



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

JOBSHEET 05

Blade View, Web Templating(AdminLTE), Datatables

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

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

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

A. Blade View

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

1. Apa Itu Blade?

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

2. Keuntungan Menggunakan Blade

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



- **Tidak Ada Overhead:** Semua kode Blade dikompilasi menjadi PHP biasa, jadi tidak ada overhead tambahan.

3. Sintaks Dasar Blade

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

- **Menampilkan Variabel:**

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

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

- **Struktur Kontrol:**

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

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

- **Komentar**

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



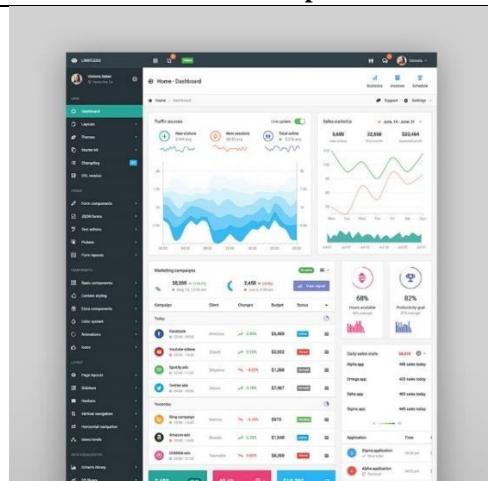
B. Web Template

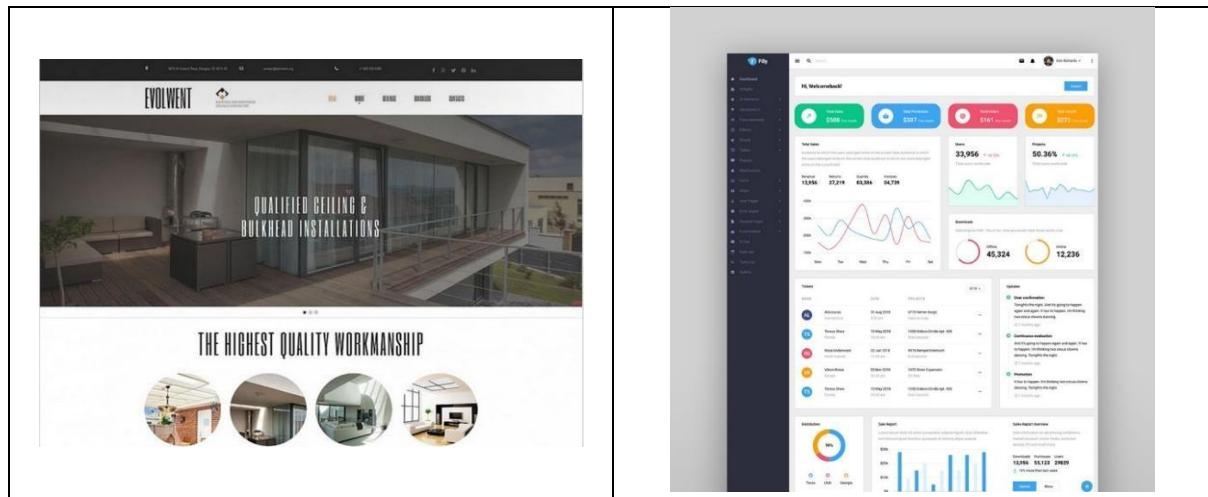
Merupakan suatu file yang sudah memiliki desain tertentu sehingga, kita tinggal memodifikasi bagian-bagian tertentu saja untuk mendapatkan hasil yang sudah bagus. Dengan menggunakan template yang sesuai dengan kebutuhan, kita bisa mengurangi waktu pembuatan suatu website secara drastis.

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

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

Contoh Web template

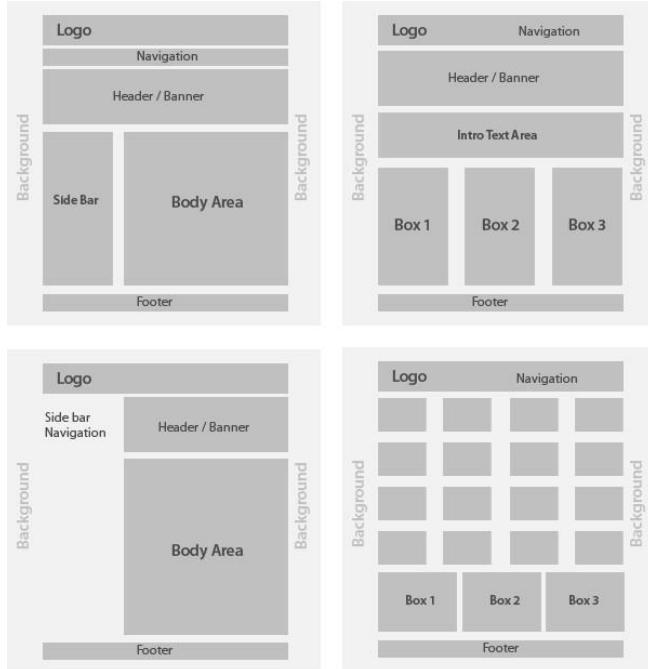
Frontend Template	Backend Template
	



A. Web Layout

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

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



C. Layouting AdminLTE

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

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

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



Praktikum 1 - Layouting AdminLTE:

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

The screenshot shows the GitHub release page for AdminLTE v3.2.0. It includes the release date (Feb 8, 2022), the author (REJack), and the commit hash (v3.2.0). The 'Assets' section is highlighted with a red checkmark over the 'Source code (zip)' link.

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

The screenshot shows a file explorer interface with the path **PWL_POS/public/adminlte** selected. A red checkmark is placed over the 'adminlte' folder.

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

The screenshot shows the AdminLTE 3 dashboard with various widgets and charts. The left sidebar shows navigation options like Dashboard, Widgets, Charts, UI Elements, Forms, and Tables.



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

The screenshot shows the AdminLTE 3 sidebar on the left with various template options. The 'Blank Page' option is highlighted with a red circle labeled '2'. A red arrow labeled '1' points to the 'Extras' dropdown menu.

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

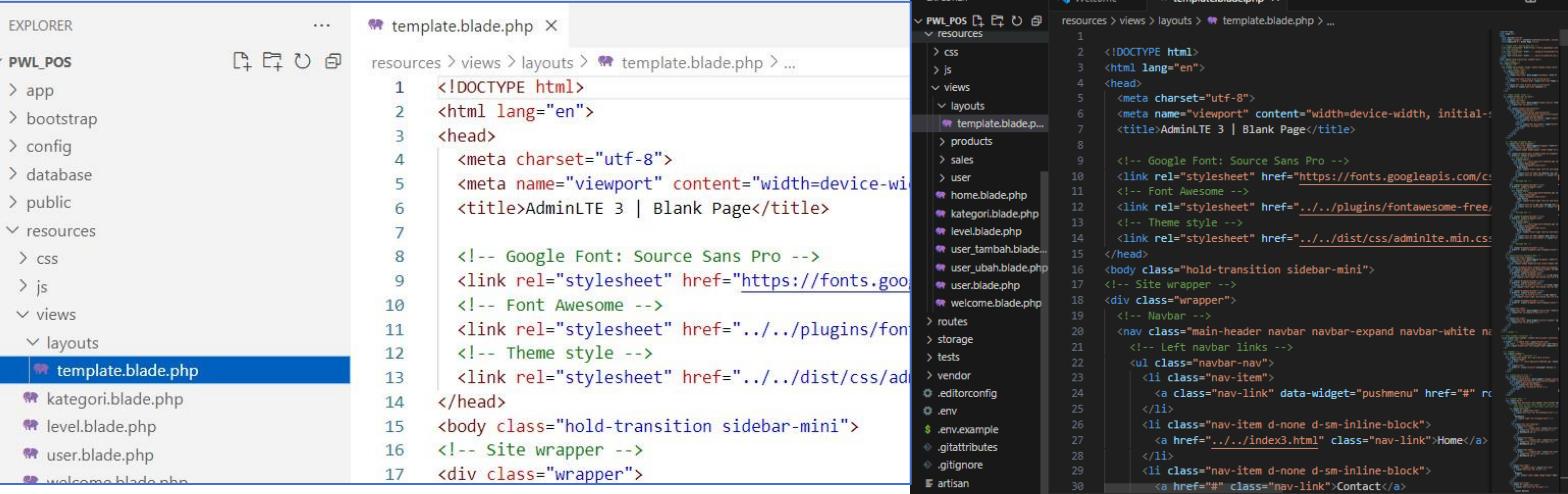
The screenshot shows the 'Blank Page' template with five numbered regions: 1 - Header / Navbar (red border), 2 - Sidebar / Navigasi (red border), 3 - Breadcrumb (red border), 4 - Content (red border), and 5 - Footer (red border). The sidebar on the left is also outlined in red and labeled '2 - Sidebar / Navigasi'.

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

The screenshot shows a right-click context menu on the 'Blank Page' content area. The 'View Page Source' option is highlighted with a red circle.



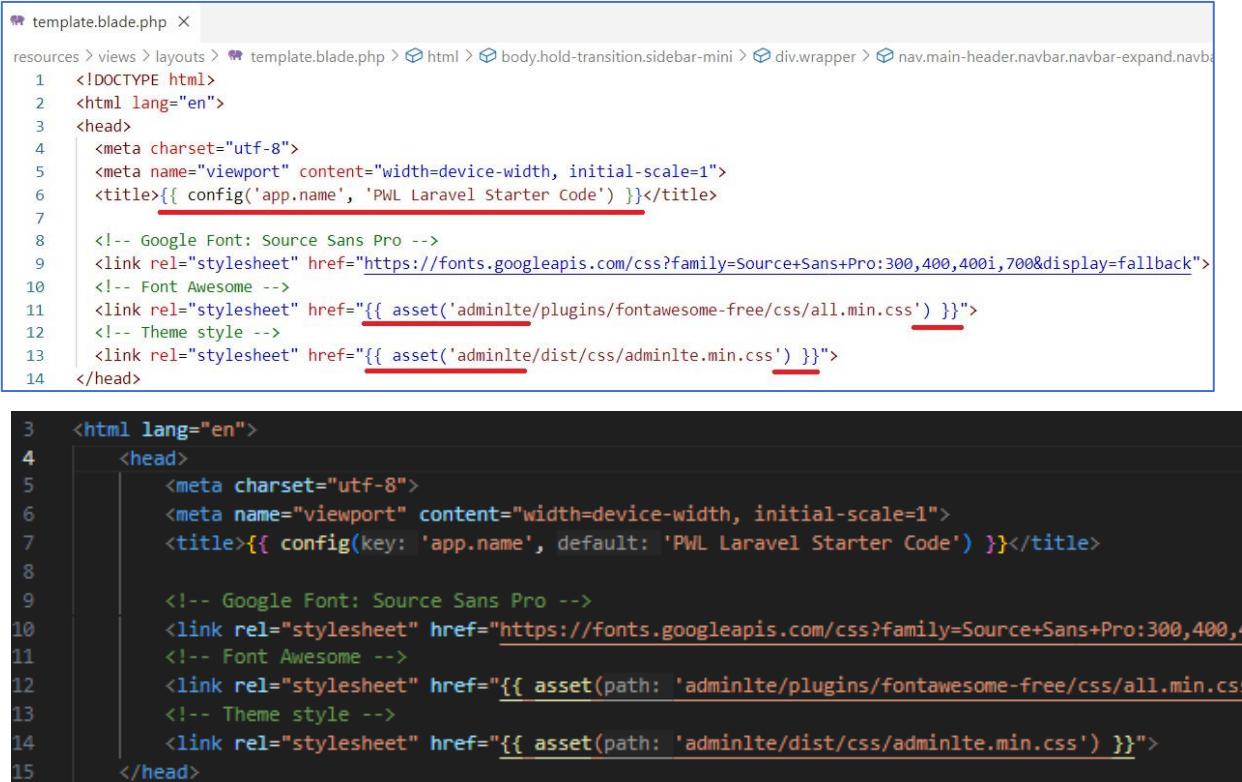
7. Selanjutnya kita copy page source dari halaman [Blank Page](#), kemudia kita *paste* pada [PWL_POS/resource/view/layouts/template.blade.php](#) (buat dulu folder `layouts` dan file `template.blade.php`)



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>AdminLTE 3 | Blank Page</title>
    <!-- Google Font: Source Sans Pro -->
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="../../plugins/fontawesome-free/css/all.min.css">
    <!-- Theme style -->
    <link rel="stylesheet" href="../../dist/css/adminlte.min.css">
</head>
<body class="hold-transition sidebar-mini">
    <!-- Site wrapper -->
    <div class="wrapper">
        <!-- Navbar -->
        <nav class="main-header navbar navbar-expand navbar-white navbar-light">
            <!-- Left navbar links -->
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" data-widget="pushmenu" href="#"><i class="fa fa-bars"></i> Menu</a>
                </li>
                <li class="nav-item d-none d-sm-inline-block">
                    <a href="#">Home</a>
                </li>
                <li class="nav-item d-none d-sm-inline-block">
                    <a href="#">Contact</a>
                </li>
            </ul>
        </nav>
        <!-- Main content -->
        <div class="content">
            <div class="container-fluid">
                <div class="row">
                    <div class="col-12">
                        <h1>Blank Page</h1>
                        <p>This is a simple hero unit, a simple jumbotron-style component for showcasing something interesting or informative.</p>
                        <p>It uses utility classes for typography and spacing by default, check out the <a href="#">Live Demo </a> for examples.</p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
```

8. File `layouts/template.blade.php` adalah file utama untuk templating website

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



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title>
    <!-- Google Font: Source Sans Pro -->
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
    <!-- Theme style -->
    <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
</head>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>{{ config('key: app.name', default: 'PWL Laravel Starter Code') }}</title>
    <!-- Google Font: Source Sans Pro -->
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="{{ asset(path: 'adminlte/plugins/fontawesome-free/css/all.min.css') }}">
    <!-- Theme style -->
    <link rel="stylesheet" href="{{ asset(path: 'adminlte/dist/css/adminlte.min.css') }}">
</head>
```



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

The screenshot shows two code editor panes. The left pane contains the `template.blade.php` file with the following code:

```
template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > aside.main
14  </head>
15  <body class="hold-transition sidebar-mini">
16  <!-- Site wrapper -->
17  <div class="wrapper">
18  <!-- Navbar -->
19  @include('layouts.header')
20  <!-- /.navbar -->
21
22 <!-- Main Sidebar Container -->
23 <aside class="main-sidebar sidebar-dark-primary elevation-4">
24  <!-- Brand Logo -->
25  <a href="../../index3.html" class="brand-link">
26    AdminLTE 3</span>
28 </a>
```

The right pane contains the `header.blade.php` file with the following code:

```
> css
> js
views
  layouts
    header.blade.php
    template.blade.php... > products
14  </head>
15  <body class="hold-transition sidebar-mini">
16  <!-- Site wrapper -->
17  <div class="wrapper">
18  <!-- Navbar -->
19  @include(view: 'layouts.header')
20  <!-- /.navbar -->
21
```

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` menjadi

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

```
24  <!-- Brand Logo -->
25  <a href="{{ url(path: '/') }}" class="brand-link">
26    PWL - Starter Code</span>
```

12. Selanjutnya kita blok baris **31-693** (baris untuk **element 2-sidebar**), lalu kita **cut**, dan **paste**-kan di file `PWL_POS/resource/view/layouts/sidebar.blade.php` (buat dulu file `sidebar.blade.php` jika belum ada). Sehingga tampilan dari file `template.blade.php` menjadi seperti berikut

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



```
38  <!-- Sidebar -->
39  @include(view: 'layouts.sidebar')
40  <!-- /.sidebar -->
```



13. Selanjutnya perhatikan baris **87-98** (baris untuk *element 5-footer*), lalu kita ***cut***, dan ***paste*-kan** di file **PWL_POS/resource/view/layouts/footer.blade.php** (buat file **footer.blade.php** jika belum ada). Sehingga tampilan dari file **template.blade.php** menjadi seperti berikut

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

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

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

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

Menjadi

```
91 <!-- jQuery -->
92 <script src="{{ asset(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

```
30 <!-- Sidebar -->
31 @include('layouts.sidebar')
32 <!-- /.sidebar -->
33 </aside>
34
35 <!-- Content Wrapper. Contains page content -->
36 <div class="content-wrapper">
37   <!-- Content Header (Page header) -->
38   @include('layouts.breadcrumb')
39
40   <!-- Main content -->
41   <section class="content">
42     <!-- Default box -->
43     <div class="card">
44       <div class="card-header">
45         <h3 class="card-title">Title</h3>
46
47       <!-- Content Header (Page header) -->
48       @include(view: 'layouts.breadcrumb')
49
50       <!-- Main content -->
```



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

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

```
17 <div class="wrapper">
18   <!-- Navbar -->
19   @include(view: 'layouts.header')
20   <!-- ./navar -->
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(section: 'content')
43     </section>
44     <!-- ./content -->
45   </div>
46   <!-- ./content-wrapper -->
47
48 @include(view: 'layouts.footer')
49 </div>
50 <!-- ./wrapper -->
```

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**

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

```
app > Http > Controllers > WelcomeController.php > PHP > WelcomeController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 0 references | 0 implementations | Codeium: Refactor | Explain | Generate Function Comment | X
6 class WelcomeController extends Controller
7
8 0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
9
10 public function index() Factory|View
11 {
12   $breadcrumb = (object) [
13     'title' => 'Selamat Datang',
14     'list' => ['Home', 'Welcome']
15   ];
16
17   $activeMenu = 'dashboard';
18
19   return view('welcome', data: ['breadcrumb' => $breadcrumb, 'acti
20
21 }
```

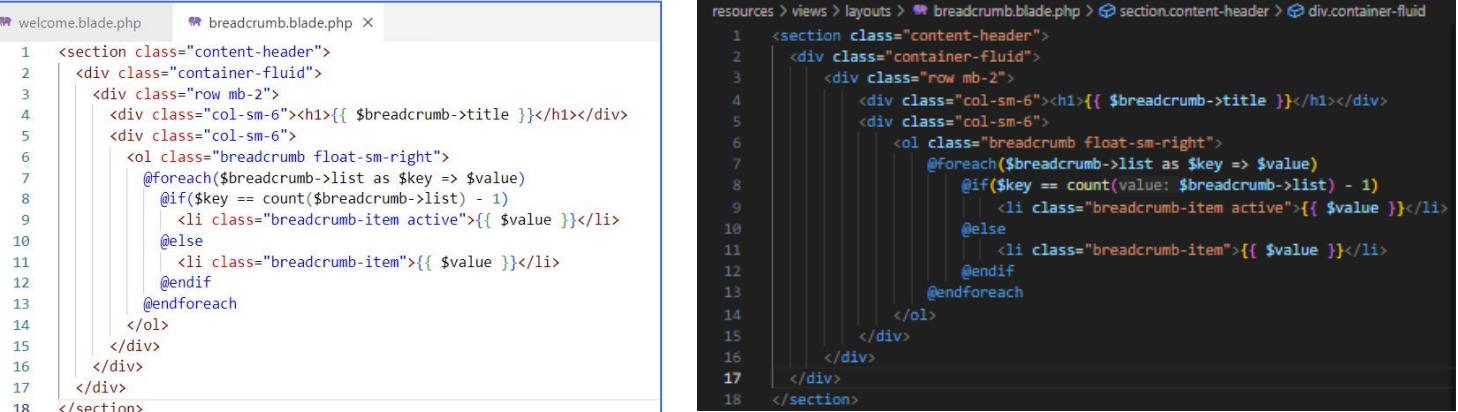
2. Kita buat file pada **PWL_POS/resources/views/welcome.blade.php**

```
welcome.blade.php x
1 @extends('layouts.template')
2
3 @section('content')
4
5 <div class="card">
6   <div class="card-header">
7     <h3 class="card-title">Halo, apakah 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 @endsection
```

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



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



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

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



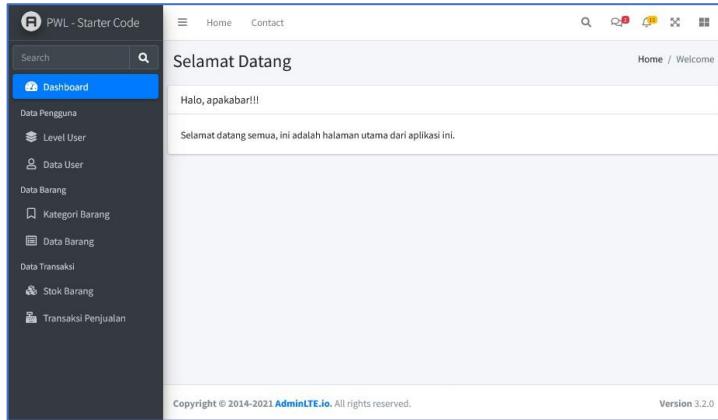
```
<a href="{{ url('/kategori') }}" class="nav-link {{ ($activeMenu == 'kategori')? 'active' : '' }} ">
    <i class="nav-icon far fa-bookmark"></i>
    <p>Kategori Barang</p>
</a>
</li>
<li class="nav-item">
    <a href="{{ url('/barang') }}" class="nav-link {{ ($activeMenu == 'barang')? 'active' : '' }} ">
        <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>
```

5. Kita tambahkan kode berikut router web.php

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

6. Sekarang kita coba jalankan di browser dengan mengetikkan url

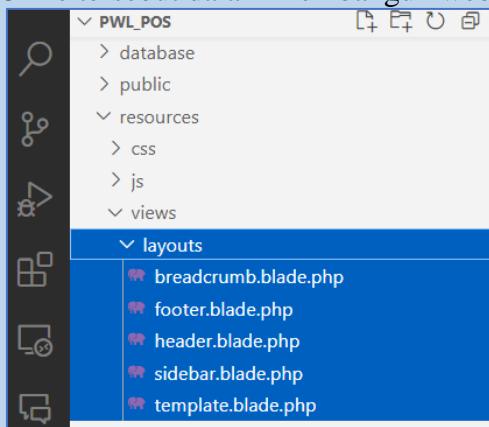
http://localhost/PWL_POS/public



7. Jangan lupa commit dan push ke github PWL_POS kalian

INFO

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





D. jQuery Datatables di Laravel

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

INFO

AdminLTE juga sudah menerapkan library Datatable dalam template nya. Datatable pada AdminLTE dapat kalian cek di
<http://localhost/PWL/public/adminlte/pages/tables/data.html>

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

E. Implementasi jQuery Datatables di Laravel

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

Praktikum 3 – Implementasi jQuery Datatable di AdminLTE :

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

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

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



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

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>
                <tbody>
                </tbody>
            </table>
        </div>
    @endsection

@push('css')
@endpush

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

```



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

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

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



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

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);
}
```

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



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

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

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

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

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

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

@section('content')


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



<form method="POST" action="{{ url('user') }}" class="form-horizontal">
    @csrf
    <div class="form-group row">
        <label class="col-1 control-label col-form-label">Level</label>
        <div class="col-11">
            <select class="form-control" id="level_id" name="level_id" required>
                <option value="">- Pilih Level -</option>
                @foreach($level as $item)
                    <option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
                @endforeach
            </select>
            @error('level_id')
                <small class="form-text text-danger">{{ $message }}</small>
            @enderror
        </div>
    </div>
    <div class="form-group row">
        <label class="col-1 control-label col-form-label">Username</label>
        <div class="col-11">
            <input type="text" class="form-control" id="username" name="username" value="{{ old('username') }}" required>
            @error('username')
                <small class="form-text text-danger">{{ $message }}</small>
            @enderror
        </div>
    </div>
    <div class="form-group row">
        <label class="col-1 control-label col-form-label">Nama</label>
        <div class="col-11">
            <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama') }}" required>
            @error('nama')
                <small class="form-text text-danger">{{ $message }}</small>
            @enderror
        </div>
    </div>
    <div class="form-group row">
        <label class="col-1 control-label col-form-label">Password</label>
    </div>


```



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

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

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

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

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

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

The screenshot shows a table titled "Daftar User" with 10 entries. The columns are ID, Username, Name, Level Pengguna, and Aksi (Actions). The "Aksi" column contains three buttons: Detail, Edit, and Hapus. The last entry in the table is "manager09". Above the table, there is a green success message: "Data user berhasil disimpan". At the top right, there is a "Tambah" button.

ID	Username	Nama	Level Pengguna	Aksi
1	admin	AdminC	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staff/Kasir	Staff	Detail Edit Hapus
4	customer-1	Pelanggan Pertama	Pelanggan	Detail Edit Hapus
5	manager_dua	Manager 2	Manager	Detail Edit Hapus
6	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
7	manager33	Manager Tiga Tiga	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	manager12	Manager11	Manager	Detail Edit Hapus
10	manager09	Manager Sembilan	Manager	Detail Edit Hapus



13. Selanjutnya, kita masuk pada bagian menampilkan detail data user (klik tombol **Detail**)

pada halaman user. Route yang bertugas untuk menangkap request detail adalah

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

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

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

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

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

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

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

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

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

@section('content')


<div class="card-header">
        <h3 class="card-title">{{ $page->title }}</h3>
        <div class="card-tools"></div>
    </div>
    <div class="card-body" > @empty($user)
        <div class="alert alert-danger alert-dismissible">
            <h5><i class="icon fas fa-ban"></i> Kesalahan! </h5> Data yang
            Anda cari tidak ditemukan.
        </div>
    @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>
@endsection

@push('css') @endpush
@push('js') @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 laporan!!!

Jawab : Melihat detail User ID 1 :

Detail User

Home / User / Detail

Detail user	
ID	1
Level	Administrator
Username	admin
Nama	AdminC
Password	*****

Kembali

Melihat ID salah misal 100 :

Detail User

Home / User / Detail

Detail user	
<p> Kesalahan! Data yang Anda cari tidak ditemukan.</p>	

Kembali

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

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

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

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

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

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

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

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

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

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

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



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 class="form-group row">
                                <label class="col-1 control-label col-form-label">Username</label>
                                <div class="col-11">
                                    <input type="text" class="form-control" id="username" name="username" value="{{
old('username', $user->username ) }}" required>
                                        @error('username')
                                            <small class="form-text text-danger">{{ $message }}</small> @enderror
                                        </div>
                                    </div>
                                    <div class="form-group row">
                                        <label class="col-1 control-label col-form-label">Nama</label>
                                        <div class="col-11">
                                            <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama', $user->nama )
}}> required>
                                                @error('nama')
                                                    <small class="form-text text-danger">{{ $message }}</small> @enderror
                                                </div>
                                                <div class="form-group row">
                                                    <label class="col-1 control-label col-form-label">Password</label>
                                                    <div class="col-11">
                                                        <input type="password" class="form-control" id="password" name="password" @error('password')
                                                            <small class="form-text text-danger">{{ $message }}</small> @else
                                                            <small class="form-text text-muted">Abaikan (jangan diisi) jika tidak ingin mengganti password
                                                        user.</small>
                                                        @enderror
                                                        </div>
                                                    </div>
                                                    <div class="form-group row">
                                                        <label class="col-1 control-label col-form-label"></label>
                                                        <div class="col-11">
                                                            <button type="submit" class="btn btn-primary btn-sm">Simpan</button>
                                                            <a class="btn btn-sm btn-default ml-1" href="{{ url('user') }}>Kembali</a>
                                                        </div>
                                                    </div>
                                                </form>
                                                @endempty
                                            </div>
                                        </div> @endsection
                                    @push('css') @endpush
                                    @push('js') @endpush
                                </div>
                            </div>
                        </div>
                    </div>
                </form>
            </div>
        </div> @endsection
    </div>
@push('css') @endpush
@push('js') @endpush
```



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

The screenshot shows a user management interface. At the top is a table with columns: ID, Username, Nama, Level Pengguna, and Aksi. One row shows ID 1, Username admin, Nama AdminC, Level Pengguna Administrator, and buttons for Detail, Edit, and Hapus. Below this is a modal window with a green header 'Data user berhasil diubah' (User data successfully changed). It contains a text input field with 'Admin' and buttons for Show (10 entries), Search, and another table identical to the one above.

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

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

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

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

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

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

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

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

Menjadi



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

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

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

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

F. Data Searching dan Filtering

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

1. *Searching* (Pencarian):

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



2. Filtering (Penyaringan):

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

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

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

The screenshot shows a Laravel application interface. On the left is a sidebar menu with items like Data Pengguna, Level User, and Data User (which is currently selected). The main content area has a header "Daftar User" and a sub-header "Daftar user yang terdaftar dalam sistem". Below this is a table with columns: ID, Username, Name, Level Pengguna, and Aksi. The "Aksi" column contains three buttons: Detail, Edit, and Hapus. At the top of the table is a "Searching" input field and a "Search:" button, both of which are highlighted with a red box. The table shows 6 entries. At the bottom, there are navigation links for "Previous" and "Next".

Praktikum 4 – Implementasi Filtering Datatables:

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

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

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

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

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

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

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



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

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

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

```
46 @push('js')
47 <script>
48 $(document).ready(function() {
49     var dataUser = $('#table_user').DataTable({
50         serverSide: true, // serverSide: true, jika ingin menggunakan server side processing
51         ajax: {
52             "url": "{{ url('user/list') }}",
53             "dataType": "json",
54             "type": "POST",
55             "data": function ( d ) {
56                 d.level_id = $('#level_id').val();
57             }
58         },
59         columns: [
60             {
61                 data: "DT_RowIndex", // nomor urut dari laravel addIndexColumn()
```

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

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



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

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

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

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

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

The screenshot shows a Laravel application interface titled "PWL - Starter Code". On the left is a sidebar with navigation links: Dashboard, Level User, Data User (which is highlighted in blue), Data Barang, Data Transaksi, Stock Barang, and Transaksi Penjualan. The main content area has a header "Daftar User" and a sub-header "Daftar user yang terdaftar dalam sistem". It includes a "Filtering" section with a dropdown set to "Semua" and a search bar. Below is a table with columns: ID, Username, Nama, Level Pengguna, and Aksi. The table contains 6 rows of data. At the bottom, there are pagination controls for "Previous" and "Next".

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

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

This screenshot shows the same application interface as above, but with a filter applied. The "Filter" dropdown is now set to "Manager". The table data has been updated to show only users with the "Manager" level. The table structure is identical to the previous screenshot, with 6 Manager-level users listed.

ID	Username	Nama	Level Pengguna	Aksi
1	manager	Manager	Manager	[Detail] [Edit] [Hapus]
2	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
3	manager22	Manager Dua Dua	Manager	[Detail] [Edit] [Hapus]
4	manager33	Manager Tiga Tiga	Manager	[Detail] [Edit] [Hapus]
5	manager56	Manager55	Manager	[Detail] [Edit] [Hapus]
6	manager12	Manager11	Manager	[Detail] [Edit] [Hapus]

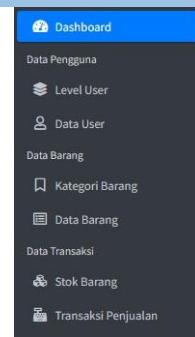
8. Jangan lupa commit dan push ke github PWL_POS kalian



G. Tugas

Implementasikan web layout dan datatables, pada menu berikut ini

1. Tabel m_level
2. Tabel m_kategori
3. Tabel m_supplier
4. Tabel m_barang



Jawab :

1. Tabel m_level

- Tampilan ketika di halaman Dashboard Level User

ID	Level Kode	Level Nama	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff	[Detail] [Edit] [Hapus]
4	CUS	Polangan	[Detail] [Edit] [Hapus]

- Tampilan ketika di halaman Detail Level User

- Tampilan ketika di halaman Tambah Level User



Tambah level baru

Level Kode	GST
Nama	Guest
Simpan Kembali	

Data level berhasil ditambahkan

ID	Level Kode	Level Name	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff	[Detail] [Edit] [Hapus]
4	CUS	Pelanggan	[Detail] [Edit] [Hapus]
5	GST	Guest	[Detail] [Edit] [Hapus]

- Tampilan ketika di halaman Mengedit Level Kode Tertentu

5	GST	Guest	Detail Edit Hapus
---	-----	-------	--

Laravel localhost/PWL2025/PWL_Week5.2/Js5/PWL_POS/public/level/7/edit

Edit Level

Kode	GST
Nama	Tamu
Simpan Kembali	

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

Laravel localhost/PWL2025/PWL_Week5.2/Js5/PWL_POS/public/level

Daftar Level

ID	Level Kode	Level Name	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff	[Detail] [Edit] [Hapus]
4	CUS	Pelanggan	[Detail] [Edit] [Hapus]
5	GST	Tamu	[Detail] [Edit] [Hapus]



- Tampilan ketika Menghapus data Level berdasarkan ID

The screenshot shows a Laravel application interface. The top navigation bar includes '5', 'GST', 'Tamu', 'Detail', 'Edit', and 'Hapus'. Below the header, a browser window displays the URL 'localhost/PWL2025/PWL_Week5.2/Js5/PWL_POS/public/level'. The left sidebar menu has 'Level User' selected. The main content area is titled 'Daftar Level' and shows a table of level data. A green success message at the top states 'Data level berhasil dihapus'. The table data is as follows:

ID	Level Kode	Level Nama	Aksi
1	ADM	Administrator	[Detail] [Edit] [Hapus]
2	MNG	Manager	[Detail] [Edit] [Hapus]
3	STF	Staff	[Detail] [Edit] [Hapus]
4	CUS	Pelanggan	[Detail] [Edit] [Hapus]

At the bottom, it says 'Showing 1 to 4 of 4 entries' and has 'Previous' and 'Next' buttons.

2. Tabel m_kategori

- Tampilan ketika di halaman Dashboard Kategori Barang

The screenshot shows a Laravel application interface. The top navigation bar includes '5', 'GST', 'Tamu', 'Detail', 'Edit', and 'Hapus'. Below the header, a browser window displays the URL 'localhost/PWL2025/PWL_Week5.2/Js5/PWL_POS/public/kategori'. The left sidebar menu has 'Kategori Barang' selected. The main content area is titled 'Daftar Kategori' and shows a table of category data. The table data is as follows:

ID	Kode Kategori	Nama Kategori	Aksi
1	KTG001	Makanan	[Detail] [Edit] [Hapus]
2	KTG002	Minuman	[Detail] [Edit] [Hapus]
3	KTG003	Elektronik	[Detail] [Edit] [Hapus]
4	KTG004	Peralatan Rumah	[Detail] [Edit] [Hapus]
5	KTG005	Pakaian	[Detail] [Edit] [Hapus]
6	KTG006	Barang	[Detail] [Edit] [Hapus]

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

- Tampilan ketika di halaman Detail Kategori Barang

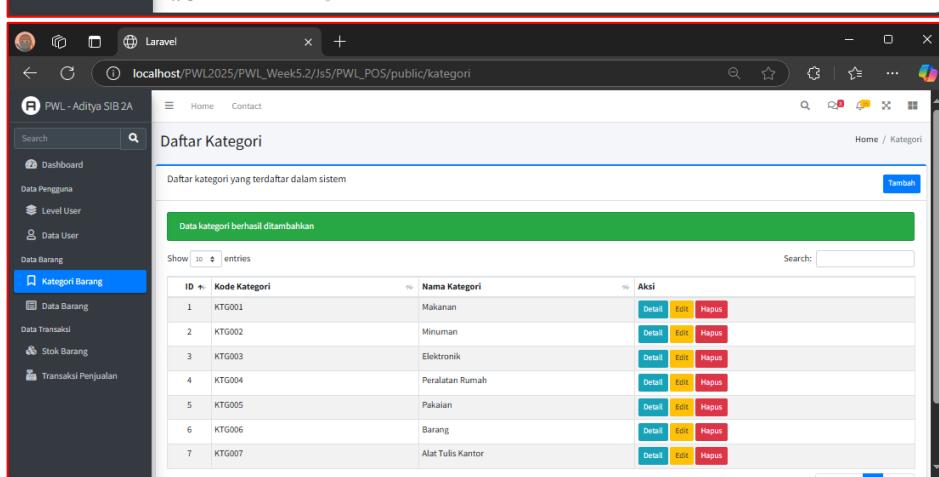
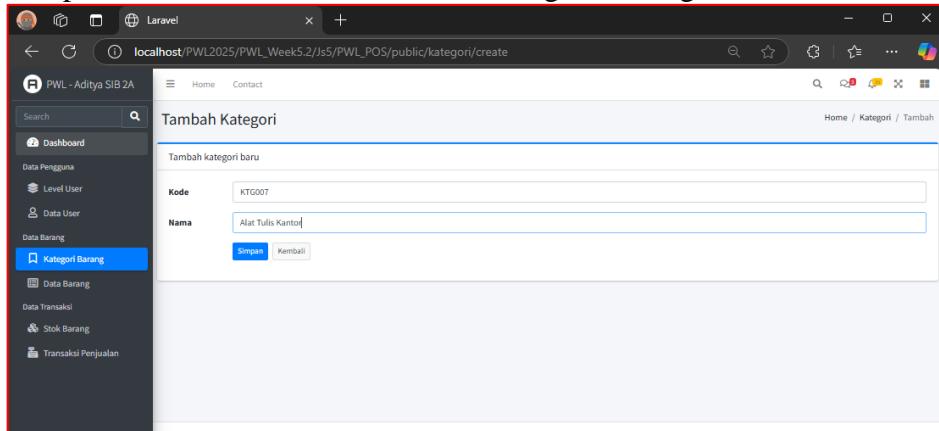
The screenshot shows a Laravel application interface. The top navigation bar includes '5', 'GST', 'Tamu', 'Detail', 'Edit', and 'Hapus'. Below the header, a browser window displays the URL 'localhost/PWL2025/PWL_Week5.2/Js5/PWL_POS/public/kategori/9'. The left sidebar menu has 'Kategori Barang' selected. The main content area is titled 'Detail Kategori' and shows a table with one row of data. The table data is as follows:

ID	Kode Kategori	Nama Kategori
9	KTG006	Barang

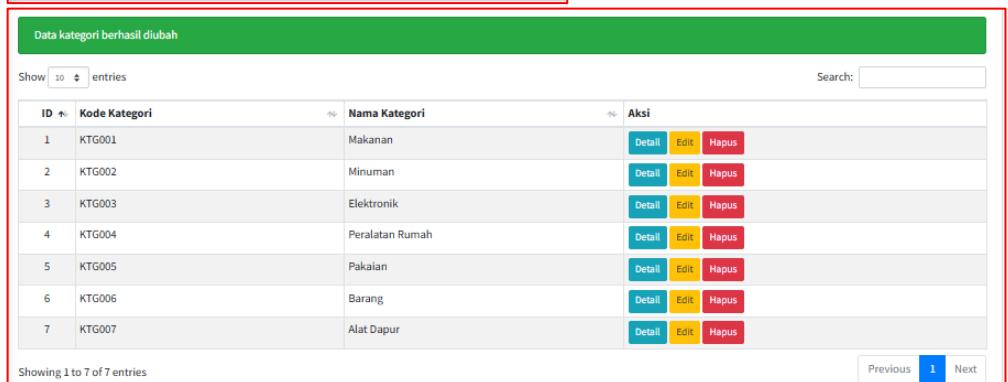
Below the table is a button labeled 'Kembali'.



- Tampilan ketika di halaman Tambah Kategori Barang



- Tampilan ketika di halaman Mengedit Kategori Barang Tertentu





- Tampilan ketika Menghapus data Kategori Barang berdasarkan ID

ID	Kode Kategori	Nama Kategori	Aksi
1	KTG001	Makanan	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	KTG002	Minuman	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	KTG003	Elektronik	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	KTG004	Peralatan Rumah	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	KTG005	Pakaian	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
6	KTG007	Alat Dapur	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

3. Tabel m_supplier

- Menambahkan Migration Baru untuk tabel m_supplier

```
INFO Migration [C:\laragon\www\PWL2025\PWL_Week5.2\Js5\PWL_POS\database\migrations\2025_03_23_193613_create_m_supplier_table.php] created successfully.

CodeView: Refactor | Explain | X
12     public function up(): void
13     {
14         Schema::create('m_supplier', function (Blueprint $table): void {
15             $table->id('supplier_id');
16             $table->string('supplier_kode', 10);
17             $table->string('supplier_nama', 100);
18             $table->string('supplier_alamat', 255);
19             $table->timestamps();
20         });
21     }
PS C:\laragon\www\PWL2025\PWL_Week5.2\Js5\PWL_POS> php artisan migrate
INFO Running migrations.
2025_03_23_193613_create_m_supplier_table ..... 139ms DONE
```

- Menambahkan Seeder Baru untuk tabel m_supplier

```
PS C:\laragon\www\PWL2025\PWL_Week5.2\Js5\PWL_POS> php artisan make:seeder SupplierSeeder
INFO Seeder [C:\laragon\www\PWL2025\PWL_Week5.2\Js5\PWL_POS\database\seeders\SupplierSeeder.php] created successfully.

public function run(): void
{
    $data_supplier = [
        [
            'supplier_kode' => 'SUP001',
            'supplier_nama' => 'PT Indomarco Prismatama',
            'supplier_alamat' => 'Jl. Ancol Barat IV No. 1, Jakarta Utara',
            'created_at' => Carbon::now(),
            'updated_at' => Carbon::now(),
        ],
        [
            'supplier_kode' => 'SUP002',
            'supplier_nama' => 'PT Sumber Alfaria Trijaya',
            'supplier_alamat' => 'Jl. M.H. Thamrin No. 9, Tangerang',
            'created_at' => Carbon::now(),
            'updated_at' => Carbon::now(),
        ],
        [
            'supplier_kode' => 'SUP003',
            'supplier_nama' => 'PT Mayora Indah Tbk',
            'supplier_alamat' => 'Jl. Daan Mogot KM 18, Jakarta Barat',
            'created_at' => Carbon::now(),
            'updated_at' => Carbon::now(),
        ],
    ];
}
```



```
PS C:\laragon\www\PWL2025\PWL_Week5.2\js5\PWL_POS> php artisan db:seed --class=SupplierSeeder
INFO Seeding database.
```

- Menambahkan kode program untuk sidebar.blade.php

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

- Tampilan ketika di halaman Dashboard Supplier

The screenshot shows a Laravel application interface. On the left is a sidebar with navigation links: Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Data Barang, Data Supplier, Supplier Barang (which is highlighted in blue), Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area has a header "Daftar Supplier" and a sub-header "Daftar supplier yang terdaftar dalam sistem". Below this is a table with columns: ID, Kode Supplier, Nama Supplier, Alamat Supplier, and Aksi (Actions). The table contains three entries:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SUP001	PT Indomarco Prismatama	JL. Ancol Barat IV No. 1, Jakarta Utara	[Detail] [Edit] [Hapus]
2	SUP002	PT Sumber Alfaria Trijaya	JL. MH. Thamrin No. 9, Tangerang	[Detail] [Edit] [Hapus]
3	SUP003	PT Mayora Indah Tbk	Jl. Daan Mogot KM 18, Jakarta Barat	[Detail] [Edit] [Hapus]

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

- Tampilan ketika di halaman Detail Supplier

The screenshot shows a Laravel application interface. The sidebar is identical to the previous one. The main content area has a header "Detail Supplier" and a sub-header "Detail Supplier". Below this is a table with columns: ID, Kode supplier, Nama supplier, and Alamat supplier. The table contains one entry:

ID	Kode supplier	Nama supplier	Alamat supplier
1	SUP001	PT Indomarco Prismatama	JL. Ancol Barat IV No. 1, Jakarta Utara

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

- Tampilan ketika di halaman Tambah Supplier



Laravel

localhost/PWL2025/PWL_Week5.2/Js5/PWL_POS/public/supplier/create

PWL - Aditya SIB 2A

Search

Tambah Supplier

Tambah supplier baru

Kode: SUP004

Nama: PT Ultra Prima Abadi

Alamat: Jl. Rawa Sumur II No.3, Kawasan Industri Pulogadung, Jakarta Timur

Simpan Kembali

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

Version 3.2.0

Daftar supplier yang terdaftar dalam sistem

Tambah

Data supplier berhasil ditambahkan

Show 10 entries Search:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SUP001	PT Indomarco Prismatama	Jl. Ancol Barat IV No. 1, Jakarta Utara	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	SUP002	PT Sumber Alfaria Trijaya	Jl. MH. Thamrin No. 9, Tangerang	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	SUP003	PT Mayora Indah Tbk	Jl. Daan Mogot KM 18, Jakarta Barat	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	SUP004	PT Ultra Prima Abadi	Jl. Rawa Sumur II No.3, Kawasan Industri Pulogadung, Jakarta Timur	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Previous 1 Next

➤ Tampilan ketika di halaman Mengedit Supplier Tertentu

4 SUP004 PT Ultra Prima Abadi Jl. Rawa Sumur II No.3, Kawasan Industri Pulogadung, Jakarta Timur Detail Edit Hapus

Edit Supplier

Home / Supplier / Edit

Edit Supplier

Kode: SUP004

Nama: PT Unilever Indonesia Tbk

Alamat: Jl. BSD Boulevard Barat, Green Office Park 3, BSD City, Tangerang Selatan

Simpan Kembali

Daftar supplier yang terdaftar dalam sistem

Tambah

Data supplier berhasil diubah

Show 10 entries Search:

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SUP001	PT Indomarco Prismatama	Jl. Ancol Barat IV No. 1, Jakarta Utara	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	SUP002	PT Sumber Alfaria Trijaya	Jl. MH. Thamrin No. 9, Tangerang	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	SUP003	PT Mayora Indah Tbk	Jl. Daan Mogot KM 18, Jakarta Barat	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	SUP004	PT Unilever Indonesia Tbk	Jl. BSD Boulevard Barat, Green Office Park 3, BSD City, Tangerang Selatan	<button>Detail</button> <button>Edit</button> <button>Hapus</button>



- Tampilan ketika Menghapus data Supplier berdasarkan ID

The screenshot shows a table of suppliers. A message at the top says "Data supplier berhasil dihapus". The table has columns: ID, Kode Supplier, Nama Supplier, Alamat Supplier, and Aksi (Actions). The last row, which is highlighted, represents SUP003. After the deletion, the table shows only three entries.

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SUP001	PT Indomarco Prismatama	Jl. Ancol Barat IV No. 1, Jakarta Utara	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	SUP002	PT Sumber Alfaria Trijaya	Jl. MH. Thamrin No. 9, Tangerang	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	SUP003	PT Mayora Indah Tbk	Jl. Daan Mogot KM 18, Jakarta Barat	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

4. Tabel m_barang

- Tampilan ketika di halaman Dashboard Data Barang

The screenshot shows a table of items. The table has columns: ID, Kode Barang, Nama Barang, Nama Kategori, Harga Beli, Harga Jual, and Aksi (Actions). The table shows 10 entries from BRG001 to BRG010. Each entry includes a "Delete" button in the actions column.

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
1	BRG001	Biskuit	Makanan	3000	5000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	BRG002	Roti	Makanan	5000	7000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	BRG003	Teh Botol	Minuman	2000	3000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	BRG004	Kopi Instan	Minuman	3000	4000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	BRG005	Mouse Wireless	Elektronik	100000	150000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
6	BRG006	Keyboard Mechanical	Elektronik	250000	350000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
7	BRG007	Sapu	Peralatan Rumah	15000	20000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
8	BRG008	Pel	Peralatan Rumah	20000	25000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
9	BRG009	Kaos Polos	Pakaian	40000	50000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
10	BRG010	Jaket Hoodie	Pakaian	100000	120000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

- Tampilan ketika di halaman Detail Data Barang

The screenshot shows a table of item details. The table has columns: ID, Kode barang, Nama barang, Kode Kategori, Harga Beli, and Harga Jual. The table shows one entry for ID 3, which corresponds to BRG003. The entry details are: Kode barang: BRG003, Nama barang: Teh Botol, Kode Kategori: Minuman, Harga Beli: 2000, and Harga Jual: 3000. There is a "Kembali" button at the bottom.

ID	Kode barang	Nama barang	Kode Kategori	Harga Beli	Harga Jual
3	BRG003	Teh Botol	Minuman	2000	3000



- Tampilan ketika di halaman Tambah Data Barang

Tambah barang baru

Kode: BRG011
Nama: Sendok
Kode Kategori: Peralatan Rumah
Harga Beli: 1000
Harga Jual: 2500

Simpan Kembali

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

Version 3.2.0

Daftar barang yang terdaftar dalam sistem

Data barang berhasil disimpan

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
11	BRG011	Sendok	Peralatan Rumah	1000	2500	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Showing 11 to 11 of 11 entries

Previous 1 2 Next

- Tampilan ketika di halaman Mengedit Data Barang Tertentu

Edit barang

Kode: BRG011
Nama: Sendok Sayur
Level: Peralatan Rumah
Harga Beli: 3000
Harga Jual: 6500

Simpan Kembali

Data barang berhasil diubah

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
11	BRG011	Sendok Sayur	Peralatan Rumah	3000	6500	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Showing 11 to 11 of 11 entries

Previous 1 2 Next



- Tampilan ketika Menghapus Data Barang berdasarkan ID

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
11	BRG011	Sendok	Peralatan Rumah	1000	2500	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Data barang berhasil dihapus

Filter: - Semua -

Kategori Barang

Show 10 entries

Search:

ID	Kode Barang	Nama Barang	Nama Kategori	Harga Beli	Harga Jual	Aksi
1	BRG001	Biskuit	Makanan	3000	5000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
2	BRG002	Roti	Makanan	5000	7000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
3	BRG003	Teh Botol	Minuman	2000	3000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
4	BRG004	Kopi Instan	Minuman	3000	4000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
5	BRG005	Mouse Wireless	Elektronik	100000	150000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
6	BRG006	Keyboard Mechanical	Elektronik	250000	350000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
7	BRG007	Sapu	Peralatan Rumah	15000	20000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
8	BRG008	Pel	Peralatan Rumah	20000	25000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
9	BRG009	Kaos Polos	Pakaian	40000	50000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
10	BRG010	Jaket Hoodie	Pakaian	100000	120000	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Showing 1 to 10 of 10 entries

Previous 1 Next