

LAPORAN PRAKTIKUM

JOBSHEET 5.2 : BLADE VIEW, WEB TEMPLATING (ADMINLTE), DATATABLES

MATA KULIAH PEMROGRAMAN WEB LANJUT

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Disusun oleh :

Nama : Vita Eka Saraswati

NIM : 2341760082

No. Absen : 29

JURUSAN TEKNOLOGI INFORMASI

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

POLITEKNIK NEGERI MALANG

2025



Nama : Vita Eka Saraswati
NIM/Kelas : 2341760082 / SIB 2A
No. Absen : 29
Mata Kuliah : Pemrograman Web Lanjut (PWL)

JOBSHEET 05

Blade View, Web Templating(AdminLTE), Datatables

Sesuai dengan **studi Kasus PWL.pdf**.

Jadi project Laravel 10 kita masih sama dengan menggunakan repositori **PWL_POS**.

Project PWL_POS akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

A. Blade View

Laravel Blade adalah template engine yang digunakan secara default oleh Laravel untuk mempermudah pembuatan tampilan (view) dalam aplikasi web. Blade menawarkan fitur-fitur yang powerful dan mudah digunakan untuk memisahkan logika bisnis dari tampilan antarmuka pengguna (UI). Berikut adalah penjelasan lengkap tentang Laravel Blade View:

1. Apa Itu Blade?

Blade adalah template engine yang disertakan dengan Laravel. Dengan Blade, Anda bisa menulis sintaks HTML yang terintegrasi dengan PHP dengan cara yang lebih bersih dan terstruktur. Meskipun menggunakan Blade, semua file template Blade di Laravel tetap memiliki ekstensi *.blade.php*.

2. Keuntungan Menggunakan Blade



- **Inheritance (Pewarisan Template):** Blade memungkinkan Anda untuk membuat layout dasar yang bisa di-extend oleh halaman lain. Ini mempermudah pembuatan struktur halaman yang konsisten.
- **Komponen:** Anda dapat membuat komponen yang reusable untuk elemen UI yang sering digunakan.
- **Control Structures:** Blade mendukung struktur kontrol seperti if, for, while, dan foreach yang mirip dengan sintaks PHP, namun dengan tampilan yang lebih bersih.
- **Tidak Ada Overhead:** Semua kode Blade dikompilasi menjadi PHP biasa, jadi tidak ada overhead tambahan.

3. Sintaks Dasar Blade

Blade memiliki sintaks yang sangat sederhana dan intuitif. Berikut adalah beberapa contoh penggunaan sintaks Blade:

- **Menampilkan Variabel:**

```
{{ $name }}
```

Blade akan secara otomatis melindungi (escape) output untuk mencegah serangan XSS (Cross-Site Scripting).

- **Struktur Kontrol:**

```
@if($condition)
    <p>Condition is true</p>
@endif
```

```
@foreach($users as $user)
    <li>{{ $user->name }}</li>
@endforeach
```

- **Komentar**

```
{{!-- Ini adalah komentar yang tidak akan terlihat dalam output HTML --}}
```



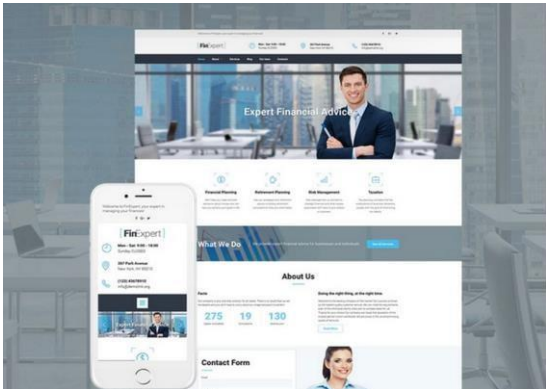
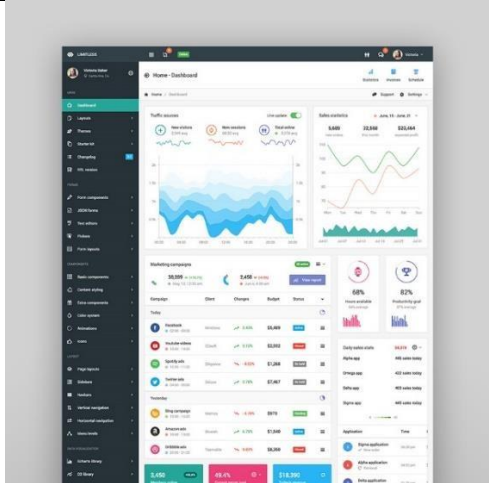
B. Web Template

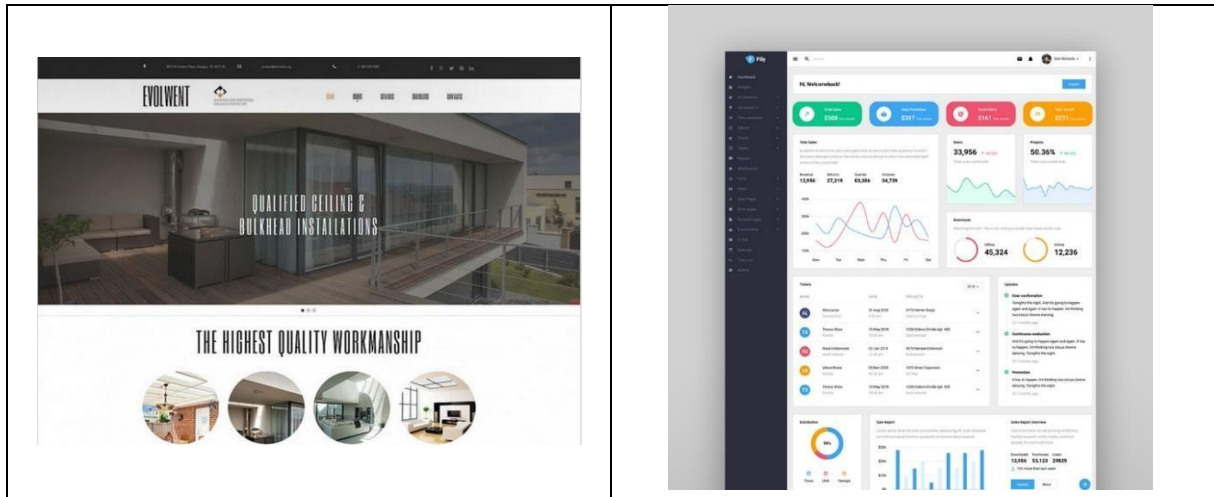
Merupakan suatu file yang sudah memiliki desain tertentu sehingga, kita tinggal memodifikasi bagian-bagian tertentu saja untuk mendapatkan hasil yang sudah bagus. Dengan menggunakan template yang sesuai dengan kebutuhan, kita bisa mengurangi waktu pembuatan suatu website secara drastis.

Secara umum kita bisa menggunakan template website yang sudah ada di internet untuk mempermudah kita dalam membuat website. Ada banyak sekali template yang ada di internet dan kita ambil satu contoh template untuk halaman administrator, seperti **AdminLTE** template. **AdminLTE** adalah salah satu template yang sering digunakan oleh web developer sebagai template backend/admin pada proyek yang sering dikerjakan. Template ini dibuat menggunakan framework bootstrap yang merupakan framework CSS yang paling banyak digunakan di kalangan web desainer

Template AdminLTE dapat di download di <https://adminlte.io>

Contoh Web template

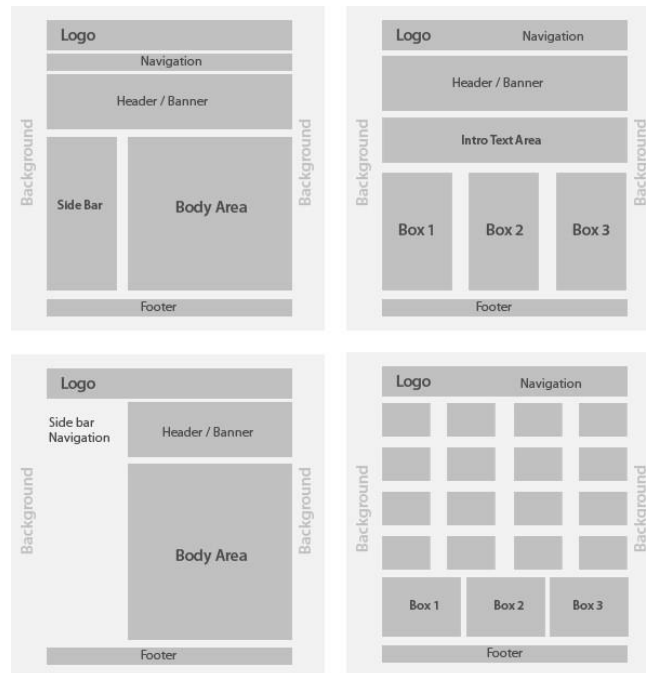
Frontend Template	Backend Template
	



A. Web Layout

Merupakan sekumpulan elemen atau variabel desain yang terkait dengan media tertentu untuk mendukung konsep pembuatan website. Pada dasarnya tujuan penggunaan layout website adalah untuk menggabungkan elemen yang digunakan untuk membangun website. Selain elemen di dalamnya, layout website juga harus memuat beberapa elemen, mulai dari header, navigasi, sidebar, hingga footer. Untuk memahami perbedaan elemen sebuah layout website, berikut penjelasan singkatnya

- **Header** → Pada bagian header, layout designer dapat mengisi bagian ini dengan logo website, navigasi situs, ikon media sosial, dan menu pencarian.
- **Navigasi** → Navigasi dapat dipahami sebagai panduan. Di website, navigasi dapat berkisar dari menu yang muncul di bagian atas halaman web hingga menu pendukung lainnya yang biasa ditemukan di bawah situs.
- **Body/Konten** → Selain itu, terdapat elemen body/konten yang biasanya diisi dengan informasi produk, fitur produk, dan deskripsi produk yang dijual.
- **Sidebar** → elemen ini merupakan elemen yang berada di samping kanan/kiri sebuah website. Element ini kadang sudah tidak dipakai, terutama pada Web Frontend. Namun, sidebar masih digunakan di banyak halaman artikel, yang kemudian akan diisi dengan informasi produk, produk terpopuler, dan navigasi tambahan.
- **Footer** → Elemen terakhir adalah footer. Footer merupakan elemen yang berada dibawah sendiri sebagai menu pendukung atau sebagai identitas dari sebuah website.



C. Layouting AdminLTE

Pada kali ini kita akan melakukan *layouting* template AdminLTE. Tujuan dari *layouting* ini adalah untuk memecah bagian-bagian dari template adminLTE menjadi berbagai elemen. Hal ini dapat mempermudah kita dalam membuat website karena kita bisa menggunakan berulang-ulang elemen yang sudah kita buat nantinya.

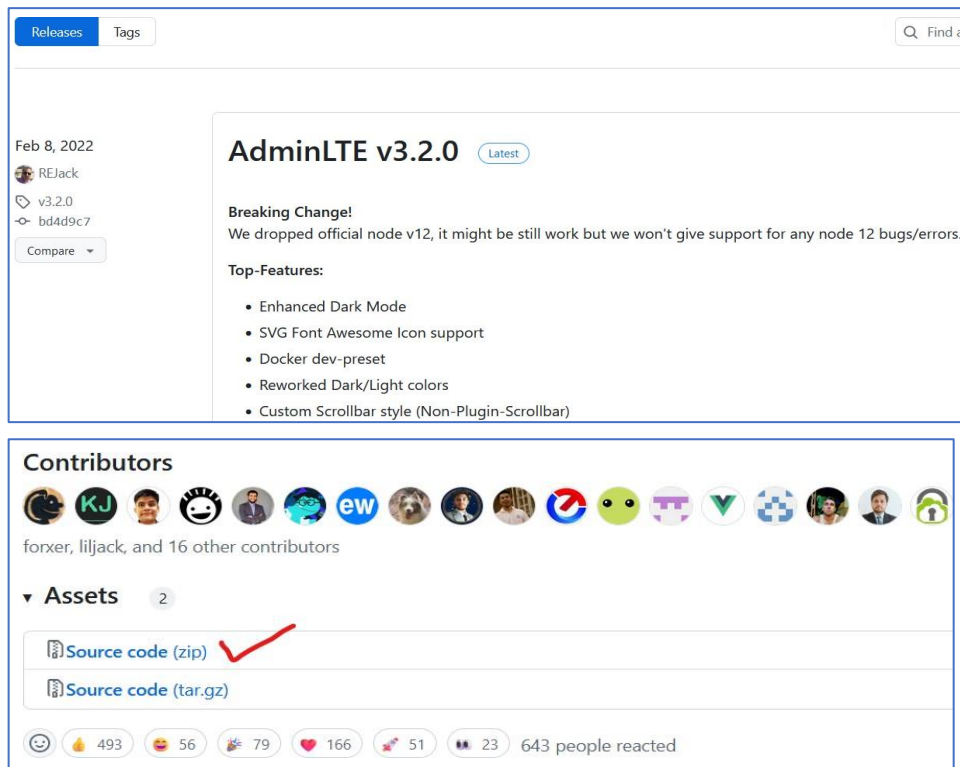
Pada layouting kali ini, kita membutuhkan **Blade Template Engine** dari Laravel untuk memecah web menjadi beberapa elemen. Komponen blade yang kita butuhkan untuk proses layouting pada pertemuan kali ini adalah

- `@include()`
- `@extend()`
- `@section()`
- `@push()`
- `@yield()`
- `@stack()`

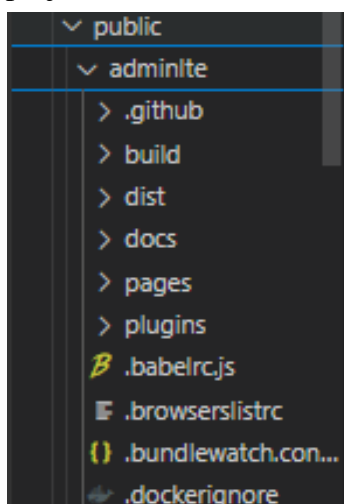


Praktikum 1 - Layouting AdminLTE:

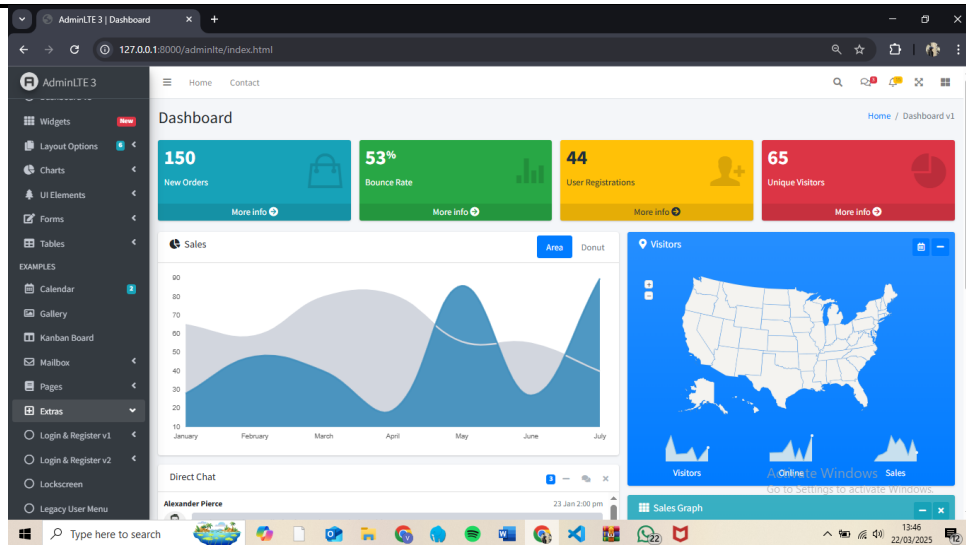
1. Kita download **AdminLTE v3.2.0** yang rilis pada 8 Feb 2022



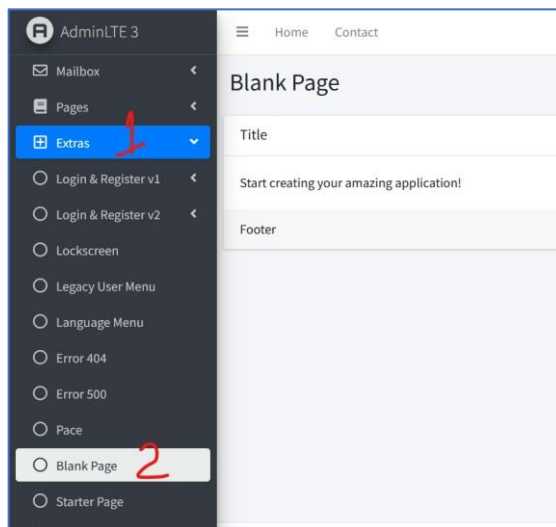
2. Setelah kita berhasil download, kita ekstrak file yang sudah di download ke folder project **PWL_POS/public**, kemudian kita **rename** folder cukup menjadi **adminlte**



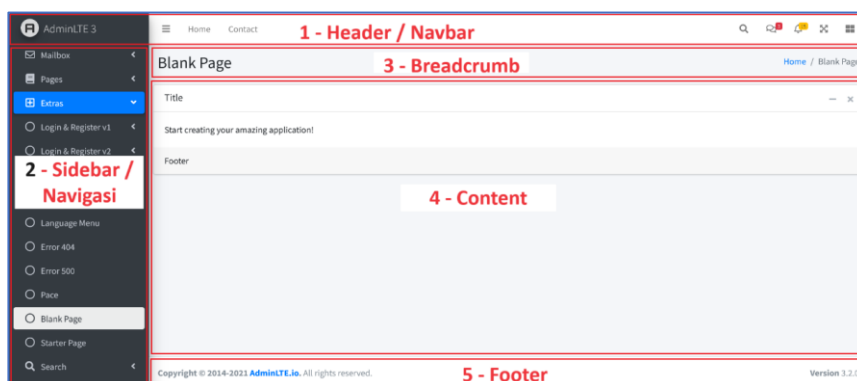
3. Selanjutnya kita buka di browser dengan alamat http://localhost/PWL_POS/public/adminlte maka akan muncul tampilan seperti berikut



1. Kita klik menu **Extras > Blank Page**, page inilah yang akan menjadi dasar web template

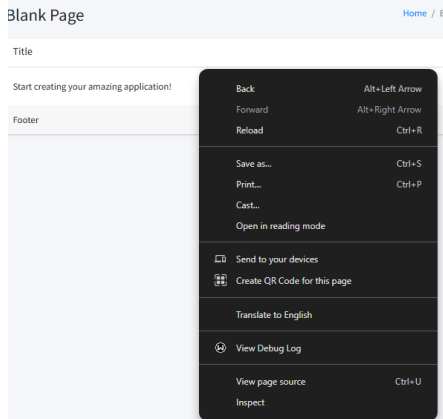


2. Dari sini kita bisa melakukan layouting halaman Blank Page ini menjadi 4 element seperti pada gambar berikut





3. Selanjutnya kita klik kanan halaman **Blank Page** dan klik *view page source*



4. Selanjutnya kita copy page source dari halaman **Blank Page**, kemudian kita *paste* pada **PWL_POS/resource/view/layouts/template.blade.php** (buat dulu folder **layouts** dan file **template.blade.php**)

```
template.blade.php X
Jobsheet5.2 > resources > views > layouts > template.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>AdminLTE 3 | Blank Page</title>
7
8 <!-- Google Font: Source Sans Pro -->
9 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
10 <!-- Font Awesome -->
11 <link rel="stylesheet" href="../../plugins/fontawesome-free/css/all.min.css">
12 <!-- Theme style -->
13 <link rel="stylesheet" href="../../dist/css/adminlte.min.css">
14 </head>
15 <body class="hold-transition sidebar-mini">
16 <!-- Site wrapper -->
17 <div class="wrapper">
18 <!-- Navbar -->
19 <nav class="main-header navbar navbar-expand navbar-white navbar-light">
20 <!-- Left navbar links -->
21 <ul class="navbar-nav">
```

5. File **layouts/template.blade.php** adalah file utama untuk templating website

6. Pada baris **1-14** file **template.blade.php**, kita modifikasi menjadi

```
Minggu5.2 > Jobsheet5.2 > resources > views > layouts > template.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>{{ config(key: 'app.name', default: 'PWL Laravel Starter Code') }}</title>
7
8 <meta name="csrf-token" content="{{ csrf_token() }}"> <!-- Untuk mengirimkan token Laravel CSRF pada setiap request ajax -->
9
10 <!-- Google Font: Source Sans Pro -->
11 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
12 <!-- Font Awesome -->
13 <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/fontawesome-free/css/all.min.css') }}">
14 <!-- Datatables -->
15 <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}">
16 <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-responsive/css/responsive.bootstrap.min.css') }}">
17 <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-buttons/css/buttons.bootstrap.min.css') }}">
18 <!-- Theme style -->
19 <link rel="stylesheet" href="{{ asset(path: 'adminlte/dist/css/adminlte.min.css') }}">
20
21 @stack('css') <!-- Digunakan untuk memanggil custom css dari perintah push('css') pada masing-masing view -->
22 </head>
```

7. Kemudian kita blok baris **19-153** (baris untuk **element I-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

```
template.blade.php U x header.blade.php U
Minggu5.2 > Jobsheet5.2 > resources > views > layouts > template.blade.php
2 <html lang="en">
3 <head>
4
5 <!-- Google Font: Source Sans Pro -->
6 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700,900">
7 <!-- Font Awesome -->
8 <link rel="stylesheet" href="{{ asset(path: 'plugins/fontawesome/css/all.min.css') }}">
9 <!-- Theme style -->
10 <link rel="stylesheet" href="{{ asset(path: 'dist/css/adminlte.min.css') }}">
11 </head>
12 <body class="hold-transition sidebar-mini">
13 <!-- Site wrapper -->
14 <div class="wrapper">
15 <!-- Navbar -->
16 @include('layouts.header')
17 <!-- /.navbar -->
18
19 <!-- Main Sidebar Container -->
20 <aside class="main-sidebar sidebar-dark-primary elevation-4">
```

Baris 19 adalah komponen Blade untuk memanggil elemen `layouts/header.blade.php` agar menjadi satu dengan `template.blade.php` saat di-render nanti.

8. Kita modifikasi baris 25 - 27 pada `template.blade.php`

```
22 <!-- Main Sidebar Container -->
23 <aside class="main-sidebar sidebar-dark-primary elevation-4">
24 <!-- Brand Logo -->
25 <a href="{{ url(path: '/') }}" class="brand-link">
26 
27 <span class="brand-text font-weight-light">PWL - Starter Code</span>
28 </a>
```

9. 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

```
22 <!-- Main Sidebar Container -->
23 <aside class="main-sidebar sidebar-dark-primary elevation-4">
24 <!-- Brand Logo -->
25 <a href="{{ url(path: '/') }}" class="brand-link">
26 
27 <span class="brand-text font-weight-light">PWL - Starter Code</span>
28 </a>
29
30 <!-- Sidebar -->
31 @include('layouts.sidebar')
32 <!-- /.sidebar -->
33 </aside>
```

10. 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

```
82 </section>
83 <!-- /.content -->
84 </div>
85 <!-- /.content-wrapper -->
86
87 @include('layouts.footer')
88 </div>
89 <!-- /.wrapper -->
90
91 <!-- jQuery -->
92 <script src="{{ asset(path: 'plugins/jquery/jquery.min.js') }}"></script>
```



11. Kemudian kita modifikasi file `template.blade.php` baris **91-100**

```

91 <!-- jQuery -->
92 <script src="{{asset(path: 'adminlte/plugins/jquery/jquery.min.js')}}"></script>
93 <!-- Bootstrap 4 -->
94 <script src="{{asset(path: 'adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js')}}"></script>
95 <!-- AdminLTE App -->
96 <script src="{{asset(path: 'adminlte/dist/js/adminlte.min.js')}}"></script>
97 </body>
98 </html>

```

12. Sekarang masuk pada bagian konten. Konten kita bagi menjadi 2, yaitu elemen untuk *breadcrumb* dan elemen untuk *content*.

13. 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

```

30 <!-- Sidebar -->
31 @include(view: 'layouts.sidebar')
32 <!-- /.sidebar -->
33 </aside>
34
35 <!-- Content Wrapper. Contains page content -->
36 <div class="content-wrapper">
37   <!-- Content Header (Page header) -->
38   @include(view: 'layouts.breadcrumb')
39
40   <!-- Main content -->
41   <section class="content"> ...
42     <!-- Default box -->
43     <div class="card">
44       <div class="card-header">
45         <h3 class="card-title">Title</h3>
46

```

14. Layout terakhir adalah pada bagian konten. Layout untuk konten bisa kita buat dinamis, sesuai dengan apa yang ingin kita sajikan pada web yang kita bangun.

15. Untuk *content*, kita akan menghapus baris **42-66** pada file `template.blade.php`. dan kita ganti dengan kode seperti ini `@yield('content')`

16. Hasil akhir pada file utama `layouts/template.blade.php` adalah seperti berikut



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
 Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
 Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>

```

15 <body class="hold-transition sidebar-mini">
16 <!-- Site wrapper -->
17 <div class="wrapper">
18   <!-- Navbar -->
19   @include(view: 'layouts.header')
20   <!-- /.navbar -->
21
22   <!-- Main Sidebar Container -->
23   <aside class="main-sidebar sidebar-dark-primary elevation-4">
24     <!-- Brand Logo -->
25     <a href="{{url(path: '/')}}" class="brand-link">
26       
27       <span class="brand-text font-weight-light">PWL - Starter Code</span>
28     </a>
29
30     <!-- Sidebar -->
31     @include(view: 'layouts.sidebar')
32     <!-- /.sidebar -->
33   </aside>
34
35   <!-- Content Wrapper. Contains page content -->
36   <div class="content-wrapper">
37     <!-- Content Header (Page header) -->
38     @include(view: 'layouts.breadcrumb')
39
40     <!-- Main content -->
41     <section class="content">
42       @yield('content')
43     </section>
44     <!-- /.content -->
45   </div>
46   <!-- /.content-wrapper -->
47
48   @include(view: 'layouts.footer')
49 </div>

```

Activate Windows

17. Selamat kalian sudah selesai dalam melakukan layouting website di laravel.
18. Jangan lupa commit dan push ke github untuk praktikum 1 ini



Praktikum 2 - Penerapan Layouting:

Sekarang kita akan mencoba melakukan penerapan terhadap layouting yang sudah kita lakukan.

1. Kita buat file controller dengan nama `WelcomeController.php`

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5.2\Jobsheet5.2> php artisan make:controller WelcomeController

INFO Controller [F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5.2\app\Http\Controllers\WelcomeController.php] created successfully.

WelcomeController.php u x
Minggu5.2 > Jobsheet5.2 > app > Http > Controllers > WelcomeController.php > ...
1  <?php
2  namespace App\Http\Controllers;
3
4  class WelcomeController extends Controller
5  {
6      0 references | 0 overrides
7      public function index(): Factory|View
8      {
9          $breadcrumb = (object) [
10             'title' => 'Selamat Datang',
11             'list' => ['Home', 'Welcome']
12         ];
13         $activeMenu = 'dashboard';
14
15         return view(view: 'welcome', data: ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu]);
16     }
17 }
```

2. Kita buat file pada `PWL_POS/resources/views/welcome.blade.php`

```
welcome.blade.php M x
Minggu5.2 > Jobsheet5.2 > resources > views > welcome.blade.php
1  @extends('layouts.template')
2
3  @section('content')
4
5      <div class="card">
6          <div class="card-header">
7              <h3 class="card-title">Halo, apakabar!!!</h3>
8              <div class="card-tools"></div>
9          </div>
10         <div class="card-body">
11             Selamat datang semua, ini adalah halaman utama dari aplikasi ini.
12         </div>
13     </div>
14
15 @endsection
```



3. Kita modifikasi file `PWL_POS/resources/views/layouts/breadcrumb.blade.php`

```

Minggu5.2 > Jobsheet5.2 > resources > views > layouts > breadcrumb.blade.php
1 <section class="content-header">
2   <div class="container-fluid">
3     <div class="row mb-2">
4       <div class="col-sm-6"><h1>{{ $breadcrumb->title }}</h1></div>
5       <div class="col-sm-6">
6         <ol class="breadcrumb float-sm-right">
7           @foreach($breadcrumb->list as $key => $value)
8             @if($key == count($value: $breadcrumb->list) - 1)
9               <li class="breadcrumb-item active">{{ $value }}</li>
10              @else
11                <li class="breadcrumb-item">{{ $value }}</li>
12              @endif
13            @endforeach
14          </ol>
15        </div>
16      </div>
17    </div>
18  </section>
  
```

4. Kita modifikasi file `PWL_POS/resources/views/layouts/sidebar.blade.php`

```

Minggu5.2 > Jobsheet5.2 > resources > views > layouts > sidebar.blade.php
1 <div class="sidebar">
2   <!-- Sidebar Search Form -->
3   <div class="form-inline mt-2">
4     <div class="input-group" data-widget="sidebar-search">
5       <input class="form-control form-control-sidebar" type="search" placeholder="Search" aria-label="Search">
6       <div class="input-group-append">
7         <button class="btn btn-sidebar">
8           <i class="fas fa-search fa-fw"></i>
9         </button>
10      </div>
11    </div>
12  </div>
13
14  <!-- Sidebar Menu -->
15  <nav class="mt-2">
16    <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">
17      <li class="nav-item">
18        <a href="{{ url(path: '/') }}" class="nav-link {{ ($activeMenu == 'dashboard') ? 'active' : '' }}">
19          <i class="nav-icon fas fa-tachometer-alt"></i>
20          <p>Dashboard</p>
21        </a>
22      </li>
23
24      <li class="nav-item">
25        <a href="{{ url(path: '/level') }}" class="nav-link {{ ($activeMenu == 'level') ? 'active' : '' }}">
26          <i class="nav-icon fas fa-layer-group"></i>
27          <p>Level User</p>
28        </a>
29      </li>
30
31      <li class="nav-item">
32        <a href="{{ url(path: '/user') }}" class="nav-link {{ ($activeMenu == 'user') ? 'active' : '' }}">
33          <i class="nav-icon far fa-user"></i>
34          <p>Data User</p>
35        </a>
36      </li>
37
38      <li class="nav-item">
39        <a href="{{ url(path: '/kategori') }}" class="nav-link {{ ($activeMenu == 'kategori') ? 'active' : '' }}">
40          <i class="nav-icon far fa-bookmark"></i>
41          <p>Kategori Barang</p>
42        </a>
43      </li>
44
45      <li class="nav-item">
46        <a href="{{ url(path: '/barang') }}" class="nav-link {{ ($activeMenu == 'barang') ? 'active' : '' }}">
47          <i class="nav-icon far fa-list-alt"></i>
48          <p>Data Barang</p>
49        </a>
50      </li>
51    </ul>
  
```



```

52 | <li class="nav-header">Data Transaksi</li>
53 | <li class="nav-item">
54 |   <a href="{{ url(path: '/stok') }}" class="nav-link {{ ($activeMenu == 'stok')? 'active' : '' }}">
55 |     <i class="nav-icon fas fa-cubes"></i>
56 |     <p>Stok Barang</p>
57 |   </a>
58 | </li>
59 | <li class="nav-item">
60 |   <a href="{{ url(path: '/barang') }}" class="nav-link {{ ($activeMenu == 'penjualan')? 'active' : '' }}">
61 |     <i class="nav-icon fas fa-cash-register"></i>
62 |     <p>Transaksi Penjualan</p>
63 |   </a>
64 | </li>
65 | </ul>
66 | </nav>
67 | </div>

```

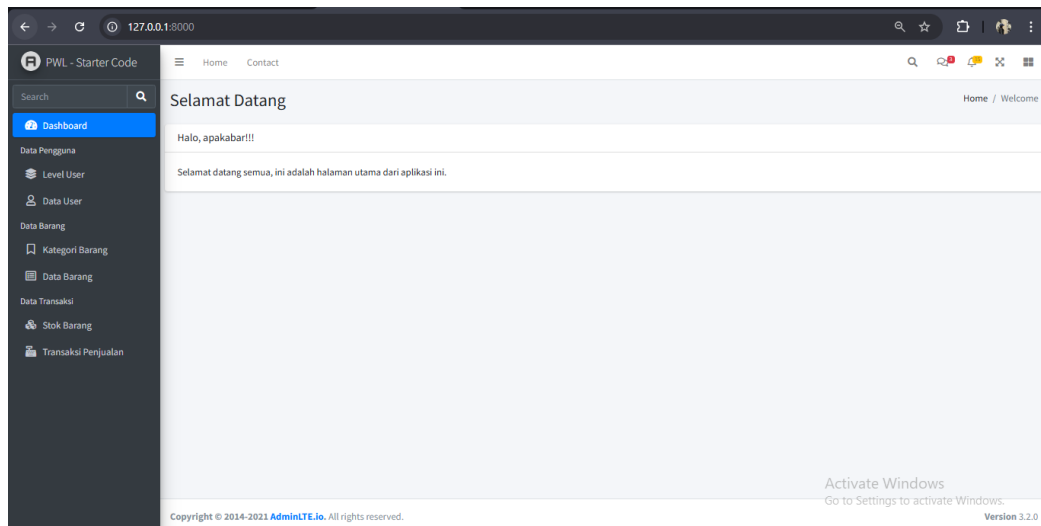
5. Kita tambahkan kode berikut router web.php

```

11 | use App\Http\Controllers\WelcomeController;
39 | //PRAKTIKUM 5.2
40 | //praktikum 2 - no 5
41 | Route::get('/', [WelcomeController::class, 'index']);

```

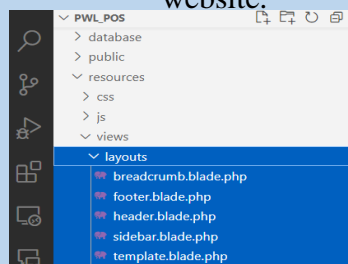
6. Sekarang kita coba jalankan di browser dengan mengetikkan url localhost atau php artisan serve



7. Jangan lupa commit dan push ke github PWL_POS kalian

INFO

Terdapat 5 file utama pada folder *layouts* yang digunakan dalam membangun sebuah website, yaitu **template**, **header**, **sidebar**, **breadcrumb**, **footer**. Kalian bisa memodifikasi ke-5 file tersebut dalam membangun website.





D. jQuery Datatables di Laravel

jQuery DataTables adalah sebuah plugin jQuery yang sangat populer dan powerful untuk menampilkan dan mengelola data dalam bentuk tabel di halaman web. Untuk menampilkan banyak data, kita bisa menampilkan data tersebut dalam format tabel. DataTable ini merupakan plugin **jQuery** yang dibuat untuk mengelola data informasi dalam bentuk grid / table.

INFO

AdminLTE juga sudah menerapkan library Datatable dalam template nya. Datatable pada AdminLTE dapat kalian cek di

<http://localhost/PWL/public/adminlte/pages/tables/data.html>

Pada praktikum kali ini kita **tidak menggunakan Vite/NodeJS** dalam mengelola datatables, kita cukup menggunakan jQuery datatables bawaan AdminLTE dan library Yajra *laravel-datatables*.

E. Implementasi jQuery Datatables di Laravel

Kita akan menerapkan menggunakan jQuery datatable dari AdminLTE dengan Laravel. Dalam penerapan ini, kita menggunakan library Yajra *laravel-datatables*.

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` atau
`composer require yajra/laravel-datatables-oracle`

```
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE
yajra/laravel-datatables-buttons ..... DONE
yajra/laravel-datatables-editor ..... DONE
yajra/laravel-datatables-fractal ..... DONE
yajra/laravel-datatables-html ..... DONE
yajra/laravel-datatables-oracle ..... DONE

86 packages you are using are looking for funding.
Use the "composer fund" command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force
[INFO] No publishable resources for tag [laravel-assets].
Found 1 security vulnerability advisory affecting 1 package.
Run "composer audit" for a full list of advisories.
```

3. Kita modifikasi route `web.php` untuk proses CRUD user

```
11 use App\Http\Controllers\UserController; //praktikum3 minggu5.2 no 2

42 //praktikum 3 - no 2
43 Route::group(['prefix' => 'user'], function () { void [
44
45     Route::get('/', [UserController::class, 'index']); // menampilkan halaman awal user
46     Route::post('/list', [UserController::class, 'list']); // menampilkan data user dalam bentuk json untuk datatables
47     Route::get('/create', [UserController::class, 'create']); // menampilkan halaman form tambah user
48     Route::post('/', [UserController::class, 'store']); // menyimpan data user baru
49     Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
50     Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
51     Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
52     Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
53 });
```




4. Kita buat atau modifikasi penuh untuk `UserController.php`. Kita buat fungsi `index()` untuk menampilkan halaman awal user

```
Minggu5.2 > Jobsheet5.2 > app > Http > Controllers > UserController.php > PHP Intelephense > UserController
1  <?php
2  namespace App\Http\Controllers;
3
4  use App\Models\LevelModel;
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Yajra\DataTables\Facades\DataTables;
8
9  class UserController extends Controller
10 {
11     //Menampilkan halaman awal user
12     public function index(): Factory|View
13     {
14         //week 5.2 - praktikum 3 - no 4
15         $breadcrumb = (object) [
16             'title' => 'User',
17             'list' => ['Home', 'User']
18         ];
19
20         $page = (object) [
21             'title' => 'Daftar user yang terdaftar dalam sistem',
22         ];
23
24         $activeMenu = 'user'; // set menu yang aktif
25
26         return view('user.index', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
27     }
28 }
29
```

5. Lalu kita buat view pada `PWL/resources/views/user/index.blade.php`

```
Minggu5.2 > Jobsheet5.2 > resources > views > user > index.blade.php
1  @extends('layouts.template')
2
3  @section('content')
4  <div class="card card-outline card-primary">
5      <div class="card-header">
6          <h3 class="card-title">{{ $page->title }}</h3>
7          <div class="card-tools">
8              <a class="btn btn-sm btn-primary mt-1" href="{{ url(path: 'user/create') }}">Tambah</a>
9          </div>
10     </div>
11     <div class="card-body">
12         <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
13             <thead>
14                 <tr>
15                     <th>ID</th>
16                     <th>Username</th>
17                     <th>Nama</th>
18                     <th>Level Pengguna</th>
19                     <th>Aksi</th>
20                 </tr>
21             </thead>
22         </table>
23     </div>
24 </div>
25 @endsection
26
27 @push('css')
28 @endpush
29
30 @push('js')
31 <script>
32     $(document).ready(function() {
33         var dataUser = $('#table_user').DataTable({
34             // serverSide: true, jika ingin menggunakan server side processing serverSide: true,
35             ajax: {
36                 "url": "{{ url(path: 'user/list') }}",
37                 "dataType": "json",
38                 "type": "POST"
39             },
36
```



```

40         columns: [
41             {
42                 // nomor urut dari laravel datatable addIndexColumn()
43                 data: "DT_RowIndex",
44                 className: "text-center",
45                 orderable: false,
46                 searchable: false
47             },
48             {
49                 data: "username",
50                 className: "",
51                 // orderable: true, jika ingin kolom ini bisa diurutkan
52                 orderable: true,
53                 // searchable: true, jika ingin kolom ini bisa dicari
54                 searchable: true
55             },
56             {
57                 data: "nama",
58                 className: "",
59                 orderable: true,
60                 searchable: true
61             },
62             {
63                 // mengambil data level hasil dari ORM berelasi
64                 data: "level.level_nama",
65                 className: "",
66                 orderable: false,
67                 searchable: false
68             },
69             {
70                 data: "aksi",
71                 className: "",
72                 orderable: false,
73                 searchable: false
74             }
75         ]
76     });
77 });
78 </script>
79 @endpush

```

6. Kemudian kita modifikasi file `template.blade.php` untuk menambahkan library jquery datatables dari template AdminLTE yang kita download dan berada di folder `public`

```

Minggu52 > Jobsheet5.2 > resources > views > layouts > template.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>{{ config(key: 'app.name', default: 'PWL Laravel Starter Code') }}</title>
7     <meta name="csrf-token" content="{{ csrf_token() }}"> <!-- Untuk mengirimkan token Laravel CSRF pada setiap request ajax -->
8
9     <!-- Google Font: Source Sans Pro -->
10    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,700&display=block">
11    <!-- Font Awesome -->
12    <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/fontawesome-free/css/all.min.css') }}">
13    <!-- DataTables -->
14    <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}">
15    <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-responsive/css/dataTables.bootstrap4.min.css') }}">
16    <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-responsive/css/dataTables.bootstrap4.min.css') }}">
17    <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-buttons/css/dataTables.bootstrap4.min.css') }}">
18    <!-- Theme style -->
19    <link rel="stylesheet" href="{{ asset(path: 'adminlte/dist/css/adminlte.min.css') }}">
20
21    @stack('css') <!-- Digunakan untuk memanggil custom css dari perintah push('css') pada masing-masing view -->
22 </head>
23
24 <!-- jQuery -->
25 <script src="{{ asset(path: 'adminlte/plugins/jquery/jquery.min.js') }}"></script>
26 <!-- Bootstrap 4 -->
27 <script src="{{ asset(path: 'adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
28 <!-- DataTables & Plugins -->
29 <script src="{{ asset(path: 'adminlte/plugins/datatables/jquery.dataTables.min.js') }}"></script>
30 <script src="{{ asset(path: 'adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}"></script>
31 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
32 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
33 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
34 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
35 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
36 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
37 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
38 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
39 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
40 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
41 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
42 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
43 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
44 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
45 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
46 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
47 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
48 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
49 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
50 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
51 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
52 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
53 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
54 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
55 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
56 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
57 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
58 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
59 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
60 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
61 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
62 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
63 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
64 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
65 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
66 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
67 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
68 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
69 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
70 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
71 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
72 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
73 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
74 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
75 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
76 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
77 <script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.bootstrap4.min.js') }}"></script>
78 <script src="{{ asset(path: 'adminlte/dist/js/adminlte.min.js') }}"></script>
79 <script>
80     //Untuk pengiriman token laravel CSRF pada setiap request AJAX
81     $.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
82 </script>
83 @stack('js') <!-- Digunakan untuk memanggil custom js dari perintah push('js') pada masing-masing view -->
84 </body>
85 </html>

```



7. Untuk bisa menangkap request data untuk datatable, kita buat fungsi `list()` pada `UserController.php` seperti berikut

```

33 // Ambil data user dalam bentuk json untuk datatable
34 1 reference | 0 overrides
35 public function list(Request $request): JsonResponse
36 {
37     $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
38     ->with('level');
39
40     return DataTables::of($users)
41     // Menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
42     ->addIndexColumn()
43     ->addColumn('aksi', function ($user): string { // Menambahkan kolom aksi
44         $btn = '<a href="'.url(path: './user/'. $user->user_id.'" class="btn btn-info btn-sm">Detail</a> ';
45         $btn .= '<a href="'.url(path: './user/'. $user->user_id.'" class="btn btn-warning btn-sm">Edit</a> ';
46         $btn .= '<form class="d-inline-block" method="POST" action="'.url(path: './user/'. $user->user_id.'">
47             .csrf_field()
48             .method_field(method: 'DELETE')
49             .button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\Apakah Anda yakin menghapus data ini?');">Hapus</button>
50         </form>;
51         return $btn;
52     });
53     ->rawColumns(['aksi']) // Memberitahu bahwa kolom aksi adalah HTML
54     ->make(true);
55 }
  
```

8. Sekarang coba jalankan browser, dan klik menu **Data User..!!!** perhatikan dan amati apa yang terjadi.

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staf/Kasir	Staf/Kasir	Detail Edit Hapus
4	customer-1	Customer 1	Pelanggan	Detail Edit Hapus
5	customer-2	Pelanggan Kedua	Pelanggan	Detail Edit Hapus
6	manager_dua	Manager 2	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus

Menampilkan daftar user yang datanya diambil dari database tabel `m_user`

9. Selanjutnya kita modifikasi `UserController.php` untuk form tambah data user

```

56 // Menampilkan halaman form tambah user
57 1 reference | 0 overrides
58 public function create(): Factory|View
59 {
60     $breadcrumb = (object) [
61         'title' => 'Tambah User',
62         'list' => ['Home', 'User', 'Tambah']
63     ];
64
65     $page = (object) [
66         'title' => 'Tambah user baru'
67     ];
68
69     $level = LevelModel::all(); // ambil data level untuk ditampilkan di form
70     $activeMenu = 'user'; // set menu yang sedang aktif
71
72     return view(view: 'user.create', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
73 }
  
```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>



10. Sekarang kita buat form untuk menambah data, kita buat file
 PWL/resources/views/user/create.blade.php

```

Minggus2 > Jobsheets2 > resources > views > user > create.blade.php
1  @extends('layouts.template')
2
3  @section('content')
4  <div class="card card-outline card-primary">
5    <div class="card-header">
6      <h3 class="card-title">{{ $page->title }}</h3>
7      <div class="card-tools"></div>
8    </div>
9    <div class="card-body">
10     <form method="POST" action="{{ url(path: 'user') }}" class="form-horizontal">
11       @csrf
12       <div class="form-group row">
13         <label class="col-1 control-label col-form-label">Level</label>
14         <div class="col-11">
15           <select class="form-control" id="level_id" name="level_id" required>
16             <option value="">- Pilih Level -</option>
17             @foreach($level as $item)
18               <option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
19             @endforeach
20           </select>
21           @error('level_id')
22             <small class="form-text text-danger">{{ $message }}</small>
23           @enderror
24         </div>
25       </div>
26
27       <div class="form-group row">
28         <label class="col-1 control-label col-form-label">Username</label>
29         <div class="col-11">
30           <input type="text" class="form-control" id="username" name="username" value="{{ old(key: 'username') }}" required>
31           @error('username')
32             <small class="form-text text-danger">{{ $message }}</small>
33           @enderror
34         </div>
35       </div>
36
37       <div class="form-group row">
38         <label class="col-1 control-label col-form-label">Nama</label>
39         <div class="col-11">
40           <input type="text" class="form-control" id="nama" name="nama" value="{{ old(key: 'nama') }}" required>
41           @error('nama')
42             <small class="form-text text-danger">{{ $message }}</small>
43           @enderror
44         </div>
45       </div>
46
47       <div class="form-group row">
48         <label class="col-1 control-label col-form-label">Password</label>
49         <div class="col-11">
50           <input type="password" class="form-control" id="password" name="password" required>
51           @error('password')
52             <small class="form-text text-danger">{{ $message }}</small>
53           @enderror
54         </div>
55       </div>
56
57       <div class="form-group row">
58         <label class="col-1 control-label col-form-label"></label>
59         <div class="col-11">
60           <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
61           <a class="btn btn-sm btn-default ml-1" href="{{ url(path: 'user') }}">Kembali</a>
62         </div>
63       </div>
64     </form>
65   </div>
66 </div>
67 @endsection
68
69 @push('css')
70 @endpush
71
72 @push('js')
73 @endpush
  
```

11. Kemudian untuk bisa *menng-handle* data yang akan disimpan ke database, kita buat fungsi `store()` di `UserController.php`

```

75 // Menyimpan data user baru
76 1 reference | 0 overrides
77 public function store(Request $request): mixed|RedirectResponse
78 {
79     $request->validate([
80         // username harus diisi, berupa string, minimal 3 karakter, dan bernilai unik di tabel m_user kolom username
81         'username' => 'required|string|min:3|unique:m_user,username',
82         'nama' => 'required|string|max:100', // nama harus diisi, berupa string, dan maksimal 100 karakter
83         'password' => 'required|min:5', // password harus diisi dan minimal 5 karakter
84         'level_id' => 'required|integer' // level_id harus diisi dan berupa angka
85     ]);
86     UserModel::create([
87         'username' => $request->username,
88         'nama' => $request->nama,
89         'password' => bcrypt(value: $request->password), // password dienkripsi sebelum disimpan
90         'level_id' => $request->level_id
91     ]);
92     return redirect(to: '/user')->with('success', 'Data user berhasil disimpan');
93 }
94
  
```



12. Sekarang coba kalian buka form tambah data user dengan klik tombol tambah.

Saat user mengklik button Tambah maka akan menampilkan form untuk menambahkan user baru. Untuk menyimpan data, user dapat mengklik button Simpan.

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	customer-1	Customer 1	Pelanggan	Detail Edit Hapus
5	customer-2	Pelanggan Kedua	Pelanggan	Detail Edit Hapus
6	manager_dua	Manager 2	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus
10	vita16	Vita	Manager	Detail Edit Hapus

Setelah user menyimpan data user baru, maka data terbaru juga akan ditampilkan pada baris/rows paling bawah seperti gambar di atas.

13. Selanjutnya, kita masuk pada bagian menampilkan detail data user (klik tombol **Detail**) pada halaman user. Route yang bertugas untuk menangkap request detail adalah

```
48 Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
```

14. Jadi kita buat/modifikasi fungsi **show()** pada **UserController.php** seperti berikut

```
96 //Menampilkan detail user
97 2 references | 0 overrides
98 public function show($id): Factory|View
99 {
100     $user = UserModel::find($id);
101     $breadcrumb = (object)[
102         'title' => 'Detail User',
103         'list' => ['Home', 'User', 'Detail']
104     ];
105     $page = (object)[
106         'title' => 'Detail user'
107     ];
108     $activeMenu = 'user'; // set menu yang sedang aktif
109     return view('user.show', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
110 }
111
112
113
```

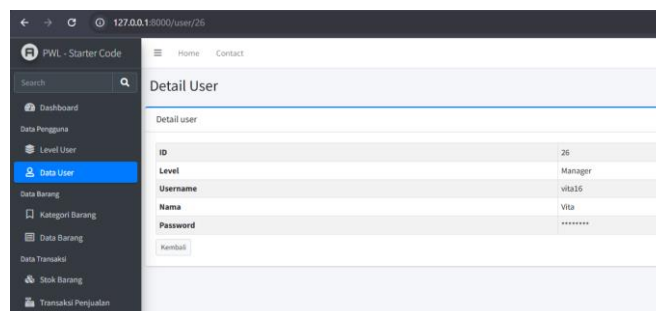


15. Kemudian kita buat *view* di <PWL/resources/views/user/show.blade.php>

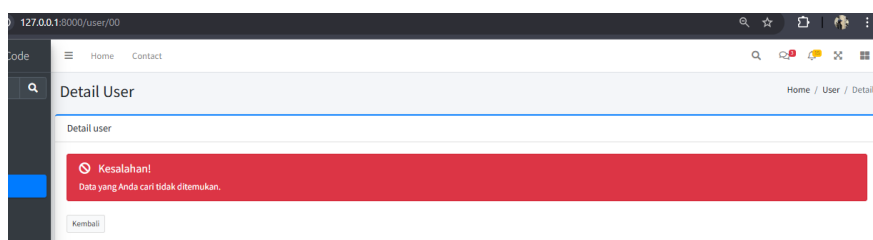
```

1 @extends('layouts.template')
2
3 @section('content')
4 <div class="card card-outline card-primary">
5   <div class="card-header">
6     <h3 class="card-title">{{ $page->title }}</h3>
7     <div class="card-tools"></div>
8   </div>
9   <div class="card-body">
10    @empty($user)
11      <div class="alert alert-danger alert-dismissible">
12        <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
13        Data yang Anda cari tidak ditemukan.
14      </div>
15    @else
16      <table class="table table-bordered table-striped table-hover table-sm">
17        <tr>
18          <th>ID</th>
19          <td>{{ $user->user_id }}</td>
20        </tr>
21        <tr>
22          <th>Level</th>
23          <td>{{ $user->level->level_nama }}</td>
24        </tr>
25        <tr>
26          <th>Username</th>
27          <td>{{ $user->username }}</td>
28        </tr>
29        <tr>
30          <th>Nama</th>
31          <td>{{ $user->nama }}</td>
32        </tr>
33        <tr>
34          <th>Password</th>
35          <td>*****</td>
36        </tr>
37      </table>
38      @empty
39        <a href="{{ url(path: 'user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
40      </div>
41    </div>
42  @endsection
43
44 @push('css')
45 @endpush
46
47 @push('js')
48 @endpush
  
```

16. 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!!!



Jika data user tidak ada dalam database maka tampilannya muncul pesan kesalahan





17. Selanjutnya, kita masuk pada bagian untuk memodifikasi data user. Route yang bertugas untuk menangkap request edit adalah

```
49 Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
50 Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
```

18. Jadi kita buat fungsi `edit()` dan `update()` pada `UserController.php`

```
121 // Menampilkan halaman form edit user
122 1 reference | 0 overrides
123 public function edit(string $id): Factory|View
124 {
125     $user = UserModel::find($id);
126     $level = LevelModel::all();
127
128     $breadcrumb = (object) [
129         'title' => 'Edit User',
130         'list' => ['Home', 'User', 'Edit']
131     ];
132
133     $page = (object) [
134         'title' => 'Edit user'
135     ];
136
137     $activeMenu = 'user'; // set menu yang sedang aktif
138
139     return view(view: 'user.edit', data: [
140         'breadcrumb' => $breadcrumb,
141         'page' => $page,
142         'user' => $user,
143         'level' => $level,
144         'activeMenu' => $activeMenu
145     ]);
146 }
147
148 // Menyimpan perubahan data user
149 1 reference | 0 overrides
150 public function update(Request $request, string $id): mixed|RedirectResponse
151 {
152     $request->validate([
153         // username harus diisi, berupa string, minimal 3 karakter,
154         // dan bernilai unik di tabel t_user kolom username kecuali untuk user dengan id yang sedang diedit
155         'username' => 'required|string|min:3|unique:m_user,username,' . $id . ',user_id',
156         'nama' => 'required|string|max:100', // nama harus diisi, berupa string, dan maksimal 100 karakter
157         'password' => 'nullable|min:5', // password bisa diisi (minimal 5 karakter) dan bisa tidak diisi
158         'level_id' => 'required|integer' // level_id harus diisi dan berupa angka
159     ]);
160
161     UserModel::find($id)->update([
162         'username' => $request->username,
163         'nama' => $request->nama,
164         'password' => $request->password ? bcrypt(value: $request->password) : UserModel::find($id)->password,
165         'level_id' => $request->level_id,
166     ]);
167
168     return redirect(to: '/user')->with('success', 'Data user berhasil diubah');
```

19. Selanjutnya, kita buat `view` untuk melakukan proses edit data user di `PWL/resources/views/user/edit.blade.php`

```
Minggu52 > Jobsheet5.2 > resources > views > user > edit.blade.php
1 @extends('layouts.template')
2
3 @section('content')
4 <div class="card card-outline card-primary">
5 <div class="card-header">
6 <h3 class="card-title">{{ $page->title }}</h3>
7 <div class="card-tools"></div>
8 </div>
9 <div class="card-body">
10 @empty($user)
11 <div class="alert alert-danger alert-dismissible">
12 <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
13 Data yang Anda cari tidak ditemukan.
14 </div>
15 <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
16 @else
17 <form method="POST" action="{{ url('/user/' . $user->user_id) }}" class="form-horizontal">
18 @csrf
19 <!-- method_field('PUT') !! <!-- tambahkan baris ini untuk proses edit yang butuh method PUT -->
20 <div class="form-group row">
21 <label class="col-1 control-label col-form-label">Level</label>
22 <div class="col-11">
23 <select class="form-control" id="level_id" name="level_id" required>
24 <option value="">- Pilih Level -</option>
25 @foreach($level as $item)
26 <option value="{{ $item->level_id }}"
27 @if($item->level_id == $user->level_id) selected @endif
28 {{ $item->level_nama }}
29 </option>
30 @endforeach
```




```

31 </select>
32 @error('level_id')
33 <small class="form-text text-danger">{{ $message }}</small>
34 @enderror
35 </div>
36 </div>
37 <div class="form-group row">
38 <label class="col-1 control-label col-form-label">Username</label>
39 <div class="col-11">
40 <input type="text" class="form-control" id="username" name="username" value="{{ old('username', $user->username) }}" required>
41 @error('username')
42 <small class="form-text text-danger">{{ $message }}</small>
43 @enderror
44 </div>
45 </div>
46 <div class="form-group row">
47 <label class="col-1 control-label col-form-label">Nama</label>
48 <div class="col-11">
49 <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama', $user->nama) }}" required>
50 @error('nama')
51 <small class="form-text text-danger">{{ $message }}</small>
52 @enderror
53 </div>
54 </div>
55 <div class="form-group row">
56 <label class="col-1 control-label col-form-label">Password</label>
57 <div class="col-11">
58 <input type="password" class="form-control" id="password" name="password">
59 @error('password')
60 <small class="form-text text-danger">{{ $message }}</small>
61 @else
62 <small class="form-text text-muted">Abaikan (jangan diisi) jika tidak ingin mengganti password user.</small>
63 @enderror
64 </div>
65 </div>
66 <div class="form-group row">
67 <label class="col-1 control-label col-form-label"></label>
68 <div class="col-11">
69 <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
70 <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}">Kembali</a>
71 </div>
72 </div>
73 </form>
74 </div>
75 </div>
76 </div>
77 @endsection
78
79 @push('css')
80 @endpush
81
82 @push('js')
83 @endpush

```

20. Sekarang kalian coba untuk mengedit data user di browser, amati, pahami, dan laporkan!

Home

Contact

Edit User

Edit user

Level

Manager

Username

vita16

Erka Ekaterina Ekta

Nama

Vita Eka

Password

Abaikan (jangan diisi) jika tidak ingin mengganti password user.

Simpan

Kembali

10	vita16	Vita Eka	Manager	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
----	--------	----------	---------	--

Showing 1 to 10 of 10 entries

21. Selanjutnya kita akan membuat penanganan untuk tombol hapus. Router `web.php` yang berfungsi untuk menangkap request hapus dengan method DELETE adalah

```
Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
```

22. Jadi kita buat fungsi `destroy()` pada `UserController.php`

23. Selanjutnya kita modifikasi file `PWL/resources/views/user/index.blade.php` untuk menambahkan tampilan jika ada pesan error

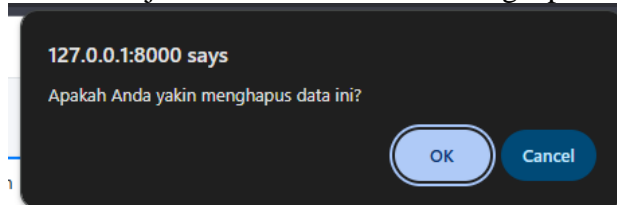


```

11 <div class="card-body">
12   @if (session(key: 'success'))
13     <div class="alert alert-success">{{ session(key: 'success') }}</div>
14   @endif
15   @if (session(key: 'error'))
16     <div class="alert alert-danger">{{ session(key: 'error') }}</div>
17   @endif
18   <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
19     <thead>
20       <tr>
21         <th>ID</th>
22         <th>Username</th>
23         <th>Nama</th>
24         <th>Level Pengguna</th>
25         <th>Aksi</th>
26       </tr>
27     </thead>
28   </table>
29 </div>

```

24. Kemudian jalankan browser untuk menghapus salah satu data user. Amati dan laporkan!



Saat user menekan button Hapus maka akan muncul alert seperti gambar di atas. Jika user memilih OK, maka data akan terhapus dan memunculkan pesan data berhasil dihapus.

Data yang telah dihapus oleh user akan hilang dari tabel dan database m_user

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	customer-1	Customer 1	Pelanggan	Detail Edit Hapus
5	customer-2	Pelanggan Kedua	Pelanggan	Detail Edit Hapus
6	manager_dua	Manager 2	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus

25. Selamat, kalian sudah membuat Laravel Starter Code untuk proses CRUD dengan menggunakan template AdminLTE dan plugin jQuery Datatables.

26. Jangan lupa commit dan push ke github PWL_POS kalian



F. Data Searching dan Filtering

Dalam menampilkan sebuah data, kadang kita perlu yang namanya melakukan pencarian (*searching*) berdasarkan kata-kunci (*keyword*) tertentu ataupun penyaringan data berdasarkan suatu kategori (*filtering*).

1. *Searching* (Pencarian):

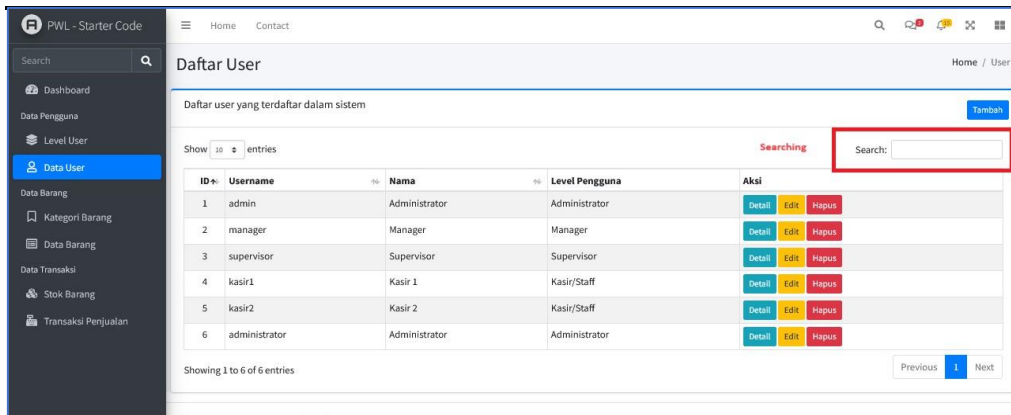
- Searching adalah proses mencari informasi atau data tertentu yang sesuai dengan kriteria yang diberikan.
- Biasanya, pencarian dilakukan dengan menggunakan kata kunci atau frasa tertentu untuk mencocokkan dengan data yang ada.
- Tujuan dari pencarian adalah untuk menemukan entitas yang cocok dengan kriteria pencarian, baik itu dalam basis data, dalam dokumen teks, atau dalam konteks lainnya.
- Pencarian sering kali melibatkan pencocokan pola atau kata kunci dalam teks atau data yang ada, dan hasilnya mungkin mencakup entitas yang sebagian cocok atau mirip dengan kriteria pencarian.

2. *Filtering* (Penyaringan):

- Filtering adalah proses memilih atau membatasi sekumpulan data atau entitas berdasarkan kriteria tertentu.
- Biasanya, filtering dilakukan untuk menyaring data yang sudah ada berdasarkan atribut atau karakteristik tertentu, seperti tanggal, kategori, atau atribut lainnya.
- Tujuan dari penyaringan adalah untuk menyajikan data yang relevan atau relevan dengan kebutuhan pengguna atau kriteria tertentu.
- Filtering dapat dilakukan dengan memilih data berdasarkan nilai yang sesuai dengan kriteria tertentu atau dengan mengecualikan data yang tidak memenuhi kriteria tersebut.

Perbedaan utama antara *searching* dan *filtering* adalah bahwa *searching* berkaitan dengan pencarian data yang sesuai dengan kriteria tertentu, sementara *filtering* berkaitan dengan penyaringan data yang sudah ada berdasarkan kriteria tertentu. Meskipun keduanya sering digunakan bersama-sama dalam aplikasi atau sistem informasi, mereka melayani tujuan yang berbeda dalam proses pengelolaan dan analisis data.

Secara default, jQuery datatables sudah memiliki fitur searching yang sudah terintegrasi dengan laravel yajra-datatables. Akan tetapi, belum ada untuk proses *filtering* datanya. Untuk itu kita akan membuat filtering data pada Laravel Starter Code kita.



Praktikum 4 – Implementasi Filtering Datatables:

Kita akan menerapkan filtering pada datatable yang sudah kita buat. Hal ini akan mempermudah kita dalam mengelompokkan suatu data sesuai kategori tertentu. Langkah-langkah yang kita kerjakan sebagai berikut

1. Kita modifikasi fungsi `index()` di `UserController.php` untuk menambahkan data yang ingin dijadikan kategori untuk data *filtering*

```

9  class UserController extends Controller
10 {
11     // Menampilkan halaman awal user
12     public function index(): Factory\View
13     {
14         $breadcrumb = (object) [
15             'title' => 'Daftar User',
16             'list' => ['Home', 'User']
17         ];
18
19         $page = (object) [
20             'title' => 'Daftar user yang terdaftar dalam sistem'
21         ];
22
23         $activeMenu = 'user'; // set menu yang sedang aktif
24
25         $level = LevelModel::all(); // ambil data level untuk filter level
26
27         return view('user.index', data: [
28             'breadcrumb' => $breadcrumb,
29             'page' => $page,
30             'level' => $level,
31             'activeMenu' => $activeMenu
32         ]);
33     }
34 }
  
```

2. Kemudian kita modifikasi view untuk menampilkan data filtering pada `PWL/resources/views/user/index.blade.php`

```

16     <div class="alert alert-danger">{{ session(key: 'error') }}</div>
17     @endif
18     <div class="row">
19         <div class="col-md-12">
20             <div class="form-group row">
21                 <label class="col-1 control-label col-form-label">Filter</label>
22                 <div class="col-3">
23                     <select class="form-control" id="level_id" name="level_id" required>
24                         <option value="">Semua</option>
25                         @foreach($level as $item)
26                             <option value="{{ $item->level_id }}">{{ $item->level_name }}</option>
27                         @endforeach
28                     </select>
29                     <small class="form-text text-muted">Level Pengguna</small>
30                 </div>
31             </div>
32         </div>
33     </div>
34     <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
  
```



3. Selanjutnya, tetap pada view `index.blade.php`, kita tambahkan kode berikut pada deklarasi ajax di datatable. Kode ini digunakan untuk mengirimkan data untuk filtering

```

53 @push('js')
54 <script>
55     $(document).ready(function() {
56         var dataUser = $('#table_user').DataTable({
57             // serverSide: true, jika ingin menggunakan server side processing serverSide: true,
58             ajax: {
59                 "url": "{{ url(path: 'user/list') }}",
60                 "dataType": "json",
61                 "type": "POST",
62                 "data": function(d) {
63                     d.level_id = $('#level_id').val();
64                 }
65             },
66         });
67     });
68 </script>
69 @endpush

```

4. Kemudian kita edit pada bagian akhir script `@push('js')` untuk menambahkan listener jika data filtering dipilih

```

96         data: "aksi",
97         className: "",
98         orderable: false,
99         searchable: false
100     }
101 }
102 });
103 $('#level_id').on('change', function() {
104     dataUser.ajax.reload();
105 });
106 });
107 </script>
108 @endpush

```

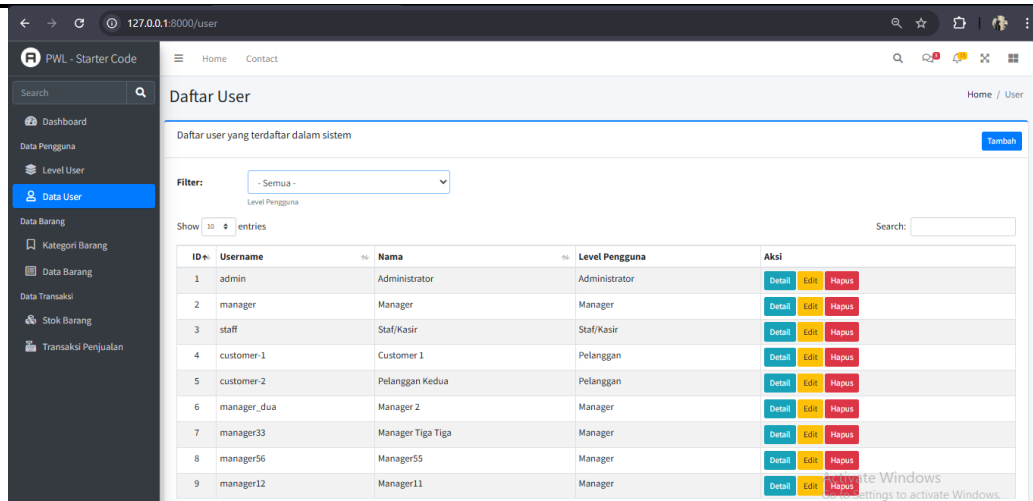
5. Tahapan akhir adalah memodifikasi fungsi `list()` pada `UserController.php` yang digunakan untuk menampilkan data pada datatable

```

37 public function list(Request $request): JsonResponse
38 {
39     $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
40         ->with('level');
41
42     // Filter data user berdasarkan level_id
43     if ($request->level_id) {
44         $users->where('level_id', $request->level_id);
45     }
46
47     Chat (CTRL + I) / Edit (CTRL + L)
48     return DataTables::of($users)
49         // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
50         ->addIndexColumn()
51         ->addColumn('aksi', function ($user): string { // menambahkan kolom aksi
52             $btn = '<a href="' . url(path: '/user/' . $user->user_id) . '" class="btn btn-info btn-sm">Detail</a> ';
53             $btn .= '<a href="' . url(path: '/user/' . $user->user_id . '/edit') . '" class="btn btn-warning btn-sm">Edit</a> ';
54             $btn .= '<form class="d-inline-block" method="POST" action="' . url(path: '/user/' . $user->user_id) . '">';
55                 csrf_field() . method_field('DELETE') .
56                 '<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\'' . 'Apakah Anda yakin menghapus data ini?' . '\');">Hapus</button></form>';
57             return $btn;
58         })
59         ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
60         ->make(true);
61 }

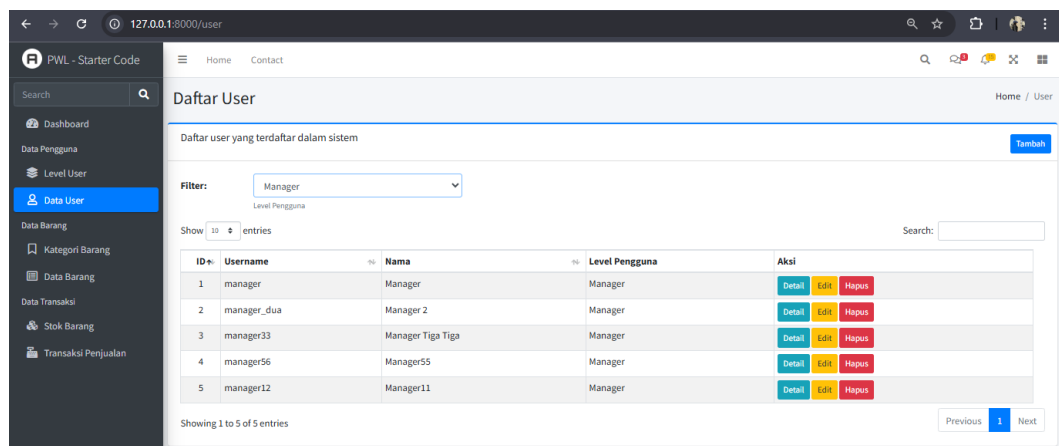
```

6. Bagian akhir adalah kita coba jalankan di browser dengan akses menu user, maka akan tampil seperti berikut

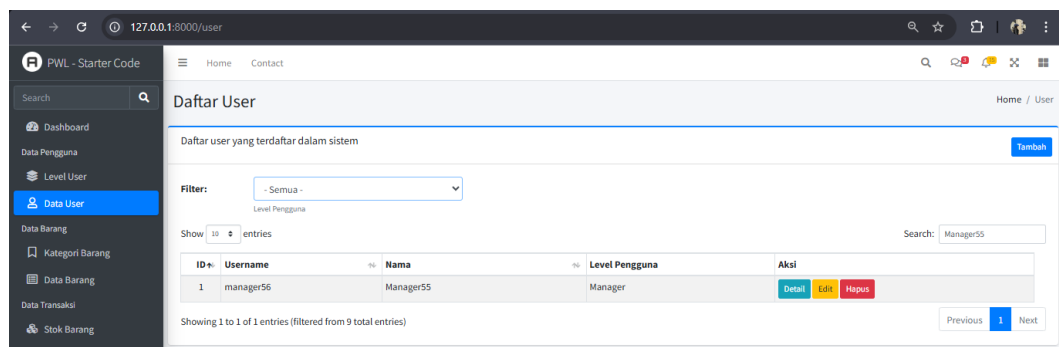


7. Selamat, sekarang Laravel Starter Code sudah ada filtering dan searching data. Starter Code sudah bisa digunakan dalam membangun sebuah sistem berbasis website.

- Contoh filtering untuk level manager



- Searching data manager56



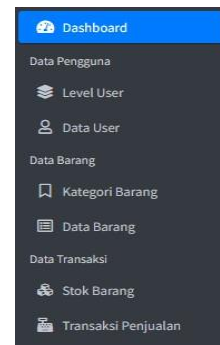
8. Jangan lupa commit dan push ke github PWL_POS kalian



G. Tugas

Implementasikan web layout dan datatables, pada menu berikut ini

- ✓ Tabel m_level
- ✓ Tabel m_kategori
- ✓ Tabel m_supplier
- ✓ Tabel m_barang



JAWAB:

1. M_level

- Tambahkan route prefix /level pada file web.php

```
8 use App\Http\Controllers\LevelController;

54 Route::group(['prefix' => 'level'], function () { void {
55     Route::get('/', [LevelController::class, 'index']); // menampilkan halaman awal level
56     Route::post('/list', [LevelController::class, 'list']); // menampilkan data level dalam bentuk json untuk datatables
57     Route::get('/create', [LevelController::class, 'create']); // menampilkan halaman form tambah level
58     Route::post('/', [LevelController::class, 'store']); // menyimpan data level baru
59     Route::get('/{id}', [LevelController::class, 'show']); // menampilkan detail level
60     Route::get('/{id}/edit', [LevelController::class, 'edit']); // menampilkan halaman form edit level
61     Route::put('/{id}', [LevelController::class, 'update']); // menyimpan perubahan data level
62     Route::delete('/{id}', [LevelController::class, 'destroy']); // menghapus data level
63 });
```

- Modifikasi file model LevelModel.php (tambahkan function seperti di bawah ini)

```
web.php M LevelModel.php M X
Minggu5.2 > Jobsheet5.2 > app > Models > LevelModel.php > PHP Intelephense > LevelModel > users
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\Relations\HasMany;
8
9 10 references | 0 implementations
10 class LevelModel extends Model
11 {
12     use HasFactory;
13
14     0 references
15     protected $table = 'm_level'; // Mendefinisikan nama tabel yang digunakan oleh model ini
16     0 references
17     protected $primaryKey = 'level_id'; // Mendefinisikan primary key dari tabel yang digunakan
18
19     0 references
20     protected $fillable = ['level_kode', 'level_nama']; // FK
21
22     0 references | 0 overrides
23     public function users(): HasMany
24     {
25         return $this->hasMany(UserModel::class, 'level_id', 'level_id');
26     }
27 }
```

- Modifikasi file LevelController.php

```
Minggu5.2 > Jobsheet5.2 > app > Http > Controllers > LevelController.php > PHP > LevelController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\LevelModel;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\DB;
8 use Yajra\DataTables\Facades\DataTables;
```




```

10 class LevelController extends Controller
11 {
12     @Override
13     public function index(): View
14     {
15         // DB::insert('insert into m_level(level_kode, level_nama, created_at) values(?, ?, ?)', ['CUS', 'Pelanggan', now()]);
16         // return 'Insert data baru berhasil';
17
18         // $row = DB::update('update m_level set level_nama = ? where level_kode = ?', ['Customer', 'CUS']);
19         // return 'Update data berhasil. Jumlah data yang diupdate: "'.$row.'" baris';
20
21         // $row = DB::delete('delete from m_level where level_kode = ?', ['CUS']);
22         // return 'Delete data berhasil. Jumlah data yang dihapus: "'.$row.'" baris';
23
24         // $data = DB::select('select * from m_level');
25         // return view('level', ['data' => $data]);
26
27         $breadcrumb = (object) [
28             'title' => 'Daftar Level',
29             'list' => ['Home', 'Level'],
30         ];
31
32         $page = (object) [
33             'title' => 'Daftar level yang terdaftar dalam sistem',
34         ];
35
36         $activeMenu = 'level';
37
38         $level = LevelModel::all(); // ambil semua data level dari db pml_pos
39
40         return view('level.index', data: [ 'breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu, 'page' => $page, 'level' => $level ]);
41     }
42 }

```

- ✓ Tambahkan function list untuk mengambil semua data user

```

60 //ambil data level dan smn dari json untuk details
61 reference [document]
62 public function List(Request $request): JsonResponse
63 {
64     $levels = LevelModel::select('level_id', 'level_kode', 'level_name');
65
66     if ($request->level_id) {
67         $levels=where('level_id', $request->level_id);
68     }
69
70 return DataTables::of($levels)
71
72     ->addColumnColumn() // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
73     ->addColumn('smn', function ($level) { return ($level->smn); }) // menambahkan kolom aksi
74
75     $btn = '<a href=""></a>' . url(path_./level/' . $level->level_id . '" class="btn btn-info btn-sm">Detail</a>' ;
76     $btn = '<a href=""></a>' . url(path_./level/' . $level->level_id . '/edit') . '" class="btn btn-warning btn-sm">Edit</a>' ;
77     $btn = '<form class="inline-block method="POST" action="">' . url(path_./level/' . $level->level_id . '"></form>' ;
78     $csrfField = method_field('DELETE') ;
79
80     ">button type='submit' class='btn btn-danger btn-sm' onclick='return confirm(\"Apakah Anda yakin menghapus data ini?\");>hapus</button></form>" ;
81 return $btn ;
82
83 )
84 ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
85 ->make(true);

```

- ✓ Tambahkan function create untuk menampilkan form tambah level

```
65 // Menampilkan form tambah level
66 // 1 reference | 0 overrides
67 public function create(): Factory\View
68 {
69     $breadcrumb = (object) [
70         'title' => 'Tambah Level',
71         'list' => ['Home', 'Level', 'Tambah'],
72     ];
73
74     $page = (object) [
75         'title' => 'Tambah level baru',
76     ];
77
78     $activeMenu = 'level';
79
80     return view('level.create', data: ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu, 'page' => $page]);
81 }
```

- ✓ Tambahkan function store untuk menyimpan data level baru yang ditambahkan

```

82 | // Menyimpan data level baru
    | 1 reference | 0 overrides
83 | public function store(Request $request): mixed|RedirectResponse
    | {
84 |
85 |     $request->validate([
86 |         'level_kode' => 'required|string|max:5',
87 |         'level_nama' => 'required|string|max:100'
88 |     ]);
89 |
90 |     LevelModel::create([
91 |         'level_kode' => $request->level_kode,
92 |         'level_nama' => $request->level_nama
93 |     ]);
94 |
95 |     return redirect(to: '/level')->with('success', 'Data level berhasil ditambahkan');
96 | }

```

- ✓ Tambahkan function show untuk menampilkan detail level

```

90 // @param {KamDetail level}
91 // reference @ overrides
92 public function show(string $id): Factory\View
93 {
94     $level = LevelModel::find($id);
95
96     $breadcrumb = (object) [
97         'title' => 'Detail level',
98         'list' => ['Home', 'Level', 'Detail'],
99     ];
100
101     $page = (object) [
102         'title' => 'Detail level',
103     ];
104
105     $activeMenu = 'level';
106
107     return view('level.show', data: ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu, 'page' => $page, 'level' => $level]);
108 }

```




- ✓ Tambahkan function edit untuk menampilkan form edit level

```

117 // Menampilkan form edit level
118 1 reference | 0 overrides
119 public function edit(string $id): Factory\View
120 {
121     $level = LevelModel::find($id);
122
123     $breadcrumb = (object) [
124         'title' => 'Edit level',
125         'list' => ['Home', 'Level', 'Edit'],
126     ];
127
128     $page = (object) [
129         'title' => 'Edit level',
130     ];
131     $activeMenu = 'level';
132
133     return view('level.edit', data: ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu, 'page' => $page, 'level' => $level]);
134 }
  
```

- ✓ Tambahkan function update untuk mengupdate data level yang telah di edit

```

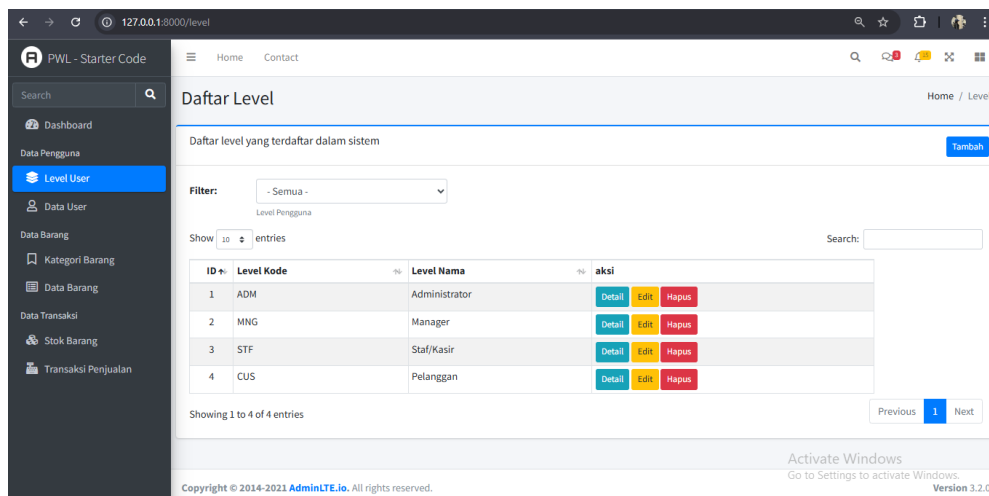
136 // Memperbarui data level
137 1 reference | 0 overrides
138 public function update(Request $request, string $id): mixed\RedirectResponse
139 {
140     $request->validate([
141         'level_kode' => 'required|string|max:5',
142         'level_nama' => 'required|string|max:100'
143     ]);
144
145     LevelModel::find($id)->update([
146         'level_kode' => $request->level_kode,
147         'level_nama' => $request->level_nama
148     ]);
149
150     return redirect(to: '/level')->with('success', 'Data level berhasil diubah');
151 }
  
```

- ✓ Tambahkan function destroy untuk menghapus data level

```

152 // Menghapus data level
153 1 reference | 0 overrides
154 public function destroy(string $id): mixed\RedirectResponse
155 {
156     $check = LevelModel::find($id);
157     if (!$check) {
158         return redirect(to: '/level')->with('error', 'Data level tidak ditemukan');
159     }
160
161     try {
162         LevelModel::destroy($id);
163         return redirect(to: '/level')->with('success', 'Data level berhasil dihapus');
164     } catch (\Illuminate\Database\QueryException $e) {
165         return redirect(to: '/level')->with('error', 'Data level gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini');
166     }
167 }
  
```

- Buat view untuk CRUD level (index blade, create blade, show blade, edit blade)
<https://github.com/vitasaraswati/PWL25/tree/master/Minggu5.2/Jobsheet5.2/resources/views/level>
- Tampilan pada web browser





✓ Menambahkan level pengguna baru

Tambah Level

Tambah level baru

Level Kode: Owners Ownership Owner's

Nama:

Daftar Level

Daftar level yang terdaftar dalam sistem

Data level berhasil ditambahkan

Filter:

Show 10 entries

ID	Level Kode	Level Nama	aksi
1	ADM	Administrator	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	MNG	Manager	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3	STF	Staff/Kasir	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
4	CUS	Pelanggan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
5	OWN	Owner	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

✓ Menampilkan detail level

Detail Level

Detail level

ID	6
Level Kode	OWN
Level Nama	Owner

✓ Mengedit level pengguna

Edit Level

Edit level

Kode: Owners Ownership Owner's

Nama:

Daftar Level

Daftar level yang terdaftar dalam sistem

Data level berhasil diupdate

Filter:

Show 10 entries

ID	Level Kode	Level Nama	aksi
1	ADM	Administrator	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	MNG	Manager	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3	STF	Staff/Kasir	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
4	CUS	Pelanggan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
5	OWN	Owner	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

✓ Filtering level pengguna

Daftar Level

Daftar level yang terdaftar dalam sistem

Filter:

Show 10 entries

ID	Level Kode	Level Nama	aksi
3	STF	Staff/Kasir	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

✓ Searching level pengguna

Daftar Level

Daftar level yang terdaftar dalam sistem

Filter:

Show 10 entries

Search:

ID	Level Kode	Level Nama	aksi
1	OWN	Owner	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Showing 1 to 1 of 1 entries (filtered from 5 total entries)

Previous 1 Next

✓ Menghapus level pengguna

Daftar Level

Daftar level yang terdaftar dalam sistem

Filter:

Show 10 entries

Search:

ID	Level Kode	Level Nama	aksi
1	ADM	Administrator	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	MNG	Manager	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3	STF	Staff/Kasir	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
4	CUS	Pelanggan	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Showing 1 to 4 of 4 entries

127.0.0.1:8000 says
Apakah Anda yakin menghapus data ini?



2. M_kategori

- Tambahkan route prefix /kategori pada file web.php

```
9 use App\Http\Controllers\KategoriController; //tugas2 m_kategori

//tugas 2 m_kategori
Route::group(['prefix' => 'kategori'], function () { void {
Route::get('/', [KategoriController::class, 'index']); // menampilkan halaman awal kategori
Route::post('/list', [KategoriController::class, 'list']); // menampilkan data kategori dalam bentuk json untuk datatables
Route::get('/create', [KategoriController::class, 'create']); // menampilkan halaman form tambah kategori
Route::post('/', [KategoriController::class, 'store']); // menyimpan data kategori baru
Route::get('/{id}', [KategoriController::class, 'show']); // menampilkan detail kategori
Route::get('/{id}/edit', [KategoriController::class, 'edit']); // menampilkan halaman form edit kategori
Route::put('/{id}', [KategoriController::class, 'update']); // menyimpan perubahan data kategori
Route::delete('/{id}', [KategoriController::class, 'destroy']); // menghapus data kategori
});
```

- Modifikasi controller pada file KategoriController.php

<https://github.com/vitasaraswati/PWL25/blob/master/Minggu5.2/Jobsheet5.2/app/Http/Controllers/KategoriController.php>

- Modifikasi models file KategoriModel.php (jika belum ada buat terlebih dahulu)

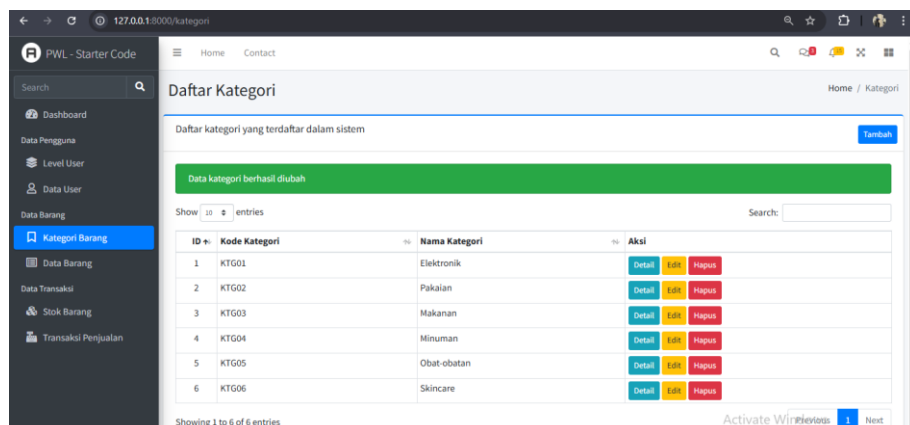
```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5.2\Jobsheet5.2> php artisan make:model KategoriModel
INFO Model [F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5.2\app\Models\KategoriModel.php] created successfully.

Minggu5.2 > JobSheet5.2 > app > Models > KategoriModel.php > PHP Intelephense > KategoriModel
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\Relations\HasMany;
8
9 class KategoriModel extends Model
10 {
11     use HasFactory;
12     protected $table = 'm_kategori';
13     protected $primaryKey = 'kategori_id';
14     protected $fillable = ['kategori_kode', 'kategori_nama'];
15
16     public function barang(): HasMany
17     {
18         return $this->hasMany(BarangModel::class);
19     }
20 }
```

- Buat view untuk CRUD kategori blade (index, create, show, edit)

<https://github.com/vitasaraswati/PWL25/tree/master/Minggu5.2/Jobsheet5.2/resources/views/kategori>

- Tampilan halaman web browser



V



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
 Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
 Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>

✓ Menambahkan kategori baru

Home Contact

Tambah Kategori

Tambah kategori baru

Kode Kategori: KTG007

Nama Kategori: Haircare

[Simpan](#) [Kembali](#)

Home Contact

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Data kategori berhasil ditambahkan

Show 10 entries

ID	Kode Kategori	Nama Kategori	Aksi
1	KTG001	Elektronik	Detail Edit Hapus
2	KTG002	Pakaian	Detail Edit Hapus
3	KTG003	Makanan	Detail Edit Hapus
4	KTG004	Minuman	Detail Edit Hapus
5	KTG005	Obat-obatan	Detail Edit Hapus
6	KTG006	Skincare	Detail Edit Hapus
7	KTG007	Haircare	Detail Edit Hapus

✓ Menampilkan detail kategori

Home Contact

Detail Kategori

Detail kategori

ID	9
Kode Level	KTG007
Nama Level	Haircare

[Kembali](#)

✓ Mengedit kategori

Home Contact

Edit Kategori

Edit kategori

Kode Kategori: KTG007

Nama Kategori: Haircare wanita

[Simpan](#) [Kembali](#)

Home Contact

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Data kategori berhasil diubah

Show 10 entries

ID	Kode Kategori	Nama Kategori	Aksi
1	KTG001	Elektronik	Detail Edit Hapus
2	KTG002	Pakaian	Detail Edit Hapus
3	KTG003	Makanan	Detail Edit Hapus
4	KTG004	Minuman	Detail Edit Hapus
5	KTG005	Obat-obatan	Detail Edit Hapus
6	KTG006	Skincare	Detail Edit Hapus
7	KTG007	Haircare wanita	Detail Edit Hapus

✓ Searching kategori

Home Contact

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Search: haircare

ID	Kode Kategori	Nama Kategori	Aksi
1	KTG07	Haircare wanita	Detail Edit Hapus

Showing 1 to 1 of 1 entries (filtered from 7 total entries)

Previous 1 Next

✓ Menghapus kategori

Home Contact

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Data kategori berhasil dihapus

Show 10 entries

ID	Kode Kategori	Nama Kategori	Aksi
1	KTG001	Elektronik	Detail Edit Hapus
2	KTG002	Pakaian	Detail Edit Hapus
3	KTG003	Makanan	Detail Edit Hapus
4	KTG004	Minuman	Detail Edit Hapus
5	KTG005	Obat-obatan	Detail Edit Hapus
6	KTG006	Skincare	Detail Edit Hapus

Showing 1 to 6 of 6 entries

127.0.0.1:8000 says

Apakah Anda yakin menghapus data ini?

[OK](#) [Cancel](#)



3. M_barang

- Tambahkan route prefix /kategori pada file web.php

```
10 use App\Http\Controllers\BarangController; //tugas3 m_barang

//tugas3 m_barang
Route::group(['prefix' => 'barang'], function () { void {
Route::get('/', [BarangController::class, 'index']); // menampilkan halaman awal barang
Route::post('/list', [BarangController::class, 'list']); // menampilkan data barang dalam bentuk json untuk datatables
Route::get('/create', [BarangController::class, 'create']); // menampilkan halaman form tambah barang
Route::post('/', [BarangController::class, 'store']); // menyimpan data barang baru
Route::get('/{id}', [BarangController::class, 'show']); // menampilkan detail barang
Route::get('/{id}/edit', [BarangController::class, 'edit']); // menampilkan halaman form edit barang
Route::put('/{id}', [BarangController::class, 'update']); // menyimpan perubahan data barang
Route::delete('/{id}', [BarangController::class, 'destroy']); // menghapus data barang
});
```

- Modifikasi controller pada file BarangController.php

<https://github.com/vitasaraswati/PWL25/blob/master/Minggu5.2/Jobsheet5.2/app/Http/Controllers/BarangController.php>

- Modifikasi models file BarangModel.php (jika belum ada buat terlebih dahulu)

```
PS F:\> KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5.2\Jobsheet5.2> php artisan make:model BarangModel
INFO Model [F:\> KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5.2\Jobsheet5.2\app\Models\BarangModel.php] created successfully.

Minggu5.2 > Jobsheet5.2 > app > Models > BarangModel.php > PHP > BarangModel > File
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\Relations\BelongsTo;
8
9 10 references | 0 implementations
11 class BarangModel extends Model
12 {
13     use HasFactory;
14
15     0 references $table = "m_barang";
16     0 references $primaryKey = "barang_id";
17     0 references $fillable = [
18         'kategori_id',
19         'barang_kode',
20         'barang_nama',
21         'harga_beli',
22         'harga_jual'
23     ];
24
25     0 references | 0 overrides
26     public function kategori(): BelongsTo
27     {
28         return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
29     }
30 }
```

- Buat view untuk CRUD barang blade (index, create, show, edit)

<https://github.com/vitasaraswati/PWL25/tree/master/Minggu5.2/Jobsheet5.2/resources/views/barang>

- Tampilan halaman web browser

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	BRG001	Samsung Smart TV	3000000	3500000	Elektronik	Detail Edit Hapus
2	BRG002	Air Conditioner Sharp	4000000	4200000	Elektronik	Detail Edit Hapus
3	BRG003	Hair Dryer Dyson	4000000	4500000	Elektronik	Detail Edit Hapus
4	BRG004	Cardigan Rajut	120000	125000	Pakaian	Detail Edit Hapus
5	BRG005	Jaket Polos	150000	155000	Pakaian	Detail Edit Hapus
6	BRG006	Voal	400000	550000	Pakaian	Detail Edit Hapus
7	BRG007	Sarden	15000	20000	Makanan	Detail Edit Hapus
8	BRG008	Selai Strawberry	15000	17000	Makanan	Detail Edit Hapus



✓ Menambahkan barang baru

Home Contact

Tambah Barang

Tambah barang baru

Kategori Skincare

Kode Barang KT06

Nama Barang Azarine Sunscreen Calm My Acne 20g

Harga Beli 35000

Harga Jual 22000

Simpan Kembali

Data barang berhasil disimpan

Filter: - Semua -

Kategori Barang

Show 10 entries

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
11	BRG011	Kopi Aren	12000	18000	Minuman	Detail Edit Hapus
12	BRG012	Jasmine Tea	15000	20000	Minuman	Detail Edit Hapus
13	BRG013	Paracetamol 500mg	2500	5000	Obat-obatan	Detail Edit Hapus
14	BRG014	Vitamin C 1000mg	10000	15000	Obat-obatan	Detail Edit Hapus
15	BRG015	Antiseptik Cair 100ml	18000	25000	Obat-obatan	Detail Edit Hapus
16	KT06	Azarine Sunscreen Calm My Acne 20g	35000	22000	Skincare	Detail Edit Hapus

✓ Menampilkan detail barang

Detail Barang

Detail barang

ID	16
Kategori Barang	Skincare
Kode Barang	KT06
Nama Barang	Azarine Sunscreen Calm My Acne 20g
Harga Beli	35000
Harga Jual	22000

Kembali

✓ Mengedit data barang

Edit Barang

Edit barang

Kategori Skincare

Kode Barang BRG16

Nama Barang Azarine Sunscreen Calm My Acne 20g

Harga Beli 35000

Harga Jual 22000

Simpan Kembali

Data barang berhasil diubah

Filter: - Semua -

Kategori Barang

Show 10 entries

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
11	BRG011	Kopi Aren	12000	18000	Minuman	Detail Edit Hapus
12	BRG012	Jasmine Tea	15000	20000	Minuman	Detail Edit Hapus
13	BRG013	Paracetamol 500mg	2500	5000	Obat-obatan	Detail Edit Hapus
14	BRG014	Vitamin C 1000mg	10000	15000	Obat-obatan	Detail Edit Hapus
15	BRG015	Antiseptik Cair 100ml	18000	25000	Obat-obatan	Detail Edit Hapus
16	BRG16	Azarine Sunscreen Calm My Acne 20g	35000	22000	Skincare	Detail Edit Hapus

✓ Filtering data barang

Daftar Barang

Daftar barang yang terdaftar dalam sistem

Filter: Skincare

Kategori Barang

Show 10 entries

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	BRG16	Azarine Sunscreen Calm My Acne 20g	35000	22000	Skincare	Detail Edit Hapus

✓ Searching data barang

Daftar Barang

Daftar barang yang terdaftar dalam sistem

Filter: - Semua -

Kategori Barang

Show 10 entries

Search:

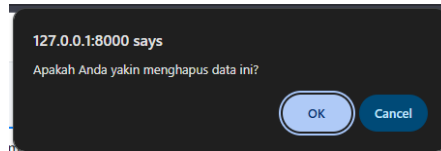
ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	BRG16	Azarine Sunscreen Calm My Acne 20g	35000	22000	Skincare	Detail Edit Hapus

Showing 1 to 1 of 1 entries (filtered from 16 total entries)

Previous 1 Next



✓ Menghapus data barang



Daftar Barang

Daftar barang yang terdaftar dalam sistem

Data barang berhasil dihapus

Filter: Semua

Show 20 entries

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
11	BRG011	Kopi Aren	12000	18000	Minuman	Detail Edit Hapus
12	BRG012	Jasmine Tea	15000	20000	Minuman	Detail Edit Hapus
13	BRG013	Paracetamol 500mg	2000	5000	Obat-obatan	Detail Edit Hapus
14	BRG014	Vitamin C 1000mg	10000	15000	Obat-obatan	Detail Edit Hapus
15	BRG015	Antisepik Cuci 100ml	18000	25000	Obat-obatan	Detail Edit Hapus

4. M_supplier

- Make migration untuk m_supplier

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PMW25\Minggu5.2> php artisan make:migration create_m_supplier_table
```

INFO Migration [F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PMW25\Minggu5.2\database\Migrations\2025_03_23_060334_create_m_supplier_table.php] created successfully.

```
Minggu5.2 > Jobsheet5.2 > database > migrations > 2025_03_23_060334_create_m_supplier_table.php
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('m_supplier', function (Blueprint $table): void {
15             $table->id('supplier_id');
16             $table->string('supplier_kode', 10)->unique();
17             $table->string('supplier_nama', 100);
18             $table->string('supplier_alamat', 255);
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      */
26     public function down(): void
27     {
28         Schema::dropIfExists('m_supplier');
29     }
30 };
```

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PMW25\Minggu5.2> php artisan migrate
```

INFO Running migrations.

2025_03_23_060334_create_m_supplier_table 81ms DONE

- Seed tabel m_supplier

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PMW25\Minggu5.2> php artisan make:seeder SupplierSeeder
```

INFO Seeder [F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PMW25\Minggu5.2\database\seeders\SupplierSeeder.php] created successfully.

```
Minggu5.2 > Jobsheet5.2 > database > seeders > SupplierSeeder.php > PHP > SupplierSeeder > run()
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Facades\DB;
8 use Carbon\Carbon;
9
10 class SupplierSeeder extends Seeder
11 {
12     /**
13      * Run the database seeds.
14      */
15 }
```



```

15 public function run(): void
16 {
17     $data_supplier = [
18         [
19             'supplier_kode' => 'SP01',
20             'supplier_nama' => 'PT Bina Makmur',
21             'supplier_alamat' => 'Jl. Rafflesia 30 Surabaya',
22             'created_at' => Carbon::now()
23         ],
24         [
25             'supplier_kode' => 'SP02',
26             'supplier_nama' => 'PT Adiguna Jaya Sentosa',
27             'supplier_alamat' => 'Jl. Haji Abdul Muis No. 8, Surabaya',
28             'created_at' => Carbon::now()
29         ],
30         [
31             'supplier_kode' => 'SP03',
32             'supplier_nama' => 'PT Wijaya Karya',
33             'supplier_alamat' => 'Jl. M.T Haryono No. 1 Surabaya',
34             'created_at' => Carbon::now()
35         ]
36     ];
37
38     DB::table('m_supplier')->insert($data_supplier);
39 }
40

```

```

PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5.2\Jobsheet5.2> php artisan db:seed --class=SupplierSeeder
>>
INFO Seeding database.

```

Cek pada database phpmyadmin

Showing rows 0 - 2 (3 total. Query took 0.0008 seconds)

SELECT * FROM 'm_supplier'

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

Extra options

	supplier_id	supplier_kode	supplier_nama	supplier_alamat	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	SP01	PT Bina Makmur	Jl. Rafflesia 30 Surabaya	2025-03-23 06:18:38	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	SP02	PT Adiguna Jaya Sentosa	Jl. Haji Abdul Muis No. 8, Surabaya	2025-03-23 06:18:38	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	SP03	PT Wijaya Karya	Jl. M.T Haryono No. 1 Surabaya	2025-03-23 06:18:38	NULL

- Buat controller -> SupplierController.php
- <https://github.com/vitasaraswati/PWL25/blob/master/Minggu5.2/Jobsheet5.2/app/Http/Controllers/SupplierController.php>
- Buat model -> ModelSupplier.php

```

PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5.2\Jobsheet5.2> php artisan make:model SupplierModel
INFO Model [F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5.2\Jobsheet5.2\app\Models\SupplierModel.php] created successfully.

```

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 0 references | 0 implementations
9 class SupplierModel extends Model
10 {
11     use HasFactory;
12
13     0 references
14     protected $table = "m_supplier";
15     0 references
16     protected $primaryKey = "supplier_id";
17     0 references
18     protected $fillable = [
19         "supplier_kode",
20         "supplier_nama",
21         "supplier_alamat"
22     ];
23 }

```

- Tambahkan route prefix /supplier

```

11 use App\Http\Controllers\SupplierController; //tugas4 m_supplier

```




```

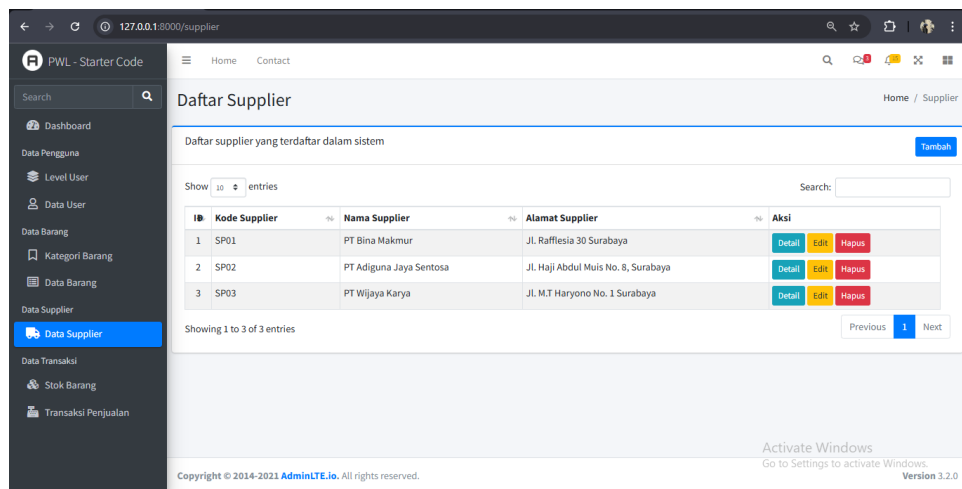
92 //tugas 4 m_supplier
93 Route::group(['prefix' => 'supplier'], function () { void {
94   Route::get('/', [SupplierController::class, 'index']); // menampilkan halaman awal supplier
95   Route::post('/list', [SupplierController::class, 'list']); // menampilkan data supplier dalam bentuk json untuk datatables
96   Route::get('/create', [SupplierController::class, 'create']); // menampilkan halaman form tambah supplier
97   Route::post('/', [SupplierController::class, 'store']); // menyimpan data supplier baru
98   Route::get('/{id}', [SupplierController::class, 'show']); // menampilkan detail supplier
99   Route::get('/{id}/edit', [SupplierController::class, 'edit']); // menampilkan halaman form edit supplier
100   Route::put('/{id}', [SupplierController::class, 'update']); // menyimpan perubahan data supplier
101   Route::delete('/{id}', [SupplierController::class, 'destroy']); // menghapus data supplier
102 });
  
```

- Tambahkan sidebar Data Supplier pada view/layouts/sidebar.blade.php

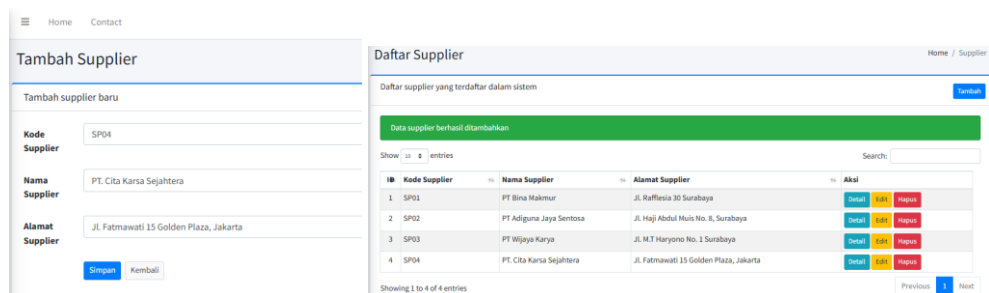
```

52 <li class="nav-header">Data Supplier</li>
53 <li class="nav-item">
54   <a href="{{ url(path: '/supplier') }}" class="nav-link {{ ($activeMenu == 'supplier') ? 'active' : '' }}">
55     <i class="nav-icon fas fa-truck"></i>
56     <p>Data Supplier</p>
57   </a>
58 </li>
  
```

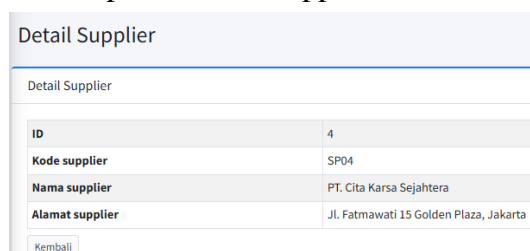
- Buat view untuk CRUD supplier blade (index, create, show, edit)
<https://github.com/vitasaraswati/PWL25/tree/master/Minggu5.2/Jobsheet5.2/resources/views/supplier>
- Tampilan pada web browser



- ✓ Menambahkan supplier baru



- ✓ Menampilkan detail supplier



V



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
 Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
 Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>

✓ Mengedit data supplier

Home Contact

Edit Supplier

Edit Supplier

Kode

Nama

Alamat

Daftar Supplier

Daftar supplier yang terdaftar dalam sistem

Data supplier berhasil diubah

Show 10 entries

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP01	PT Bina Makmur	Jl. Rafflesia 30 Surabaya	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	SP02	PT Adiguna Jaya Sentosa	Jl. Haji Abdul Muis No. 8, Surabaya	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3	SP03	PT Wijaya Karya	Jl. MT Haryono No. 1 Surabaya	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
4	SP04	PT. Budi Utama Sejahtera	Jl. Fatmawati 15 Golden Plaza, Jakarta	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Showing 1 to 4 of 4 entries

Previous 1 Next

✓ Searching data supplier

Daftar Supplier

Daftar supplier yang terdaftar dalam sistem

Show 10 entries

Search:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP03	PT Wijaya Karya	Jl. MT Haryono No. 1 Surabaya	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Showing 1 to 1 of 1 entries (filtered from 4 total entries)

Previous 1 Next

✓ Menghapus supplier

127.0.0.1:8000 says

Apakah Anda yakin menghapus data ini?

Daftar Supplier

Daftar supplier yang terdaftar dalam sistem

Data supplier berhasil dihapus

Show 10 entries

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP01	PT Bina Makmur	Jl. Rafflesia 30 Surabaya	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	SP02	PT Adiguna Jaya Sentosa	Jl. Haji Abdul Muis No. 8, Surabaya	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3	SP03	PT Wijaya Karya	Jl. MT Haryono No. 1 Surabaya	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Showing 1 to 3 of 3 entries

Previous 1 Next



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>
