



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

JOBSHEET 05

Blade View, Web Templating(AdminLTE), Datatables

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

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

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

A. Blade View

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

1. Apa Itu Blade?

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

2. Keuntungan Menggunakan Blade



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

3. Sintaks Dasar Blade

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

- **Menampilkan Variabel:**

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

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

- **Struktur Kontrol:**

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

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

- **Komentar**

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



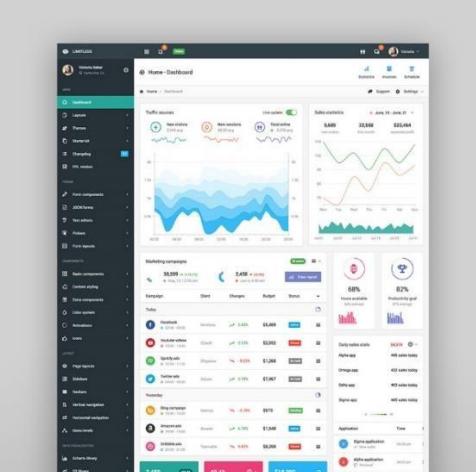
B. Web Template

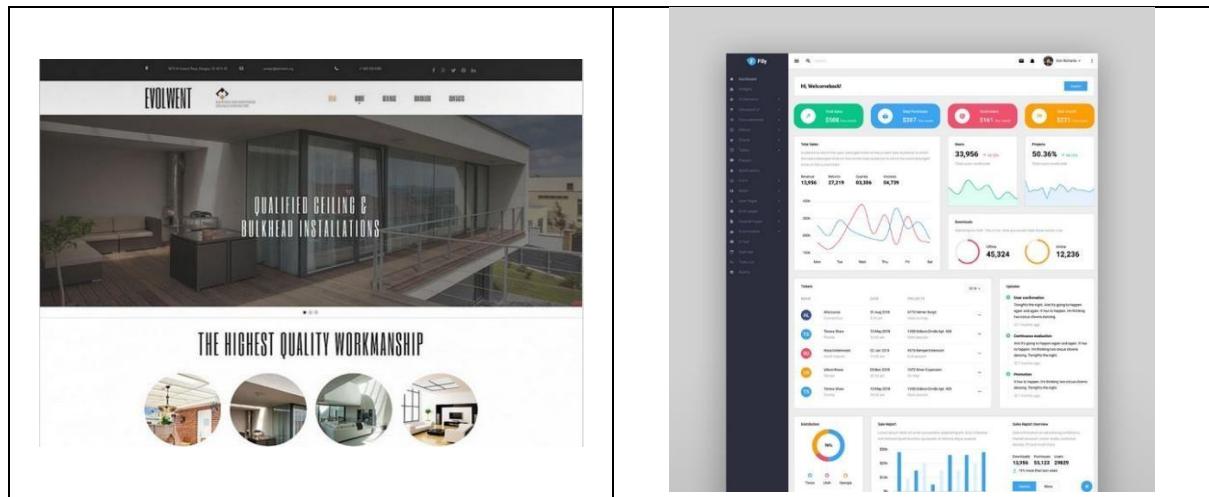
Merupakan suatu file yang sudah memiliki desain tertentu sehingga, kita tinggal memodifikasi bagian-bagian tertentu saja untuk mendapatkan hasil yang sudah bagus. Dengan menggunakan template yang sesuai dengan kebutuhan, kita bisa mengurangi waktu pembuatan suatu website secara drastis.

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

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

Contoh Web template

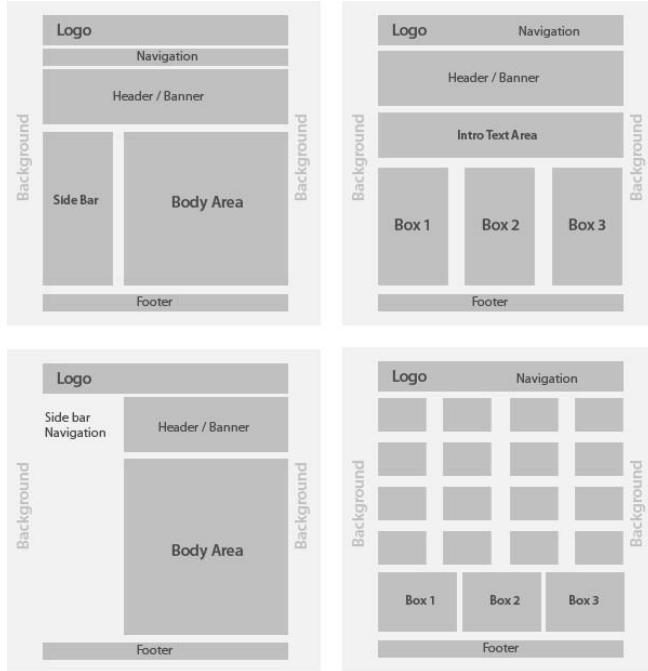
Frontend Template	Backend Template
	



A. Web Layout

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

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



C. Layouting AdminLTE

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

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

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

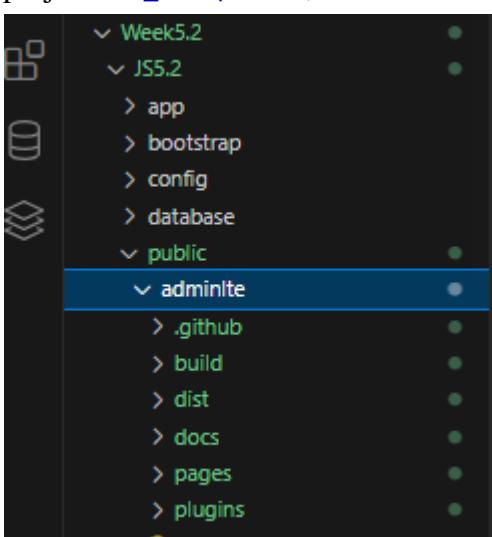


Praktikum 1 - Layouting AdminLTE:

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

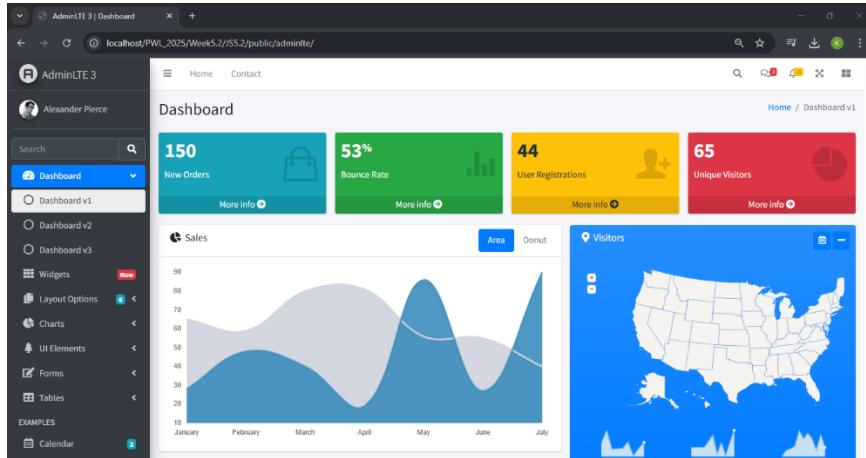
The screenshot shows the GitHub release page for AdminLTE v3.2.0. It displays the release date (Feb 8, 2022), the author (REJack), and the commit hash (v3.2.0). The main content area shows the release title "AdminLTE v3.2.0" and a "Latest" button. Below it, there's a "Breaking Change!" note stating that node v12 has been dropped and support will not be given for bugs/errors. The "Top-Features:" section lists several improvements. At the bottom, there are sections for "Contributors" (with 17 contributors shown as icons) and "Assets" (with two download links: "Source code (zip)" and "Source code (tar.gz)", where the "Source code (zip)" link is highlighted with a red checkmark). Below the assets, there are reaction counts (493, 56, 79, 166, 51, 23) and a total of 643 people reacted.

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

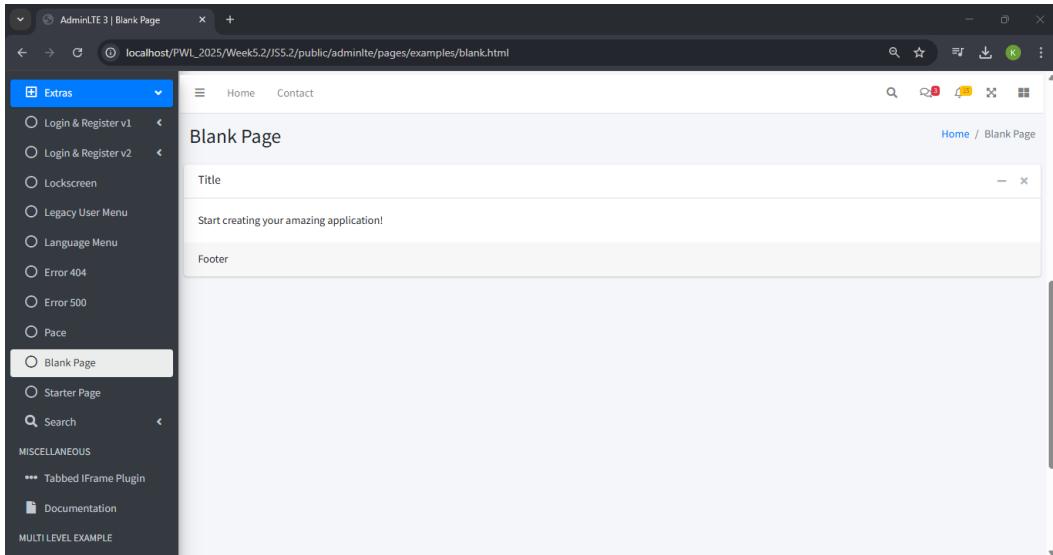


3. Selanjutnya kita buka di browser dengan alamat

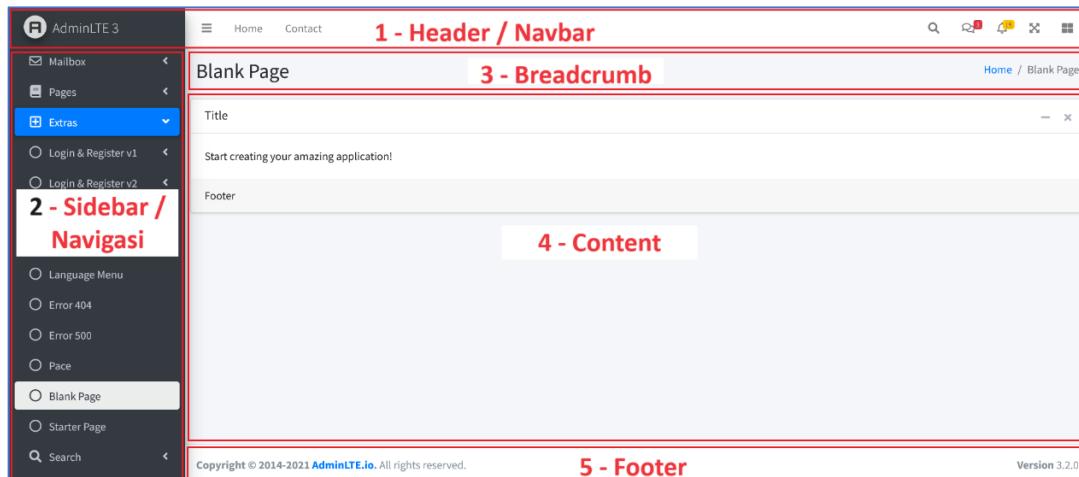
http://localhost/PWL_POS/public/adminlte maka akan muncul tampilan seperti berikut



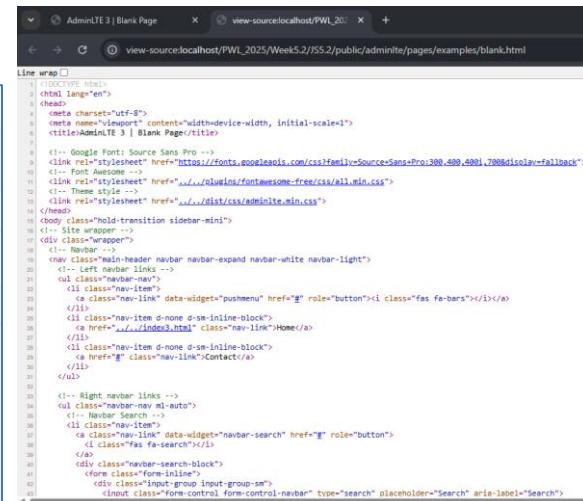
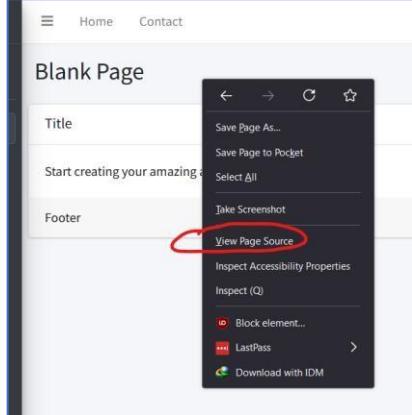
4. Kita klik menu **Extras > Blank Page**, page inilah yang akan menjadi dasar web template



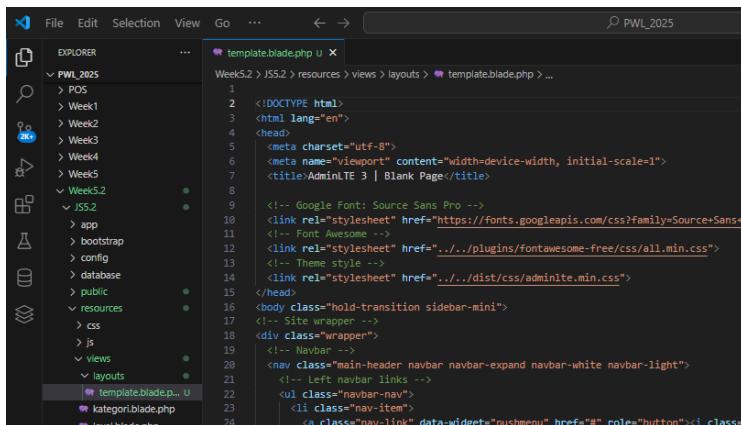
5. Dari sini kita bisa melakukan layouting halaman Blank Page ini menjadi 4 element seperti pada gambar berikut



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



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



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

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>{{config(key: 'app.name', default: 'PWL Laravel Starter Code')}}</title>
7
8      <!-- 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(path: 'adminlte/plugins/fontawesome-free/css/all.min.css') }}>
12     <!-- Theme style -->
13     <link rel="stylesheet" href="{{ asset(path: '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

`template.blade.php`

```
15 <body class="hold-transition sidebar-mini">
16   !-- Site wrapper -->
17 <div class="wrapper">
18   !-- Navbar -->
19   @include(view: 'layouts.header')
20   !-- /.navbar -->
21
22   !-- Main Sidebar Container -->
23 <aside class="main-sidebar sidebar-dark-primary elevation-4">
24   !-- Brand Logo -->
25   <a href="/" />index3.html" class="brand-link">
```

`headre.blade.php`

```
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 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="fas fa-search"></i>
21      </a>
22      <div class="navbar-search-block">
23        <form class="form-inline">
24          <div class="input-group input-group-sm">
25            <input class="form-control form-control-navbar" type="search" placeholder="Search" aria-label="Search">
26            <div class="input-group-append">
27              <button class="btn btn-navbar" type="submit">
28                <i class="fas fa-search"></i>
29              </button>
30            </div>
31          </div>
32        </form>
33      </div>
34    </li>
35  </ul>
```

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`

```
22   !-- Main Sidebar Container -->
23   <aside class="main-sidebar sidebar-dark-primary elevation-4">
24     !-- Brand Logo -->
25     <a href="{{ url(path: '/') }}" class="brand-link">
26       
31     @include(view: 'layouts.sidebar')
32     <!-- ./sidebar -->
33   </aside>
34
```

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

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

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

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

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

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

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')**

```
40 |     <!-- Main content -->
41 |     <section class="content">
42 |         @yield('content')
43 |     </section>
```

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

```
17 |     <div class="wrapper">
18 |         <!-- Navbar -->
19 |         @include(view: 'layouts.header')
20 |     <!-- ./navbar -->
21 |
22 |         <!-- Main Sidebar Container -->
23 |         <aside class="main-sidebar sidebar-dark-primary elevation-4">
24 |             <!-- Brand Logo -->
25 |             <a href="{{ url(path: '/') }}" class="brand-link">
26 |                 PWL - Starter Code</span>
28 |             </a>
29 |
30 |             <!-- Sidebar -->
31 |             @include(view: 'layouts.sidebar')
32 |         <!-- ./sidebar -->
33 |     </aside>
34 |
35 |         <!-- Content Wrapper. Contains page content -->
36 |         <div class="content-wrapper">
37 |             <!-- Content Header (Page header) -->
38 |             @include(view: 'layouts.breadcrumb')
39 |
40 |             <!-- Main content -->
41 |             <section class="content">
42 |                 @yield('content')
43 |             </section>
44 |             <!-- ./content -->
45 |         </div>
46 |         <!-- ./content-wrapper -->
47 |
48 |         @include(view: 'layouts.footer')
49 |     </div>
50 |     <!-- ./wrapper -->
51 | 
```

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](#)

```
Week5.2 > JS5.2 > app > Http > Controllers > WelcomeController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 3 references | 0 implementations
6 class WelcomeController extends Controller
7 {
8     0 references | 0 overrides
9     public function index(): Factory|View
10    {
11        $breadcrumb = (object) [
12            'title' => 'Selamat Datang',
13            'list'  => ['Home', 'Welcome']
14        ];
15
16        $activeMenu = 'dashboard';
17
18        return view('welcome', ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu]);
19    }
20 }
```

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

```
Week5.2 > JS5.2 > resources > views > welcome.blade.php > ...
1 @extends('layouts.template')
2
3 @section('content')
4
5 <div class="card">
6     <div class="card-header">
7         <h3 class="card-title">Halo, apa kabar!!!</h3>
8         <div class="card-tools"></div>
9     </div>
10    <div class="card-body">
11        Selamat datang semua, ini adalah halaman utama dari aplikasi ini.
12    </div>
13 </div>
14
15 @endsection
```

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

```
Week5.2 > JS5.2 > resources > views > layouts > breadcrumb.blade.php > section.content-header
1 <section class="content-header">
2     <div class="container-fluid">
3         <div class="row mb-2">
4             <div class="col-sm-6"><h1>{{ $breadcrumb->title }}</h1></div>
5             <div class="col-sm-6">
6                 <ol class="breadcrumb float-sm-right">
7                     @foreach($breadcrumb->list as $key => $value)
8                         @if($key == count($value) - 1)
9                             <li class="breadcrumb-item active">{{ $value }}</li>
10                        @else
11                            <li class="breadcrumb-item">{{ $value }}</li>
12                        @endif
13                    @endforeach
14                </ol>
15            </div>
16        </div>
17    </div>
18 </section>
```



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

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



```
Week5.2 > J55.2 > resources > views > layouts > sidebar.blade.php > div.sidebar
 1  <div class="sidebar">
 2    <!-- SidebarSearch Form -->
 3    <div class="form-inline mt-2">
 4      <div class="input-group" data-widget="sidebar-search">
 5        <input class="form-control form-control-sidebar" type="search" placeholder="Search" aria-label="Search">
 6        <input 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  <!-- Sidebar Menu -->
13  <nav class="mt-2">
14    <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">
15      <li class="nav-item">
16        <a href="{{ url(path: '/') }}" class="nav-link {{ ($activeMenu == 'dashboard') ? 'active' : '' }} ">
17          |   <i class="nav-icon fas fa-tachometer-alt"></i>
18          |   <p>Dashboard</p>
19        </a>
20      </li>
21      <li class="nav-header">Data Pengguna</li>
22      <li class="nav-item">
23        <a href="{{ url(path: '/level') }}" class="nav-link {{ ($activeMenu == 'level') ? 'active' : '' }} ">
24          |   <i class="nav-icon fas fa-layer-group"></i>
25          |   <p>Level User</p>
26        </a>
27      </li>
28
29      <li class="nav-item">
30        <a href="{{ url(path: '/user') }}" class="nav-link {{ ($activeMenu == 'user') ? 'active' : '' }} ">
31          |   <i class="nav-icon far fa-user"></i>
32          |   <p>Data User</p>
33        </a>
34      </li>
35      <li class="nav-header">Data Barang</li>
36      <li class="nav-item">
37        <a href="{{ url(path: '/kategori') }}" class="nav-link {{ ($activeMenu == 'kategori') ? 'active' : '' }} ">
38          |   <i class="nav-icon far fa-bookmark"></i>
39          |   <p>Kategori Barang</p>
40        </a>
41      </li>
42      <li class="nav-item">
43        <a href="{{ url(path: '/barang') }}" class="nav-link {{ ($activeMenu == 'barang') ? 'active' : '' }} ">
44          |   <i class="nav-icon far fa-list-alt"></i>
45          |   <p>Data Barang</p>
46        </a>
47      </li>
48      <li class="nav-header">Data Transaksi</li>
49      <li class="nav-item">
50        <a href="{{ url(path: '/stok') }}" class="nav-link {{ ($activeMenu == 'stok') ? 'active' : '' }} ">
51          |   <i class="nav-icon fas fa-cubes"></i>
52          |   <p>Stok Barang</p>
53        </a>
54      </li>
55      <li class="nav-item">
56        <a href="{{ url(path: '/barang') }}" class="nav-link {{ ($activeMenu == 'penjualan') ? 'active' : '' }} ">
57          |   <i class="nav-icon fas fa-cash-register"></i>
58          |   <p>Transaksi Penjualan</p>
59        </a>
60      </li>
61    </ul>
62  </div>
63 </div>
```

5. Kita tambahkan kode berikut router web.php

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

6. Sekarang kita coba jalankan di browser dengan mengetikkan url

http://localhost/PWL_POS/public

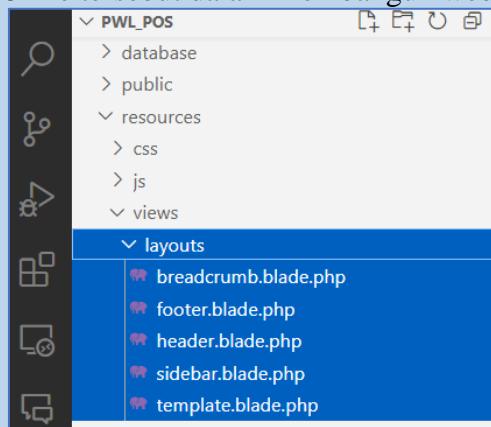


The screenshot shows a Laravel application interface. On the left is a dark sidebar with a navigation menu. The menu items include: Dashboard (selected), Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area has a header "Selamat Datang" and a sub-header "Halo, apa kabar!!!". Below that is a message: "Selamat datang semua, ini adalah halaman utama dari aplikasi ini.". At the bottom of the page, there is a copyright notice "Copyright © 2014-2021 AdminLTE.io. All rights reserved." and a "Version 3.2.0" link.

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`

```
C:\laragon\www\PWL_2025\Week5.2>composer require yajra/laravel-datatables:^10.0
The "10.0" constraint for "yajra/laravel-datatables" appears too strict and will likely not match what you want. See https://getcomposer.org/constraints
./composer.json has been updated
Running composer update yajra/laravel-datatables
Loading composer repositories with package information
Updating dependencies
Lock file operations: 7 installs, 0 updates, 0 removals
- Locking league/fractal (v0.20.2)
- Locking yajra/laravel-datatables (v10.0.0)
- Locking yajra/laravel-datatables-buttons (v10.0.9)
- Locking yajra/laravel-datatables-editor (v1.25.4)
- Locking yajra/laravel-datatables-fractal (v10.0.0)
- Locking yajra/laravel-datatables-html (v10.12.0)
- Locking yajra/laravel-datatables-oracle (v10.11.4)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 7 installs, 0 updates, 0 removals
- Installing yajra/laravel-datatables-oracle (v10.11.4): Extracting archive
- Installing yajra/laravel-datatables-html (v10.12.0): Extracting archive
- Installing League/fractal (v0.20.2): Extracting archive
```

```
(?) composer.json      M
(?) composer.lock      M
```



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

```
Week5.2 > JS5.2 > routes > 📄 web.php > ...
1  <?php
2
3  use App\Http\Controllers\WelcomeController;
4  use Illuminate\Support\Facades\Route;
5  use App\Http\Controllers\UserController;
6
7  /*
8  |-----
9  | Web Routes
10 |
11 |
12 | Here is where you can register web routes for your application. These
13 | routes are loaded by the RouteServiceProvider and all of them will
14 | be assigned to the "web" middleware group. Make something great!
15 |
16 */
17
18 Route::get('/', [WelcomeController::class, 'index']);
19
20 Route::group(['prefix' => 'user'], function () {
21     Route::get('/', [UserController::class, 'index']);           // menampilkan halaman awal user
22     Route::post('/list', [UserController::class, 'list']);       // menampilkan data user dalam bentuk json untuk datatables
23     Route::get('/create', [UserController::class, 'create']);   // menampilkan halaman form tambah user
24     Route::post('/', [UserController::class, 'store']);         // menyimpan data user baru
25     Route::get('/{id}', [UserController::class, 'show']);        // menampilkan detail user
26     Route::get('/{id}/edit', [UserController::class, 'edit']);   // menampilkan halaman form edit user
27     Route::put('/{id}', [UserController::class, 'update']);     // menyimpan perubahan data user
28     Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
29 });
30
31
32
```

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

```
Week5.2 > JS5.2 > app > Http > Controllers > 📄 UserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\LevelModel;
6  use App\Models\UserModel;
7  use Illuminate\Http\Request;
8  use Yajra\DataTables\Facades\DataTables;
9
10
11 class UserController extends Controller
12 {
13     // Menampilkan halaman awal user
14     public function index(): Factory|View
15     {
16         $breadcrumb = (object) [
17             'title' => 'Daftar User',
18             'list'  => ['Home', 'User']
19         ];
20
21         $page = (object) [
22             'title' => 'Daftar user yang terdaftar dalam sistem'
23         ];
24
25         $activeMenu = 'user'; // set menu yang sedang aktif
26
27         return view('user.index', data: [
28             'breadcrumb' => $breadcrumb,
29             'page'        => $page,
30             'activeMenu' => $activeMenu
31         ]);
32     }
33 }
```



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
            }
        ]
    });
}); @endpush
</script> @endpush
```



```
Week5.2 > JSS.2 > resources > views > users > index.blade.php > ...
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title"><{{ $page->title }}></h3>
7              <div class="card-tools">
8                  <a class="btn btn-sm btn-primary mt-1" href="{{ url(path: 'user/create') }}>Tambah</a>
9              </div>
10         </div>
11         <div class="card-body">
12             <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
13                 <thead>
14                     <tr>
15                         <th>ID</th>
16                         <th>Username</th>
17                         <th>Nama</th>
18                         <th>Level Pengguna</th>
19                         <th>Aksi</th>
20                     </tr>
21                 </thead>
22                 </table>
23             </div>
24         </div> @endsection
25     @push('css') @endpush
26
27     @push('js')
28         <script>
29             $(document).ready(function () {
30                 var dataUser = $('#table_user').DataTable({
31                     // serverSide: true, jika ingin menggunakan server side processing
32                     serverSide: true,
33                     ajax: {
34                         "url": "{{ url(path: 'user/list') }}",
35                         "dataType": "json",
36                         "type": "POST"
37                     },
38                     columns: [
39                         {
40                             // nomor urut dari laravel datatables addIndexColumn()
41                             data: "DT_RowIndex",
42                             className: "text-center",
43                             orderable: false,
44                             searchable: false
45                         },
46                         {
47                             data: "username",
48                             className: "",
49                             // orderable: true, jika ingin kolom ini bisa diurutkan
50                             orderable: true,
51                             // searchable: true, jika ingin kolom ini bisa dicari
52                             searchable: true
53                         },
54                         {
55                             data: "nama",
56                             className: "",
57                             orderable: true,
58                             searchable: true
59                         },
60                         {
61                             // mengambil data level hasil dari ORM berelasi data: "level.level_nama",
62                             data: "level",
63                             className: "",
64                             orderable: false,
65                             searchable: false
66                         },
67                         {
68                             data: "aksi",
69                             className: "",
70                             orderable: false,
71                             searchable: false
72                         }
73                     ]
74                 });
75             });
76         </script>
77     @endpush
78 
```

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

```
Week5.2 > JSS.2 > resources > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > aside.main-sidebar.sidebar-dark-pr
1  <!DOCTYPE html>
2  <html lang="en">
3
4      <head>
5          <meta charset="utf-8">
6          <meta name="viewport" content="width=device-width, initial-scale=1">
7          <title>{{config(key: 'app.name', default: 'PWL Laravel Starter Code')}}</title>
8
9          <meta name="csrf-token" content="{{ csrf_token() }}>
10         <!-- untuk mengirimkan token laravel CSRF pada setiap request ajax-->
11
12         <!-- Google Font: Source Sans Pro -->
13         <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
14
15         <!-- Font Awesome -->
16         <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/Fontawesome-free/css/all.min.css') }}>
17
18         <!-- DataTables -->
19         <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}>
20         <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}>
21         <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}>
22
23         <link rel="stylesheet" href="{{ asset(path: 'adminlte/dist/css/adminlte.min.css') }}>
24
25     @stack('css') <!-- digunakan untuk memanggil custom css dari perintah push('css') pada masing-masing view-->
26 
```



```
26 |<body class="hold-transition sidebar-mini">
27 |<!-- Site wrapper -->
28 |<div class="wrapper">
29 |<!-- Navbar -->
30 |<@include(view: 'layouts.header')>
31 |<!-- ./navbar -->
32 |
33 |<!-- Main Sidebar Container -->
34 |<aside class="main-sidebar sidebar-dark-primary elevation-4">
35 |<!-- Brand Logo -->
36 |<a href="{{ url(path: '/') }}" class="brand-link">
37 |
39 |<span class="brand-image img-circle elevation-3" style="opacity: .8">
40 |<span class="brand-text font-weight-light">PWL - Starter Code</span>
41 |</a>
42 |
43 |<!-- Sidebar -->
44 |<@include(view: 'layouts.sidebar')>
45 |<!-- ./sidebar -->
46 |</aside>
47 |
48 |<!-- Content Wrapper. Contains page content -->
49 |<div class="content-wrapper">
50 |<!-- Content Header (Page header) -->
51 |<@include(view: 'layouts.breadcrumb')>
52 |
53 |<!-- Main content -->
54 |<section class="content">
55 |<@yield('content')>
56 |</section>
57 |<!-- /.content -->
58 |</div>
59 |<!-- /.content-wrapper -->
60 |
61 |<@include(view: 'layouts.footer')>
62 |</div>
63 |<!-- ./wrapper -->
64 |
65 |<!-- jQuery -->
66 |<script src="{{ asset(path: 'adminlte/plugins/jquery/jquery.min.js') }}"></script>
67 |<!-- Bootstrap 4 -->
68 |<script src="{{ asset(path: 'adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
69 |<!-- DataTables & Plugins -->
70 |<script src="{{ asset(path: 'adminlte/plugins/datatables/jquery.dataTables.min.js') }}"></script>
71 |<script src="{{ asset(path: 'adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}"></script>
72 |<script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.responsive.bootstrap4.min.js') }}"></script>
73 |<script src="{{ asset(path: 'adminlte/plugins/datatables-responsive/js/dataTables.responsive.bootstrap4.min.js') }}"></script>
74 |<script src="{{ asset(path: 'adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}"></script>
75 |<script src="{{ asset(path: 'adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}"></script>
76 |<script src="{{ asset(path: 'adminlte/plugins/jspdf/jspdf.min.js') }}"></script>
77 |<script src="{{ asset(path: 'adminlte/plugins/pdfmake/pdfmake.min.js') }}"></script>
78 |<script src="{{ asset(path: 'adminlte/plugins/pdfmake/vfs_fonts.js') }}"></script>
79 |<script src="{{ asset(path: 'adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>
80 |<script src="{{ asset(path: 'adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>
81 |<script src="{{ asset(path: 'adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>
82 |<!-- AdminLTE App -->
83 |<script src="{{ asset(path: 'adminlte/dist/js/adminlte.min.js') }}"></script>
84 |<script>
85 |// Untuk mengirimkan token Laravel CSRF pada setiap request ajax
86 |$.ajaxSetup({ headers: { 'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content') } });
87 |</script>
88 |
89 |@stack('js') <!-- Digunakan untuk memanggil custom js dari perintah push('js') pada masing-masing view -->
90 |</body>
91 |
92 |</html>
```

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

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

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



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

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

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

```
54 // Menampilkan halaman form tambah user
55 1 reference | 0 overrides
56 public function create(): Factory|View
57 {
58     $breadcrumb = (object) [
59         'title' => 'Tambah User',
60         'list' => ['Home', 'User', 'Tambah']
61     ];
62
63     $page = (object) [
64         'title' => 'Tambah user baru'
65     ];
66
67     $level = LevelModel::all(); // ambil data level untuk ditampilkan di form
68     $activeMenu = 'user'; // set menu yang sedang aktif
69
70     return view(view: 'user.create', data: ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $activeMenu]);
71 }
72 }
```



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 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>
                </div>
            </div> @endsection
            @push('css') @endpush
            @push('js') @endpush
        </div>
    </div>
</div>
```



```
Week5.2 > JS5.2 > resources > views > user > create.blade.php > div.card.card-outline.card-primary > div.card-body > form.form-horizontal > div.form-group.row
1  @extends('layouts.template')
2
3  @section('content')
4  <div class="card card-outline card-primary">
5      <div class="card-header">
6          <h3 class="card-title">{{ $page->title }}</h3>
7          <div class="card-tools"></div>
8      </div>
9      <div class="card-body">
10         <form method="POST" action="{{ url(path: 'user') }}" class="form-horizontal">
11             @csrf
12             <div class="form-group row">
13                 <label class="col-1 control-label col-form-label">Level</label>
14                 <div class="col-11">
15                     <select class="form-control" id="level_id" name="level_id" required>
16                         <option value=""> Pilih Level </option>
17                         @foreach($level as $item)
18                             <option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
19                         @endforeach
20                     </select>
21                     @error('level_id')
22                         <small class="form-text text-danger">{{ $message }}</small>
23                     @enderror
24                 </div>
25             </div>
26
27             <div class="form-group row">
28                 <label class="col-1 control-label col-form-label">Username</label>
29                 <div class="col-11">
30                     <input type="text" class="form-control" id="username" name="username" value="{{ old(key: 'username') }}" required>
31                     @error('username')
32                         <small class="form-text text-danger">{{ $message }}</small>
33                     @enderror
34                 </div>
35             </div>
36
37             <div class="form-group row">
38                 <label class="col-1 control-label col-form-label">Nama</label>
39                 <div class="col-11">
40                     <input type="text" class="form-control" id="nama" name="nama" value="{{ old(key: 'nama') }}" required>
41                     @error('nama')
42                         <small class="form-text text-danger">{{ $message }}</small>
43                     @enderror
44                 </div>
45             </div>
46
47             <div class="form-group row">
48                 <label class="col-1 control-label col-form-label">Password</label>
49                 <div class="col-11">
50                     <input type="password" class="form-control" id="password" name="password" required>
51                     @error('password')
52                         <small class="form-text text-danger">{{ $message }}</small>
53                     @enderror
54                 </div>
55             </div>
56
57             <div class="form-group row">
58                 <label class="col-1 control-label col-form-label"></label>
59                 <div class="col-11">
60                     <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
61                     <a class="btn btn-sm btn-default ml-1" href="{{ url(path: 'user') }}>Kembali</a>
62                 </div>
63             </div>
64         </form>
65     </div>
66     @endsection
67
68     @push('css')
69     @endpush
70
71     @push('js')
72     @endpush
73 }
```

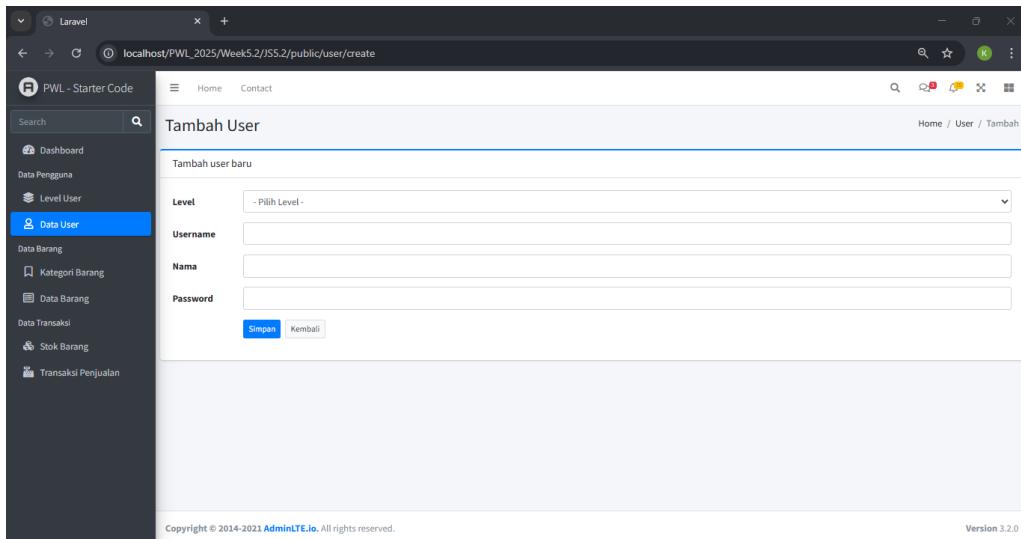
11. Kemudian untuk bisa menng-handle data yang akan disimpan ke database, kita buat fungsi **store()** di **UserController.php**

```
72     // Menyimpan data user baru
73     1 reference | 0 overrides
74     public function store(Request $request): mixed|RedirectResponse
75     {
76         $request->validate([
77             // username harus diisi, berupa string, minimal 3 karakter, dan bernilai unik di tabel m_user kolom username
78             'username' => 'required|string|min:3|unique:m_user,username',
79             'nama' => 'required|string|max:100', // nama harus diisi, berupa string, dan maksimal 100 karakter
80             'password' => 'required|min:5', // password harus diisi dan minimal 5 karakter
81             'level_id' => 'required|integer', // level_id harus diisi dan berupa angka
82         ]);
83
84         UserModel::create([
85             'username' => $request->username,
86             'name' => $request->name,
87             'password' => bcrypt(value: $request->password), // password dienkripsi sebelum disimpan
88             'level_id' => $request->level_id
89         ]);
90
91         return redirect(to: '/user')->with('success', 'Data user berhasil disimpan');
92
93     }
94 }
```



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

Hasil: akan menampilkan form untuk tambah data user



Tambah User

Tambah user baru

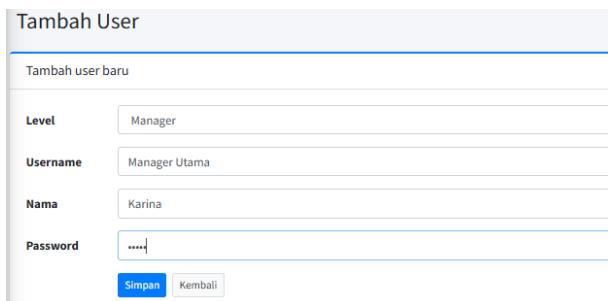
Level: - Pilih Level -

Username:

Nama:

Password:

Mencoba menambah data baru



Tambah User

Tambah user baru

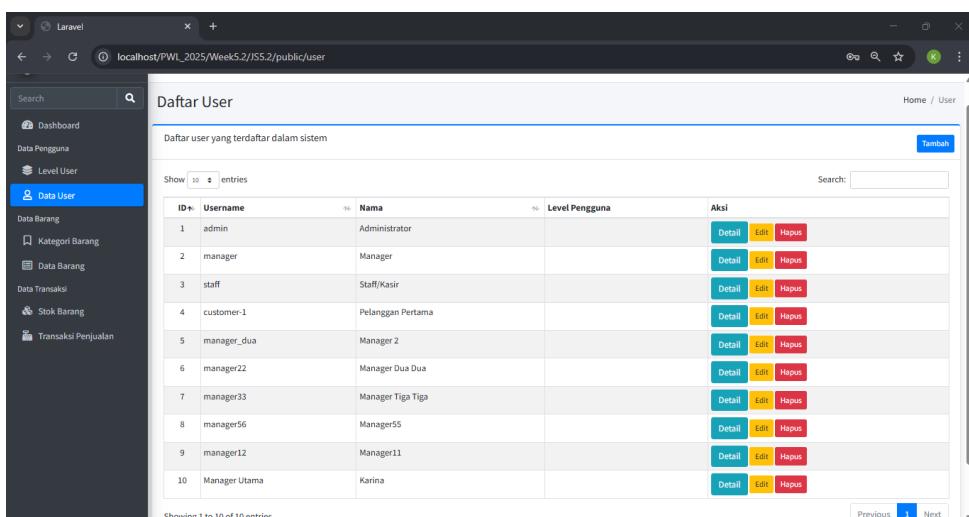
Level: Manager

Username: Manager Utama

Nama: Karina

Password:

Berhasil



Daftar User

Daftar user yang terdaftar dalam sistem

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

Showing 1 to 10 of 10 entries

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



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

```
93 // Menampilkan detail user
94 1 reference | 0 overrides
95 public function show(string $id): Factory|View
96 {
97     $user = UserModel::with('level')->find($id);
98
99     $breadcrumb = (object) [
100         'title' => 'Detail User',
101         'list' => ['Home', 'User', 'Detail']
102     ];
103
104     $page = (object) [
105         'title' => 'Detail user'
106     ];
107
108     $activeMenu = 'user'; // set menu yang sedang aktif
109
110     return view('user.show', data: [
111         'breadcrumb' => $breadcrumb,
112         'page' => $page,
113         'user' => $user,
114         'activeMenu' => $activeMenu
115     ]);
116 }
117 }
```

15. Kemudian kita buat `view` di `PWL/resources/views/user/show.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">
                <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>
        </div>
    </div> @endsection

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



```
Week5.2 > SS5.2 > resources > views > user > show.blade.php > ...
1  @extends('layouts.template')
2  @section('content')
3  <div class="card card-outline card-primary">
4      <div class="card-header">
5          <h3 class="card-title">{{ $page->title }}</h3>
6          <div class="card-tools"></div>
7      </div>
8      <div class="card-body">
9          @empty($user)
10             <div class="alert alert-danger alert-dismissible">
11                 <h5><i class="fas fa-ban"></i> Kesalahan!</h5>
12                 Data yang Anda cari tidak ditemukan.
13             </div>
14         @else
15             <table class="table table-bordered table-striped table-hover table-sm">
16                 <tr>
17                     <th>ID</th>
18                     <td>{{ $user->user_id }}</td>
19                 </tr>
20                 <tr>
21                     <th>Level</th>
22                     <td>{{ $user->level->level_nama }}</td>
23                 </tr>
24                 <tr>
25                     <th>Username</th>
26                     <td>{{ $user->username }}</td>
27                 </tr>
28                 <tr>
29                     <th>Nama</th>
30                     <td>{{ $user->nama }}</td>
31                 </tr>
32                 <tr>
33                     <th>Password</th>
34                     <td>*****</td>
35                 </tr>
36             </table>
37         @endempty
38         <a href="{{ url(path: 'user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
39     </div>
40   </div>
41   @endsection
42   @push('css')
43   @endpush
44   @push('js')
45   @endpush
46   @push('js')
47   @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!!!

Contoh id yang salah.

The screenshot shows a browser window with the URL http://localhost/PWL_2025/Week5.2/SS5.2/public/user/100. The page title is 'Detail User'. A red error box contains the message 'Kesalahan! Data yang Anda cari tidak ditemukan.' (Error! The data you searched for was not found.).

Menampilkan detail data user berdasarkan id yang kita pilih.

The screenshot shows a browser window with the URL http://localhost/PWL_2025/Week5.2/SS5.2/public/user/15. The page title is 'Detail User'. The user details are listed in a table:

ID	15
Level	Manager
Username	Manager Utama
Nama	Ketut
Password	*****



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

```
117 |     // Menampilkan halaman form edit user
118 |     1 reference | overrides
119 |     public function edit(string $id): Factory|View
120 |     {
121 |         $user = UserModel::find($id);
122 |         $level = LevelModel::all();
123 |
124 |         $breadcrumb = (object) [
125 |             'title' => 'Edit User',
126 |             'list' => ['Home', 'User', 'Edit']
127 |         ];
128 |
129 |         $page = (object) [
130 |             'title' => 'Edit user'
131 |         ];
132 |
133 |         $activeMenu = 'user'; // set menu yang sedang aktif
134 |
135 |         return view('user.edit', data: [
136 |             'breadcrumb' => $breadcrumb,
137 |             'page' => $page,
138 |             'user' => $user,
139 |             'level' => $level,
140 |             'activeMenu' => $activeMenu
141 |         ]);
142 |
143 |     } // Menyimpan perubahan data user
144 |     1 reference | overrides
145 |     public function update(Request $request, string $id): mixed\RedirectResponse
146 |     {
147 |         $request->validate([
148 |             // username harus diisi, berupa string, minimal 3 karakter,
149 |             // dan bernilai unik di tabel m_user kolom username kecuali untuk user dengan id yang sedang diedit
150 |             'username' => 'required|string|min:3|unique:m_user,username,' . $id . ',user_id',
151 |             'nama' => 'required|string|max:100', // nama harus diisi, berupa string, dan maksimal 100 karakter
152 |             'password' => 'nullable|min:5', // password bisa diisi (minimal 5 karakter) dan bisa tidak diisi
153 |             'level_id' => 'required|integer' // level_id harus diisi dan berupa angka
154 |         ]);
155 |
156 |         UserModel::find($id)->update([
157 |             'username' => $request->username,
158 |             'nama' => $request->nama,
159 |             'password' => $request->password ? bcrypt(value: $request->password) : UserModel::find($id)->password,
160 |             'level_id' => $request->level_id
161 |         ]);
162 |
163 |         return redirect(to: '/user')->with('success', 'Data user berhasil diubah');
164 |
165 |     }
166 }
```

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!</h5> Data yang Anda cari
                tidak ditemukan.
            </div>
            <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a> @else
            <form method="POST" action="{{ url('/user/'.$user->user_id) }}" class="form- horizontal">
                @csrf
                {!! method_field('PUT') !!}<!-- tambahkan baris ini untuk proses edit yang butuh method PUT -->
                <div class="form-group row">
                    <label class="col-1 control-label col-form-label">Level</label>
                    <div class="col-11">
                        <select class="form-control" id="level_id" name="level_id" required>
                            <option value=""> Pilih Level </option> @foreach($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>
            </form>
        @endempty
    </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') required>
                <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
```

```
Week52 > JS52 > 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 }}
7     <div class="card-tools">
```

```
49   <div class="form-group row">
50     <label class="col-1 control-label col-form-label">Nama</label>
51     <div class="col-11">
52       <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama', $user->nama) }}" required>
53       @error('nama')
54         <small class="form-text text-danger">{{ $message }}</small>
55       @enderror
56     </div>
57   </div>
58 
59   <div class="form-group row">
60     <label class="col-1 control-label col-form-label">Password</label>
61     <div class="col-11">
62       <input type="password" class="form-control" id="password" name="password" @error('password') required>
63           <small class="form-text text-danger">{{ $message }}</small> @else
64             <small class="form-text text-muted">Abaikan (jangan diisi) jika tidak ingin mengganti password user.</small>
65           @enderror
66     </div>
67   </div>
68 
69   <div class="form-group row">
70     <label class="col-1 control-label col-form-label"></label>
71     <div class="col-11">
72       <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
73       <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}>Kembali</a>
74     </div>
75   </div>
76   @endempty
77   @endsection
78 
79   @push('css') @endpush
80   @push('js') @endpush
81 
82   @push('css') @endpush
83   @push('js') @endpush
84 
85   @push('css') @endpush
86   @push('js') @endpush
87 
88   @push('css') @endpush
89   @push('js') @endpush
90 
91   @push('css') @endpush
92   @push('js') @endpush
```



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

The screenshot shows two pages from a Laravel application. The top part is a modal titled 'Edit User' with fields for Level (Manager), Username (Manager Utama), Name (Karina Ika), and Password (empty). Below it is a 'Simpan' (Save) button. The bottom part is a browser window titled 'localhost/PWL_2025/Week5.2/JSS.2/public/User' showing a table of users with columns ID, Username, Name, Level Pengguna, and Aksi (Action). The table contains 10 entries, including 'Manager Utama' which corresponds to the edited user in the modal.

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

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

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

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

```
165 // Menghapus data user
166 1 reference | 0 overrides
167 public function destroy(string $id): mixed|RedirectResponse
168 {
169     $check = UserModel::find($id);
170     if (!$check) { // untuk mengecek apakah data user dengan id yang dimaksud ada atau tidak
171         return redirect(to: '/user')->with('error', 'Data user tidak ditemukan');
172     }
173
174     try {
175         UserModel::destroy($id); // Hapus data level
176
177         return redirect(to: '/user')->with('success', 'Data user berhasil dihapus');
178     } catch (\Illuminate\Database\QueryException $e) {
179
180         // Jika terjadi error ketika menghapus data, redirect kembali ke halaman dengan membawa pesan error
181         return redirect(to: '/user')->with('error', 'Data user gagal dihapus karena masih terdapat tabel lain yang terkait dengan data ini');
182     }
183 }
184 }
```

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



```
4 <div><!-- Content -->
5 <div class="card card-outline card-primary">
6   <div class="card-header">
7     <h3>{{ $page->title }}</h3>
8     <div class="card-tools">
9       <a class="btn btn-sm btn-primary mt-1" href="{{ url(path: 'user/create') }}>Tambah</a>
10    </div>
11  </div>
12  <div class="card-body">
13    @if (session(key: 'success'))
14      <div class="alert alert-success">{{ session(key: 'success') }}</div>
15    @endif
16    @if (session(key: 'error'))
17      <div class="alert alert-danger">{{ session(key: 'error') }}</div>
18    @endif
19    <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
20      <thead>
21        <tr>
22          <th>ID</th>
23          <th>Username</th>
24          <th>Nama</th>
25          <th>Level Pengguna</th>
26          <th>Aksi</th>
27        </tr>
28      </thead>
29      </table>
30    </div>
31  @endsection
```

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

A screenshot of a web browser displaying a modal dialog box. The dialog box has a title "localhost says" and the message "Apakah Anda yakin menghapus data ini?". There are two buttons at the bottom: "OK" and "Cancel". The background shows a table of user data with one row highlighted in red, corresponding to the user "Manager Utama".

A screenshot of a web browser showing the "Daftar User" page. A green success message at the top states "Data user berhasil dihapus". The table below shows the remaining users. The row for "Manager Utama" is missing from the list.

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 web application interface titled "PWL - Starter Code". On the left, there is a sidebar with navigation links: Home, Contact, Dashboard, Data Pengguna, Level User, and Data User (which is currently selected). The main content area is titled "Daftar User" and displays a table of users. The table has columns: ID, Username, Nama, Level Pengguna, and Aksi (Actions). The "Aksi" column contains three buttons: Detail, Edit, and Hapus. A search bar is located at the top right of the table. The table data is as follows:

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

Showing 1 to 6 of 6 entries.

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 // Menampilkan halaman awal user
13 4 references | 0 overrides
14 public function index(): Factory|View
15 {
16     $breadcrumb = (object) [
17         'title' => 'Daftar User',
18         'list' => ['Home', 'User']
19     ];
20
21     $page = (object) [
22         'title' => 'Daftar user yang terdaftar dalam sistem'
23     ];
24
25     $activeMenu = 'user'; // set menu yang sedang aktif
26
27     $level = LevelModel::all(); // ambil data level untuk filter level
28
29     return view('user.index', data: [
30         'breadcrumb' => $breadcrumb,
31         'page' => $page,
32         'level' => $level,
33         'activeMenu' => $activeMenu
34     ]);
35 }
```

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

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

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

```
58         ajax: {
59             "url": "{{ url(path: 'user/list') }}",
60             "dataType": "json",
61             "type": "POST",
62             "data" : function(d) {
63                 d.level_id = $('#level_id').val();
64             }
65         },
66         columns: [
```



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

```
91     data: "aksi",
92     className: "",
93     orderable: false,
94     searchable: false
95   }
96 }
97 $('#level_id').on('change', function() {
98   dataUser.ajax.reload();
99 })
100
101
102 </script>
103 @endpush
```

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

```
37 // Ambil data user dalam bentuk json untuk datatables
38 1 reference | 0 overrides
39 public function list(Request $request): JsonResponse
40 {
41   $users = UserModel::select('user_id', 'username', 'nama', 'level_id')
42     ->with('level');
43
44   //Filter data user berdasarkan level_id
45   if ($request->level_id) {
46     $users->where('level_id', $request->level_id);
47   }
48
49   return DataTables::of($users)
50     ->addColumn('aksi', 'button', 'buttonDelete', 'buttonEdit', 'buttonUpdate');
51 }
```

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		Detail Edit Delete
2	manager	Manager		Detail Edit Delete
3	staff	Staff/Kasir		Detail Edit Delete
4	customer-1	Pelanggan Pertama		Detail Edit Delete

Mencoba:

ID	Username	Nama	Level Pengguna	Aksi
1	manager	Manager		Detail Edit Delete
2	manager_dua	Manager 2		Detail Edit Delete
3	manager22	Manager Dua Dua		Detail Edit Delete
4	manager33	Manager Tiga Tiga		Detail Edit Delete
5	manager96	Manager 96		Detail Edit Delete
6	manager12	Manager 11		Detail Edit Delete

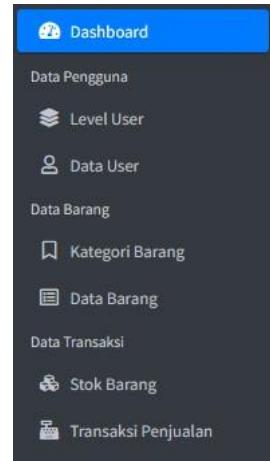
7. Selamat, sekarang Laravel Starter Code sudah ada filtering dan searching data. Starter Code sudah bisa digunakan dalam membangun sebuah sistem berbasis website.
8. Jangan lupa commit dan push ke github PWL_POS kalian



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

- Tambah



The screenshot shows a table titled 'Daftar Level' (List Level) with the following data:

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	CUS	Customer	[Detail] [Edit] [Hapus]
5	CO	CEO	[Detail] [Edit] [Hapus]

Below the table, it says 'Data level berhasil ditambahkan' (Data level has been added successfully). The page includes a sidebar with various menu items like Dashboard, Data Pengguna, Level User, Data User, Data Barang, etc.

- Detail

The screenshot shows a table titled 'Detail level' (Detail Level) with the following data:

ID	Level Kode	Level Nama
1	ADM	Administrator

Below the table, there is a 'Kembali' (Back) button. The page includes a sidebar with various menu items like Dashboard, Data Pengguna, Level User, Data User, Data Barang, etc.

- Edit

The screenshot shows a form titled 'Edit Level' (Edit Level) with the following fields:

Kode	C
Nama	CEO

Below the form, there are 'Simpan' (Save) and 'Kembali' (Back) buttons. The page includes a sidebar with various menu items like Dashboard, Data Pengguna, Level User, Data User, Data Barang, etc.



The screenshot shows a Laravel application interface. The left sidebar has a 'Level User' section selected. The main content area is titled 'Daftar Level' and displays a table of user levels. A green success message at the top says 'Data level berhasil dihapus'. The table includes columns for ID, Level Kode, Level Nama, and aksii (with buttons for Detail, Edit, and Hapus). The row for 'C' (CEO) is highlighted with a red border. At the bottom, there's a copyright notice and a 'Version 3.2.0' link.

- Hapus

This screenshot shows the same application interface after a user has selected the 'Hapus' (Delete) button for the 'C' (CEO) level. A modal dialog box appears in the center asking 'Apakah Anda yakin menghapus data ini?' (Are you sure you want to delete this data?). There are 'OK' and 'Cancel' buttons. The rest of the page remains the same, showing the list of levels and the successful deletion message.

In this final screenshot, the 'OK' button has been clicked in the confirmation dialog. The 'C' (CEO) row is now completely removed from the 'Daftar Level' table. The green success message 'Data level berhasil dihapus' is still visible at the top. The table structure remains the same with columns for ID, Level Kode, Level Nama, and aksii.



2. Tabel m_kategori

- Tampilan

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	CML	Cemilan	[Detail] [Edit] [Hapus]
7	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]
8	SPT	Sepatu	[Detail] [Edit] [Hapus]

- Tambah

Tambah Kategori

Tambah kategori baru

Kode: ACS

Nama: Accessories

[Simpan] [Kembali]

Data kategori berhasil ditambahkan

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	CML	Cemilan	[Detail] [Edit] [Hapus]
7	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]
8	SPT	Sepatu	[Detail] [Edit] [Hapus]
9	ACS	Accessories	[Detail] [Edit] [Hapus]



- Detail

The screenshot shows a web application interface for managing categories. On the left is a sidebar with various menu items: Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang (which is highlighted in blue), Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area has a title 'Detail Kategori'. Below it is a table with one row containing the following data:

ID	12
Kode Kategori	ACS
Nama Kategori	Accessories

At the bottom of the main content area, there is a 'Kembali' button. The footer of the page includes the text 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

- Edit

This screenshot shows the 'Edit Kategori' page. The sidebar and overall layout are identical to the 'Detail' page. The main content area has a title 'Edit Kategori'. It displays a form with two fields: 'Kode' (containing 'AC') and 'Nama' (containing 'Accessories'). Below the form are two buttons: 'Simpan' (in blue) and 'Kembali'. The footer includes the same copyright and version information as the previous page.

This screenshot shows the 'Daftar Kategori' page. The sidebar and layout remain consistent. The main content area has a title 'Daftar Kategori' and a sub-header 'Daftar kategori yang terdaftar dalam sistem'. Below this is a table titled 'Data kategori berhasil diubah' (Data category successfully changed). The table has columns: ID, Kode Kategori, Nama Kategori, and Aksi (Actions). The data in the table is as follows:

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	CML	Cemilan	[Detail] [Edit] [Hapus]
7	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]
8	SPT	Sepatu	[Detail] [Edit] [Hapus]
9	AC	Accessories	[Detail] [Edit] [Hapus]

At the bottom of the table, it says 'Showing 1 to 9 of 9 entries'. The footer includes the same copyright and version information.



- Hapus

The screenshot shows two consecutive screenshots of a Laravel application interface. Both screenshots are titled 'Daftar Kategori' and show a table of categories. A modal dialog box is open in both cases, asking 'Apakah Anda yakin menghapus data ini?' (Are you sure you want to delete this data?). The first screenshot shows a green success message at the top: 'Data kategori berhasil dihapus' (Category data has been successfully deleted). The second screenshot shows a similar green message: 'Data kategori berhasil dihapus' (Category data has been successfully deleted). The table lists nine categories with columns for ID, Kode Kategori, Name, and Actions (Detail, Edit, Hapus).

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	CML	Cemilan	[Detail] [Edit] [Hapus]
7	MNR	Minuman Ringan	[Detail] [Edit] [Hapus]
8	SPT	Sepatu	[Detail] [Edit] [Hapus]
9	AC	Accessories	[Detail] [Edit] [Hapus]

3. Tabel m_barang

- Tampilan

The screenshot shows a Laravel application interface titled 'Data Barang'. It displays a table of items with columns for ID, Kode Barang, Name, Category, Purchase Price, Sale Price, and Actions (Detail, Edit, Hapus). A filter dropdown is visible, and a search bar is at the top right. The table contains ten entries.

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]
8	B008	Susu Formula	Baby & Kids	60000	65000	[Detail] [Edit] [Hapus]
9	B009	Lampu LED	Electronics	25000	30000	[Detail] [Edit] [Hapus]
10	B010	Charger HP	Electronics	35000	40000	[Detail] [Edit] [Hapus]



- Tambah

Tambah barang baru

Kode: B011

Nama: Moizturizer

Kode Kategori: Beauty & Health

Harga Beli: 35000

Harga Jual: 40000

Simpan **Kembali**

Data barang yang terdapat dalam sistem

Tambah

Data barang berhasil dihapus

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]
8	B008	Susu Formula	Baby & Kids	60000	65000	[Detail] [Edit] [Hapus]
9	B009	Lampu LED	Electronics	25000	30000	[Detail] [Edit] [Hapus]
10	B010	Charger HP	Electronics	15000	20000	[Detail] [Edit] [Hapus]

- Detail

Detail barang

ID: 11

Kode barang: B011

Nama barang: Moizturizer

Kode Kategori: Beauty & Health

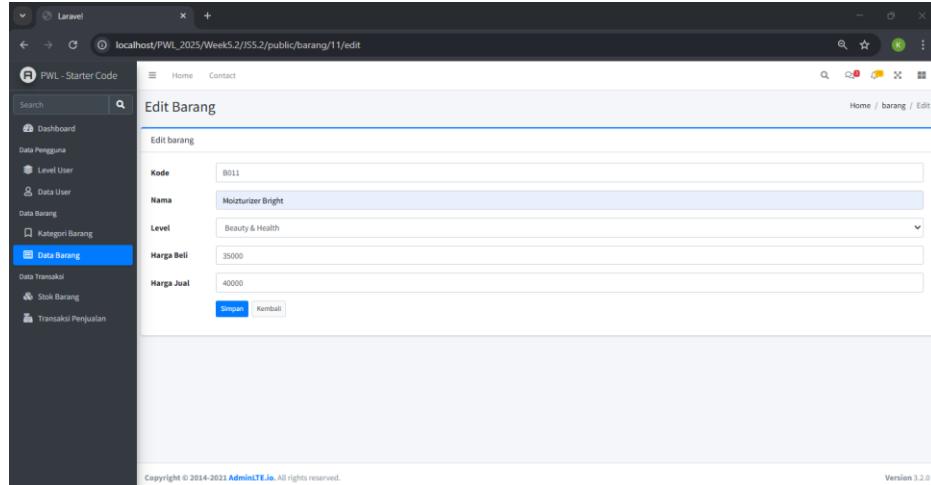
Harga Beli: 35000

Harga Jual: 40000

Kembali

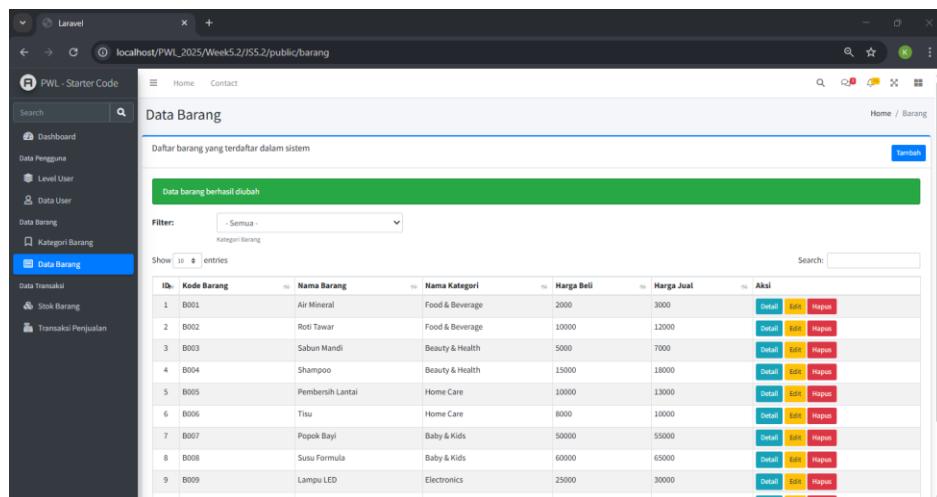


- Edit



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

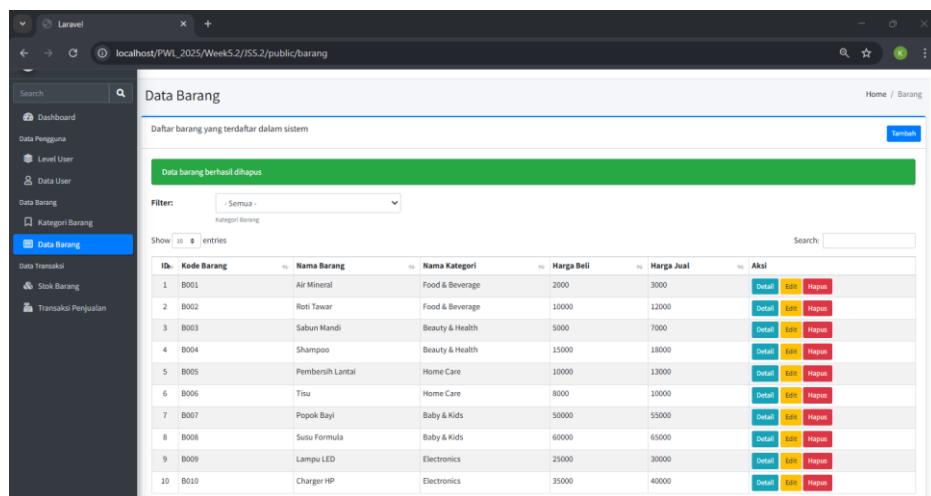
Versi 3.2.0



Data barang berhasil dihapus

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]
8	B008	Susu Formula	Baby & Kids	60000	65000	[Detail] [Edit] [Hapus]
9	B009	Lampu LED	Electronics	25000	30000	[Detail] [Edit] [Hapus]
10	B010	Charger HP	Electronics	35000	40000	[Detail] [Edit] [Hapus]

- Hapus



Data barang berhasil dihapus

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]
8	B008	Susu Formula	Baby & Kids	60000	65000	[Detail] [Edit] [Hapus]
9	B009	Lampu LED	Electronics	25000	30000	[Detail] [Edit] [Hapus]
10	B010	Charger HP	Electronics	35000	40000	[Detail] [Edit] [Hapus]



4. Tabel m_supplier

- Tampilan

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

- Tambah

Tambah supplier baru

Kode: SP004

Nama: PT Sumber Rejeki

Alamat: Jl. Merdeka No. 10, Jakarta

[Simpan] [Kembali]

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 Sejatera	Jl. Pahlawan No. 25, Surabaya	[Detail] [Edit] [Hapus]
3	SP003	PT Sentosa Makmur Bersama	Jl. Gatot Subroto No. 15, Bandung	[Detail] [Edit] [Hapus]
4	SP004	PT Sumber Rejeki	Jl. Merdeka No. 10, Jakarta	[Detail] [Edit] [Hapus]

- Detail

Detail Supplier

ID: 4

Kode supplier: SP004

Nama supplier: PT Sumber Rejeki

Alamat supplier: Jl. Merdeka No. 10, Jakarta

[Kembali]



- Edit

The screenshot shows two browser windows of a Laravel application. The left window displays the 'Edit Supplier' form with fields for Kode (SP004), Nama (PT Sumber Rejeki Makmur), and Alamat (Jl. Merdeka No. 10, Jakarta). The right window shows the 'Daftar Supplier' list with four entries:

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]
4	SP004	PT Sumber Rejeki Makmur	Jl. Merdeka No. 10, Jakarta	[Detail] [Edit] [Hapus]

- Hapus

The screenshot shows two browser windows. The top window displays a confirmation dialog: "localhost says Apakah Anda yakin menghapus data ini?". The bottom window shows the 'Daftar Supplier' list with three entries remaining:

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]

*** Sekian, dan selamat belajar ***