sudah di download ke folder project PWL\_POS/public, kemudian kita ***rename*** folder cukup menjadi adminlte    Selanjutnya kita buka di browser dengan alamat  http://localhost/PWL\_POS/public/adminlte maka akan muncul tampilan seperti  berikut    Kita klik menu Extras > Blank Page, page inilah yang akan menjadi dasar web template    Selanjutnya kita klik kanan halaman Blank Page dan klik *view page source*    Selanjutnya kita copy page source dari halaman Blank Page, kemudia kita *paste* pada **PWL\_POS/resource/view/layouts/template.blade.php** (buat dulu folder layouts dan file template.blade.php)    Pada baris **1-14** file template.blade.php, kita modifikasi    Kemudian kita blok baris **19-153** (baris untuk *element* ***1-header***), lalu kita ***cut***, dan ***paste***-kan di file **PWL\_POS/resource/view/layouts/header.blade.php** (buat dulu file header.blade.php jika belum ada). Sehingga tampilan dari file template.blade.php menjadi seperti berikut    Kita modifikasi baris **25** dan **26** pada template.blade.php    Selanjutnya kita blok baris **31-693** (baris untuk *element* ***2-sidebar***), lalu kita ***cut***, dan ***paste***-kan di file **PWL\_POS/ resource/ view/layouts/ sidebar.blade.php** (buat dulu file sidebar.blade.php jika belum ada). Sehingga tampilan dari file template.blade.php menjadi seperti berikut    Selanjutnya perhatikan baris **87-98** (baris untuk *element* ***5-footer***), lalu kita ***cut***, dan ***paste***-kan di file **PWL\_POS/resource/view/layouts/footer.blade.php** (buat file footer.blade.php jika belum ada). Sehingga tampilan dari file template.blade.php menjadi seperti berikut    Kemudian kita modifikasi file template.blade.php baris **91-100**    Sekarang masuk pada bagian konten. Konten kita bagi menjadi 2, yaitu elemen untuk ***breadcrumb*** dan elemen untuk ***content*.**  Perhatikan file template.blade.php pada baris **38-52** kita jadikan sebagai elemen ***4- breadcrumb***. Kita blok baris **38-52** lalu kita ***cut***, dan ***paste***-kan di file **PWL\_POS/resource/view/layouts/breadcrumb.blade.php** (buat file  breadcrumb.blade.php jika belum ada). Sehingga tampilan dari file  template.blade.php menjadi seperti berikut    Untuk ***content***, kita akan menghapus baris **42-66** pada file template.blade.php. dan kita ganti dengan kode seperti ini @yield('content')    Jangan lupa commit dan push ke github untuk praktikum 1 ini |
| 2. | **Praktikum 2 - Penerapan Layouting:**  Kita buat file controller dengan nama WelcomeController.php    Kita buat file pada PWL\_POS/resources/views/welcome.blade.php    Kita modifikasi file PWL\_POS/resources/views/layouts/breadcrumb.blade.php    Kita modifikasi file PWL\_POS/resources/views/layouts/sidebar.blade.php    Kita tambahkan kode berikut router web.php    Sekarang kita coba jalankan di browser dengan mengetikkan url    Commit Github |
| 3. | **Praktikum 3 – Implementasi jQuery Datatable di AdminLTE :**  1. Kita modifikasi proses CRUD pada tabel m\_user pada praktikum ini  2. Kita gunakan library Yajra-datatable dengan mengetikkan perintah pada CMD   |  |  | | --- | --- | | composer require yajra/laravel-datatables:^10.0 composer require yajra/laravel-datatables-oracle | Atau |     Kita modifikasi route web.php untuk proses CRUD user    Kita buat atau modifikasi penuh untuk UserController.php. Kita buat fungsi **index()** untuk menampilkan halaman awal user    **Lalu kita buat view pada PWL/resources/views/user/index.blade.php**    **Kemudian kita modifikasi file template.blade.php untuk menambahkan library jquery datatables dari template AdminLTE yang kita download dan berada di folder public**    **Untuk bisa menangkap request data untuk datatable, kita buat fungsi list() pada UserController.php seperti berikut**    **Sekarang coba jalankan browser, dan klik menu Data User..!!! perhatikan dan amati apa yang terjadi.**    **Selanjutnya kita modifikasi UserController.php untuk form tambah data user**    **Kemudian untuk bisa menng-*handle* data yang akan disimpan ke database, kita buat fungsi store() di UserController.php**    Sekarang coba kalian buka form tambah data user dengan klik tombol tambah. Amati dan pelajari..!!!      Selanjutnya, kita masuk pada bagian menampilkan detail data user (klik tombol ) pada halaman user. Route yang bertugas untuk menangkap request detail adalah    Jadi kita buat/modifikasi fungsi show() pada UserController.php seperti berikut    **Kemudian kita buat *view* di PWL/resources/views/user/show.blade.php**    Sekarang kalian coba untuk melihat detail data user di browser, dan coba untuk mengetikkan id yang salah contoh http://localhost/PWL/public/user/100 amati apa yang terjadi, dan laporkan!!    Selanjutnya, kita masuk pada bagian untuk memodifikasi data user. Route yang bertugasuntuk menangkap request edit adalah    Jadi kita buat fungsi **edit()** dan **update()** pada UserController.php    Selanjutnya, kita buat *view* untuk melakukan proses edit data user di  PWL/resources/views/user/edit.blade.php    Sekarang kalian coba untuk mengedit data user di browser, amati, pahami, dan laporkan!      Selanjutnya kita akan membuat penanganan untuk tombol hapus. Router web.php yang berfungsi untuk menangkap request hapus dengan method DETELE adalah  Route::delete('/{id}', [UserController::class, 'destroy'])  Jadi kita buat fungsi **destroy()** pada UserController.php    Kemudian jalankan browser untuk menghapus salah satu data user. Amati dan laporkan!      Jangan lupa commit dan push ke github PWL\_POS kalian |
| 4. | **Praktikum 4 – Implementasi *Filtering* Datatables:**  Kita modifikasi fungsi **index()** di UserController.php untuk menambahkan data yang ingin dijadikan kategori untuk data ***filtering***    Kemudian kita modifikasi view untuk menampilkan data filtering pada  PWL/resources/views/user/index.blade.php    Selanjutnya, tetap pada view index.blade.php, kita tambahkan kode berikut pada deklarasi ajax di datatable. Kode ini digunakan untuk mengirimkan data untuk filtering      Tahapan akhir adalah memodifikasi fungsi **list()** pada UserController.php yang  digunakan untuk menampilkan data pada datatable    Hasil:    Selamat, sekarang Laravel Starter Code sudah ada filtering dan searching data. Starter Code sudah bisa digunakan dalam membangun sebuah sistem berbasis website.  Jangan lupa commit dan push ke github PWL\_POS kalian |
| 5. |  |
| 6. |  |