



Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis
Semester : 4 (empat) / 5 (lima)
Pertemuan ke- : 5 (lima)

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 --}}
```



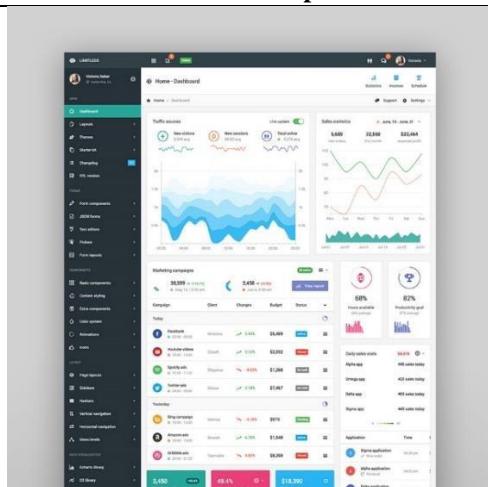
B. Web Template

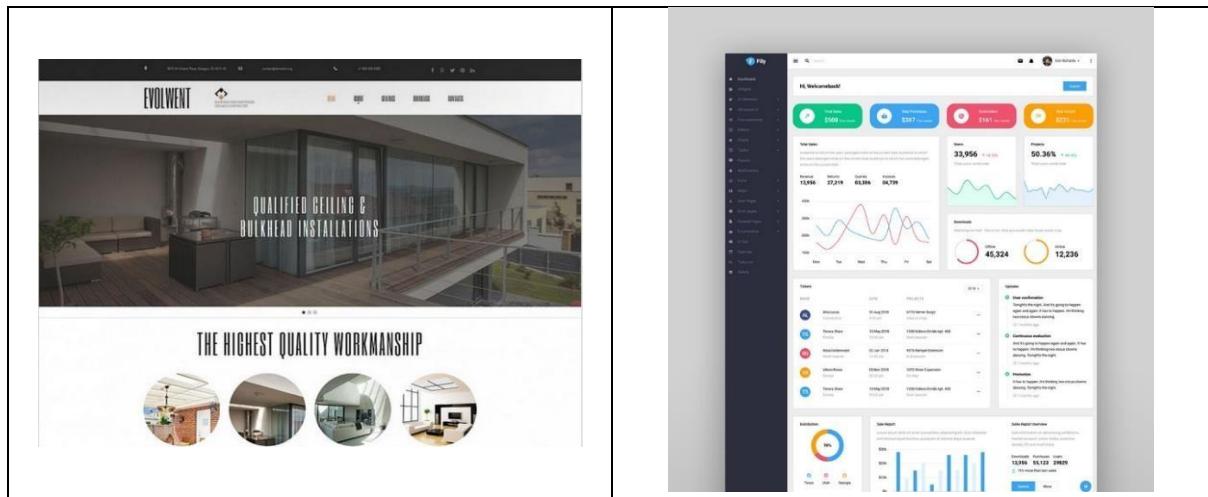
Merupakan suatu file yang sudah memiliki desain tertentu sehingga, kita tinggal memodifikasi bagian-bagian tertentu saja untuk mendapatkan hasil yang sudah bagus. Dengan menggunakan template yang sesuai dengan kebutuhan, kita bisa mengurangi waktu pembuatan suatu website secara drastis.

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

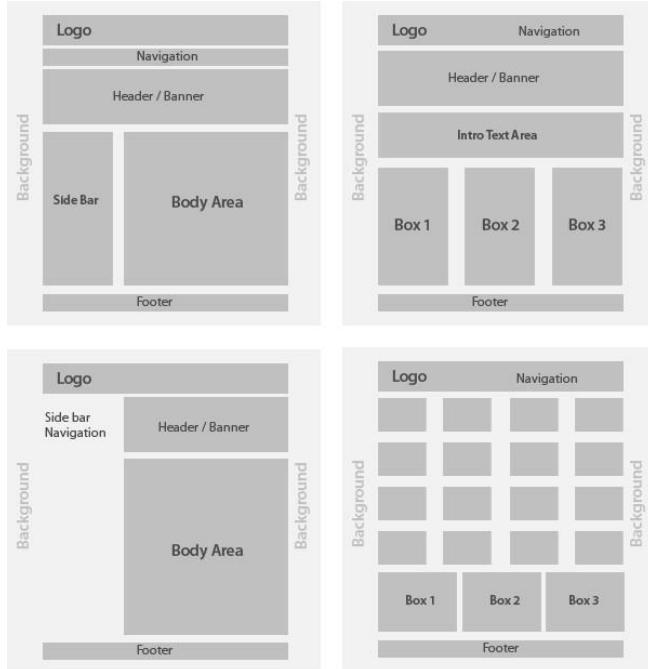
Frontend Template	Backend Template
	



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()`



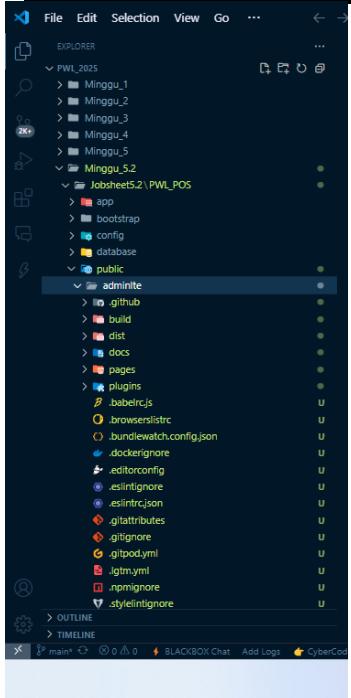
Praktikum 1 - Layouting AdminLTE:

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

The screenshot shows the GitHub release page for AdminLTE v3.2.0. It includes the release date (Feb 8, 2022), contributors (REJack, v3.2.0, bd4d9c7), and a list of top features: Enhanced Dark Mode, SVG Font Awesome Icon support, Docker dev-preset, Reworked Dark/Light colors, and Custom Scrollbar style (Non-Plugin-Scrollbar). Below this is a section for contributors and a list of assets, with a red checkmark pointing to the 'Source code (zip)' link.

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**

The screenshot shows the VS Code Explorer interface. A folder named 'adminlte' is selected and highlighted with a blue bar, indicating it has been renamed from its original name. Other subfolders like .github, build, dist, docs, pages, and plugins are also visible.



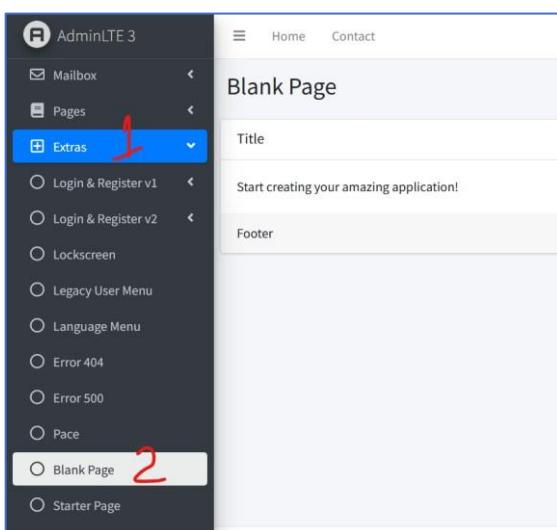
3. Selanjutnya kita buka di browser dengan alamat

http://localhost/PWL_POS/public/adminlte maka akan muncul tampilan seperti berikut



The screenshot shows two instances of the AdminLTE 3 dashboard side-by-side. Both instances have a similar layout with a left sidebar and a main dashboard area. The sidebar includes a user profile for 'Alexander Pierce' and a navigation menu with various options like 'Dashboard', 'Widgets', 'Layout Options', 'Charts', etc. The main dashboard area features four prominent cards: 'New Orders' (150), 'Bounce Rate' (53%), 'User Registrations' (44), and 'Unique Visitors' (65). Below these cards are two large, interactive charts: a line chart for 'Sales' showing fluctuations over time, and a map of the United States labeled 'Visitors'.

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



The screenshot shows the AdminLTE 3 sidebar with the 'Extras' menu expanded. The 'Blank Page' option is highlighted with a red number 2. Other options in the 'Extras' menu include 'Login & Register v1', 'Login & Register v2', 'Lockscreen', 'Legacy User Menu', 'Language Menu', 'Error 404', 'Error 500', 'Page', 'Starter Page', and 'Blank Page'. The main content area shows a 'Blank Page' template with sections for 'Title', 'Start creating your amazing application!', and 'Footer'.

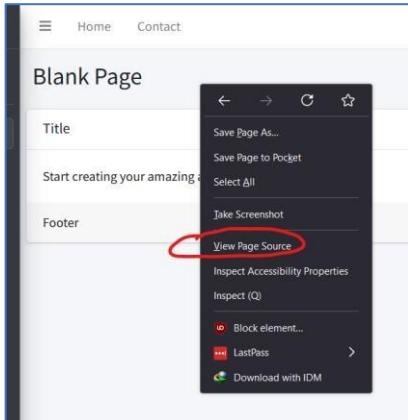
5. Dari sini kita bisa melakukan layouting halaman Blank Page ini menjadi 4 element seperti pada gambar berikut



The screenshot illustrates the AdminLTE 3 template structure:

- 1 - Header / Navbar:** Located at the top right, showing the title "Blank Page" and a breadcrumb trail "Home / Blank Page".
- 2 - Sidebar / Navigasi:** Located on the left side, containing a navigation menu with items like Mailbox, Pages, Extras (selected), Login & Register v1, Login & Register v2, Language Menu, Error 404, Error 500, Page, Blank Page (selected), Starter Page, and Search.
- 3 - Breadcrumb:** Located above the content area, showing the current page path.
- 4 - Content:** The main content area where you can start creating your amazing application.
- 5 - Footer:** Located at the bottom, displaying copyright information and the version number "Version 3.2.0".

6. Selanjutnya kita klik kanan halaman [Blank Page](#) dan klik [view page source](#)



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

```
EXPLORER
└ PWL_POS
  └ views
    └ layouts
      template.blade.php

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="https://use.fontawesome.com/releases/v5.11.2/assets/webfonts/all.css">
12 | <!-- Theme style -->
13 | <link rel="stylesheet" href="https://adminlte.io/themes/AdminLTE/dist/css/adminlte.min.css">
14 </head>
15 <body class="hold-transition sidebar-mini">
16 <!-- Site wrapper -->
17 <div class="wrapper">
```

```
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > nav.main-header.navbar.navbar-expand.navbar
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="https://use.fontawesome.com/releases/v5.11.2/assets/webfonts/all.css">
12 | <!-- Theme style -->
13 | <link rel="stylesheet" href="https://adminlte.io/themes/AdminLTE/dist/css/adminlte.min.css">
14 </head>
```



The screenshot shows a code editor window with the following details:

- Title Bar:** PWL_2025
- File Explorer:** Shows the project structure:
 - PWL_2025
 - Minggu_1
 - Minggu_2
 - Minggu_3
 - Minggu_4
 - Minggu_5
 - Minggu_5.2
 - Jobsheet5.2\PWL_POS
 - app
 - bootstrap
 - config
 - database
 - public
 - resources
 - css
 - js
 - views
 - layouts
 - template.blade.php
 - index.blade.php
 - kategori.blade.php
 - level.blade.php
 - user_tambah.blade.php
 - user_ubah.blade.php
 - user.blade.php
 - welcome.blade.php
 - routes
 - storage
 - tests
 - vendor
 - .editorconfig
 - env
 - .env.example
- Code View:** The template.blade.php file content is displayed, showing PHP code with Blade syntax and AdminLTE CSS/JS imports.
- Bottom Bar:** Includes tabs for main, BLACKBOX Chat, Add Logs, CyberCoder, Improve Code, Share Code Link, Generate Commit Message, and various status indicators like Ln 909, Col 1, Spaces: 4, UFT-8, CR LF, PHP, Go Live, and AI Code Chat.

8. File **layouts/template.blade.php** adalah file utama untuk templating website
9. Pada baris **1-14** file **template.blade.php**, kita modifikasi

Menjadi



```
template.blade.php x
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > nav.main-header.navbar.navbar-expand.navba
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('app.name', 'PWL Laravel Starter Code') }}</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="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}>
12     <!-- Theme style -->
13     <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}>
14 </head>
```

```
Minggu,52 > Jobsheet52 > PWL_POS > resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > script
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('app.name', 'PWL Laravel Starter Code') }}</title>
```

10. Kemudian kita blok baris **19-153** (baris untuk **element 1-header**), lalu kita **cut**, dan **paste**-kan di file **PWL_POS/resource/view/layouts/header.blade.php** (buat dulu file **header.blade.php** jika belum ada). Sehingga tampilan dari file **template.blade.php** menjadi seperti berikut

```
17 <div class="wrapper">
18     <!-- Navbar -->
19     @include('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="../../index3.html" class="brand-link">
26             
27             <span class="brand-text font-weight-light">AdminLTE 3</span>
28         </a>
29
30         <!-- Sidebar -->
31         <div class="sidebar">
```

23	<aside class="main-sidebar sidebar-dark-primary elevation-4">
24	<!-- Brand Logo -->
25	
26	
27	AdminLTE 3
28	

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

```
25 <body class="hold-transition sidebar-mini">
26     <!-- Site wrapper -->
27     <div class="wrapper">
28         <!-- Navbar -->
29         @include('layouts.header')
30         <!-- /.navbar -->
```

11. Kita modifikasi baris **25** dan **26** pada **template.blade.php**



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>

Menjadi



```
17 <div class="wrapper">
18   <!-- Navbar -->
19   @include('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('/') }}" class="brand-link">
26       
27       <span class="brand-text font-weight-light">PWL - Starter Code</span>
28     </a>
29
30   <!-- Sidebar -->
31   <div class="sidebar">
```

```
34   <!-- Brand Logo -->
35   <a href="{{ url('/') }}" class="brand-link">
36     
37     <span class="brand-text font-weight-light">PWL - Starter Code</span>
38   </a>
```

12. 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('/') }}" class="brand-link">
26       PWL - Starter Code</span>
28     </a>
29
30     <!-- Sidebar -->
31     @include('layouts.sidebar')
32     <!-- /.sidebar -->
33   </aside>
```

```
1 <div class="sidebar">
2   <!-- Sidebar user (optional) -->
3   <div class="user-panel mt-3 pb-3 mb-3 d-flex">
4     <div class="image">
5       
6     </div>
7     <div class="info">
8       <a href="#" class="d-block">Alexander Pierce</a>
9     </div>
10   </div>
11
12   <!-- SidebarSearch Form -->
13   <div class="form-inline">
14     <div class="input-group" data-widget="sidebar-search">
15       <input class="form-control form-control-sidebar" type="search" placeholder="Search" aria-label="Search">
16       <div class="input-group-append">
17         <button class="btn btn-sidebar">
18           | <i class="fas fa-search fa-fw"></i>
19         </button>
20       </div>
21     </div>
22   </div>
23
24   <!-- Sidebar Menu -->
25   <nav class="mt-2">
26     <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">
27       <!-- Add icons to the links using the .nav-icon class
28           with font-awesome or any other icon font library -->
29       <li class="nav-item">
30         <a href="#" class="nav-link">
31           <i class="nav-icon fas fa-tachometer-alt"></i>
32           <p>
33             Dashboard
34             <i class="right fas fa-angle-left"></i>
35           </p>
36         </a>
37         <ul class="nav nav-treeview">
38           <li class="nav-item">
39             <a href=".../index.html" class="nav-link">
```

13. 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 Jobsheet 05 – PWL 2023/2024



footer.blade.php jika belum ada). Sehingga tampilan dari file template.blade.php menjadi seperti berikut

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

57
58
59
60
61
62

14. Kemudian kita modifikasi file template.blade.php baris 91-100



```
91  <!-- jQuery -->
92  <script src="../../plugins/jquery/jquery.min.js"></script>
93  <!-- Bootstrap 4 -->
94  <script src="../../plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
95  <!-- AdminLTE App -->
96  <script src="../../dist/js/adminlte.min.js"></script>
97  <!-- AdminLTE for demo purposes -->
98  <script src="../../dist/js/demo.js"></script>
99  </body>
100 </html>
```

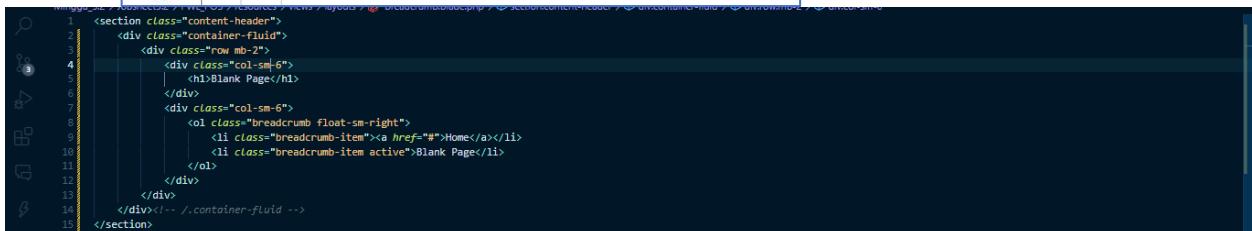
Menjadi

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

```
61  <!-- jQuery -->
62  <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script>
63  <!-- Bootstrap 4 -->
64  <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
65  <!-- AdminLTE App -->
66  <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>
67  <!-- DataTables & Plugins -->
68  <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}"></script>
```

15. Sekarang masuk pada bagian konten. Konten kita bagi menjadi 2, yaitu elemen untuk **breadcrumb** dan elemen untuk **content**.
16. 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('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('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>
```



```
1  <section class="content-header">
2  <div class="container-fluid">
3  <div class="row mb-2">
4  <div class="col-sm-6">
5  | <h1>Blank Page</h1>
6  </div>
7  <div class="col-sm-6">
8  | <ol class="breadcrumb float-sm-right">
9  | | <li class="breadcrumb-item"><a href="#">Home</a></li>
10 | | <li class="breadcrumb-item active">Blank Page</li>
11 | </ol>
12 </div>
13 </div><!-- /.container-fluid -->
14 </section>
```

17. Layout terakhir adalah pada bagian konten. Layout untuk konten bisa kita buat dinamis,



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>

sesuai dengan apa yang ingin kita sajikan pada web yang kita bangun.

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

```
50 | <!-- Main content -->
51 | <section class="content">
52 | | @yield('content')
53 | </section>
54 | <!-- /content -->
```

19. Hasil akhir pada file utama **layouts/template.blade.php** adalah seperti berikut



```
 15 <body class="hold-transition sidebar-mini">
 16   <!-- Site wrapper -->
 17   <div class="wrapper">
 18     <!-- Navbar -->
 19     @include('layouts.header') 1
 20     <!-- /.navbar -->
 21
 22     <!-- Main Sidebar Container -->
 23     <aside class="main-sidebar sidebar-dark-primary elevation-4">
 24       <!-- Brand Logo -->
 25       <a href="{{ url('/') }}" class="brand-link">
 26         PWL - Starter Code</span>
 28       </a>
 29
 30       <!-- Sidebar -->
 31       @include('layouts.sidebar') 2
 32       <!-- /.sidebar -->
 33     </aside>
 34
 35     <!-- Content Wrapper. Contains page content -->
 36     <div class="content-wrapper">
 37       <!-- Content Header (Page header) -->
 38       @include('layouts.breadcrumb') 3
 39
 40       <!-- Main content -->
 41       <section class="content">
 42         @yield('content') 4
 43       </section>
 44       <!-- /.content -->
 45     </div>
 46     <!-- /.content-wrapper -->
 47
 48     @include('layouts.footer') 5
 49   </body>
```

Minggu,52 > Jobsheet52 > PWLPOS > resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > script

```
19 <!-- Theme style -->
20 <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}>
21
22 @stack('css') <!-- Digunakan untuk memanggil custom css dari perintah push('css') pada masing-masing view -->
23 </head>
24
25 <body class="hold-transition sidebar-mini">
26   <!-- Site wrapper -->
27   <div class="wrapper">
28     <!-- Navbar -->
29     @include('layouts.header')
30     <!-- /.navbar -->
31
32     <!-- Main Sidebar Container -->
33     <aside class="main-sidebar sidebar-dark-primary elevation-4">
34       <!-- Brand Logo -->
35       <a href="{{ url('/') }}" class="brand-link">
36         
37         <span class="brand-text font-weight-light">PWL - Starter Code</span>
38       </a>
39
40       <!-- Sidebar -->
41       @include('layouts.sidebar')
42       <!-- /.sidebar -->
43     </aside>
44
45     <!-- Content Wrapper. Contains page content -->
46     <div class="content-wrapper">
47       <!-- Content Header (Page header) -->
48       @include('layouts.breadcrumb')
49
50       <!-- Main content -->
51       <section class="content">
52         @yield('content')
53       </section>
54       <!-- /.content -->
55     </div>
56     <!-- /.content-wrapper -->
57
58     @include('layouts.footer')
59
60   <!-- /.wrapper -->
61
62   <!-- Scripts -->
63   <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}></script>
```

20. Selamat kalian sudah selesai dalam melakukan layoutting website di laravel.

21. Jangan lupa commit dan push ke github untuk praktikum 1 ini

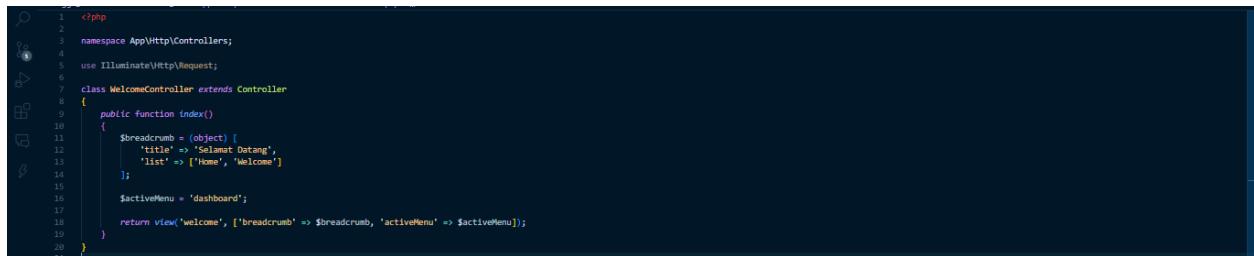


Praktikum 2 - Penerapan Layouting:

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

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

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



```
1 <?php
2 namespace App\Http\Controllers;
3
4 use Illuminate\Http\Request;
5
6 class WelcomeController extends Controller
7 {
8     public function index()
9     {
10        $breadcrumb = (object) [
11            'title' => 'Selamat Datang',
12            'list'  => ['Home', 'Welcome']
13        ];
14
15        $activeMenu = 'dashboard';
16
17        return view('welcome', ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu]);
18    }
19
20 }
```

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

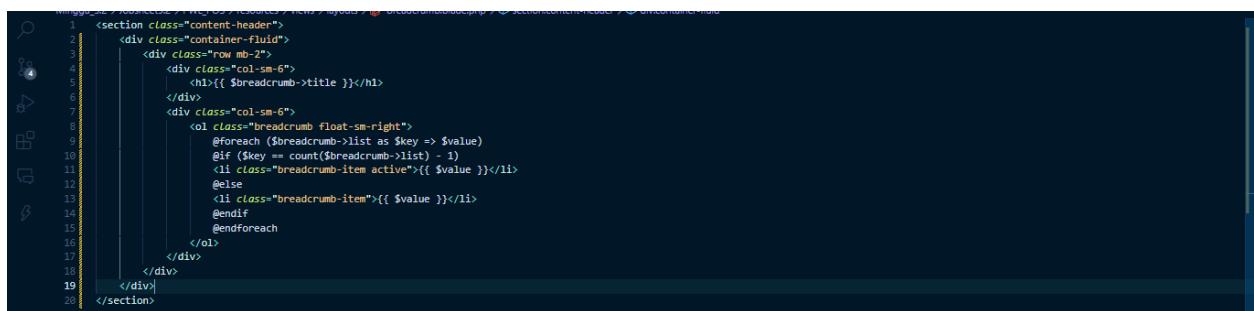


```
welcome.blade.php ×
1  @extends('layouts.template')
2
3  @section('content')
4
5  <div class="card">
6    <div class="card-header">
7      <h3 class="card-title">Halo, apakahbar!!!</h3>
8      <div class="card-tools"></div>
9    </div>
10   <div class="card-body">
11     | Selamat datang semua, ini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



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

```
welcome.blade.php 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($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>
```



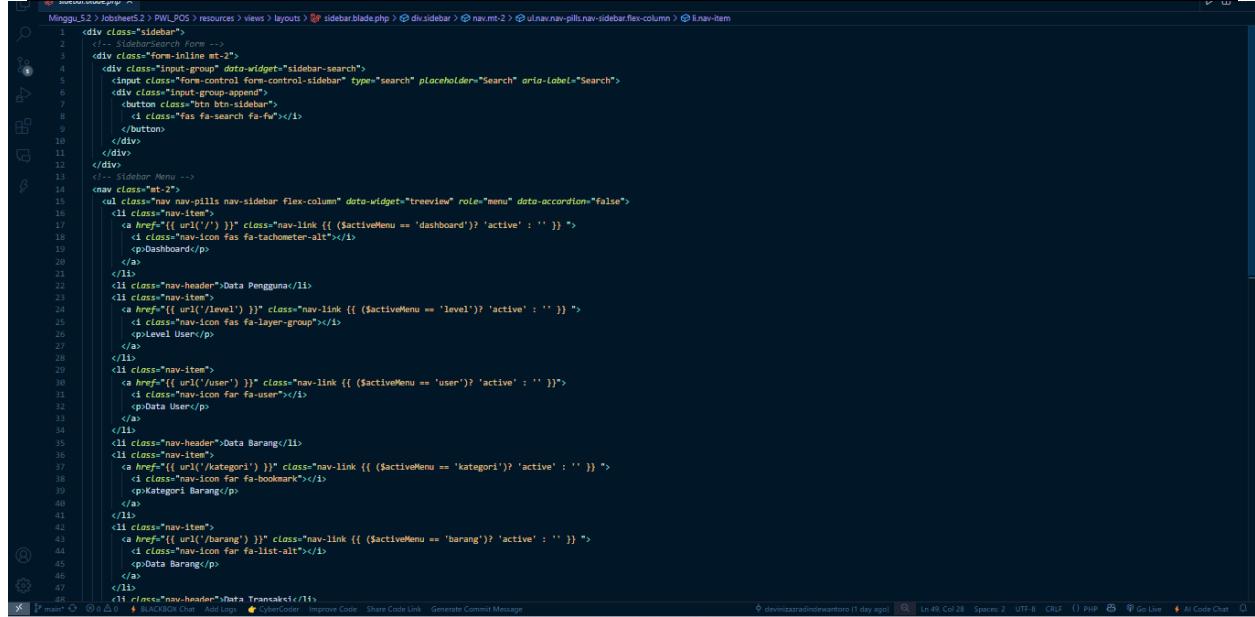


4. Kita modifikasi file [PWL_POS/resources/views/layouts/sidebar.blade.php](#)

```
<div class="sidebar">
    <!-- SidebarSearch Form -->
    <div class="form-inline mt-2">
        <div class="input-group" data-widget="sidebar-search">
            <input class="form-control form-control-sidebar" type="search"
placeholder="Search" aria-label="Search">
            <div class="input-group-append">
                <button class="btn btn-sidebar">
                    <i class="fas fa-search fa-fw"></i>
                </button>
            </div>
        </div>
    </div>
    <!-- Sidebar Menu -->
    <nav class="mt-2">
        <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview"
role="menu" data-accordion="false">
            <li class="nav-item">
                <a href="{{ url('/') }}" class="nav-link {{ ($activeMenu == 'dashboard')?
'active' : '' }} ">
                    <i class="nav-icon fas fa-tachometer-alt"></i>
                    <p>Dashboard</p>
                </a>
            </li>
        </ul>
    </nav>
</div>
```



```
<li class="nav-header">Data Pengguna</li>
<li class="nav-item">
    <a href="{{ url('/level') }}" class="nav-link {{ ($activeMenu == 'level')?
'active' : '' }} ">
        <i class="nav-icon fas fa-layer-group"></i>
        <p>Level User</p>
    </a>
</li>
<li class="nav-item">
    <a href="{{ url('/user') }}" class="nav-link {{ ($activeMenu == 'user')?
'active' : '' }} ">
        <i class="nav-icon far fa-user"></i>
        <p>Data User</p>
    </a>
</li>
<li class="nav-header">Data Barang</li>
<li class="nav-item">
    <a href="{{ url('/kategori') }}" class="nav-link {{ ($activeMenu ==
'kategori')? 'active' : '' }} ">
        <i class="nav-icon far fa-bookmark"></i>
        <p>Kategori Barang</p>
    </a>
</li>
<li class="nav-item">
    <a href="{{ url('/barang') }}" class="nav-link {{ ($activeMenu ==
'barang')? 'active' : '' }} ">
        <i class="nav-icon far fa-list-alt"></i>
        <p>Data Barang</p>
    </a>
</li>
<li class="nav-header">Data Transaksi</li>
<li class="nav-item">
    <a href="{{ url('/stok') }}" class="nav-link {{ ($activeMenu == 'stok')?
'active' : '' }} ">
        <i class="nav-icon fas fa-cubes"></i>
        <p>Stok Barang</p>
    </a>
</li>
<li class="nav-item">
    <a href="{{ url('/barang') }}" class="nav-link {{ ($activeMenu ==
'penjualan')? 'active' : '' }} ">
        <i class="nav-icon fas fa-cash-register"></i>
        <p>Transaksi Penjualan</p>
    </a>
</li>
</ul>
</nav>
</div>
```



```
Jobsheet 5.2 > Jobsheet5.2 > PWL_POS > resources > views > layouts > sidebar.blade.php > sidebar > nav-item > ul.nav.nav-pills.nav-sidebar.flex-column > li.nav-item

<div class="sidebar">
  <div class="form-search form">
    <div class="form-group mt-2">
      <div class="input-group data-widget">
        <input class="form-control form-control-sidebar" type="search" placeholder="Search" aria-label="Search">
        <div class="input-group-append">
          <button class="btn btn-sidebar">
            <i class="fas fa-search fa-fw"></i>
          </button>
        </div>
      </div>
    </div>
  </div>
  <!-- Sidebar Menu -->
  <nav class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">
    <ul class="nav-list">
      <li class="nav-item">
        <a href="#" class="nav-link" {{ ($activeMenu == 'dashboard')? 'active' : '' }}>
          <ci class="nav-icon fas fa-tachometer-alt"></ci>
          <p>Dashboard</p>
        </a>
      </li>
      <li class="nav-item">
        <a href="#" class="nav-link" {{ ($activeMenu == 'level')? 'active' : '' }}>
          <ci class="nav-icon fas fa-layer-group"></ci>
          <p>Level User</p>
        </a>
      </li>
      <li class="nav-item">
        <a href="#" class="nav-link" {{ ($activeMenu == 'user')? 'active' : '' }}>
          <ci class="nav-icon far fa-user"></ci>
          <p>Data User</p>
        </a>
      </li>
      <li class="nav-item">
        <a href="#" class="nav-link" {{ ($activeMenu == 'barang')? 'active' : '' }}>
          <ci class="nav-icon far fa-bookmark"></ci>
          <p>Data Barang</p>
        </a>
      </li>
      <li class="nav-item">
        <a href="#" class="nav-link" {{ ($activeMenu == 'kategori')? 'active' : '' }}>
          <ci class="nav-icon far fa-list-alt"></ci>
          <p>Data Barang</p>
        </a>
      </li>
      <li class="nav-item">
        <a href="#" class="nav-link" {{ ($activeMenu == 'transaksi')? 'active' : '' }}>
          <ci class="nav-icon far fa-list-alt"></ci>
          <p>Data Transaksi</p>
        </a>
      </li>
    </ul>
  </nav>
</div>
```

5. Kita tambahkan kode berikut router web.php

```
Route::get('/', [WelcomeController::class, 'index']);
```

6. Sekarang kita coba jalankan di browser dengan mengetikkan url

http://localhost/PWL_POS/public



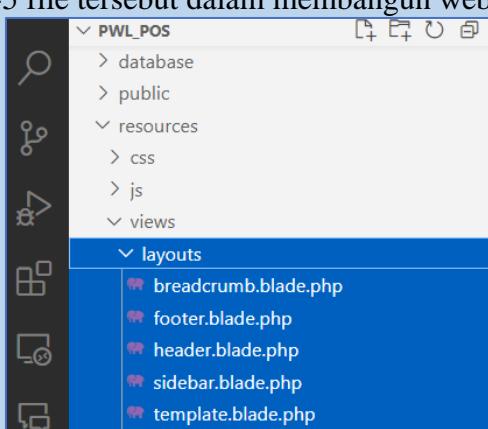
Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

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 penggunaan 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-datable dengan mengetikkan perintah pada CMD
`composer require yajra/laravel-datatables:^10.0` atau
`composer require yajra/laravel-datatables-oracle`
3. Kita modifikasi route `web.php` untuk proses CRUD user

```
6  use App\Http\Controllers\WelcomeController;
7  use Illuminate\Support\Facades\Route;
8  use App\Http\Controllers\UserController;
9
10
11 Route::get('/', [WelcomeController::class, 'index']);
12
13 Route::group(['prefix' => 'user'], function () {
14     Route::get('/', [UserController::class, 'index']);           // menampilkan halaman awal user
15     Route::post('/list', [UserController::class, 'list']);       // menampilkan data user dalam bentuk json untuk datatables
16     Route::get('/create', [UserController::class, 'create']);   // menampilkan halaman form tambah user
17     Route::post('/', [UserController::class, 'store']);         // menyimpan data user baru
18     Route::get('/{id}', [UserController::class, 'show']);        // menampilkan detail user
19     Route::get('/{id}/edit', [UserController::class, 'edit']);   // menampilkan halaman form edit user
20     Route::put('/{id}', [UserController::class, 'update']);     // menyimpan perubahan data user
21     Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
22 });
23 }
```



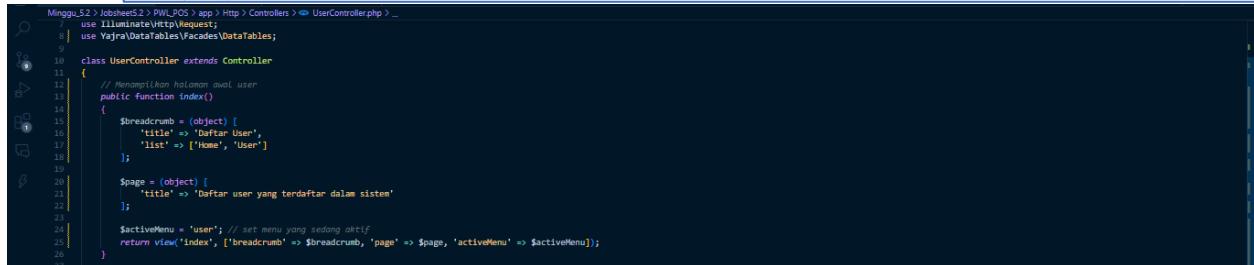
```
31 | // Jobsheet 5
32 | Route::get('/', [WelcomeController::class, 'index']);
33 |
34 | Route::group(['prefix' => 'user'], function () {
35 |     Route::get('/', [UserController::class, 'index']);           // menampilkan halaman awal user
36 |     Route::post('list', [UserController::class, 'list']);       // menampilkan data user dalam bentuk json untuk datatables
37 |     Route::get('create', [UserController::class, 'create']);   // menampilkan form tambah user
38 |     Route::post('/', [UserController::class, 'store']);        // menyimpan data user
39 |     Route::get('/{id}', [UserController::class, 'show']);       // menampilkan detail user
40 |     Route::get('/{id}/edit', [UserController::class, 'edit']);  // menampilkan halaman form edit user
41 |     Route::put('{id}', [UserController::class, 'update']);    // menyimpan perubahan data user
42 |     Route::delete('{id}', [UserController::class, 'destroy']); // menghapus data user
43 |
44 | });

45 | }
```

4. Kita buat atau modifikasi penuh untuk [UserController.php](#). Kita buat fungsi [index\(\)](#) untuk menampilkan halaman awal user



```
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()
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         return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
26     }
}
```



The screenshot shows a code editor window with the following details:

- File path: Maggu_S2 > Jobsheet2 > PWL_PCS > app > Http > Controllers > UserController.php
- Code content:

```
use Illuminate\Http\Request;
use Yajra\DataTables\Facades\DataTables;
class UserController extends Controller
{
    // Menampilkan halaman awal user
    public function index()
    {
        $breadcrumb = (object) [
            'title' => 'Daftar User',
            'list' => ['Home', 'User']
        ];
        $page = (object) [
            'title' => 'Daftar user yang terdaftar dalam sistem'
        ];
        $activeMenu = 'user'; // set menu yang sedang aktif
        return view('index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
    }
}
```

5. Lalu kita buat view pada [PWL/resources/views/user/index.blade.php](#)



```
@extends('layouts.template')

@section('content')


### {{ $page->title }}



Tambah



| ID | Username | Nama | Level Pengguna | Aksi |
|----|----------|------|----------------|------|
|----|----------|------|----------------|------|


@endsection

@push('css')
@endpush

@push('js')
<script>
$(document).ready(function() {
    var dataUser = $('#table_user').DataTable({
        // serverSide: true, jika ingin menggunakan server side processing
        serverSide: true,
        ajax: {
            "url": "{{ url('user/list') }}",
            "dataType": "json",
            "type": "POST"
        },
        columns: [
        {
```



```
// nomor urut dari laravel datatable addIndexColumn()
data: "DT_RowIndex",
className: "text-center",
orderable: false,
searchable: false
},{
data: "username",
className: "",
// orderable: true, jika ingin kolom ini bisa diurutkan
orderable: true,
// searchable: true, jika ingin kolom ini bisa dicari
searchable: true
},{
data: "nama",
className: "",
orderable: true,
searchable: true
},{
// mengambil data level hasil dari ORM berelasi
data: "level.level_nama",
className: "",
orderable: false,
searchable: false
},{
data: "aksi",
className: "",
orderable: false,
searchable: false
}
]);
});
});
</script>
@endpush
```

```
UserController.php M index.blade.php M template.blade.php M
Minggu, 22 Januari 2023 Pada 10:45 AM Dalam direktori C:\xampp\htdocs\pwit\src\app\resources\views\user
Jobsheet 2 PWI_DoS resources > views > index.blade.php > script > ready() callback > el dataUser > columns

1 //extends('layouts.template')
2
3 @section('content')
4 <div class="card card-outline card-primary">
5   <div class="card-header">
6     <h3>{{ $page->title }}</h3>
7     <div class="card-tools">
8       <a class="btn btn-sm btn-primary mt-1" href="{{ url('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 @endsection
25 @push('css') @endpush
26 @push('js') @endpush
27 <script>
28 $(document).ready(function() {
29   var dataUser = $('#table_user').DataTable({
30     //serverSide: true, jika ingin menggunakan server side processing
31     //true,
32     //url: {{ url('user/list') }},
33     //dataType: "json",
34     //type: "POST"
35   },
36   columns: [
37     //buat untuk datatables addIndexColumn() data: "DT_RowIndex",
38     //className: "text-center",
39     //orderable: false,
40     //searchable: false
41   ],
42   {
43     data: "username",
44     className: "",
45     //orderable: true, jika ingin kolom ini bisa diurutkan orderable: true,
46     //searchable: true, jika ingin kolom ini bisa dicari searchable: true
47   },
48   {
49     data: "name",
50     className: ""
51   }
52 });
53 </script>
```



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

```
template.blade.php X
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('app.name', '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('adminlte/plugins/fontawesome-free/css/all.min.css') }}>
14  <!-- DataTables -->
15  <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}>
16  <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}>
17  <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}>
18  <!-- Theme style -->
19  <link rel="stylesheet" href="{{ asset('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>
```



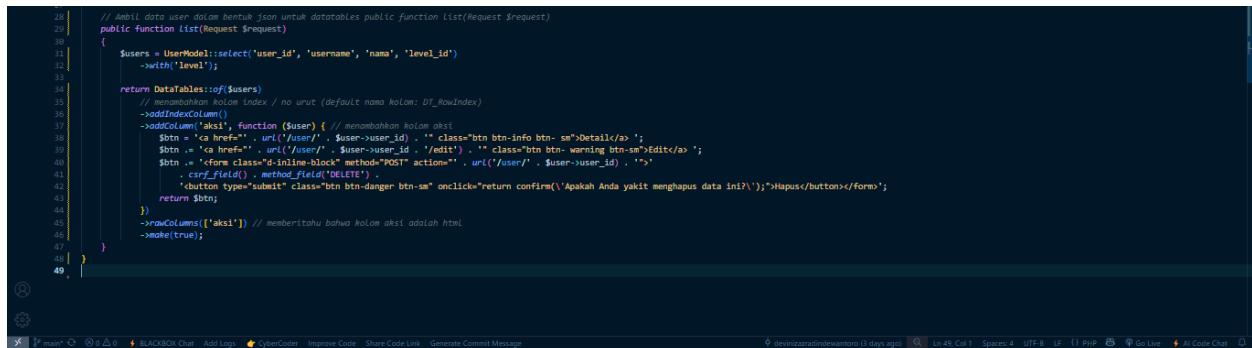
```
4   <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title>
8
9     <meta name="csrf-token" content="{{ csrf_token() }}><!-- Untuk mengirimkan token Laravel CSRF pada setiap request ajax -->
10
11     <!-- Google Font: Source Sans Pro -->
12     <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
13     <!-- Font Awesome -->
14     <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}>
15
16     <!-- datatables -->
17     <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}>
18     <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-responsive/css/responsive.bootstrap.min.css') }}>
19     <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-button/css/buttons.bootstrap4.min.css') }}>
20
21     <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}>
22
23   </head>
24
25   60   <!-- jQuery -->
26   61   <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}></script>
27   62   <!-- Bootstrap 4 -->
28   63   <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}></script>
29   64   <!-- DataTables & Plugins -->
30   65   <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}></script>
31   66   <script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}></script>
32   67   <script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}></script>
33   68   <script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}></script>
34   69   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}></script>
35   70   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}></script>
36   71   <script src="{{ asset('adminlte/plugins/jszip/jszip.min.js') }}></script>
37   72   <script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}></script>
38   73   <script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}></script>
39   74   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}></script>
40   75   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}></script>
41   76   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}></script>
42
43   77   <!-- AdminLTE App -->
44   78   <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}></script>
45
46   79   <script>
47   // Untuk mengirimkan token Laravel CSRF pada setiap request ajax
48   $.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
49
50
51   80   <!-- Digunakan untuk memanggil custom js dari perintah push('js') pada masing-masing view -->
52   81   </script>
53
54   82   </body>
55   83   </html>
56
57
58   84   <!-- jQuery -->
59   <script src="{{ asset('/plugins/jquery/jquery.min.js') }}></script>
60   <!-- Bootstrap 4 -->
61   <script src="{{ asset('/plugins/bootstrap/js/bootstrap.bundle.min.js') }}></script>
62
63   <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}></script>
64   <script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}></script>
65   <script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}></script>
66   <script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}></script>
67   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}></script>
68   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}></script>
69   <script src="{{ asset('adminlte/plugins/jszip/jszip.min.js') }}></script>
70   <script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}></script>
71   <script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}></script>
72   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}></script>
73   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}></script>
74   <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}></script>
75
76   <!-- AdminLTE App -->
77   <script src="{{ asset('/dist/js/adminlte.min.js') }}></script>
78
79   <!-- Untuk mengirimkan token Laravel CSRF pada setiap request ajax
80   $.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
81
82   </script>
83
84   <!-- AdminLTE For demo purposes -->
85   <script src="{{ asset('../..../dist/js/demo.js') }}></script>
86
87   </body>
88   </html>
```

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



```
// Ambil data user dalam bentuk json untuk datatables
public function list(Request $request)
{
    $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
        ->with('level');

    return DataTables::of($users)
        // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
        ->addIndexColumn()
        ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
            $btn = '<a href="'.url('/user/' . $user->user_id).'" class="btn btn-info btn-sm">Detail</a> ';
            $btn .= '<a href="'.url('/user/' . $user->user_id . '/edit').'" class="btn btn-warning btn-sm">Edit</a> ';
            $btn .= '<form class="d-inline-block" method="POST" action="'.url('/user/' . $user->user_id).'">
                . csrf_field() . method_field('DELETE') .
                <button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\'Apakah Anda yakin menghapus data ini?\');">Hapus</button></form>';
            return $btn;
        })
        ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
        ->make(true);
}
```



The screenshot shows the same PHP code as the previous code block, but it is displayed within a code editor interface. The editor has a dark theme with syntax highlighting. The code is numbered from 28 to 49 on the left. Below the code, there is a toolbar with various icons for file operations like save, open, and copy, as well as tabs for different code files and a status bar at the bottom.

```
28 // Ambil data user dalam bentuk json untuk datatables public function list(Request $request)
29 {
30     $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
31         ->with('level');
32
33     return DataTables::of($users)
34         // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
35         ->addIndexColumn()
36         ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
37             $btn = '<a href="'.url('/user/' . $user->user_id).'" class="btn btn-info btn-sm">Detail</a> ';
38             $btn .= '<a href="'.url('/user/' . $user->user_id . '/edit').'" class="btn btn-warning btn-sm">Edit</a> ';
39             $btn .= '<form class="d-inline-block" method="POST" action="'.url('/user/' . $user->user_id).'">
40                 . csrf_field() . method_field('DELETE') .
41                 <button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Apakah Anda yakin menghapus data ini?');">Hapus</button></form>';
42             return $btn;
43         })
44         ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
45         ->make(true);
46     }
47 }
```

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



PWL - Starter Code

Home Contact

Daftar User

Daftar user yang terdaftar dalam sistem

Tambah

Search:

Filter: - Semua - Level Pengguna

Show 10 entries

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	manager	Manager	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	staff	Staff/Kasir	Staff	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	customer-1	Pelanggan Pertama	Pelanggan	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	manager_dua	Manager 2	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
6	manager22	Manager Dua Dua	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
7	manager33	Manager Tiga Tiga	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
8	manager56	Manager55	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
9	manager12	Manager11	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

9. Selanjutnya kita modifikasi [UserController.php](#) untuk form tambah data user



```
// Menampilkan halaman form tambah user
public function create()
{
    $breadcrumb = (object) [
        'title' => 'Tambah User',
        'list' => ['Home', 'User', 'Tambah']
    ];

    $page = (object) [
        'title' => 'Tambah user baru'
    ];

    $level = LevelModel::all(); // ambil data level untuk ditampilkan di form
    $activeMenu = 'user'; // set menu yang sedang aktif

    return view('user.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
}

// Menampilkan halaman form tambah user
public function create()
{
    $breadcrumb = (object) [
        'title' => 'Tambah User',
        'list' => ['Home', 'User', 'Tambah']
    ];

    $page = (object) [
        'title' => 'Tambah user baru'
    ];

    $level = LevelModel::all();
    $activeMenu = 'user'; // set menu yang sedang aktif

    return view('user.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu ]);
}
```

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

```
@extends('layouts.template')

@section('content')


### {{ $page->title }}



@csrf


<label class="col-1 control-label col-form-label">Level</label>


<select class="form-control" id="level_id" name="level_id" required>
<option value="">- Pilih Level -</option>
@foreach($level as $item)
<option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
@endforeach
</select>
@error('level_id')
<small class="form-text text-danger">{{ $message }}</small>
@enderror



<label class="col-1 control-label col-form-label">Username</label>


<input type="text" class="form-control" id="username" name="username" value="{{ old('username') }}" required>
@error('username')
<small class="form-text text-danger">{{ $message }}</small>
@enderror



<label class="col-1 control-label col-form-label">>Nama</label>


<input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama') }}" required>
@error('nama')
<small class="form-text text-danger">{{ $message }}</small>


```



```
<div class="col-11">
    <input type="password" class="form-control" id="password" name="password"
required>
    @error('password')
        <small class="form-text text-danger">{{ $message }}</small>
    @enderror
</div>
</div>
<div class="form-group row">
    <label class="col-1 control-label col-form-label"></label>
    <div class="col-11">
        <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
        <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}">Kembali</a>
    </div>
</div>
</form>
</div>
</div>
@endsection
@push('css')
@endpush
@push('js')
@endpush
```

```
Minggu_52 > Jobsheet52 > PWL_POS > resources > views > user > create.blade.php ...  
1  @extends('layouts.template')
2  @section('content')
3  <div class="card card-outline card-primary">
4      <div class="card-header">
5          <h3 class="card-title">{{ $page->title }}Simpan
```



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

```
// Menyimpan data user baru
public function store(Request $request)
{
    $request->validate([
        // username harus diisi, berupa string, minimal 3 karakter, dan bernilai unik di tabel m_user kolom username
        'username' => 'required|string|min:3|unique:m_user,username',
        'nama'      => 'required|string|max:100', // nama harus diisi, berupa string, dan maksimal 100 karakter
        'password'  => 'required|min:5',           // password harus diisi dan minimal 5 karakter
        'level_id'   => 'required|integer'         // level_id harus diisi dan berupa angka
    ]);

    UserModel::create([
        'username' => $request->username,
        'nama'      => $request->nama,
        'password'  => bcrypt($request->password), // password dienkripsi sebelum disimpan
        'level_id'   => $request->level_id
    ]);

    return redirect('/user')->with('success', 'Data user berhasil disimpan');
}

// Menyimpan data user baru
public function store(Request $request)
{
    $request->validate([
        // username harus diisi, berupa string, minimal 3 karakter, dan bernilai unik di tabel m_user kolom username
        'username' => 'required|string|min:3|unique:m_user,username',
        'nama'      => 'required|string|max:100', // nama harus diisi, berupa string, dan maksimal 100 karakter
        'password'  => 'required|min:5',           // password harus diisi dan minimal 5 karakter
        'level_id'   => 'required|integer',         // Level_id harus diisi dan berupa angka
    ]);

    UserModel::create([
        'username' => $request->username,
        'nama'      => $request->nama,
        'password'  => bcrypt($request->password), // password dienkripsi sebelum disimpan
        'level_id'   => $request->level_id
    ]);

    return redirect('/user')->with('success', 'Data user berhasil disimpan');
}
```

12. Sekarang coba kalian buka form tambah data user dengan klik tombol tambah. Amati dan pelajari..!!!

Tambah User

Tambah user baru

Level - Pilih Level -

Username

Nama

Password

Simpan Kembali

Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0



PWL - Starter Code

Home Contact

Daftar User

Daftar user yang terdaftar dalam sistem

Tambah

Show 10 entries Search:

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	Detail Edit Hapus
4	customer-1	Pelanggan Pertama	Pelanggan	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager22	Manager Dua Dua	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	Beni	Beni Santoso	Administrator	Detail Edit Hapus

phpMyAdmin

Server: localhost:3306 Database: pwl_pos Table: m_user

Browse Structure SQL Search Insert Export Privileges Operations Triggers

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

#	user_id	username	nama	password	level_id	created_at	updated_at
1	1	admin	Administrator	\$2y\$12\$GIVmOuWtTdulMAfomWOU6U2hBuHzDbTRFwX...	1	NULL	NULL
2	2	manager	Manager	\$2y\$12\$JDS23lOzLnlMKRm0f05Qe8yRCYuwHuzrN1bxqJ8Z...	2	NULL	NULL
3	3	staff	Staff/Kasir	\$2y\$12\$2efFB/Wmvtc7D2syRkp2vuS42DGoddv0pjBEOoYAK...	3	NULL	NULL
16	16	customer-1	Pelanggan Pertama	\$2y\$12\$t.sBvA3ZYxDR2WuEJlhHONIBcB.206mLuaEZgv5gSH...	5	NULL	2025-03-05 02:42:14
17	17	manager_dua	Manager 2	\$2y\$12\$5a0YrAf4v2C5wOEBOGrusu0G67nDILPEdETJ...e...	2	2025-03-06 06:51:27	2025-03-06 06:51:27
18	18	manager22	Manager Dua Dua	\$2y\$12\$WjdtdJzeNhY0Ygzf78lOEgmerHATZQe4lVTR7go...	2	2025-03-06 11:45:08	2025-03-06 11:45:08
19	19	manager33	Manager Tiga Tiga	\$2y\$12\$FqrPTnGWUEhizgf2kcGuF5O1OX/p2U3uG9NqCx1F...	2	2025-03-06 12:02:23	2025-03-06 12:02:23
20	20	manager56	Manager55	\$2y\$12\$nl5rDKL4oRo'nkGlinedC7FKyxeVFUT5w340FQjn...	2	2025-03-06 12:09:26	2025-03-06 12:09:26
21	21	manager12	Manager11	\$2y\$12\$meBTSK29NQob.Kmf0o9Tkuu5xA7vk3ns8A5ecZ...	2	2025-03-06 12:17:55	2025-03-06 12:17:55
23	23	Beni	Beni Santoso	\$2y\$12\$juVZJlVnmYa5pNqmVYUOeE21tNeD0ZBwAWatPxoF...	1	2025-03-23 22:00:31	2025-03-23 22:00:31

Terdapat tampilan untuk menambah user dan jika sudah melakukan pengisian di dalam form tambah user, user baru akan otomatis bertambah pada database dan juga ke dalam tampilan data user.

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

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



Detail User

ID	23
Level	Administrator
Username	Beni
Nama	Beni Santoso
Password	*****

Kembali

Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0

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

```
// Menampilkan detail user
public function show(string $id)
{
    $user = UserModel::with('level')->find($id);

    $breadcrumb = (object) [
        'title' => 'Detail User',
        'list' => ['Home', 'User', 'Detail']
    ];

    $page = (object) [
        'title' => 'Detail user'
    ];

    $activeMenu = 'user'; // set menu yang sedang aktif

    return view('user.show', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
}
```



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>

```
Minggu_52 > Jobsheet5 > PWL_POC > app > Http > Controllers > UserController.php > UserController > @__list > Closure
16     class UserController extends Controller
17     {
18         public function store(Request $request)
19         {
20             User::create([
21                 'username' => $request->username,
22                 'name' => $request->name,
23                 'password' => bcrypt($request->password), // password dienkripsi sebelum disimpan
24                 'level_id' => $request->level_id
25             ]);
26
27             return redirect('/user')->with('success', 'Data user berhasil disimpan');
28         }
29
30         // Menampilkan detail user
31
32         public function show(string $id)
33         {
34             $user = UserModel::with('level')->find($id);
35
36             $breadcrumb = (object) [
37                 'title' => 'Detail User',
38                 'list' => ['Home', 'User', 'Detail']
39             ];
40
41             $page = (object) [
42                 'title' => 'Detail user'
43             ];
44
45             $activeMenu = 'user'; // set menu yang sedang aktif
46
47             return view('user.show', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
48         }
49
50         // Menampilkan halaman form edit user
51
52         public function edit(string $id)
53         {
54             $user = UserModel::find($id);
55             $level = LevelModel::all();
56
57             $breadcrumb = (object) [
58                 'title' => 'Edit User',
59                 'list' => ['Home', 'User', 'Edit']
60             ];
61
62             $page = (object) [
63                 'title' => 'Edit User'
64             ];
65
66             $activeMenu = 'user'; // set menu yang sedang aktif
67
68             return view('user.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
69         }
70
71         // Menampilkan halaman form tambah user
72
73         public function create()
74         {
75             $level = LevelModel::all();
76
77             $breadcrumb = (object) [
78                 'title' => 'Tambah User',
79                 'list' => ['Home', 'User', 'Create']
80             ];
81
82             $page = (object) [
83                 'title' => 'Tambah User'
84             ];
85
86             $activeMenu = 'user'; // set menu yang sedang aktif
87
88             return view('user.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
89         }
90
91         // Menampilkan halaman list user
92
93         public function index()
94         {
95             $users = UserModel::all();
96
97             $breadcrumb = (object) [
98                 'title' => 'List User',
99                 'list' => ['Home', 'User', 'Index']
100            ];
101
102            $page = (object) [
103                'title' => 'List User'
104            ];
105
106            $activeMenu = 'user'; // set menu yang sedang aktif
107
108            return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'users' => $users, 'activeMenu' => $activeMenu]);
109        }
110
111        // Menampilkan halaman detail user
112
113        public function detail(string $id)
114        {
115            $user = UserModel::with('level')->find($id);
116
117            $breadcrumb = (object) [
118                'title' => 'Detail User',
119                'list' => ['Home', 'User', 'Detail']
120            ];
121
122            $page = (object) [
123                'title' => 'Detail User'
124            ];
125
126            $activeMenu = 'user'; // set menu yang sedang aktif
127
128            return view('user.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
129        }
130
131        // Menampilkan halaman edit user
132
133        public function editUser()
134        {
135            $user = UserModel::all();
136
137            $breadcrumb = (object) [
138                'title' => 'Edit User',
139                'list' => ['Home', 'User', 'Edit']
140            ];
141
142            $page = (object) [
143                'title' => 'Edit User'
144            ];
145
146            $activeMenu = 'user'; // set menu yang sedang aktif
147
148            return view('user.editUser', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
149        }
150
151        // Menampilkan halaman tambah user
152
153        public function createUser()
154        {
155            $level = LevelModel::all();
156
157            $breadcrumb = (object) [
158                'title' => 'Tambah User',
159                'list' => ['Home', 'User', 'Create']
160            ];
161
162            $page = (object) [
163                'title' => 'Tambah User'
164            ];
165
166            $activeMenu = 'user'; // set menu yang sedang aktif
167
168            return view('user.createUser', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
169        }
170
171        // Menampilkan halaman list level
172
173        public function indexLevel()
174        {
175            $levels = LevelModel::all();
176
177            $breadcrumb = (object) [
178                'title' => 'List Level',
179                'list' => ['Home', 'Level', 'Index']
180            ];
181
182            $page = (object) [
183                'title' => 'List Level'
184            ];
185
186            $activeMenu = 'level'; // set menu yang sedang aktif
187
188            return view('level.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'levels' => $levels, 'activeMenu' => $activeMenu]);
189        }
190
191        // Menampilkan halaman detail level
192
193        public function detailLevel(string $id)
194        {
195            $level = LevelModel::with('user')->find($id);
196
197            $breadcrumb = (object) [
198                'title' => 'Detail Level',
199                'list' => ['Home', 'Level', 'Detail']
200            ];
201
202            $page = (object) [
203                'title' => 'Detail Level'
204            ];
205
206            $activeMenu = 'level'; // set menu yang sedang aktif
207
208            return view('level.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
209        }
210
211        // Menampilkan halaman edit level
212
213        public function editLevel()
214        {
215            $levels = LevelModel::all();
216
217            $breadcrumb = (object) [
218                'title' => 'Edit Level',
219                'list' => ['Home', 'Level', 'Edit']
220            ];
221
222            $page = (object) [
223                'title' => 'Edit Level'
224            ];
225
226            $activeMenu = 'level'; // set menu yang sedang aktif
227
228            return view('level.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
229        }
230
231        // Menampilkan halaman tambah level
232
233        public function createLevel()
234        {
235            $level = LevelModel::all();
236
237            $breadcrumb = (object) [
238                'title' => 'Tambah Level',
239                'list' => ['Home', 'Level', 'Create']
240            ];
241
242            $page = (object) [
243                'title' => 'Tambah Level'
244            ];
245
246            $activeMenu = 'level'; // set menu yang sedang aktif
247
248            return view('level.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
249        }
250
251        // Menampilkan halaman list role
252
253        public function indexRole()
254        {
255            $roles = RoleModel::all();
256
257            $breadcrumb = (object) [
258                'title' => 'List Role',
259                'list' => ['Home', 'Role', 'Index']
260            ];
261
262            $page = (object) [
263                'title' => 'List Role'
264            ];
265
266            $activeMenu = 'role'; // set menu yang sedang aktif
267
268            return view('role.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'roles' => $roles, 'activeMenu' => $activeMenu]);
269        }
270
271        // Menampilkan halaman detail role
272
273        public function detailRole(string $id)
274        {
275            $role = RoleModel::with('level')->find($id);
276
277            $breadcrumb = (object) [
278                'title' => 'Detail Role',
279                'list' => ['Home', 'Role', 'Detail']
280            ];
281
282            $page = (object) [
283                'title' => 'Detail Role'
284            ];
285
286            $activeMenu = 'role'; // set menu yang sedang aktif
287
288            return view('role.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'role' => $role, 'activeMenu' => $activeMenu]);
289        }
290
291        // Menampilkan halaman edit role
292
293        public function editRole()
294        {
295            $roles = RoleModel::all();
296
297            $breadcrumb = (object) [
298                'title' => 'Edit Role',
299                'list' => ['Home', 'Role', 'Edit']
300            ];
301
302            $page = (object) [
303                'title' => 'Edit Role'
304            ];
305
306            $activeMenu = 'role'; // set menu yang sedang aktif
307
308            return view('role.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
309        }
310
311        // Menampilkan halaman tambah role
312
313        public function createRole()
314        {
315            $level = LevelModel::all();
316
317            $breadcrumb = (object) [
318                'title' => 'Tambah Role',
319                'list' => ['Home', 'Role', 'Create']
320            ];
321
322            $page = (object) [
323                'title' => 'Tambah Role'
324            ];
325
326            $activeMenu = 'role'; // set menu yang sedang aktif
327
328            return view('role.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
329        }
330
331        // Menampilkan halaman list permission
332
333        public function indexPermission()
334        {
335            $permissions = PermissionModel::all();
336
337            $breadcrumb = (object) [
338                'title' => 'List Permission',
339                'list' => ['Home', 'Permission', 'Index']
340            ];
341
342            $page = (object) [
343                'title' => 'List Permission'
344            ];
345
346            $activeMenu = 'permission'; // set menu yang sedang aktif
347
348            return view('permission.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'permissions' => $permissions, 'activeMenu' => $activeMenu]);
349        }
350
351        // Menampilkan halaman detail permission
352
353        public function detailPermission(string $id)
354        {
355            $permission = PermissionModel::with('role')->find($id);
356
357            $breadcrumb = (object) [
358                'title' => 'Detail Permission',
359                'list' => ['Home', 'Permission', 'Detail']
360            ];
361
362            $page = (object) [
363                'title' => 'Detail Permission'
364            ];
365
366            $activeMenu = 'permission'; // set menu yang sedang aktif
367
368            return view('permission.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'permission' => $permission, 'activeMenu' => $activeMenu]);
369        }
370
371        // Menampilkan halaman edit permission
372
373        public function editPermission()
374        {
375            $permissions = PermissionModel::all();
376
377            $breadcrumb = (object) [
378                'title' => 'Edit Permission',
379                'list' => ['Home', 'Permission', 'Edit']
380            ];
381
382            $page = (object) [
383                'title' => 'Edit Permission'
384            ];
385
386            $activeMenu = 'permission'; // set menu yang sedang aktif
387
388            return view('permission.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
389        }
390
391        // Menampilkan halaman tambah permission
392
393        public function createPermission()
394        {
395            $role = RoleModel::all();
396
397            $breadcrumb = (object) [
398                'title' => 'Tambah Permission',
399                'list' => ['Home', 'Permission', 'Create']
400            ];
401
402            $page = (object) [
403                'title' => 'Tambah Permission'
404            ];
405
406            $activeMenu = 'permission'; // set menu yang sedang aktif
407
408            return view('permission.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
409        }
410
411        // Menampilkan halaman list menu
412
413        public function indexMenu()
414        {
415            $menus = MenuModel::all();
416
417            $breadcrumb = (object) [
418                'title' => 'List Menu',
419                'list' => ['Home', 'Menu', 'Index']
420            ];
421
422            $page = (object) [
423                'title' => 'List Menu'
424            ];
425
426            $activeMenu = 'menu'; // set menu yang sedang aktif
427
428            return view('menu.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'menus' => $menus, 'activeMenu' => $activeMenu]);
429        }
430
431        // Menampilkan halaman detail menu
432
433        public function detailMenu(string $id)
434        {
435            $menu = MenuModel::with('parent')->find($id);
436
437            $breadcrumb = (object) [
438                'title' => 'Detail Menu',
439                'list' => ['Home', 'Menu', 'Detail']
440            ];
441
442            $page = (object) [
443                'title' => 'Detail Menu'
444            ];
445
446            $activeMenu = 'menu'; // set menu yang sedang aktif
447
448            return view('menu.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'menu' => $menu, 'activeMenu' => $activeMenu]);
449        }
450
451        // Menampilkan halaman edit menu
452
453        public function editMenu()
454        {
455            $menus = MenuModel::all();
456
457            $breadcrumb = (object) [
458                'title' => 'Edit Menu',
459                'list' => ['Home', 'Menu', 'Edit']
460            ];
461
462            $page = (object) [
463                'title' => 'Edit Menu'
464            ];
465
466            $activeMenu = 'menu'; // set menu yang sedang aktif
467
468            return view('menu.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
469        }
470
471        // Menampilkan halaman tambah menu
472
473        public function createMenu()
474        {
475            $parent = MenuModel::all();
476
477            $breadcrumb = (object) [
478                'title' => 'Tambah Menu',
479                'list' => ['Home', 'Menu', 'Create']
480            ];
481
482            $page = (object) [
483                'title' => 'Tambah Menu'
484            ];
485
486            $activeMenu = 'menu'; // set menu yang sedang aktif
487
488            return view('menu.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
489        }
490
491        // Menampilkan halaman list role
492
493        public function indexRole()
494        {
495            $roles = RoleModel::all();
496
497            $breadcrumb = (object) [
498                'title' => 'List Role',
499                'list' => ['Home', 'Role', 'Index']
500            ];
501
502            $page = (object) [
503                'title' => 'List Role'
504            ];
505
506            $activeMenu = 'role'; // set menu yang sedang aktif
507
508            return view('role.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'roles' => $roles, 'activeMenu' => $activeMenu]);
509        }
510
511        // Menampilkan halaman detail role
512
513        public function detailRole(string $id)
514        {
515            $role = RoleModel::with('level')->find($id);
516
517            $breadcrumb = (object) [
518                'title' => 'Detail Role',
519                'list' => ['Home', 'Role', 'Detail']
520            ];
521
522            $page = (object) [
523                'title' => 'Detail Role'
524            ];
525
526            $activeMenu = 'role'; // set menu yang sedang aktif
527
528            return view('role.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'role' => $role, 'activeMenu' => $activeMenu]);
529        }
530
531        // Menampilkan halaman edit role
532
533        public function editRole()
534        {
535            $roles = RoleModel::all();
536
537            $breadcrumb = (object) [
538                'title' => 'Edit Role',
539                'list' => ['Home', 'Role', 'Edit']
540            ];
541
542            $page = (object) [
543                'title' => 'Edit Role'
544            ];
545
546            $activeMenu = 'role'; // set menu yang sedang aktif
547
548            return view('role.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
549        }
550
551        // Menampilkan halaman tambah role
552
553        public function createRole()
554        {
555            $level = LevelModel::all();
556
557            $breadcrumb = (object) [
558                'title' => 'Tambah Role',
559                'list' => ['Home', 'Role', 'Create']
560            ];
561
562            $page = (object) [
563                'title' => 'Tambah Role'
564            ];
565
566            $activeMenu = 'role'; // set menu yang sedang aktif
567
568            return view('role.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
569        }
570
571        // Menampilkan halaman list permission
572
573        public function indexPermission()
574        {
575            $permissions = PermissionModel::all();
576
577            $breadcrumb = (object) [
578                'title' => 'List Permission',
579                'list' => ['Home', 'Permission', 'Index']
580            ];
581
582            $page = (object) [
583                'title' => 'List Permission'
584            ];
585
586            $activeMenu = 'permission'; // set menu yang sedang aktif
587
588            return view('permission.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'permissions' => $permissions, 'activeMenu' => $activeMenu]);
589        }
590
591        // Menampilkan halaman detail permission
592
593        public function detailPermission(string $id)
594        {
595            $permission = PermissionModel::with('role')->find($id);
596
597            $breadcrumb = (object) [
598                'title' => 'Detail Permission',
599                'list' => ['Home', 'Permission', 'Detail']
600            ];
601
602            $page = (object) [
603                'title' => 'Detail Permission'
604            ];
605
606            $activeMenu = 'permission'; // set menu yang sedang aktif
607
608            return view('permission.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'permission' => $permission, 'activeMenu' => $activeMenu]);
609        }
610
611        // Menampilkan halaman edit permission
612
613        public function editPermission()
614        {
615            $permissions = PermissionModel::all();
616
617            $breadcrumb = (object) [
618                'title' => 'Edit Permission',
619                'list' => ['Home', 'Permission', 'Edit']
620            ];
621
622            $page = (object) [
623                'title' => 'Edit Permission'
624            ];
625
626            $activeMenu = 'permission'; // set menu yang sedang aktif
627
628            return view('permission.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
629        }
630
631        // Menampilkan halaman tambah permission
632
633        public function createPermission()
634        {
635            $role = RoleModel::all();
636
637            $breadcrumb = (object) [
638                'title' => 'Tambah Permission',
639                'list' => ['Home', 'Permission', 'Create']
640            ];
641
642            $page = (object) [
643                'title' => 'Tambah Permission'
644            ];
645
646            $activeMenu = 'permission'; // set menu yang sedang aktif
647
648            return view('permission.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
649        }
650
651        // Menampilkan halaman list menu
652
653        public function indexMenu()
654        {
655            $menus = MenuModel::all();
656
657            $breadcrumb = (object) [
658                'title' => 'List Menu',
659                'list' => ['Home', 'Menu', 'Index']
660            ];
661
662            $page = (object) [
663                'title' => 'List Menu'
664            ];
665
666            $activeMenu = 'menu'; // set menu yang sedang aktif
667
668            return view('menu.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'menus' => $menus, 'activeMenu' => $activeMenu]);
669        }
670
671        // Menampilkan halaman detail menu
672
673        public function detailMenu(string $id)
674        {
675            $menu = MenuModel::with('parent')->find($id);
676
677            $breadcrumb = (object) [
678                'title' => 'Detail Menu',
679                'list' => ['Home', 'Menu', 'Detail']
680            ];
681
682            $page = (object) [
683                'title' => 'Detail Menu'
684            ];
685
686            $activeMenu = 'menu'; // set menu yang sedang aktif
687
688            return view('menu.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'menu' => $menu, 'activeMenu' => $activeMenu]);
689        }
690
691        // Menampilkan halaman edit menu
692
693        public function editMenu()
694        {
695            $menus = MenuModel::all();
696
697            $breadcrumb = (object) [
698                'title' => 'Edit Menu',
699                'list' => ['Home', 'Menu', 'Edit']
700            ];
701
702            $page = (object) [
703                'title' => 'Edit Menu'
704            ];
705
706            $activeMenu = 'menu'; // set menu yang sedang aktif
707
708            return view('menu.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
709        }
710
711        // Menampilkan halaman tambah menu
712
713        public function createMenu()
714        {
715            $parent = MenuModel::all();
716
717            $breadcrumb = (object) [
718                'title' => 'Tambah Menu',
719                'list' => ['Home', 'Menu', 'Create']
720            ];
721
722            $page = (object) [
723                'title' => 'Tambah Menu'
724            ];
725
726            $activeMenu = 'menu'; // set menu yang sedang aktif
727
728            return view('menu.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
729        }
730
731        // Menampilkan halaman list role
732
733        public function indexRole()
734        {
735            $roles = RoleModel::all();
736
737            $breadcrumb = (object) [
738                'title' => 'List Role',
739                'list' => ['Home', 'Role', 'Index']
740            ];
741
742            $page = (object) [
743                'title' => 'List Role'
744            ];
745
746            $activeMenu = 'role'; // set menu yang sedang aktif
747
748            return view('role.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'roles' => $roles, 'activeMenu' => $activeMenu]);
749        }
750
751        // Menampilkan halaman detail role
752
753        public function detailRole(string $id)
754        {
755            $role = RoleModel::with('level')->find($id);
756
757            $breadcrumb = (object) [
758                'title' => 'Detail Role',
759                'list' => ['Home', 'Role', 'Detail']
760            ];
761
762            $page = (object) [
763                'title' => 'Detail Role'
764            ];
765
766            $activeMenu = 'role'; // set menu yang sedang aktif
767
768            return view('role.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'role' => $role, 'activeMenu' => $activeMenu]);
769        }
770
771        // Menampilkan halaman edit role
772
773        public function editRole()
774        {
775            $roles = RoleModel::all();
776
777            $breadcrumb = (object) [
778                'title' => 'Edit Role',
779                'list' => ['Home', 'Role', 'Edit']
780            ];
781
782            $page = (object) [
783                'title' => 'Edit Role'
784            ];
785
786            $activeMenu = 'role'; // set menu yang sedang aktif
787
788            return view('role.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
789        }
790
791        // Menampilkan halaman tambah role
792
793        public function createRole()
794        {
795            $level = LevelModel::all();
796
797            $breadcrumb = (object) [
798                'title' => 'Tambah Role',
799                'list' => ['Home', 'Role', 'Create']
800            ];
801
802            $page = (object) [
803                'title' => 'Tambah Role'
804            ];
805
806            $activeMenu = 'role'; // set menu yang sedang aktif
807
808            return view('role.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
809        }
810
811        // Menampilkan halaman list permission
812
813        public function indexPermission()
814        {
815            $permissions = PermissionModel::all();
816
817            $breadcrumb = (object) [
818                'title' => 'List Permission',
819                'list' => ['Home', 'Permission', 'Index']
820            ];
821
822            $page = (object) [
823                'title' => 'List Permission'
824            ];
825
826            $activeMenu = 'permission'; // set menu yang sedang aktif
827
828            return view('permission.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'permissions' => $permissions, 'activeMenu' => $activeMenu]);
829        }
830
831        // Menampilkan halaman detail permission
832
833        public function detailPermission(string $id)
834        {
835            $permission = PermissionModel::with('role')->find($id);
836
837            $breadcrumb = (object) [
838                'title' => 'Detail Permission',
839                'list' => ['Home', 'Permission', 'Detail']
840            ];
841
842            $page = (object) [
843                'title' => 'Detail Permission'
844            ];
845
846            $activeMenu = 'permission'; // set menu yang sedang aktif
847
848            return view('permission.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'permission' => $permission, 'activeMenu' => $activeMenu]);
849        }
850
851        // Menampilkan halaman edit permission
852
853        public function editPermission()
854        {
855            $permissions = PermissionModel::all();
856
857            $breadcrumb = (object) [
858                'title' => 'Edit Permission',
859                'list' => ['Home', 'Permission', 'Edit']
860            ];
861
862            $page = (object) [
863                'title' => 'Edit Permission'
864            ];
865
866            $activeMenu = 'permission'; // set menu yang sedang aktif
867
868            return view('permission.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
869        }
870
871        // Menampilkan halaman tambah permission
872
873        public function createPermission()
874        {
875            $role = RoleModel::all();
876
877            $breadcrumb = (object) [
878                'title' => 'Tambah Permission',
879                'list' => ['Home', 'Permission', 'Create']
880            ];
881
882            $page = (object) [
883                'title' => 'Tambah Permission'
884            ];
885
886            $activeMenu = 'permission'; // set menu yang sedang aktif
887
888            return view('permission.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
889        }
890
891        // Menampilkan halaman list menu
892
893        public function indexMenu()
894        {
895            $menus = MenuModel::all();
896
897            $breadcrumb = (object) [
898                'title' => 'List Menu',
899                'list' => ['Home', 'Menu', 'Index']
900            ];
901
902            $page = (object) [
903                'title' => 'List Menu'
904            ];
905
906            $activeMenu = 'menu'; // set menu yang sedang aktif
907
908            return view('menu.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'menus' => $menus, 'activeMenu' => $activeMenu]);
909        }
910
911        // Menampilkan halaman detail menu
912
913        public function detailMenu(string $id)
914        {
915            $menu = MenuModel::with('parent')->find($id);
916
917            $breadcrumb = (object) [
918                'title' => 'Detail Menu',
919                'list' => ['Home', 'Menu', 'Detail']
920            ];
921
922            $page = (object) [
923                'title' => 'Detail Menu'
924            ];
925
926            $activeMenu = 'menu'; // set menu yang sedang aktif
927
928            return view('menu.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'menu' => $menu, 'activeMenu' => $activeMenu]);
929        }
930
931        // Menampilkan halaman edit menu
932
933        public function editMenu()
934        {
935            $menus = MenuModel::all();
936
937            $breadcrumb = (object) [
938                'title' => 'Edit Menu',
939                'list' => ['Home', 'Menu', 'Edit']
940            ];
941
942            $page = (object) [
943                'title' => 'Edit Menu'
944            ];
945
946            $activeMenu = 'menu'; // set menu yang sedang aktif
947
948            return view('menu.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
949        }
950
951        // Menampilkan halaman tambah menu
952
953        public function createMenu()
954        {
955            $parent = MenuModel::all();
956
957            $breadcrumb = (object) [
958                'title' => 'Tambah Menu',
959                'list' => ['Home', 'Menu', 'Create']
960            ];
961
962            $page = (object) [
963                'title' => 'Tambah Menu'
964            ];
965
966            $activeMenu = 'menu'; // set menu yang sedang aktif
967
968            return view('menu.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
969        }
970
971        // Menampilkan halaman list role
972
973        public function indexRole()
974        {
975            $roles = RoleModel::all();
976
977            $breadcrumb = (object) [
978                'title' => 'List Role',
979                'list' => ['Home', 'Role', 'Index']
980            ];
981
982            $page = (object) [
983                'title' => 'List Role'
984            ];
985
986            $activeMenu = 'role'; // set menu yang sedang aktif
987
988            return view('role.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'roles' => $roles, 'activeMenu' => $activeMenu]);
989        }
990
991        // Menampilkan halaman detail role
992
993        public function detailRole(string $id)
994        {
995            $role = RoleModel::with('level')->find($id);
996
997            $breadcrumb = (object) [
998                'title' => 'Detail Role',
999                'list' => ['Home', 'Role', 'Detail']
1000            ];
1001
1002            $page = (object) [
1003                'title' => 'Detail Role'
1004            ];
1005
1006            $activeMenu = 'role'; // set menu yang sedang aktif
1007
1008            return view('role.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'role' => $role, 'activeMenu' => $activeMenu]);
1009        }
1010
1011        // Menampilkan halaman edit role
1012
1013        public function editRole()
1014        {
1015            $roles = RoleModel::all();
1016
1017            $breadcrumb = (object) [
1018                'title' => 'Edit Role',
1019                'list' => ['Home', 'Role', 'Edit']
1020            ];
1021
1022            $page = (object) [
1023                'title' => 'Edit Role'
1024            ];
1025
1026            $activeMenu = 'role'; // set menu yang sedang aktif
1027
1028            return view('role.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
1029        }
1030
1031        // Menampilkan halaman tambah role
1032
1033        public function createRole()
1034        {
1035            $level = LevelModel::all();
1036
1037            $breadcrumb = (object) [
1038                'title' => 'Tambah Role',
1039                'list' => ['Home', 'Role', 'Create']
1040            ];
1041
1042            $page = (object) [
1043                'title' => 'Tambah Role'
1044            ];
1045
1046            $activeMenu = 'role'; // set menu yang sedang aktif
1047
1048            return view('role.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
1049        }
1050
1051        // Menampilkan halaman list permission
1052
1053        public function indexPermission()
1054        {
1055            $permissions = PermissionModel::all();
1056
1057            $breadcrumb = (object) [
1058                'title' => 'List Permission',
1059                'list' => ['Home', 'Permission', 'Index']
1060            ];
1061
1062            $page = (object) [
1063                'title' => 'List Permission'
1064            ];
1065
1066            $activeMenu = 'permission'; // set menu yang sedang aktif
1067
1068            return view('permission.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'permissions' => $permissions, 'activeMenu' => $activeMenu]);
1069        }
1070
1071        // Menampilkan halaman detail permission
1072
1073        public function detailPermission(string $id)
1074        {
1075            $permission = PermissionModel::with('role')->find($id);
1076
1077            $breadcrumb = (object) [
1078                'title' => 'Detail Permission',
1079                'list' => ['Home', 'Permission', 'Detail']
1080            ];
1081
1082            $page = (object) [
1083                'title' => 'Detail Permission'
1084            ];
1085
1086            $activeMenu = 'permission'; // set menu yang sedang aktif
1087
1088            return view('permission.detail', ['breadcrumb' => $breadcrumb, 'page' => $page, 'permission' => $permission, 'activeMenu' => $activeMenu]);
1089        }
1090
1091        // Menampilkan halaman edit permission
1092
1093        public function editPermission()
1094        {
1095            $permissions = PermissionModel::all();
1096
1097            $breadcrumb = (object) [
1098                'title' => 'Edit Permission',
1099                'list' => ['Home', 'Permission', 'Edit']
1100            ];
1101
1102            $page = (object) [
1103                'title' => 'Edit Permission'
1104            ];
1105
1106            $activeMenu = 'permission'; // set menu yang sedang aktif
1107
1108            return view('permission.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
1109        }
1110
1111        // Menampilkan halaman tambah permission
1112
1113        public function createPermission()
1114        {
1115            $role = RoleModel::all();
1116
1117            $breadcrumb = (object) [
1118                'title' => 'Tambah Permission',
1119                'list' => ['Home', 'Permission', 'Create']
1120            ];
1121
1122            $page = (object) [
1123                'title' => 'Tambah Permission'
1124            ];
1125
1126            $activeMenu = 'permission'; // set menu yang sedang aktif
1127
1128            return view('permission.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
1129        }
1130
1131        // Menampilkan halaman list menu
1132
1133        public function indexMenu()
1134        {
1135            $menus = MenuModel::all();
1136
1137            $breadcrumb = (object) [
1138                'title' => 'List Menu',
1139                'list' => ['Home', 'Menu', 'Index']
1140            ];
1141
1142            $page = (object) [
1143                'title' => 'List Menu'
1144            ];
1145
1146            $activeMenu = 'menu'; // set menu yang sedang aktif
1147
1148            return view('menu.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'menus' => $menus, 'activeMenu' => $activeMenu]);
1149        }
1150
1151        // Menampilkan halaman detail menu
1152
1153        public function detailMenu(string $id)
1154        {
1155            $menu = MenuModel::with('parent')->find($id);
1156
1157            $breadcrumb = (object) [
1158                'title' => 'Detail Menu',
1159                'list' => ['Home', 'Menu', 'Detail']
1160            ];
1161
1162            $page = (object) [
1163
```

```
@extends('layouts.template')

@section('content')


### <{{ $page->title }}>



@empty($user)


<h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
Data yang Anda cari tidak ditemukan.


@else


| ID       | <{{ \$user->user_id }}>           |
|----------|-----------------------------------|
| Level    | <{{ \$user->level->level_nama }}> |
| Username | <{{ \$user->username }}>          |
| Nama     | <{{ \$user->nama }}>              |
| Password | *****                             |


@endempty
Kembali</a>


```



```
// Menampilkan halaman form edit user
public function edit(string $id)
{
    $user = UserModel::find($id);
    $level = LevelModel::all();

    $breadcrumb = (object) [
        'title' => 'Edit User',
        'list'  => ['Home', 'User', 'Edit']
    ];

    $page = (object) [
        'title' => 'Edit user'
    ];

    $activeMenu = 'user'; // set menu yang sedang aktif

    return view('user.edit', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'level' => $level, 'activeMenu' => $activeMenu]);
}

// Menyimpan perubahan data user
public function update(Request $request, string $id)
{
    $request->validate([
        // username harus diisi, berupa string, minimal 3 karakter,
        // dan bernilai unik di tabel m_user kolom username kecuali untuk user dengan id yang sedang diedit
        'username' => 'required|string|min:3|unique:m_user,username,'.$id.',user_id',
        'nama'      => 'required|string|max:100', // nama harus diisi, berupa string, dan maksimal 100 karakter
        'password'  => 'nullable|min:5',           // password bisa diisi (minimal 5 karakter) dan bisa tidak diisi
        'level_id'   => 'required|integer'         // level_id harus diisi dan berupa angka
    ]);

    UserModel::find($id)->update([
        'username' => $request->username,
        'nama'      => $request->nama,
        'password'  => $request->password ? bcrypt($request->password) : UserModel::find($id)->password,
        'level_id'   => $request->level_id
    ]);

    return redirect('/user')->with('success', 'Data user berhasil diubah');
}
```

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

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

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



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>

19. Selanjutnya, kita buat *view* untuk melakukan proses edit data user di [PWL/resources/views/user/edit.blade.php](#)

```
@extends('layouts.template')  
@section('content')
```



```
<div class="card card-outline card-primary">
    <div class="card-header">
        <h3 class="card-title">{{ $page->title }}</h3>
        <div class="card-tools"></div>
    </div>
    <div class="card-body">
        @empty($user)
            <div class="alert alert-danger alert-dismissible">
                <i class="icon fas fa-ban"></i> Kesalahan!
                Data yang Anda cari tidak ditemukan.
            </div>
            <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
        @else
            <form method="POST" action="{{ url('/user/'.$user->user_id) }}" class="form-horizontal">
                @csrf
                {!! method_field('PUT') !!}<!-- tambahkan baris ini untuk proses edit yang butuh
method PUT -->
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label">Level</label>
                    <div class="col-11">
                        <select class="form-control" id="level_id" name="level_id" required>
                            <option value="">- Pilih Level -</option>
                            @foreach($level as $item)
                                <option value="{{ $item->level_id }}" @if($item->level_id == $user-
level_id) selected @endif>{{ $item->level_nama }}</option>
                            @endforeach
                        </select>
                        @error('level_id')
                            <small class="form-text text-danger">{{ $message }}</small>
                        @enderror
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label">Username</label>
                    <div class="col-11">
                        <input type="text" class="form-control" id="username" name="username"
value="{{ old('username', $user->username) }}" required>
                        @error('username')
                            <small class="form-text text-danger">{{ $message }}</small>
                        @enderror
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label">Nama</label>
                    <div class="col-11">
                        <input type="text" class="form-control" id="nama" name="nama" value="{{
old('nama', $user->nama) }}" required>
                        @error('nama')
                            <small class="form-text text-danger">{{ $message }}</small>
                        @enderror
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label">Password</label>
                    <div class="col-11">
                        <input type="password" class="form-control" id="password" name="password">
                        @error('password')
                            <small class="form-text text-danger">{{ $message }}</small>
                        @else
                            <small class="form-text text-muted">Abaikan (jangan diisi) jika tidak ingin
mengganti password user.</small>
                        @enderror
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label"></label>
```



```
<div class="col-11">
    <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
    <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}>Kembali</a>
</div>
</div>
</form>
@endempty
</div>
</div>
@endsection

@push('css')
@endpush
@push('js')
@endpush
```

```
Minggu,52 > Jobsheet5.2 > PWL.POS > resources > views > user > edit.blade.php -.
@extends('layouts.template')

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
            <div class="card-tools"></div>
        </div>
        <div class="card-body">
            @empty($user)
                <div class="alert alert-danger alert-dismissible">
                    <span class="icon fa fa-ban"></span> Kesalahan!
                    Data yang Anda cari tidak ditemukan.
                </div>
            @else
                <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
                <form method="POST" action="{{ url('/user/'.$user->user_id) }}" class="form-horizontal">
                    @csrf
                    (!!! method_field("PUT") !!) <!-- Tambahkan baris ini untuk proses edit yang butuh method PUT -->
                    <div class="form-group row">
                        <label class="col-1 control-label col-form-label">Level</label>
                        <div class="col-1">
                            <select class="form-control" id="level_id" name="level_id" required>
                                <option value=""> Pilih level ...</option>
                                @foreach($level as $item)
                                    <option value="{{ $item->level_id }}> @if($item->level_id == $user->level_id) selected @endif {{ $item->level_name }}</option>
                                @endforeach
                            </select>
                            @error('level_id')
                                <small class="form-text text-danger">{{ $message }}</small>
                            @enderror
                        </div>
                    </div>
                    <div class="form-group row">
                        <label class="col-1 control-label col-form-label">Username</label>
                        <div class="col-1">
                            <input type="text" class="form-control" id="username" name="username" value="{{ old('username', $user->username) }}" required>
                            @error('username')
                                <small class="form-text text-danger">{{ $message }}</small>
                            @enderror
                        </div>
                    </div>
                </div>
            @endempty
        </div>
    </div>
```

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

PWL - Starter Code

Home Contact

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Edit User

Edit user

Level

Username

Nama

Ahmad Santos

Password

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

Simpan Kembali

Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0



Screenshot of a web application showing a list of users. The sidebar on the left includes links for Dashboard, Data Pengguna, Level User, Data User (selected), Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area shows a table titled "Data user berhasil diubah" with 10 entries. The table columns are ID, Username, Nama, Level Pengguna, and Aksi (Detail, Edit, Hapus). The data includes users like admin, manager, staff, customer-1, manager_dua, manager22, manager33, manager56, manager12, and Ahmad.

```
11 <div class="card-body">
12   @if (session('success'))
13     <div class="alert alert-success">{{ session('success') }}</div>
14   @endif
15   <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
16     <thead>
17       <tr><th>ID</th><th>Username</th><th>Nama</th><th>Level Pengguna</th><th>Aksi</th></tr>
18     </thead>
19     </table>
20   </div>
```

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']);
```

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

```
// Menghapus data user
public function destroy(string $id)
{
    $check = UserModel::find($id);
    if (!$check) { // untuk mengecek apakah data user dengan id yang dimaksud ada atau tidak
        return redirect('/user')->with('error', 'Data user tidak ditemukan');
    }

    try{
        UserModel::destroy($id); // Hapus data level

        return redirect('/user')->with('success', 'Data user berhasil dihapus');
    }catch (\Illuminate\Database\QueryException $e){
        // Jika terjadi error ketika menghapus data, redirect kembali ke halaman dengan membawa pesan error
        return redirect('/user')->with('error', 'Data user gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini');
    }
}
```



```
162 // Menghapus data user
163 public function destroy(string $id)
164 {
165     $check = UserModel::find($id);
166     if ($check) { // untuk memeriksa apakah data user dengan id yang diminta ada atau tidak
167         return redirect('/user')->with('error', 'Data user tidak ditemukan');
168     }
169
170     try {
171         UserModel::destroy($id); // Hapus data Level
172         return redirect('/user')->with('success', 'Data user berhasil dihapus');
173     } catch (\Illuminate\Database\QueryException $e) {
174
175         // Jika terjadi error ketika menghapus data, redirect kembali ke halaman dengan membawa pesan error
176         return redirect('/user')->with('error', 'Data user gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini');
177     }
178 }
```

23. Selanjutnya kita modifikasi file [PWL/resources/views/user/index.blade.php](#) untuk menambahkan tampilan jika ada pesan error

Menjadi



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

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

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	manager	Manager	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	staff	Staff/Kasir	Staff	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	customer-1	Pelanggan Pertama	Pelanggan	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	manager_dua	Manager 2	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
6	manager22	Manager Dua Dua	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
7	manager33	Manager Tiga Tiga	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
8	manager56	Manager55	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
9	manager12	Manager11	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

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 and 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.

The screenshot shows a Laravel application interface. On the left is a sidebar with navigation links: PWL - Starter Code, Search, Dashboard, Data Pengguna, Level User, Data User (which is highlighted in blue), Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area has a header with Home, Contact, and a search icon. Below it is a sub-header Daftar User. A search bar with the placeholder "Search:" is highlighted with a red box. The main table displays user data with columns: ID, Username, Nama, Level Pengguna, and Aksi (with buttons for Detail, Edit, and Hapus). The table shows 6 entries. At the bottom, there are navigation buttons for Previous, Next, and a page number indicator (1).

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	supervisor	Supervisor	Supervisor	[Detail] [Edit] [Hapus]
4	kasir1	Kasir 1	Kasir/Staff	[Detail] [Edit] [Hapus]
5	kasir2	Kasir 2	Kasir/Staff	[Detail] [Edit] [Hapus]
6	administrator	Administrator	Administrator	[Detail] [Edit] [Hapus]

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**



```
// Menampilkan halaman awal user
public function index()
{
    $breadcrumb = (object) [
        'title' => 'Daftar User',
        'list'  => ['Home', 'User']
    ];

    $page = (object) [
        'title' => 'Daftar user yang terdaftar dalam sistem'
    ];

    $activeMenu = 'user'; // set menu yang sedang aktif

    $level = LevelModel::all(); // ambil data level untuk filter level

    return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
}
```

```
@if (session('error'))
| <div class="alert alert-danger">{{ session('error') }}</div>
@endif

<div class="row">
<div class="col-md-12">
<div class="form-group row">
    <label class="col-1 control-label col-form-label">Filter:</label>
    <div class="col-3">
        <select class="form-control" id="level_id" name="level_id" required>
            <option value="">- Semua -</option>
            @foreach($level as $item)
                <option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
            @endforeach
        </select>
        <small class="form-text text-muted">Level Pengguna</small>
    </div>
    </div>
</div>
</div>

<table class="table table-bordered table-striped table-hover table-sm" id="table_user">
    <thead>
        <tr><th>ID</th><th>Username</th><th>>Nama</th><th>Level Pengguna</th><th>Aksi</th></tr>
    </thead>
</table>
</div>
<iv>
<ection
```



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>

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



```
46  @push('js')
47    <script>
48      $(document).ready(function() {
49        var dataUser = $('#table_user').DataTable({
50          serverSide: true, // serverSide: true, jika ingin menggunakan server side processing
51          ajax: {
52            "url": "{{ url('user/list') }}",
53            "dataType": "json",
54            "type": "POST",
55            "data": function (d) {
56              d.level_id = $('#level_id').val();
57            }
58          },
59          columns: [
60            {
61              data: "DT_RowIndex", // nomor urut dari laravel datatable addIndexColumn()
62            }
63          ]
64        });
65        $('#table_user').DataTable({
66          serverSide: true, // jika ingin menggunakan server side processing serverSide: true,
67          ajax: {
68            "url": "{{ url('user/list') }}",
69            "datatype": "json",
70            "type": "POST",
71            "data": function (d) {
72              d.level_id = $('#level_id').val();
73            }
74          }
75        });
76      });
77    </script>
78  @endpush
```

4. Kemudian kita edit pada bagian akhir script @push('js') untuk menambahkan listener

```
81      data: "aksi",
82      className: "",
83      orderable: false, // orderable: true, jika ingin kolom ini bisa diurutkan
84      searchable: false // searchable: true, jika ingin kolom ini bisa dicari
85    }
86
87  ]);
88 });
89
90 $('#level_id').on('change', function() {
91   dataUser.ajax.reload();
92 });
93
94 });
95 </script>
96 @endpush
```

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

```
// Ambil data user dalam bentuk json untuk datatables
public function list(Request $request)
{
  $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
    ->with('level');

  // Filter data user berdasarkan level_id
  if ($request->level_id) {
    $users->where('level_id', $request->level_id);
  }

  return DataTables::of($users)
    ->addIndexColumn() // menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
    ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
      $btn = '<a href="'.url('/user/' . $user->user_id).'>Detail</a> ';
      $btn .= '<a href="'.url('/user/' . $user->user_id . '/edit').'>Edit</a> ';
      $btn .= '<form class="d-inline-block" method="POST" action="'.url('/user/' . $user->user_id).'">';
      $btn .= csrf_field() . method_field('DELETE') .
      '<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\'Apakah Anda yakin menghapus data ini?\')>Hapus</button>';
      return $btn;
    })
    ->rawColumns(['aksi']) // memberitahu bahwa kolom aksi adalah html
    ->make(true);
}
```



```
31
32     public function list(Request $request)
33     {
34         $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
35             ->with('level');
36
37         // Filter data user berdasarkan Level_id
38         if ($request->level_id) {
39             $users->where('level_id', $request->level_id);
40         }
41     }
```

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

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	supervisor	Supervisor	Supervisor	[Detail] [Edit] [Hapus]
4	kasir1	Kasir 1	Kasir/Staff	[Detail] [Edit] [Hapus]
5	kasir2	Kasir 2	Kasir/Staff	[Detail] [Edit] [Hapus]
6	administrator	Administrator	Administrator	[Detail] [Edit] [Hapus]

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

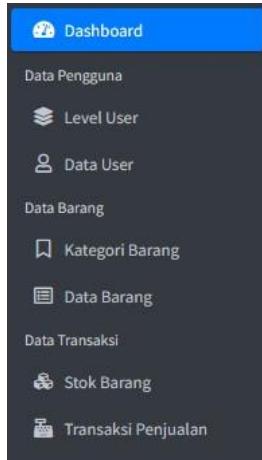
ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staff	Staff/Kasir	Staff	[Detail] [Edit] [Hapus]
4	customer-1	Pelanggan Pertama	Pelanggan	[Detail] [Edit] [Hapus]
5	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
6	manager22	Manager Dua Dua	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]

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

No	Kode Level	Nama Level	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff	[Detail] [Edit] [Hapus]
4	CUS	Pelanggan	[Detail] [Edit] [Hapus]

✓ Tabel m_kategori



PWL - Starter Code

Home Contact

Kategori Barang

Daftar kategori barang yang terdaftar dalam sistem

Tambah

Show 10 entries

No	Kode Kategori	Nama Kategori	Aksi
1	KTN001	Alat Tulis	Detail Edit Hapus
2	KTN002	Pakaian	Detail Edit Hapus
3	KTN003	Peralatan Rumah	Detail Edit Hapus
4	KTN004	Makanan	Detail Edit Hapus
5	KTN005	Minuman	Detail Edit Hapus

Showing 1 to 5 of 5 entries

Previous [1](#) Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0

✓ Tabel m_supplier

PWL - Starter Code

Home Contact

Daftar Stok

Daftar stok yang terdaftar dalam sistem

Tambah

Filter: - Semua

Search:

Show 10 entries

ID	Nama Barang	Nama User	Tanggal Stok	Jumlah Stok	Aksi
1	Keripik Kentang	Staff/Kasir	2025-03-05 01:23:22	50	Detail Edit Hapus
2	Donat Coklat	Staff/Kasir	2025-03-05 01:23:22	40	Detail Edit Hapus
3	Susu Kotak	Staff/Kasir	2025-03-05 01:23:22	60	Detail Edit Hapus
4	Mie Instan	Staff/Kasir	2025-03-05 01:23:22	30	Detail Edit Hapus
5	Headset Bluetooth	Staff/Kasir	2025-03-05 01:23:22	70	Detail Edit Hapus
6	Monitor LED	Staff/Kasir	2025-03-05 01:23:22	45	Detail Edit Hapus
7	Keset Karet	Staff/Kasir	2025-03-05 01:23:22	35	Detail Edit Hapus
8	Ember Plastik	Staff/Kasir	2025-03-05 01:23:22	55	Detail Edit Hapus
9	Celana Jeans	Staff/Kasir	2025-03-05 01:23:22	25	Detail Edit Hapus
10	Sweater Rajut	Staff/Kasir	2025-03-05 01:23:22	65	Detail Edit Hapus



✓ Tabel m_barang

The screenshot shows a web application interface titled "PWL - Starter Code". The left sidebar contains a navigation menu with options like Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Data Barang (which is selected and highlighted in blue), Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area is titled "Data Barang" and displays a table of items. The table has columns for ID, Kode Barang, Nama Barang, Kategori, Harga Beli, Harga Jual, and Aksi (with buttons for Detail, Edit, and Hapus). The table data is as follows:

ID	Kode Barang	Nama Barang	Kategori	Harga Beli	Harga Jual	Aksi
1	BRG001	Keripik Kentang	Alat Tulis	4500	7000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	BRG002	Donat Coklat	Alat Tulis	6000	8500	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	BRG003	Susu Kotak	Pakaian	3500	5000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	BRG004	Mie Instan	Pakaian	4000	5500	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	BRG005	Headset Bluetooth	Peralatan Rumah	120000	180000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
6	BRG006	Monitor LED	Peralatan Rumah	300000	450000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
7	BRG007	Keset Karet	Makanan	18000	25000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
8	BRG008	Ember Plastik	Makanan	25000	35000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
9	BRG009	Celana Jeans	Minuman	60000	75000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

*** Sekian, dan selamat belajar ***