



Mata Kuliah	:	Pemrograman Web Lanjut (PWL)
Program Studi	:	D4 – Sistem Informasi Bisnis
Semester	:	4 (empat)
Pertemuan ke-	:	5 (lima)
Nama	:	Aqila Nur Azza

## JOBSHEET 05

### Blade View, Web Templating(AdminLTE), Datatables

Sesuai dengan **Studi Kasus PWL.pdf**.

Jadi project Laravel 10 kita masih sama dengan menggunakan repositori **PWL\_POS**.

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

## A. Blade View

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

### 1. Apa Itu Blade?

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

### 2. Keuntungan Menggunakan Blade



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

### 3. Sintaks Dasar Blade

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

- **Menampilkan Variabel:**

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

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

- **Struktur Kontrol:**

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

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

- **Komentar**

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



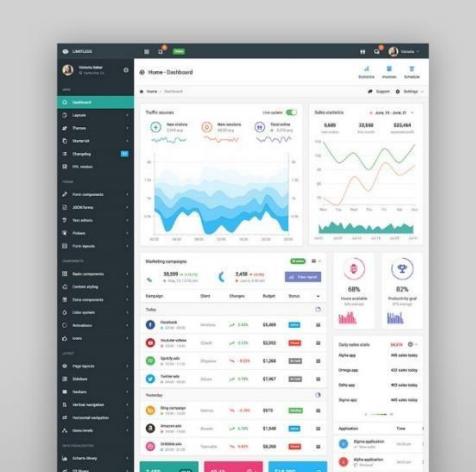
## B. Web Template

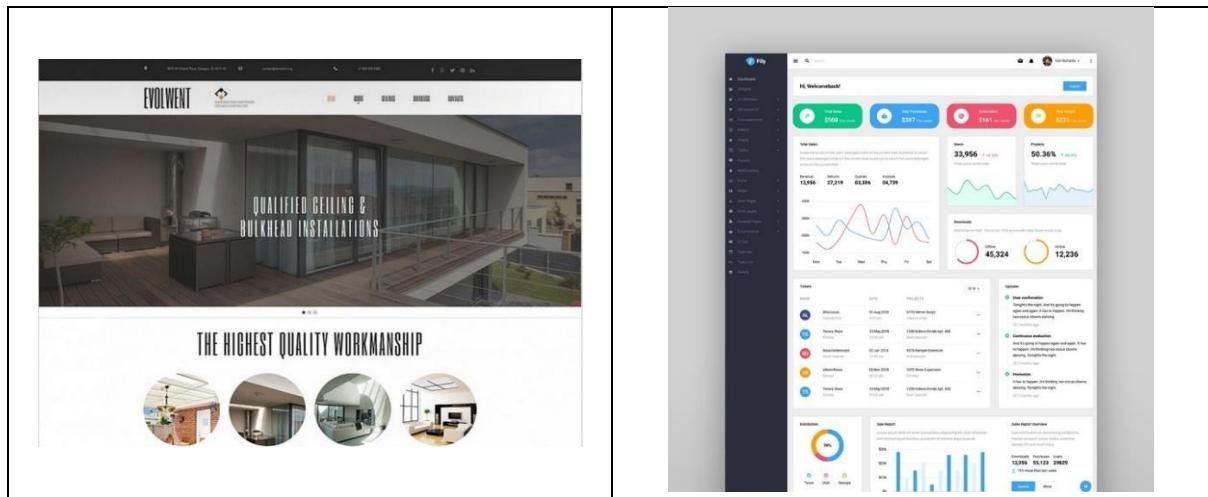
Merupakan suatu file yang sudah memiliki desain tertentu sehingga, kita tinggal memodifikasi bagian-bagian tertentu saja untuk mendapatkan hasil yang sudah bagus. Dengan menggunakan template yang sesuai dengan kebutuhan, kita bisa mengurangi waktu pembuatan suatu website secara drastis.

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

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

### Contoh Web template

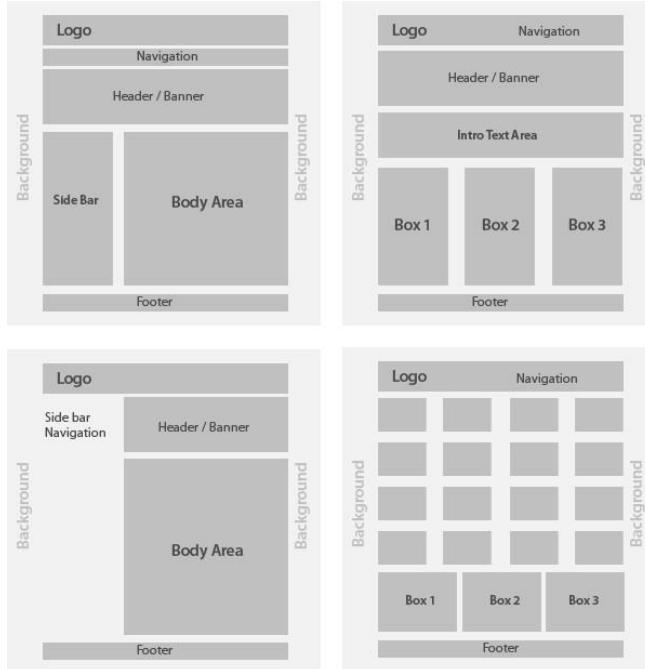
Frontend Template	Backend Template
	



## A. Web Layout

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

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



## C. Layouting AdminLTE

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

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

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



## Praktikum 1 - Layouting AdminLTE:

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

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

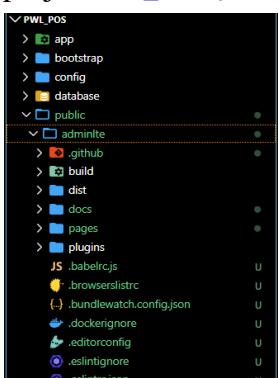
Source code (zip) Feb 8, 2022

Source code (tar.gz) Feb 8, 2022

575 68 91 197 64 39 750 people reacted

31 Join discussion

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



3. Selanjutnya kita buka di browser dengan alamat

[http://localhost/PWL\\_POS/public/adminlte](http://localhost/PWL_POS/public/adminlte) maka akan muncul tampilan seperti berikut



The screenshot shows the AdminLTE 3 dashboard interface. On the left is a dark sidebar with a user profile for 'Alexander Pierce' and a navigation menu including 'Dashboard v1' (which is selected), 'Dashboard v2', 'Dashboard v3', 'Widgets', 'Layout Options', 'Charts', 'UI Elements', 'Forms', 'Tables', and 'EXAMPLES'. The main content area features four cards: 'New Orders' (150), 'Bounce Rate' (53%), 'User Registrations' (44), and 'Unique Visitors' (65). Below these are two larger charts: a line chart titled 'Sales' showing monthly trends from January to July, and a map titled 'Visitors' showing visitor locations across the United States.

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

The screenshot shows the 'Extras' section of AdminLTE 3. The 'Blank Page' option is highlighted with a red box. The sidebar also lists other extras like 'Login & Register v1', 'Lockscreen', 'Legacy User Menu', 'Language Menu', 'Error 404', 'Error 500', 'Pace', and 'Starter Page'. The main content area displays the 'Blank Page' template, which consists of a header, a large central area with a sales chart, a 'To Do List' section with several items, and a 'Sales Graph' section at the bottom.

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



Blank Page

Home / Blank Page

Title

Start creating your amazing application!

Footer

Ask Highlight... @ +

LABELS

- Important
- Warning
- Informational

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

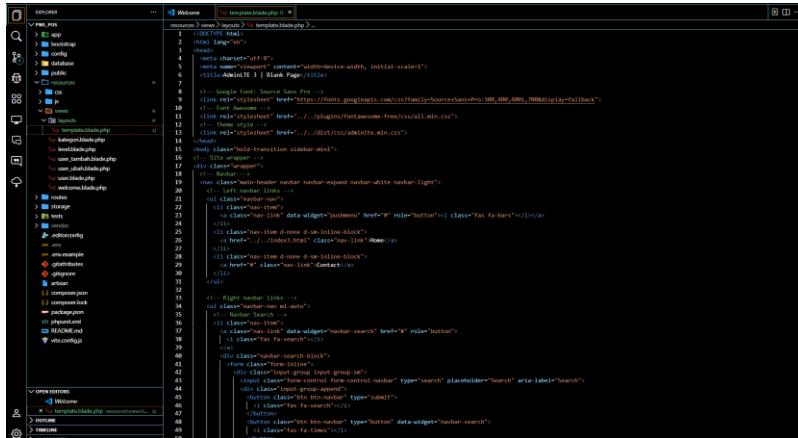
Version 3.2.0

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



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>AdminLTE 3 | Blank Page</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="https://plugins/fontawesome-free/css/all.min.css">
    <!-- Theme style -->
    <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}>
    <!-- AdminLTE -->
    <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}>
</head>
<body>
    <div class="hold-transition sidebar-mini">
        <div class="wrapper">
            <div class="main-header navbar navbar-expand navbar-light">
                ...
                <div>Left navbar Links </div>
                <ul class="navbar-nav">
                    ...
                    <li class="nav-item d-none d-sm-inline-block">
                        <a href="#" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>
                    </li>
                    <li class="nav-item d-none d-sm-inline-block">
                        <a href="{{ url('/') }}" class="nav-link">Home</a>
                    </li>
                    <li class="nav-item d-none d-sm-inline-block">
                        <a href="#" class="nav-link">Contact</a>
                    </li>
                </ul>
                ...
                <div>Right navbar Links </div>
                <ul class="navbar-nav ml-auto">
                    ...
                    <li>Navbar Search </li>
                    <li class="nav-item">
                        <a href="#" data-widget="navbar-search" href="#" role="button">
                            <i class="fas fa-search"></i>
                        </a>
                    </li>
                    <div class="navbar-search-block">
                        <form>
                            <div class="input-group input-group-sm">
                                <div class="form-control form-control-navbar" id="search" placeholder="Search" aria-label="Search">
                                    <input type="text" id="search" name="search" placeholder="Search" aria-label="Search" />
                                </div>
                            </div>
                        </form>
                    </div>
                </ul>
            </div>
        </div>
    </div>
</body>
```

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



8. File [layouts/template.blade.php](#) adalah file utama untuk templating website  
9. Pada baris [1-14](#) file [template.blade.php](#), kita modifikasi

```
resources > views > layouts > template.blade.php > html > head
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>
```



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

Header.blade.php

```
resources > views > layouts > header.blade.php > nav.main.header.navbar.navbar-expand.navbar-white.navbar-light
1 <nav class="main-header navbar navbar-expand navbar-white navbar-light"
2   <!-- Left navbar links -->
3   <ul class="navbar-nav">
4     <li class="nav-item">
5       <a href="#" class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>
6     </li>
7     <li class="nav-item d-none d-sm-inline-block">
8       <a href="../../index3.html" class="nav-link">Home</a>
9     </li>
10    <li class="nav-item d-none d-sm-inline-block">
11      <a href="#" class="nav-link">Contact</a>
12    </li>
13  </ul>
14
15  <!-- Right navbar links -->
16  <ul class="navbar-nav ml-auto">
17    <!-- Navbar Search -->
18    <li class="nav-item">
19      <a class="nav-link" data-widget="navbar-search" href="#" role="button">
20        <i class="fa fa-search"></i>
21      </a>
22    </li>
23    <div class="navbar-search-block">
24      <form class="form-inline">
25        <div class="input-group input-group-sm">
26          <input class="form-control form-control-navbar" type="search" placeholder="Search" aria-label="Search">
27          <div class="input-group-append">
28            <button class="btn btn-navbar" type="submit">
29              <i class="fas fa-search"></i>
30            </button>
31            <button class="btn btn-navbar" type="button" data-widget="navbar-search">
32              <i class="fas fa-times"></i>
33            </button>
34          </div>
35        </div>
36      </form>
37    </li>
38
39  <!-- Messages Dropdown Menu -->
40  <li class="nav-item dropdown">
41    <a class="nav-link" data-toggle="dropdown" href="#">
42      <i class="far fa-comments"></i>
43      <span class="badge badge-danger navbar-badge">3</span>
44    </a>
45    <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">
46      <a href="#" class="dropdown-item">
47        <i class="fas fa-envelope"></i> Messages <span class="float-right">3</span>
48      </a>
49      <div class="dropdown-divider">
50    </div>
```

## Template.blade.php

```
19 @include(view: 'layouts.header')
20 <!-- ./navbar -->
21
22 <!-- Main Sidebar Container -->
23 <aside class="main-sidebar sidebar-dark-primary elevation-4">
24   <!-- Brand Logo -->
25   <a href="../../index3.html" class="brand-link">
26     | 
27     | <span class="brand-text font-weight-light">AdminLTE 3</span>
28   </a>
```

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

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

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

```
84 |   </div>
85 |   <!-- /.content-wrapper -->
86 |
87 |   @include('layouts.footer')
88 | </div>
89 | <!-- ./wrapper -->
90 |
```

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>
101
```

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>
99
```

15. Sekarang masuk pada bagian konten. Konten kita bagi menjadi 2, yaitu elemen untuk **breadcrumb** dan elemen untuk **content**.
16. Perhatikan file `template.blade.php` pada baris **38-52** kita jadikan sebagai elemen **4-breadcrumb**. Kita blok baris **38-52** lalu kita **cut**, dan **paste**-kan di file `PWL_POS/resource/view/layouts/breadcrumb.blade.php` (buat file `breadcrumb.blade.php` jika belum ada). Sehingga tampilan dari file `template.blade.php` menjadi seperti berikut

```
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>
```

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')`



```
39
40     <!-- Main content -->
41     <section class="content">
42         |     @yield('content')
43     </section>
44     <!-- /.content -->
45 </div>
46 <!-- /.content-wrapper -->
```

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

```
<head>
    <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}>
</head>
<body class="hold-transition sidebar-mini">
    <!-- Site wrapper -->
    <div class="wrapper">
        <!-- Navbar -->
        @include(['layouts.header'])
        <!-- /.navbar -->

        <!-- Main Sidebar Container -->
        <aside class="main-sidebar sidebar-dark-primary elevation-4">
            <!-- Brand Logo -->
            <a href="{{ url('/') }}" class="brand-link">
                
                <span class="brand-text font-weight-light">PWL - Starter Code</span>
            </a>

            <!-- Sidebar -->
            @include(['layouts.sidebar'])
            <!-- /.sidebar -->
        </aside>

        <!-- Content Wrapper. Contains page content -->
        <div class="content-wrapper">
            <!-- Content Header (Page header) -->
            @include(['layouts.breadcrumb'])

            <!-- Main content -->
            <section class="content">
                |     @yield('content')
            </section>
            <!-- /.content -->
        </div>
        <!-- /.content-wrapper -->

        @include(['layouts.footer'])
    </div>
```

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

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



## Praktikum 2 - Penerapan Layouting:

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

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

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

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



```
resources > views > welcome.blade.php > ...
1 @extends('layouts.template')
2
3 @section('content')
4
5 <div class="card">
6   <div class="card-header">
7     <h3 class="card-title">Halo, apa kabar!!!</h3>
8     <div class="card-tools">x</div>
9   </div>
10  <div class="card-body">
11    | Selamat datang semua, ini adalah halaman utama dari aplikasi ini.
12  </div>
13 </div>
14
15 @endsection
16
```

3. Kita modifikasi file PWL\_POS/resources/views/layouts/breadcrumb.blade.php

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

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

5. Kita tambahkan kode berikut router web.php

```
13  |
14  | route::get('/', [WelcomeController::class, 'index']);
15  |
16  |
```

6. Sekarang kita coba jalankan di browser dengan mengetikkan url

[http://localhost/PWL\\_POS/public](http://localhost/PWL_POS/public)

Selamat Datang

Halo, apa kabar!!!

Selamat datang semua, ini adalah halaman utama dari aplikasi ini.

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

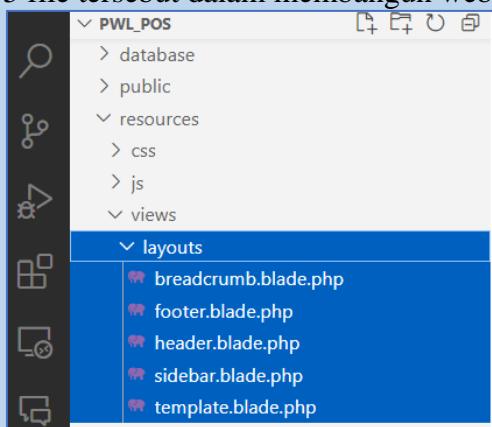
Version 3.2.0



7. Jangan lupa commit dan push ke github PWL\_POS kalian

### INFO

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





## D. jQuery Datatables di Laravel

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

### INFO

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

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

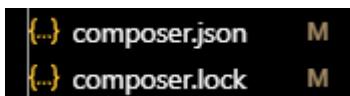
Pada praktikum kali ini kita **tidak menggunakan Vite/NodeJS** dalam mengelola datatables, kita cukup menggunakan jQuery datatables bawaan AdminLTE dan library Yajra *laravel-datatables*.

## E. Implementasi jQuery Datatables di Laravel

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

### Praktikum 3 – Implementasi jQuery Datatable di AdminLTE :

1. Kita modifikasi proses CRUD pada tabel `m_user` pada praktikum ini
2. Kita gunakan library Yajra-datable dengan mengetikkan perintah pada CMD  
`composer require yajra/laravel-datatables:^10.0` atau  
`composer require yajra/laravel-datatables-oracle`



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

```
routes > web.php > ...
15 | routes are loaded by the RouteServiceProvider and all of them will
16 | be assigned to the "web" middleware group. Make something great!
17 |
18 */
19
20 // Route::get('/', function () {
21 //     return view('welcome');
22 // });
23
24 // Route::get('/level', [LevelController::class, 'index']);
25 // Route::get('/kategori', [KategoriController::class, 'index']);
26 // Route::get('/user', [UserController::class, 'index']);
27
28 // Route::get('/user/tambah', [UserController::class, 'tambah']);
29 // Route::post('/user/tambah_simpan', [UserController::class, 'tambah_simpan']);
30 // Route::get('/user/ubah/{id}', [UserController::class, 'ubah']);
31 // Route::put('/user/ubah_simpan/{id}', [UserController::class, 'ubah_simpan']);
32 // Route::get('/user/hapus/{id}', [UserController::class, 'hapus']);
33
34 Route::get('/', [WelcomeController::class, 'index']);
35
36 Route::group(['prefix' => 'user'], function () {
37     Route::get('/', [UserController::class, 'index']); // menampilkan halaman awal user
38     Route::post('/list', [UserController::class, 'list']); // menampilkan data user dalam bentuk json untuk datatables
39     Route::get('/create', [UserController::class, 'create']); // menampilkan halaman form tambah user
40     Route::post('/', [UserController::class, 'store']); // menyimpan data user baru
41     Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
42     Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
43     Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
44     Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
45 });
46
```



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

```
app > Http > Controllers > UserController.php > UserController
1  <?php
2  namespace App\Http\Controllers;
3
4  use App\Models\LevelModel;
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Yajra\DataTables\Facades\DataTables;
8
9  class UserController extends Controller
10 [
11     // Menampilkan halaman awal user
12     public function index()
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', [
26             'breadcrumb' => $breadcrumb,
27             'page' => $page,
28             'activeMenu' => $activeMenu
29         ]);
30     }
31 ]
```



5. Lalu kita buat view pada [PWL/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

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

```



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

```
resources > views > user > index.blade.php > ...
1 @extends('layouts.template')
2
3 @section('content')
4 <div class="card card-outline card-primary">
5   <div class="card-header">
6     <h3 class="card-title">{{ $page->title }}</h3>
7     <div class="card-tools">
8       <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}>Tambah</a>
9     </div>
10   </div>
11   <div class="card-body">
12     <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
13       <thead>
14         <tr>
15           <th>ID</th>
16           <th>Username</th>
17           <th>Nama</th>
18           <th>Level Pengguna</th>
19           <th>Aksi</th>
20         </tr>
21       </thead>
22     </table>
23   </div>
24 </div>
25 @endsection
26
27 @push('css')
28 @endpush
29
30 @push('js')
31 <script>
32   $(document).ready(function() {
33     var datalUser = $('#table_user').DataTable({
34       // serverSide: true, jika ingin menggunakan server-side processing
35       serverSide: true,
36       ajax: {
37         "url": "{{ url('user/list') }}",
38         "datatype": "json",
39         "type": "POST"
40       },
41       columns: [
42         {
43           // Nomor urut dari Laravel DataTable addIndexColumn()
44           data: "DT_RowIndex",
45           className: "text-center",
46           orderable: false,
47           searchable: false
48         },
49         {
50           data: "username",
```



- 
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/jszip/jszip.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 <script>
80 // Untuk mengirimkan token Laravel CSRF pada setiap request ajax
81 $.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
82 </script>
83 @stack('js') <!-- Digunakan untuk memanggil custom js dari perintah push('js') pada masing-masing view -->
84 </body>
85 </html>
```

```
resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini
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.bootstrap.min.css') }}>
17  <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/css/buttons.bootstrap.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>
23 <body class="hold-transition sidebar-mini">
24   <!-- Site wrapper -->
25   <div class="wrapper">
26     <!-- Navbar -->
27     @include('layouts.header')
28     <!-- /.navbar -->
29
30     <!-- Main Sidebar Container -->
31     <aside class="main-sidebar sidebar-dark-primary elevation-4">
32       <!-- Brand Logo -->
33       <a href="{{ url('/') }}" class="brand-link">
34         PWL - Starter Code</span>
36       </a>
37
38     <!-- Sidebar -->
39     @include('layouts.sidebar')
40     <!-- /.sidebar -->
41   </aside>
42
43   <!-- Content Wrapper. Contains page content -->
44   <div class="content-wrapper">
45     <!-- Content Header (Page header) -->
46     @include('layouts.breadcrumb')
```



```
48 |     <!-- Main content -->
49 |     <section class="content">
50 |         @yield('content')
51 |     </section>
52 |     <!-- /.content -->
53 |     </div>
54 |     <!-- /.content-wrapper -->
55 |
56 |     @include('layouts.footer')
57 | 
```

</div>

<!-- ./wrapper -->

59

<!-- jQuery -->

60 | <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"/></script>

61 | <!-- Bootstrap 4 -->

62 | <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"/></script>

63 | <!-- DataTables & Plugins -->

64 | <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}"/></script>

65 | <script src="{{ asset('adminlte/plugins/datatables/dataTables.bootstrap4.min.js') }}"/></script>

66 | <script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}"/></script>

67 | <script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap.min.js') }}"/></script>

68 | <script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}"/></script>

69 | <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap.min.js') }}"/></script>

70 | <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap.min.js') }}"/></script>

71 | <script src="{{ asset('adminlte/plugins/jstz/lib/jstz.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 |

// Untuk mengirimkan token laravel CSRF pada setiap request AJAX

80 | \$ajaxSetup(['headers': {'X-CSRF-TOKEN': \$('meta[name="csrf-token"]').attr('content')}]);

81 |

82 |

83 |

84 |

85 |

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

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

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

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

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

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/Kasir	Staff/Kasir	[Detail] [Edit] [Hapus]
4	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
5	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
6	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
7	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]
8	manager12	Manager11	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]);
}
```

```
52 // Menampilkan halaman form tambah user
53 public function create()
54 {
55     $breadcrumb = (object) [
56         'title' => 'Tambah User',
57         'list' => ['Home', 'User', 'Tambah']
58     ];
59
60     $page = (object) [
61         'title' => 'Tambah user baru'
62     ];
63
64     $level = LevelModel::all(); // ambil data level untuk ditampilkan di form
65     $activeMenu = 'user'; // set menu yang sedang aktif
66
67     return view('user.create', [
68         'breadcrumb' => $breadcrumb,
69         'page' => $page,
70         'level' => $level,
71         'activeMenu' => $activeMenu
72     ]);
73 }
74 }
```

10. Sekarang kita buat form untuk menambah data, kita buat file [PWL/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>
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label">Password</label>
                </div>
```



```
<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/\_create.blade.php -

```
1 @extends('layouts.template')
2
3  @section('content')
4  <div class="card card-outline card-primary">
5      <div class="card-body">
6          <h4>{{ $page->title }}</h4>
7          <div class="card-tools"></div>
8      </div>
9      <div class="card-body">
10         <form method="POST" action="{{ url('user') }}" class="form-horizontal">
11             @csrf
12             <div class="form-group row">
13                 <label class="col-1 control-label col-form-label">Level:</label>
14                 <div class="col-11">
15                     <select class="form-control" id="level_id" name="level_id" required>
16                         <option value=""><option>
17                         <option value="{{ $item->level_name }}">{!! $item->level_name !!}{!! $item->level_name !!}<option>
20                         <option value=""><option>
21                         <option value=""><option>
22                         <option value=""><option>
23                         <option value=""><option>
24                         <option value=""><option>
25                     </select>
26                     @error('level_id')
27                         <small class="form-text text-danger">{{ $message }}</small>
28                     @enderror
29                 </div>
30             </div>
31             <div class="form-group row">
32                 <label class="col-1 control-label col-form-label">Username:</label>
33                 <div class="col-11">
34                     <input type="text" class="form-control" id="username" name="username"
35                         | value="{{ old('username') }}" required>
36                     @error('username')
37                         <small class="form-text text-danger">{{ $message }}</small>
38                     @enderror
39                 </div>
40             </div>
41             <div class="form-group row">
42                 <label class="col-1 control-label col-form-label">Nama:</label>
43                 <div class="col-11">
44                     <input type="text" class="form-control" id="nama" name="nama"
45                         | value="{{ old('name') }}" required>
46                     @error('name')
47                         <small class="form-text text-danger">{{ $message }}</small>
48                     @enderror
49                 </div>
50             </div>
51         </form>
52     </div>
53 
```

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');
}
```



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

Tambah User

Tambah user baru

Level: - Pilih Level -

Username: Manager Utama

Nama: Aqila

Password: \*\*\*\*\*

Simpan Kembali

Penjelasan :

Akan menampilkan form untuk mengisi data yang baru.

Tambah User

Tambah user baru

Level: Manager

Username: Manager Utama

Nama: Aqila

Password: \*\*\*\*\*

Simpan Kembali



ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staff	Staff/Kasir	Staff/Kasir	[Detail] [Edit] [Hapus]
4	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
5	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
6	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
7	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]
8	manager12	Manager11	Manager	[Detail] [Edit] [Hapus]
9	Manager Utama	Aqila	Manager	[Detail] [Edit] [Hapus]

13. Selanjutnya, kita masuk pada bagian menampilkan detail data user (klik tombol **Detail**) pada halaman user. Route yang bertugas untuk menangkap request detail adalah Penjelasan :
- Akan terjadi error karena belum ada fungsi show yang diambil dari UserController

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



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

```
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
```

```
// 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
    ]);
}
```

15. Kemudian kita buat view di `PWL/resources/views/user/show.blade.php`

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

@section('content')


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



@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>
@else
    <table class="table table-bordered table-striped table-hover table-sm">
        <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>


```



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

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

16. Sekarang kalian coba untuk melihat detail data user di browser, dan coba untuk mengetikkan id yang salah contoh <http://localhost/PWL/public/user/100> amati apa yang terjadi, dan laporkan!!!

Detail user	
ID	14
Level	Manager
Username	Manager Utama
Nama	Aqila
Password	*****

Penjelasan : akan diambil dari ID yang kita pilih

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



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

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

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

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

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



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



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

@push('css')
@endpush
@push('js')
@endpush
resources>views>user>edit.blade.php ...
1 @extends('layouts.template')
2
3  @section('content')
4  <div class="card card-outline card-primary">
5      <div class="card-header">
6          <h3 class="card-title">{{ $page->title }}
```

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

➤ Sebelum

The screenshot shows a web application interface for managing users. On the left is a sidebar with navigation links like Dashboard, Data Pengguna, Level User, Data User (which is currently selected), Data Barang, Kategori Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area has a title 'Edit User'. It contains four input fields: 'Level' set to 'Manager', 'Username' set to 'Manager Pertama', 'Nama' set to 'Aqila', and a 'Password' field which is empty. Below the password field is a note: 'Abalkan (jangan diisi) jika tidak ingin mengganti password user.' At the bottom are two buttons: 'Simpan' (Save) and 'Kembali' (Back). The browser's address bar shows the URL: localhost:PWL\_2025/PWL\_WEEKS/2/PWL\_POS/public/user/14/edit. The overall theme is dark.



➤ Sesudah

9	Manager Pertama	Aqila	Manager	<a href="#">Detail</a>	<a href="#">Edit</a>	<a href="#">Hapus</a>
---	-----------------	-------	---------	------------------------	----------------------	-----------------------

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

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

```
// Menghapus data user  
public function destroy(string $id)  
{  
    $check = UserModel::find($id); // untuk mengecek apakah data user dengan id yang dimaksud ada atau tidak  
    if (!$check) {  
        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/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 |     @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>  
21 |                 <th>ID</th>  
22 |                 <th>Username</th>  
23 |                 <th>Nama</th>  
24 |                 <th>Level Pengguna</th>  
25 |                 <th>Aksi</th>  
26 |             </tr>  
27 |         </thead>  
28 |         </table>  
29 |     </div>  
30 | 
```



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

➤ Sebelum

Daftar user yang terdaftar dalam sistem

Tambah

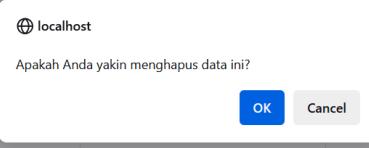
ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	manager	Manager	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	staff	Staff/Kasir	Staff/Kasir	<button>Detail</button> <button>Edit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

Showing 1 to 9 of 9 entries

Previous 1 Next

Tambah

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



➤ Sesudah



Daftar user yang terdaftar dalam sistem				
<b>Data user berhasil dihapus</b>				
Show 10 entries Search:				
ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	manager	Manager	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	staff	Staff/Kasir	Staff/Kasir	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	manager_dua	Manager 2	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	manager22	Manager Dua Dua	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
6	manager33	Manager Tiga Tiga	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
7	manager56	Manager55	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
8	manager12	Manager11	Manager	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

25. Selamat, kalian sudah membuat Laravel Starter Code untuk proses CRUD dengan menggunakan template AdminLTE dan plugin jQuery Datatables.
26. Jangan lupa commit dan push ke github PWL\_POS kalian

## F. Data Searching 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.

The screenshot shows a Laravel application interface. On the left is a sidebar with navigation links: Home, Contact, 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 'Daftar User' and a sub-header 'Daftar user yang terdaftar dalam sistem'. It features a search bar with the placeholder 'Search:' and a 'Searching' button. Below is a table with columns: ID, Username, Nama, Level Pengguna, and Aksi (with buttons for Detail, Edit, and Hapus). The table contains 6 entries. At the bottom, it says 'Showing 1 to 6 of 6 entries' and includes 'Previous' and 'Next' buttons.

#### 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**



```
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 |         $level = LevelModel::all(); // ambil data level untuk filter level
26 |
27 |         return view('user.index', [
28 |             'breadcrumb' => $breadcrumb,
29 |             'page' => $page,
30 |             'level' => $level,
31 |             'activeMenu' => $activeMenu
32 |         ]);
33 |
34 |     }

```

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

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

```

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

```
56 |         // serverSide: true, jika ingin menggunakan server-side processing
57 |         serverSide: true,
58 |         ajax: {
59 |             "url": "{{ url('user/list') }}",
60 |             "dataType": "json",
61 |             "type": "POST",
62 |             "data": function(d) [
63 |                 d.level_id = $('#level_id').val();
64 |             ],
65 |             columns: [
66 |                 {
67 |             }

```



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

```
    });
    $('#level_id').on('change', function() {
        dataUser.ajax.reload();
    });
});
</script>
@endpush
```

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

```
34  }
35
36     // Ambil data user dalam bentuk json untuk datatables
37     public function list(Request $request)
38     {
39         $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
40             ->with('level');
41
42         // Filter data user berdasarkan level_id
43         if ($request->level_id) {
44             $users->where('level_id', $request->level_id);
45         }
46     }
```



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

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	[Detail] [Edit] [Hapus]
2	manager	Manager	Manager	[Detail] [Edit] [Hapus]
3	staff	Staff/Kasir	Staff/Kasir	[Detail] [Edit] [Hapus]
4	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
5	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
6	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
7	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]

Implementasi :

ID	Username	Nama	Level Pengguna	Aksi
1	manager	Manager	Manager	[Detail] [Edit] [Hapus]
2	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
3	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
4	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
5	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]
6	manager12	Manager11	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.  
8. Jangan lupa commit dan push ke github PWL\_POS kalian

## G. Tugas



Implementasikan web layout dan datatables, pada menu berikut ini

- ✓ Tabel m\_level
- ✓ Tabel m\_kategori
- ✓ Tabel m\_supplier
- ✓ Tabel m\_barang

### 1. Tabel m\_level

#### ➤ Tampilan Dashboard Level User

ID	Level Kode	Level Nama	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff/Kasir	[Detail] [Edit] [Hapus]

#### ➤ Tampilan Detail Level User

ID	1
Level Kode	ADM
Level Nama	Administrator

#### ➤ Tampilan Tambah Level User



PWL - Starter Code

Home Contact

Tambah Level

Tambah level baru

Level Kode: CEO

Nama: CEO

Simpan Kembali

Daftar level yang terdaftar dalam sistem

Tambah

Data level berhasil ditambahkan

Filter: - Semua -

Level Pengguna

Show 10 entries Search:

ID	Level Kode	Level Nama	aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff/Kasir	[Detail] [Edit] [Hapus]
4	CEO	CEO	[Detail] [Edit] [Hapus]

Showing 1 to 4 of 4 entries Previous 1 Next

➤ Tampilan Mengedit Level Kode

PWL - Starter Code

Home Contact

Edit Level

Edit level

Kode: C

Nama: CEO

Simpan Kembali

Daftar level yang terdaftar dalam sistem

Tambah

Data level berhasil diubah

Filter: - Semua -

Level Pengguna

Show 10 entries Search:

ID	Level Kode	Level Nama	aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff/Kasir	[Detail] [Edit] [Hapus]
4	C	CEO	[Detail] [Edit] [Hapus]

Showing 1 to 4 of 4 entries Previous 1 Next



➤ Tampilan Menghapus data berdasarkan ID

Data level berhasil dihapus

ID	Level Code	Level Name	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff/Kasir	[Detail] [Edit] [Hapus]

2. Tabel m\_barang

➤ Tampilan Dashboard Kategori Barang

Daftar Kategori

ID	Kode Kategori	Nama Kategori	Aksi
1	FB	Food & Beverage	[Detail] [Edit] [Hapus]
2	BH	Beauty & Health	[Detail] [Edit] [Hapus]
3	HC	Home Care	[Detail] [Edit] [Hapus]
4	BK	Baby & Kids	[Detail] [Edit] [Hapus]
5	EL	Electronics	[Detail] [Edit] [Hapus]
6	SNK	Camilan	[Detail] [Edit] [Hapus]
7	CML	Cemilan	[Detail] [Edit] [Hapus]
8	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]
9	SPT	Sepatu	[Detail] [Edit] [Hapus]

➤ Tambah Barang Baru

Tambah Kategori

Tambah kategori baru

Kode	FSH
Nama	Fashion

[Simpan] [Kembali]



Data Pengguna      Data Transaksi      Stok Barang      Transaksi Penjualan

Tambah

Data Pengguna

Level User

Data User

Data Barang

**Kategori Barang**

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Data kategori berhasil ditambahkan

Show 10 entries Search:

ID	Kode Kategori	Nama Kategori	Aksi
1	FB	Food & Beverage	[Detail] [Edit] [Hapus]
2	BH	Beauty & Health	[Detail] [Edit] [Hapus]
3	HC	Home Care	[Detail] [Edit] [Hapus]
4	BK	Baby & Kids	[Detail] [Edit] [Hapus]
5	EL	Electronics	[Detail] [Edit] [Hapus]
6	SNK	Camilan	[Detail] [Edit] [Hapus]
7	CML	Cemilan	[Detail] [Edit] [Hapus]
8	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]
9	SPT	Sepatu	[Detail] [Edit] [Hapus]
10	FSH	Fashion	[Detail] [Edit] [Hapus]

Showing 1 to 10 of 10 entries Previous 1 Next

➤ Edit Kategori Barang

PWL - Starter Code

Home Contact

Search

Data Pengguna

Level User

Data User

Data Barang

**Kategori Barang**

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Edit Kategori

Home / Kategori / Edit

Edit kategori

Kode FH

Nama Fashion

Simpan Kembali

Data User

Data Barang

**Kategori Barang**

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Data kategori berhasil ditambahkan

Show 10 entries Search:

ID	Kode Kategori	Nama Kategori	Aksi
1	FB	Food & Beverage	[Detail] [Edit] [Hapus]
2	BH	Beauty & Health	[Detail] [Edit] [Hapus]
3	HC	Home Care	[Detail] [Edit] [Hapus]
4	BK	Baby & Kids	[Detail] [Edit] [Hapus]
5	EL	Electronics	[Detail] [Edit] [Hapus]
6	SNK	Camilan	[Detail] [Edit] [Hapus]
7	CML	Cemilan	[Detail] [Edit] [Hapus]
8	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]
9	SPT	Sepatu	[Detail] [Edit] [Hapus]
10	FH	Fashion	[Detail] [Edit] [Hapus]

Showing 1 to 10 of 10 entries Previous 1 Next

➤ Detail Kategori Barang



Detail Kategori

ID	11
Kode Kategori	FH
Nama Kategori	Fashion

➤ Delete Kategori Barang

ID	Kode Kategori	Nama Kategori	Aksi
1	FB	Food & Beverage	[Detail] [Edit] [Hapus]
2	BH	Beauty & Health	[Detail] [Edit] [Hapus]
3	HC	Home Care	[Detail] [Edit] [Hapus]
4	BK	Baby & Kids	[Detail] [Edit] [Hapus]
5	EL	Electronics	[Detail] [Edit] [Hapus]
6	SNK	Camilan	[Detail] [Edit] [Hapus]
7	CML	Cemilan	[Detail] [Edit] [Hapus]
8	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]
9	SPT	Sepatu	[Detail] [Edit] [Hapus]

Daftar kategori yang terdaftar dalam sistem

Data kategori berhasil dihapus

ID	Kode Kategori	Nama Kategori	Aksi
1	FB	Food & Beverage	[Detail] [Edit] [Hapus]
2	BH	Beauty & Health	[Detail] [Edit] [Hapus]
3	HC	Home Care	[Detail] [Edit] [Hapus]
4	BK	Baby & Kids	[Detail] [Edit] [Hapus]
5	EL	Electronics	[Detail] [Edit] [Hapus]
6	SNK	Camilan	[Detail] [Edit] [Hapus]
7	CML	Cemilan	[Detail] [Edit] [Hapus]
8	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]
9	SPT	Sepatu	[Detail] [Edit] [Hapus]

3. Tampilan Table m\_barang

➤ Tampilan Dashboard Tabel Barang



↗ **Tampilan Detail Tabel Barang**

The screenshot shows a table of items with columns: ID, Kode, Nama Kategori, Harga Beli, Harga Jual, and Aksi. A dropdown filter for 'Nama Kategori' is open, showing options like Food & Beverage, Beauty & Health, Home Care, Baby & Kids, Electronics, Camilan, Cemilan, Minuman Ringan, Sepatu, Tisu, and Popok Bayi. The table data includes:

ID	Kode	Nama Kategori	Harga Beli	Harga Jual	Aksi
1	B001	Food & Beverage	2000	3000	[Detail] [Edit] [Hapus]
2	B002	Food & Beverage	10000	12000	[Detail] [Edit] [Hapus]
3	B003	Beauty & Health	5000	7000	[Detail] [Edit] [Hapus]
4	B004	Beauty & Health	15000	18000	[Detail] [Edit] [Hapus]
5	B005	Home Care	10000	13000	[Detail] [Edit] [Hapus]
6	B006	Tisu	8000	10000	[Detail] [Edit] [Hapus]
7	B007	Popok Bayi	50000	55000	[Detail] [Edit] [Hapus]

↗ **Tampilan Tambah Tabel Barang**

The screenshot shows a form for adding a new item. The fields are: Kode (B011), Nama (Butter), Kode Kategori (Food & Beverage), Harga Beli (10000), and Harga Jual (15000). At the bottom are 'Simpan' and 'Kembali' buttons.



PWL - Starter Code

Home Contact Search

### Data Barang

Daftar barang yang terdaftar dalam sistem

Tambah

Data barang berhasil disimpan

Filter: - Semua - Kategori Barang

Show 10 entries Search:

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
11	B011	Butter	Food & Beverage	10000	15000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 11 to 11 of 11 entries Previous 1 2 Next

➤ Tampilan Edit Tabel Barang

PWL - Starter Code

Home Contact Search

### Edit Barang

Edit barang

Kode: B011

Nama: Mentega

Level: Food & Beverage

Harga Beli: 10000

Harga Jual: 15000

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

PWL - Starter Code

Home Contact Search

### Data Barang

Daftar barang yang terdaftar dalam sistem

Tambah

Data barang berhasil diubah

Filter: - Semua - Kategori Barang

Show 10 entries Search:

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
11	B011	Mentega	Food & Beverage	10000	15000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 11 to 11 of 11 entries Previous 1 2 Next

➤ Tampilan Hapus Tabel Barang



ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
1	B001	Air Mineral	Food & Beverage	2000	3000	[Detail] [Edit] [Hapus]
2	B002	Roti Tawar	Food & Beverage	10000	12000	[Detail] [Edit] [Hapus]
3	B003	Sabun Mandi	Beauty & Health	5000	7000	[Detail] [Edit] [Hapus]
4	B004	Shampoo	Beauty & Health	15000	18000	[Detail] [Edit] [Hapus]
5	B005	Pembersih Lantai	Home Care	10000	13000	[Detail] [Edit] [Hapus]
6	B006	Tisu	Home Care	8000	10000	[Detail] [Edit] [Hapus]
7	B007	Popok Bayi	Baby & Kids	50000	55000	[Detail] [Edit] [Hapus]

#### 4. Tampilan Tabel m\_supplier

##### ➤ Tampilan Dasboard Supplier

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP001	PT Sumber Jaya Abadi	Jl. Industri Raya No. 10, Jakarta Barat	[Detail] [Edit] [Hapus]
2	SP002	PT Mitra Sejahtera	Jl. Pahlawan No. 25, Surabaya	[Detail] [Edit] [Hapus]
3	SP003	PT Sentosa Makmur Bersama	Jl. Gatot Subroto No. 15, Bandung	[Detail] [Edit] [Hapus]

Penjelasan : Menambahkan Migration dan Seeder pada table m\_supplier lalu menambahkan kode program pada sidebar.blade.php

```
<li class="nav-header">Data Supplier</li>
<li class="nav-item">
    <a href="{{ url('/supplier') }}" class="nav-link {{ ($activeMenu == 'supplier') ? 'active' : '' }}>
        <i class="nav-icon fas fa-truck"></i>
        <p>Supplier Barang</p>
    </a>
</li>
```

##### ➤ Tampilan Detail Table Supplier



## Detail Supplier

[Home](#) / [Supplier](#) / [Detail](#)

### Detail Supplier

ID	1
Kode supplier	SP001
Nama supplier	PT Sumber Jaya Abadi
Alamat supplier	Jl. Industri Raya No. 10, Jakarta Barat

[Kembali](#)

## ➤ Tampilan Tambah Table Supplier

### Tambah Supplier

[Home](#) / [Supplier](#) / [Tambah](#)

#### Tambah supplier baru

Kode	SP004
Nama	PT Gudang Garam
Alamat	Jl. Semampir Kota Kediri

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

### Daftar supplier yang terdaftar dalam sistem

[Tambah](#)

Data supplier berhasil ditambahkan

Show 10 entries

Search:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP001	PT Sumber Jaya Abadi	Jl. Industri Raya No. 10, Jakarta Barat	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	SP002	PT Mitra Sejahtera	Jl. Pahlawan No. 25, Surabaya	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	SP003	PT Sentosa Makmur Bersama	Jl. Gatot Subroto No. 15, Bandung	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	SP004	PT Gudang Garam	Jl. Semampir Kota Kediri	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 4 of 4 entries

[Previous](#) [1](#) [Next](#)

## ➤ Tampilan Edit Table Supplier

### Edit Supplier

[Home](#) / [Supplier](#) / [Edit](#)

#### Edit Supplier

Kode	SP004
Nama	PT Gudang Garam
Alamat	Jl. Semampir No 20, Kota Kediri

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



## Daftar Supplier

Home / Supplier

Daftar supplier yang terdaftar dalam sistem

Tambah

Data supplier berhasil diubah

Show 10 entries

Search:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP001	PT Sumber Jaya Abadi	Jl. Industri Raya No. 10, Jakarta Barat	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	SP002	PT Mitra Sejahtera	Jl. Pahlawan No. 25, Surabaya	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	SP003	PT Sentosa Makmur Bersama	Jl. Gatot Subroto No. 15, Bandung	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	SP004	PT Gudang Garam	Jl. Semampir No 20, Kota Kediri	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Showing 1 to 4 of 4 entries

Previous 1 Next

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

Version 3.2.0

### ➤ Tampilan Hapus Table Supplier

Data supplier berhasil diubah

Show 10 entries

Search:

localhost

Apakah Anda yakin menghapus data ini?

OK

Cancel

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP001	PT Sumber Jaya Abadi	Jl. Industri Raya No. 10, Jakarta Barat	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	SP002	PT Mitra Sejahtera	Jl. Pahlawan No. 25, Surabaya	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	SP003	PT Sentosa Makmur Bersama	Jl. Gatot Subroto No. 15, Bandung	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	SP004	PT Gudang Garam	Jl. Semampir No 20, Kota Kediri	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

## Daftar Supplier

Home / Supplier

Daftar supplier yang terdaftar dalam sistem

Tambah

Data supplier berhasil dihapus

Show 10 entries

Search:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SP001	PT Sumber Jaya Abadi	Jl. Industri Raya No. 10, Jakarta Barat	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	SP002	PT Mitra Sejahtera	Jl. Pahlawan No. 25, Surabaya	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	SP003	PT Sentosa Makmur Bersama	Jl. Gatot Subroto No. 15, Bandung	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Showing 1 to 3 of 3 entries

Previous 1 Next

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