

## **JOBSHEET 7**

### **LAPORAN TUGAS**

**Diajukan untuk Memenuhi Tugas  
Pemograman Web Lanjut**

**Jurusan:**

**TEKNOLOGI INFORMASI**

**Program Studi:**

**TEKNIK INFORMATIKA**

**Disusun oleh**

**1. Afrizal Dwi Septian (2241720122) / 01 / 2H**



**POLITEKNIK NEGERI MALANG**

**TEKNIK INFORMATIKA**

**2023**



Mata Kuliah : Pemrograman Web Lanjut (PWL)  
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis  
Semester : 4 (empat) / 6 (enam)  
Pertemuan ke- : 7 (tujuh)

## **JOBSHEET 07**

### **PWL – LARAVEL STARTER CODE**

Sebelumnya kita sudah membahas mengenai *templating*, penerapan *composer library laravel-adminLTE*, dan *composer library laravel-yajra-datable*. Tentu saja library-library tersebut sangat canggih dan bisa mempermudah serta mempercepat pengembangan website yang kita kerjakan.

Akan tetapi pada pertemuan kali ini, kita akan mempelajari basic penerapan adminLTE (tanpa library) dan penerapan yajra-datable yang simpel. Hal ini diperuntukkan agar kalian lebih mendalami dasar penerapan laravel dengan menggunakan library.

Sebelum kita masuk materi, kita buat dulu project baru yang akan kita gunakan untuk membangun aplikasi sederhana dengan topik *Point of Sales (PoS)*, sesuai dengan **Studi Kasus PWL.pdf**.  
Jadi kita bikin project Laravel 10 dengan nama **PWL\_POS**.

*Project PWL\_POS* akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

### **A. 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.

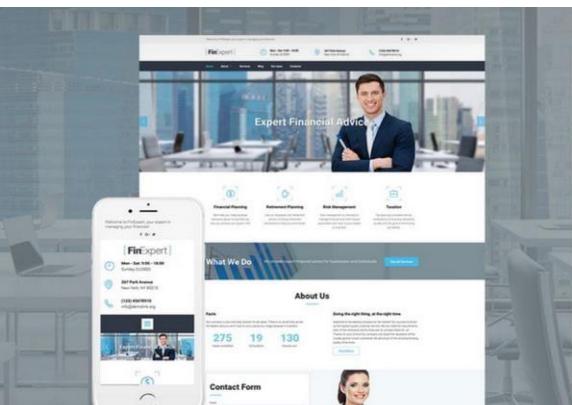
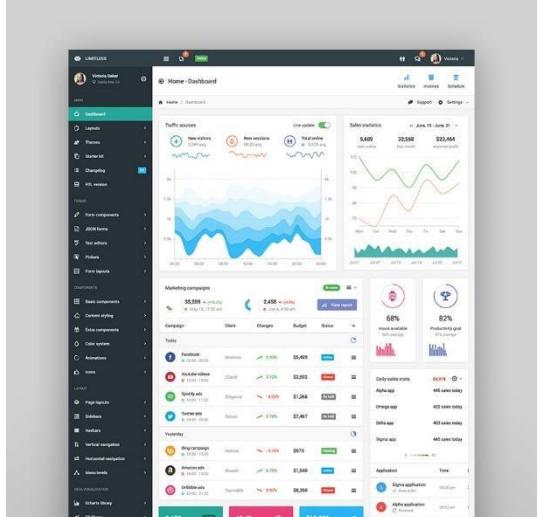


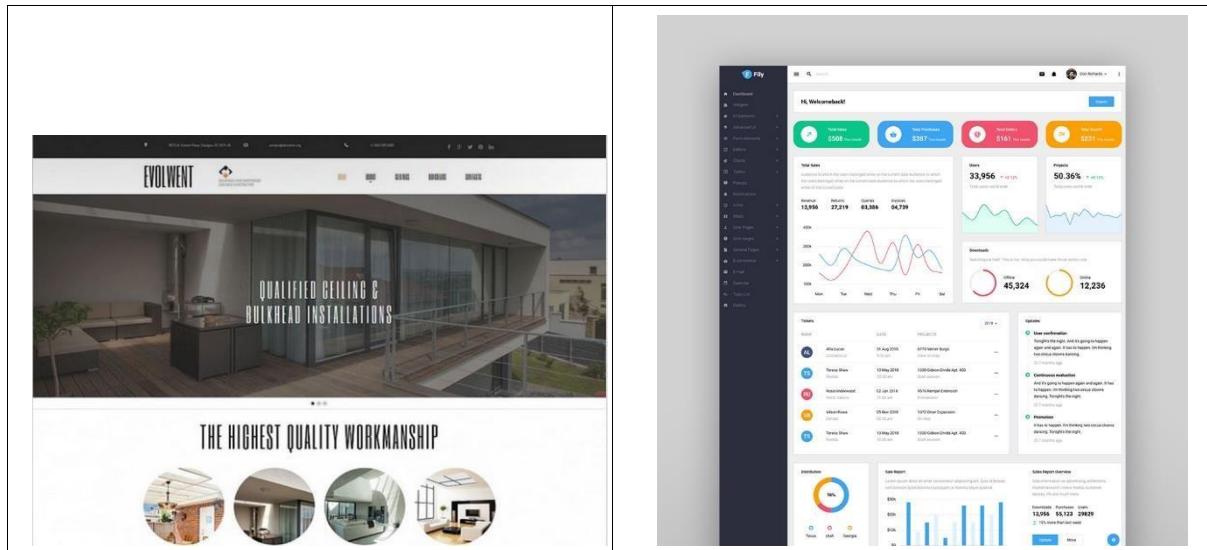
Secara umum kita bisa menggunakan template website yang sudah ada di internet untuk mempermudah kita dalam membuat website. Ada banyak sekali template yang ada di internet dan kita ambil satu contoh template untuk halaman administrator, seperti **AdminLTE** template.

**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
	



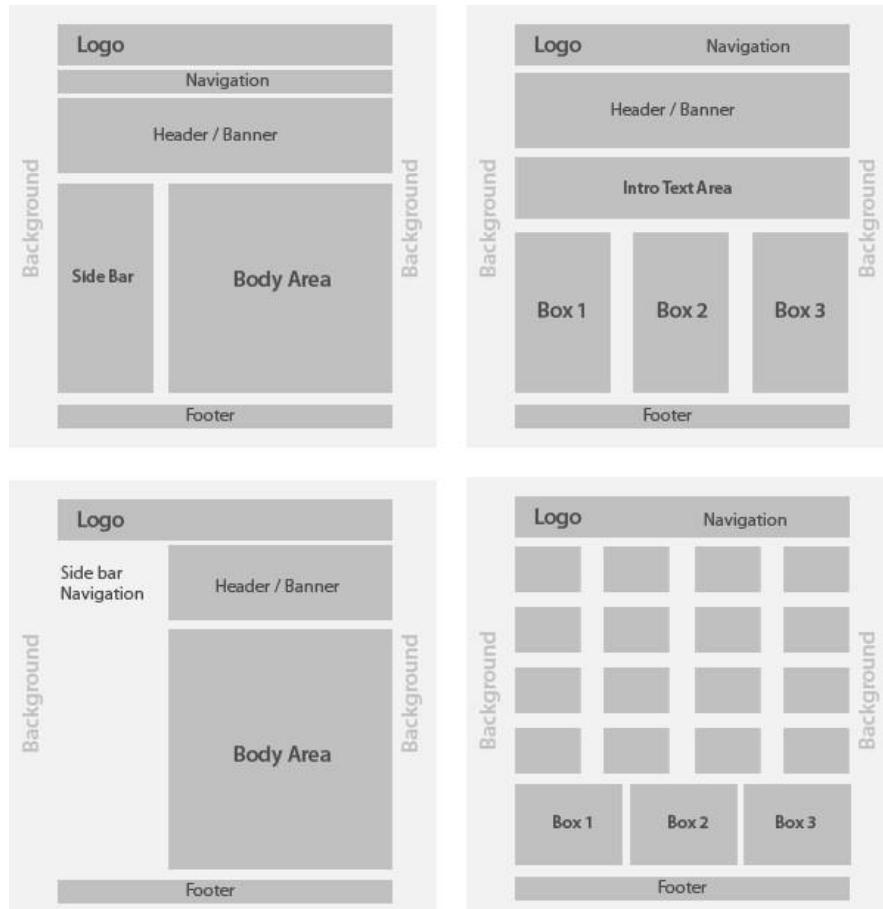
## B. WEB LAYOUT

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** ⑦ Pada bagian header, layout designer dapat mengisi bagian ini dengan logo website, navigasi situs, ikon media sosial, dan menu pencarian.
- **Navigasi** ⑦ 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** ⑦ Selain itu, terdapat elemen body/konten yang biasanya diisi dengan informasi produk, fitur produk, dan deskripsi produk yang dijual.
- **Sidebar** ⑦ 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** ⑦ 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:

Feb 8, 2022

REJack

v3.2.0

bd4d9c7

Compare ▾

### AdminLTE v3.2.0 Latest

**Breaking Change!**  
We dropped official node v12, it might be still work but we won't give support for any node 12 bugs/errors.

**Top-Features:**

- Enhanced Dark Mode
- SVG Font Awesome Icon support
- Docker dev-preset
- Reworked Dark/Light colors
- Custom Scrollbar style (Non-Plugin-Scrollbar)

### Contributors

forxer, liljack, and 16 other contributors

### Assets 2

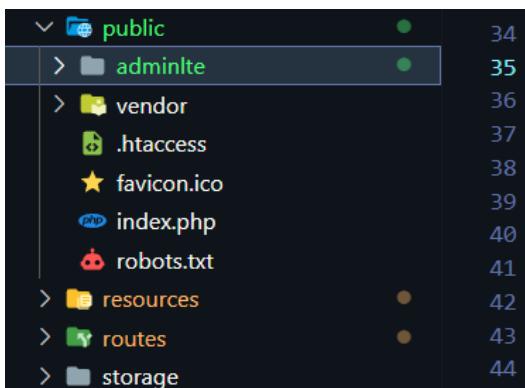
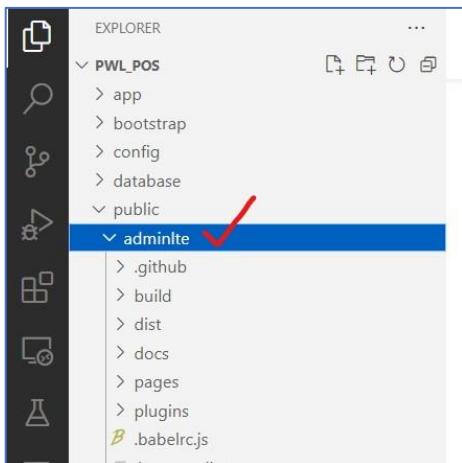
[Source code \(zip\)](#) ✓

[Source code \(tar.gz\)](#)

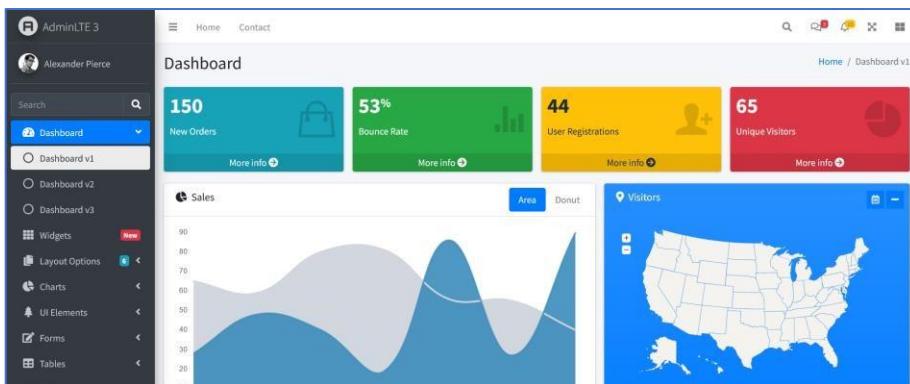
493 56 79 166 51 23 643 people reacted

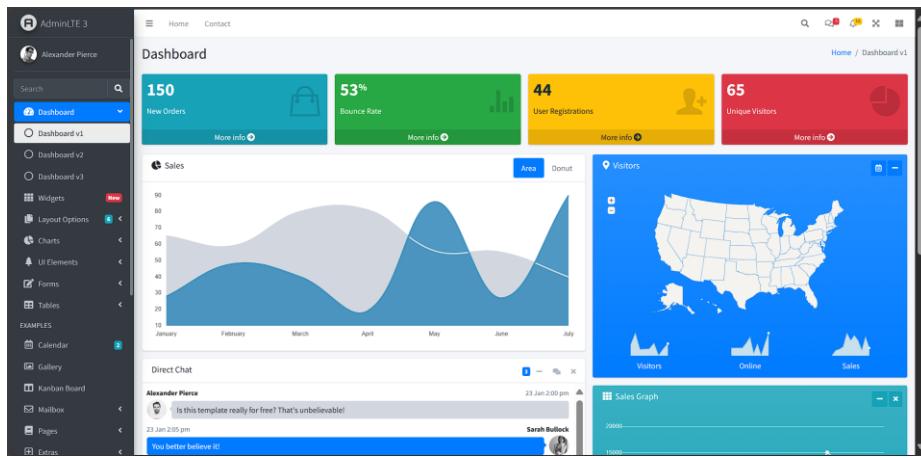
2. Setelah kita berhasil download, kita ekstrak file yang sudah di download ke folder project [PWL\\_POS/public](#), kemudian kita **rename** folder cukup menjadi [adminlte](#)

<sup>1</sup> . Kita download **AdminLTE v3.2.0** yang rilis pada 8 Feb 2022



3. Selanjutnya kita buka di browser dengan alamat [http://localhost/PWL\\_POS/public/adminlte](http://localhost/PWL_POS/public/adminlte) maka akan muncul tampilan seperti berikut





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 open. The 'Blank Page' option is highlighted with a red box and the number '2'. A red arrow points to the 'Blank Page' option. The sidebar also lists other templates like Login & Register v1, v2, Lockscreen, Legacy User Menu, Language Menu, Error 404, 500, Pace, and Starter Page.



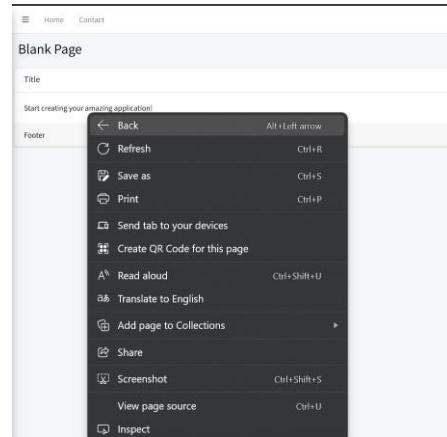
5. Dari sini kita bisa melakukan layouting halaman Blank Page ini menjadi 4 element seperti pada gambar berikut

The screenshot shows the AdminLTE 3 interface with a sidebar on the left and a main content area on the right. The sidebar is labeled '2 - Sidebar / Navigasi' and contains a list of navigation items under 'Extras'. The main content area is divided into five sections: '1 - Header / Navbar' at the top, followed by '3 - Breadcrumb', 'Title', 'Start creating your amazing application!', 'Footer', '4 - Content' (which is currently empty), and '5 - Footer' at the bottom. The footer includes copyright information and a version number.

This screenshot shows the AdminLTE 3 interface with a different sidebar configuration. The sidebar is labeled 'AdminLTE 3' and 'Alexander Pierce' at the top. It contains a search bar and a list of navigation items under 'Dashboard', 'Widgets', 'Layout Options', 'Charts', 'UI Elements', 'Forms', 'Tables', 'EXAMPLES', 'Calendar', 'Gallery', 'Kanban Board', 'Mailbox', 'Pages', 'Extras', 'Login & Register v1', and 'Login & Register v2'. The main content area is identical to the one in the previous screenshot, showing the 1-5 layout structure.

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

The screenshot shows a browser window with a context menu open over a 'Blank Page'. The menu options include 'Save Page As...', 'Save Page to Pocket', 'Select All', 'Take Screenshot', 'View Page Source' (which is circled in red), 'Inspect Accessibility Properties', 'Inspect (Q)', 'Block element...', 'LastPass', and 'Download with IDM'. The main content area of the browser shows a simple 'Blank Page' template with sections for 'Title', 'Start creating your amazing ...', 'Footer', and 'Content'.





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

```
template.blade.php X
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="../../plugins/fontawesome-free/css/all.min.css">
12    <!-- Theme style -->
13    <link rel="stylesheet" href="../../dist/css/adminlte.min.css">
14  </head>
```

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.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>{{ 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>
```

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title>

  <!-- Google Font: Source Sans Pro -->
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
  <!-- Font Awesome -->
  <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}>
  <!-- Theme style -->
  <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}>
</head>
```

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



```
template.blade.php X header.blade.php

template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > aside.main
14   </head>
15   <body class="hold-transition sidebar-mini">
16     <!-- Site wrapper -->
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           AdminLTE 3</span>
28         </a>

resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper
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>
15   <body class="hold-transition sidebar-mini">
16     <!-- Site wrapper -->
17     <div class="wrapper">
18       <!-- Navbar -->
19       @include('layouts.header')
20       <!-- ./navbar -->
```

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

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

```
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">
```



## 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">
32
33       <!-- Brand Logo -->
34       <a href="{{ url('/') }}" class="brand-link">
35         
36         <span class="brand-text font-weight-light">PWL - Starter Code</span>
37       </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       
27       <span class="brand-text font-weight-light">PWL - Starter Code</span>
28     </a>
29
30     <!-- Sidebar -->
31     @include('layouts.sidebar')
32     <!-- /.sidebar -->
33   </aside>
```

```
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper
2   <html lang="en">
3     </head>
4     <body class="hold-transition sidebar-mini">
5       <!-- Site wrapper -->
6       <div class="wrapper">
7         <!-- Navbar -->
8         @include('layouts.header')
9         <!-- /.navbar -->
10
11         <!-- Main Sidebar Container -->
12         <aside class="main-sidebar sidebar-dark-primary elevation-4">
13           <!-- Brand Logo -->
14           <a href="{{ url('/') }}" class="brand-link">
15             
16             <span class="brand-text font-weight-light">PWL - Starter Code</span>
17           </a>
18
19           <!-- Sidebar -->
20           @include('layouts.sidebar')
21           <!-- /.sidebar -->
22         </aside>
```



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 **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>
```

```
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>
```

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>
```



```
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>
```

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 **4breadcrumb**. 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>
46
```

```
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
41  <!-- Main content -->
42  <section class="content">
43
44  <!-- Default box -->
45  <div class="card">
46  <div class="card-header">
```



17. Layout terakhir adalah pada bagian konten. Layout untuk konten bisa kita buat dinamis, sesuai dengan apa yang ingin kita sajikan pada web yang kita bangun.
18. Untuk **content**, kita akan menghapus baris **42-66** pada file **template.blade.php**. dan kita ganti dengan kode seperti ini **@yield('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 </div>
```

```
resources > views > layouts > template.blade.php > @body.hold-transition.sidebar-mini > @div.wrapper > @div.content-wrapper
2 <html lang="en">
15   <body class="hold-transition sidebar-mini">
16     <!-- Site wrapper -->
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         @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         @yield('content')
42         | <!-- /.card -->
43
44       </section>
45       <!-- /.content -->
46     </div>
47     <!-- /.content-wrapper -->
48
49     @include('layouts.footer')
50   </div>
51   <!-- /.wrapper -->
52
53   <!-- jQuery -->
54   <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"/></script>
55   <!-- Bootstrap A -->
56   <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"/></script>
57   <!-- AdminLTE App -->
58   <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"/></script>
59   </body>
60 </html>
```

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



## 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 }
```

```
app > Http > Controllers >  WelcomeController.php >  WelcomeController >  index
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class WelcomeController extends Controller
8 {
9
10     /**
11      * Display a listing of the resource.
12      */
13     public function index()
14     {
15         $breadcrumb = (object) [
16             'title' => 'Selamat Datang',
17             'list'  => ['Home', 'Welcome']
18         ];
19
20         $activeMenu = 'dashboard';
21
22         return view('welcome',  data: ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu]);
23     }
}
```



2. Kita buat file pada [PWL\\_POS/resources/views/welcome.blade.php](#)

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

```
resources > views > welcome.blade.php > ...<br>
1  @extends('layouts.template')<br>
2<br>
3  @section('content')<br>
4<br>
5  <div class="card"><br>
6      <div class="card-header"><br>
7          <h3 class="card-title">Halo Apa Kabar!</h3><br>
8          <div class="card-tools"></div><br>
9      </div><br>
10     <div class="card-body"><br>
11         | Selamat Datang Semua, Ini Adalah Halaman Utama dari Aplikasi Ini.<br>
12     </div><br>
13 </div><br>
14 @endsection|
```

3. Kita modifikasi file [PWL\\_POS/resources/views/layouts/breadcrumb.blade.php](#)

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



```
resources > views > layouts > breadcrumb.blade.php > ...
1  <section class="content-header">
2    <div class="container-fluid">
3      <div class="row mb-2">
4        <div class="col-sm-6">
5          <h1>{{ $breadcrumb->title }}</h1>
6        </div>
7        <div class="col-sm-6">
8          <ol class="breadcrumb float-sm-right">
9            @foreach ($breadcrumb->list as $key => $value)
10           @if ($key == count($breadcrumb->list) - 1)
11             <li class="breadcrumb-item active">{{ $value }}</li>
12           @else
13             <li class="breadcrumb-item">{{ $value }}</li>
14           @endif
15         @endforeach
16       </ol>
17     </div>
18   </div>
19 </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 {{ ($activeMenu == 'dashboard')? 'active' : '' }} ">
    <i class="nav-icon fas fa-tachometer-alt"></i>
    <p>Dashboard</p>
</a>
</li>
<li class="nav-header">Data Pengguna</li>
<li class="nav-item">
    <a href="{{ url('/level') }}" class="nav-link {{ ($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>
```

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

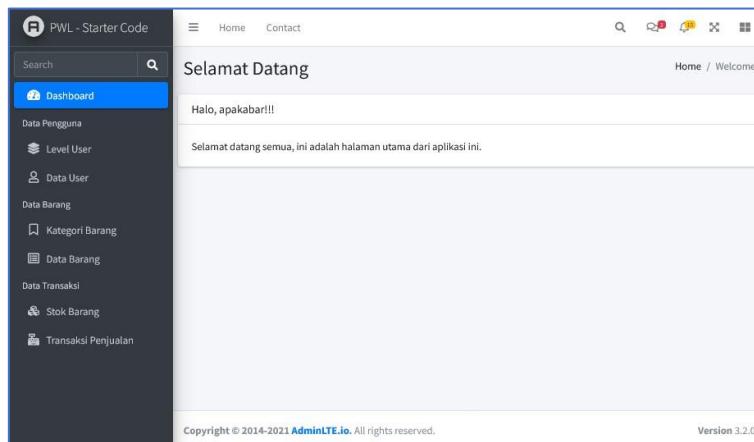


5. Kita tambahkan kode berikut router web.php

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

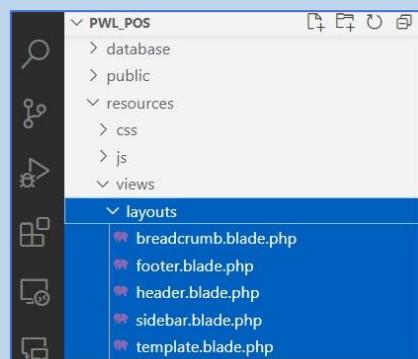
```
Route::get('uri: /', action: [WelcomeController::class, 'index']);
```

6. Sekarang kita coba jalankan di browser dengan mengetikkan url  
[http://localhost/PWL\\_POS/public](http://localhost/PWL_POS/public)



### 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.





Screenshot of a Laravel application dashboard titled "PWL - Starter Code". The dashboard features a sidebar with links like Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area displays a welcome message: "Selamat Datang" and "Halo Apa Kabar!", followed by a note: "Selamat Datang Semua, Ini Adalah Halaman Utama dari Aplikasi Ini.". The footer includes copyright information "Copyright © 2014-2021 AdminLTE.io. All rights reserved." and "Version 3.2.0".



## D. LARAVEL DATATABLE

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\\_POS/public/adminlte/pages/tables/data.html](http://localhost/PWL_POS/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*.

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 }
```



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

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     }
}
```

```
app > Http > Controllers > UserController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\UserModel;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Database\Eloquent\ModelNotFoundException;
9 use Illuminate\Http\RedirectResponse;
10 // use App\DataTables\UserDataTable;
11 use Yajra\DataTables\Facades\DataTables;
12
13 class UserController extends Controller
14 {
15     //Jobsheet 7
16
17     public function index()
18     {
19         $breadcrumb = (object) [
20             'title' => 'Daftar User',
21             'list' => ['Home', 'User']
22         ];
23
24         $page = (object) [
25             'title' => 'Daftar user yang terdaftar dalam sistem'
26         ];
27
28         $activeMenu = 'user'; // set menu yang sedang aktif
29
30         return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'activeMenu' => $activeMenu]);
31     }
32 }
```



5. Lalu kita buat view pada `PWL_POS/resources/views/user/index.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">
                <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}>Tambah</a>
            </div>
        </div>
        <div class="card-body">
            <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>
        </div>
    @endsection

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

```
resources > views > user > index.blade.php > div.card.card-outline.card-primary > div.card-body > table#table_user.table.table-bordered.table-striped
1   @extends('layouts.template')
2
3   @section('content')
4       <div class="card card-outline card-primary">
5           <div class="card-header">
6               <h3 class="card-title">{{ $page->title }}</h3>
7               <div class="card-tools">
8                   <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}>Tambah</a>
9               </div>
10          </div>
11          <div class="card-body">
12              <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
13                  <thead>
14                      <tr>
15                          <th>ID</th>
16                          <th>Username</th>
17                          <th>Nama</th>
18                          <th>Level Pengguna</th>
19                          <th>Aksi</th>
20                      </tr>
21                  </thead>
22                  </table>
23              </div>
24          </div>
25      @endsection
26
27      @push('css')
28      @endpush
29
30
```



```
@push('js')
<script>
$(document).ready(function() {
    var dataUser =
$('#table_user').DataTable({
        serverSide: true, // serverSide: true,
        jika ingin menggunakan server side processing         ajax: {
            "url": "{{ url('user/list') }}",
            "dataType": "json",
            "type": "POST"
        },
        columns: [
            {
                data: "DT_RowIndex", // nomor urut dari laravel datatable addIndexColumn()
                className: "text-center",
                orderable: false,
                searchable: false
            },
            {
                data: "username",
                className: "",
                orderable: true, // orderable: true, jika ingin kolom ini bisa diurutkan
                searchable: true // searchable: true, jika ingin kolom ini bisa dicari
            },
            {
                data: "nama",
                className: "",
                orderable: true, // orderable: true, jika ingin kolom ini bisa diurutkan
                searchable: true // searchable: true, jika ingin kolom ini bisa dicari
            },
            {
                data: "level.level_nama",
                className: "",
                orderable: false, // orderable: true, jika ingin kolom ini bisa diurutkan
                searchable: false // searchable: true, jika ingin kolom ini bisa dicari
            },
            {
                data: "aksi",
                className: "",
                orderable: false, // orderable: true, jika ingin kolom ini bisa diurutkan
            }
        ]
    });
});
```



```
        searchable: false      // searchable: true, jika ingin kolom ini bisa
dicari
    }
]
});
});
</script>
@endpush
```

```
resources > views > user > index2.blade.php > script > <function> > dataUser
31 @push('js')
32     <script>
33         $(document).ready(function() {
34             var dataUser = $('#table_user').DataTable({
35                 serverSide: true, // serverSide: true, jika ingin menggunakan server side processing
36                 ajax: [
37                     "url": "{{ url('user/list') }}",
38                     "dataType": "json",
39                     "type": "POST"
40                 ],
41                 columns: [
42                     {
43                         data: "DT_RowIndex", // nomor urut dari laravel datatable addIndexColumn()
44                         className: "text-center",
45                         orderable: false,
46                         searchable: false
47                     },
48                     {
49                         data: "username",
50                         className: "",
51                         orderable: true, // orderable: true, jika ingin kolom ini bisa diurutkan
52                         searchable: true // searchable: true, jika ingin kolom ini bisa dicari
53                     },
54                     {
55                         data: "nama",
56                         className: "",
57                         orderable: true, // orderable: true, jika ingin kolom ini bisa diurutkan
58                         searchable: true // searchable: true, jika ingin kolom ini bisa dicari
59                     },
60                     {
61                         data: "level.level_nama",
62                         className: "",
63                         orderable: false, // orderable: true, jika ingin kolom ini bisa diurutkan
64                         searchable: false // searchable: true, jika ingin kolom ini bisa dicari
65                     },
66                     {
67                         data: "aksi",
68                         className: "",
69                         orderable: false, // orderable: true, jika ingin kolom ini bisa diurutkan
70                         searchable: false // searchable: true, jika ingin kolom ini bisa dicari
71                     }
72                 ]
73             });
74         </script>
75     @endpush
```



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

`public`

```
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>

60    <!-- jQuery -->
61    <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}></script>
62    <!-- Bootstrap 4 -->
63    <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}></script>
64    <!-- DataTables & Plugins -->
65    <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}></script>
66    <script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}></script>
67    <script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}></script>
68    <script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}></script>
69    <script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}></script>
70    <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}></script>
71    <script src="{{ asset('adminlte/plugins/jspzip/jspzip.min.js') }}></script>
72    <script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}></script>
73    <script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}></script>
74    <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}></script>
75    <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}></script>
76    <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}></script>
77    <!-- AdminLTE App -->
78    <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}></script>
79
80    <script>
81        // Untuk mengirimkan token Laravel CSRF pada setiap request ajax
82        $.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
83    </script>
84    @stack('js') <!-- Digunakan untuk memanggil custom js dari perintah push('js') pada masing-masing view -->
85 </body>
86 </html>
```



```
resources > views > layouts > template.blade.php > head
1   <!DOCTYPE html>
2   <html lang="en">
3
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() }}>
10    <!-- Untuk mengirimkan token Laravel CSRF pada setiap request ajax -->
11
12    <!-- Google Font: Source Sans Pro -->
13    <link rel="stylesheet"
14      href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
15    <!-- Font Awesome -->
16    <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}>
17    <!-- DataTables -->
18    <link rel="stylesheet" href="{{ asset('Template/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}>
19    <link rel="stylesheet" href="{{ asset('Template/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}>
20    <link rel="stylesheet" href="{{ asset('Template/plugins/buttons/css/buttons.bootstrap4.min.css') }}>
21    <!-- Theme style -->
22    <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}>
23
24    @stack('css') <!-- Digunakan untuk memanggil custom css dari perintah push('css') pada masing-masing view -->
25
26  </head>
27
28  <!-- jQuery -->
29  <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}></script>
30  <!-- Bootstrap 4 -->
31  <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}></script>
32  <!-- Dataatables & Plugins -->
33  <script src="{{ asset('Template/plugins/datatables/jquery.dataTables.min.js') }}></script>
34  <script src="{{ asset('Template/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}></script>
35  <script src="{{ asset('Template/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}></script>
36  <script src="{{ asset('Template/plugins/datatables-responsive/js/dataTables.responsive.bootstrap4.min.js') }}></script>
37  <script src="{{ asset('Template/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}></script>
38  <script src="{{ asset('Template/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}></script>
39  <script src="{{ asset('Template/plugins/datatables-buttons/js/buttons.html5.min.js') }}></script>
40  <script src="{{ asset('Template/plugins/datatables-buttons/js/buttons.print.min.js') }}></script>
41  <script src="{{ asset('Template/plugins/datatables-buttons/js/buttons.colVis.min.js') }}></script>
42  <!-- AdminLTE App -->
43  <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}></script>
44  <script>
45    // Untuk mengirimkan token Laravel CSRF pada setiap request ajax
46    $.ajaxSetup({
47      headers: {
48        'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
49      }
50    });
51  </script>
52  @stack('js') <!-- Digunakan untuk memanggil custom js dari perintah push ('js') pada masing-masing view -->
53  <!-- AdminLTE for demo purposes -->
54  {{-- <script src="{{ asset('adminlte/dist/js/demo.js') }}></script> --}}
55
56  </body>
57
58  </html>
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```



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($users)
    ->addIndexColumn() // menambahkan kolom index / no urut (default nama
kolom: DT_RowIndex)
    ->addColumn('aksi', function ($user) { // menambahkan kolom aksi
$btn = '<a href="'.url('/user/' . $user->user_id).'" class="btn btninfo 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);
}
```

```
public function list(Request $request)
{
    $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
        ->with('level');

    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(path: '/user/' . $user->user_id).'" class="btn btn-info btn-sm">Detail</a> ';
            $btn .= '<a href="'. url(path: '/user/' . $user->user_id . '/edit').'" class="btn btn-warning btn-sm">Edit</a> ';
            $btn .= '<form class="d-inline-block" method="POST" action="'. url(path: '/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);
}
```



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

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail, Edit, Hapus]
2	manager	Manager	Manager	[Detail, Edit, Hapus]
3	staff	Staff	Staff/Kasir	[Detail, Edit, Hapus]
4	customer-1	Pelanggan Pertama	Pelanggan	[Detail, Edit, Hapus]
5	manager_dua	Manager 2	Manager	[Detail, Edit, Hapus]
6	manager_tiga	Manager 3	Manager	[Detail, Edit, Hapus]
7	manager22	Manager Dua Dua	Manager	[Detail, Edit, Hapus]
8	manager23	Manager Tiga Tiga	Manager	[Detail, Edit, Hapus]

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]);
}
```

```
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', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
}
```

10. Sekarang kita buat form untuk menambah data, kita buat file [PWL\\_POS/resources/views/user/create.blade.php](#)



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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
            <div class="card-tools"></div>
        </div>
        <div class="card-body">
            <form method="POST" action="{{ url('user') }}" class="form-horizontal">
                @csrf
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label">Level</label>
                    <div class="col-11">
                        <select class="form-control" id="level_id" name="level_id" required>
                            <option value="">- Pilih Level -</option>
                            @foreach($level as $item)
                                <option value="{{ $item->level_id }}>{{ $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') }}" 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') }}" required>
                        @error('nama')
                            <small class="form-text text-danger">{{ $message }}</small>
                        @enderror
                    </div>
                </div>
            </form>
        </div>
    </div>

```



```
</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" 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
```



```
resources > views > user > create2.blade.php > ...
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title">{{ $page->title }}</h3>
7          </div>
8      </div>
9      <div class="card-body">
10         <form method="POST" action="{{ url('user') }}" class="form-horizontal" @csrf
11             <div class="form-group row">
12                 <label class="col-1 control-label col-form-label">Level</label>
13                 <div class="col-11">
14                     <select class="form-control" id="level_id" name="level_id" required>
15                         <option value=""> Pilih Level </option>
16                         @foreach ($level as $item)
17                             <option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
18                         @endforeach
19                     </select> @error('level_id')
20                         <small class="form-text text-danger">{{ $message }}</small>
21                     @enderror
22                 </div>
23             </div>
24             <div class="form-group row">
25                 <label class="col-1 control-label col-form-label">Username</label>
26                 <div class="col-11">
27                     <input type="text" class="form-control" id="username" name="username"
28                         value="{{ old('username') }}" required>
29                     @error('username')
30                         <small class="form-text text-danger">{{ $message }}</small>
31                     @enderror
32                 </div>
33             </div>
34             <div class="form-group row">
35                 <label class="col-1 control-label col-form-label">Nama</label>
36                 <div class="col-11">
37                     <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama') }}"
38                         required>
39                     @error('nama')
40                         <small class="form-text text-danger">{{ $message }}</small>
41                     @enderror
42                 </div>
43             </div>
44             <div class="form-group row">
45                 <label class="col-1 control-label col-form-label">Password</label>
46                 <div class="col-11">
47                     <input type="password" class="form-control" id="password" name="password" required>
48                     @error('password')
49                         <small class="form-text text-danger">{{ $message }}</small>
50                     @enderror
51                 </div>
52             </div>
53             <div class="form-group row">
54                 <label class="col-1 control-label col-form-label"></label>
55                 <div class="col-11">
56                     <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
57                     <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}>Kembali</a>
58                 </div>
59             </div>
60         </form>
61     </div>
62     </div>
63     @endsection
64     @push('css')
65     @endpush
66     @push('js')
67     @endpush
68 |
```

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
    ]);

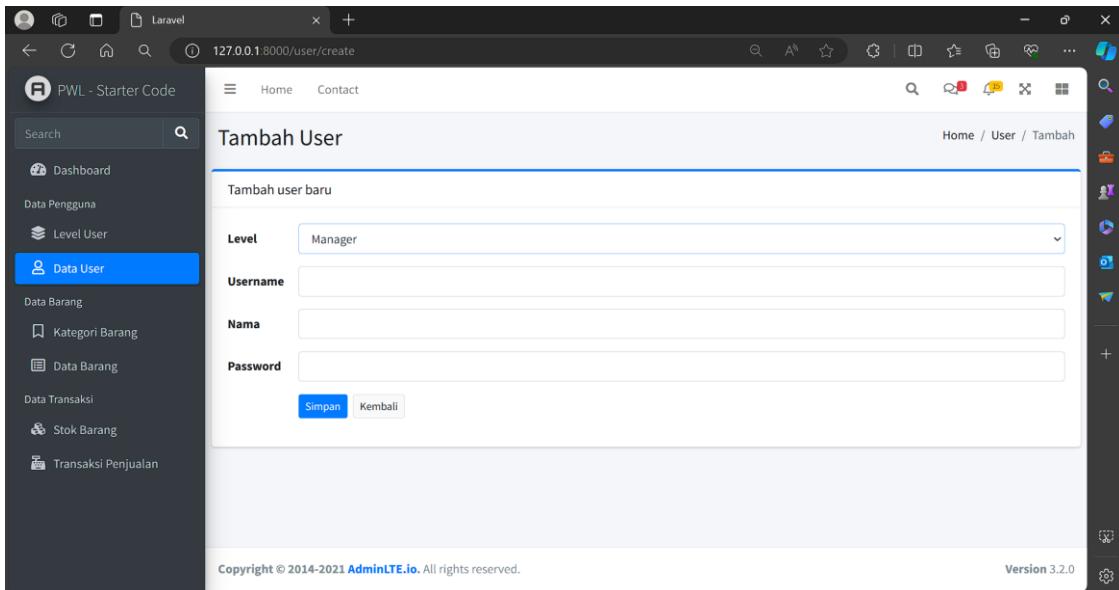
    return redirect('/user')->with('success', 'Data user berhasil disimpan');
}
```



```
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(['key': 'success', 'value': 'Data user berhasil disimpan']);
}
```

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



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
});
```



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]);
}
```

```
public function show(String $id)
{
    $user = UserModel::with( relations: 'level')->find( id: $id);

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

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

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

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

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

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

@section('content')


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



@if($user)
    

##### Kesalahan!



Data yang Anda cari tidak ditemukan.


@else
    <table class="table table-bordered table-striped table-hover tablesm">
        <tr>
            <th>ID</th>


```



```
<td>{{ $user->user_id }}</td>
</tr>
<tr>
    <th>Level</th>
    <td>{{ $user->level->level_nama }}</td>
</tr>
<tr>
    <th>Username</th>
    <td>{{ $user->username }}</td>
</tr>
<tr>
    <th>Nama</th>
    <td>{{ $user->nama }}</td>
</tr>
<tr>
    <th>Password</th>
    <td>*****</td>
</tr>
</table>
@endempty
<a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
</div>
</div>
@endsection

@push('css')
@endpush

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



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

16. Sekarang kalian coba untuk melihat detail data user di browser, dan coba untuk mengetikkan id yang salah contoh [http://localhost/PWL\\_POS/public/user/100](http://localhost/PWL_POS/public/user/100) amati apa yang terjadi, dan laporan!!!

Detail User	
ID	12
Level	Manager
Username	manager56
Nama	Manager55
Password	*****



17. Selanjutnya, kita masuk pada bagian untuk memodifikasi data user. Route yang bertugas untuk menangkap request edit 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
});
```

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

```
// 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');
}
```



```
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]);
}

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('key: 'success', 'value: 'Data user berhasil diubah');
}
```

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

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

@section('content')


<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">
                <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
                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 --&gt;
<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
```



```
resources > views > user > edit.blade.php > ...
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3>{{ $page->title }}</h3>
7              <div class="card-tools"></div>
8          </div>
9          <div class="card-body">
10             @empty($user)
11                 <div class="alert alert-danger alert-dismissible">
12                     <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
13                     Data yang Anda cari tidak ditemukan.
14                 </div>
15                 <a href="{{ url('user') }}" class="btn btn-sm btn-default mt- 2">Kembali</a>
16             @else
17                 <form method="POST" action="{{ url('/user/' . $user->user_id) }}" class="form-horizontal">
18                     @csrf
19                     (@ method_field('PUT')) <!-- tambahkan baris ini untuk proses edit yang butuh method PUT -->
20                     <div class="form-group row">
21                         <label class="col-1 control-label col-form-label">Level</label>
22                         <div class="col-11">
23                             <select class="form-control" id="level_id" name="level_id" required>
24                                 <option value="">- Pilih Level -</option>
25                                 @foreach ($level as $item)
26                                     <option value="{{ $item->level_id }}> {{ $item->level_nama }}</option>
27                                 @endforeach
28                             </select> @error('level_id')
29                                 <small class="form-text text-danger">{{ $message }}</small>
30                             @enderror
31                         </div>
32                     </div>
33                     <div class="form-group row">
34                         <label class="col-1 control-label col-form-label">Username</label>
35                         <div class="col-11">
36                             <input type="text" class="form-control" id="username" name="username"
37                                 value="{{ old('username', $user->username) }}" required>
38                             @error('username')
39                                 <small class="form-text text-danger">{{ $message }}</small>
40                             @enderror
41                         </div>
42                     </div>
43                     <div class="form-group row">
44                         <label class="col-1 control-label col-form-label">Nama</label>
45                         <div class="col-11">
46                             <input type="text" class="form-control" id="nama" name="nama"
47                                 value="{{ old('nama', $user->nama) }}" required>
48                             @error('nama')
49                                 <small class="form-text text-danger">{{ $message }}</small>
50                             @enderror
51                         </div>
52                     </div>
53                     <div class="form-group row">
54                         <label class="col-1 control-label col-form-label">Password</label>
55                         <div class="col-11">
56                             <input type="password" class="form-control" id="password" name="password">
57                             @error('password')
58                                 <small class="form-text text-danger">{{ $message }}</small>
59                             @else
60                                 <small class="form-text text-muted">
61                                     Abaikan (jangan diisi) jika tidak ingin mengganti password user.</small>
62                             @enderror
63                         </div>
64                     </div>
65                     <div class="form-group row">
66                         <label class="col-1 control-label col-form-label"></label>
67                         <div class="col-11">
68                             <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
69                             <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}>Kembali</a>
70                         </div>
71                     </div>
72                     </form>
73             @endempty
74         </div>
75     </div>
76     @endsection
77     @push('css')
78     @endpush
79 
```



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

The screenshot shows a web application interface for editing a user. On the left is a sidebar with various menu items: 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 'Edit user'. It contains a form with the following fields: 'Level' set to 'Administrator', 'Username' set to 'adminRizal', 'Nama' set to 'Administrator', and a 'Password' field which is empty. Below the form is a note: 'Abaikan (jangan diisi) jika tidak ingin mengganti password user.' At the bottom of the form are two buttons: 'Simpan' (Save) and 'Kembali' (Back). The footer of the page includes 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

The screenshot shows a list of users. The sidebar on the left is identical to the previous screenshot. The main content area has a header 'Daftar User'. Below it is a message: 'Daftar User yang terdaftar dalam sistem' and a 'Tambah' button. A green banner at the top of the table area says 'Data user berhasil diubah'. The table has columns: 'ID', 'Username', 'Nama', 'Level Pengguna', and 'Aksi'. The data in the table is as follows:

ID	Username	Nama	Level Pengguna	Aksi
1	adminRizal	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staff	Staff	Staff/Kasir	[Detail] [Edit] [Hapus]
4	customer-1	Pelanggan Pertama	Pelanggan	[Detail] [Edit] [Hapus]
5	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
6	manager_tiga	Manager 3	Manager	[Detail] [Edit] [Hapus]
7	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
8	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]

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');
    }
}
```

```
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');
    }
}
```

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

```
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>
```

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>
```



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

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

The screenshot shows a Laravel application interface. On the left is a sidebar with navigation links: Dashboard, Data Pengguna, Level User, Data User (which is currently selected), Data Barang, Kategori Barang, Data Barang, Transaksi, Stok Barang, and Transaksi Penjualan. The main content area displays a table titled "Daftar User yang terdaftar" with 10 entries. The columns are ID, Username, Nama, Level Pengguna, and Aksi (Action). The table includes buttons for Detail, Edit, and Hapus (Delete). A modal dialog box is centered over the table, asking "Apakah Anda yakin menghapus data ini?" with OK and Cancel buttons. At the bottom of the page, there is a footer with the text "Showing 1 to 10 of 10 entries" and navigation buttons for Previous, Next, and a page number indicator.

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



ID	Username	Nama	Level Pengguna	Aksi
1	adminRizal	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staff	Staff	Staff/Kasir	[Detail] [Edit] [Hapus]
4	customer-1	Pelanggan Pertama	Pelanggan	[Detail] [Edit] [Hapus]
5	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
6	manager_tiga	Manager 3	Manager	[Detail] [Edit] [Hapus]
7	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
8	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]

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

## E. DATA SEARCHING DAN FILTERING

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

### 1. *Searching* (Pencarian):

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

### 2. *Filtering* (Penyaringan):



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

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

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

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]);
}
```

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



2. Kemudian kita modifikasi view untuk menampilkan data filtering pada [PWL\\_POS/resources/views/user/index.blade.php](#)

```
@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>
    <tbody>
        <tr><td>1</td><td>admin</td><td>Admin</td><td>Administrator</td><td><a href="#">Edit</a> <a href="#">Delete</a></td></tr>
    </tbody>
</table>
</div>
</div>
@endsection

<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" name="level_id" id="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>
```



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 })
66 
```

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

```
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 </script>
95 @endpush
96 
```



```
        searchables: false // searchables: true, jika kita ingin mencari di kolom tersebut
    });
    $('#level_id').on('change', function() {
        dataUser.ajax.reload();
    });
});
```

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).'" 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);
}
```

```
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).'" 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);
}
```

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



The screenshot shows the 'Daftar User' (User List) page. On the left is a sidebar with navigation links: Dashboard, Data Pengguna, Level User, **Data User**, Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main area has a title 'Daftar User' and a subtitle 'Daftar user yang terdaftar dalam sistem'. It features a 'Filtering' section with dropdowns for 'Filter:' (set to '- Semua -') and 'Jenis Pengguna'. A 'Searching' input field is also present. Below is a table with columns: ID#, Username, Nama, Level Pengguna, and Aksi (Detail, Edit, Hapus). The table contains 6 entries:

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]

At the bottom, it says 'Showing 1 to 6 of 6 entries' and includes 'Previous' and 'Next' buttons.

This screenshot shows the same 'Daftar User' page but with a different set of data. The table now contains 7 entries:

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

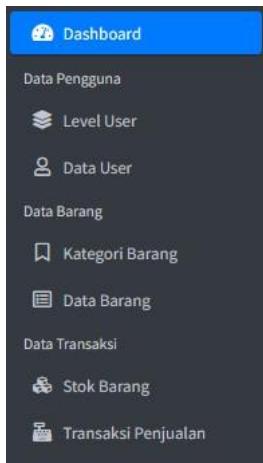


## F. PERTANYAAN

Jawablah pertanyaan berikut sesuai pemahaman materi di atas

1. Apa perbedaan *frontend template* dengan *backend template*?
2. Apakah *layouting* itu penting dalam membangun sebuah website?
3. Jelaskan fungsi dari komponen laravel blade berikut : `@include()`, `@extend()`, `@section()`, `@push()`, `@yield()`, dan `@stack()`
4. Apa fungsi dan tujuan dari variable `$activeMenu` ?

## G. TUGAS



Implementasikan menu yang belum ada di laravel starter code ini sesuai dengan Studi Kasus Point of Sales Sederhana. Silahkan terapkan kode di laravel starter code untuk menu-menu yang sesuai dengan menu di samping ini.

Jawab: Tugasnya untuk UTS

ID	Kategori Kode	Kategori Nama	Aksi
1	KTG01	Peralatan Rumah Tangga	[Detail] [Edit] [Hapus]
2	KTG02	Makanan	[Detail] [Edit] [Hapus]
3	KTG03	Minuman	[Detail] [Edit] [Hapus]
4	KTG04	Peralatan Bayi	[Detail] [Edit] [Hapus]
5	KTG05	Produk Kecantikan	[Detail] [Edit] [Hapus]
6	mkrr	makanan ringan	[Detail] [Edit] [Hapus]



**Daftar Barang**

Daftar Barang yang Terdaftar Dalam Sistem

Filter: - Semua - Kategori Barang

Show 10 entries Search:

ID	Kategori Nama	Barang Kode	Barang Nama	Harga Beli	Harga Jual	Aksi
1	Peralatan Rumah Tangga	BRG01	Alat Pel	41000	43000	[Detail] [Edit] [Hapus]
2	Peralatan Rumah Tangga	BRG02	Sikat Lantai	6000	10000	[Detail] [Edit] [Hapus]
3	Peralatan Rumah Tangga	BRG03	Kemoceng	15000	20000	[Detail] [Edit] [Hapus]
4	Makanan	BRG04	Platoz	8000	12000	[Detail] [Edit] [Hapus]
5	Makanan	BRG05	Kanzler Chicken Nugget	35000	45000	[Detail] [Edit] [Hapus]
6	Minuman	BRG06	Fanta 500 ml	5000	7000	[Detail] [Edit] [Hapus]
7	Minuman	BRG07	Teh Pucuk 250 ml	2300	3000	[Detail] [Edit] [Hapus]
8	Minuman	BRG08	Tebz	6000	9000	[Detail] [Edit] [Hapus]
9	Peralatan Bayi	BRG09	Sabun Cusson Baby	2500	3500	[Detail] [Edit] [Hapus]
10	Produk Kecantikan	BRG10	Lipstick make over	50000	70000	[Detail] [Edit] [Hapus]

Showing 1 to 10 of 10 entries Previous Next

**Daftar Stok**

Daftar Stok yang terdaftar dalam sistem

Filter: - Semua - Nama Barang

Show 10 entries Search:

ID	Nama Barang	Penyetok	Tanggal Stok	Jumlah	Aksi
1	Alat Pel	Bima galang ardiansyah	2024-04-24 00:00:00	19	[Detail] [Edit] [Hapus]
2	Sikat Lantai	Manager 3	2024-04-24 00:00:00	0	[Detail] [Edit] [Hapus]
3	Kemoceng	Staff	2024-03-10 00:00:00	10	[Detail] [Edit] [Hapus]
4	Platoz	Staff	2024-03-10 00:00:00	15	[Detail] [Edit] [Hapus]
5	Kanzler Chicken Nugget	Staff	2024-03-10 00:00:00	7	[Detail] [Edit] [Hapus]
6	Fanta 500 ml	Staff	2024-03-10 00:00:00	40	[Detail] [Edit] [Hapus]
7	Teh Pucuk 250 ml	Staff	2024-03-10 00:00:00	30	[Detail] [Edit] [Hapus]
8	Tebz	Staff	2024-03-10 00:00:00	50	[Detail] [Edit] [Hapus]
9	Sabun Cusson Baby	staff	2024-04-24 00:00:00	3	[Detail] [Edit] [Hapus]
10	Lipstick make over	Staff	2024-03-10 00:00:00	22	[Detail] [Edit] [Hapus]

Showing 1 to 10 of 10 entries Previous Next



The screenshot shows a web application interface for a Point of Sale (POS) system. The left sidebar contains navigation links for Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area is titled "Daftar Penjualan" and displays a table of sales entries. The table columns are ID, Pembeli, Petugas, Total Harga, Kode Penjualan, Tanggal, and Aksi. There are four entries listed:

ID	Pembeli	Petugas	Total Harga	Kode Penjualan	Tanggal	Aksi
1	aci	bima galang ardiansyah	215000	PJ01	2024-04-24 00:00:00	[Detail] [Edit] [Hapus]
2	ana	staff	70000	PJ02	2024-04-24 00:00:00	[Detail] [Edit] [Hapus]
3	abe	staff	50000	PJ03	2024-04-24 00:00:00	[Detail] [Edit] [Hapus]
4	aji	Manager 3	150000	PJ04	2024-04-24 00:00:00	[Detail] [Edit] [Hapus]

At the bottom of the page, there is a copyright notice: "Copyright © 2014-2021 AdminLTE.io. All rights reserved." and a version information: "Version 3.2.0".

\*\*\* Sekian, dan selamat belajar \*\*\*