



Nama : Muhammad Bagus Indrawan

Kelas : TI-2H/20

NIM : 2241720217

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

## JOBSHEET 07

### PWL – LARAVEL STARTER CODE

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

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

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

Jadi kita bikin project Laravel 10 dengan nama **PWL\_POS**.

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

## A. WEB TEMPLATE

Merupakan suatu file yang sudah memiliki desain tertentu sehingga, kita tinggal memodifikasi bagian-bagian tertentu saja untuk mendapatkan hasil yang sudah bagus. Dengan



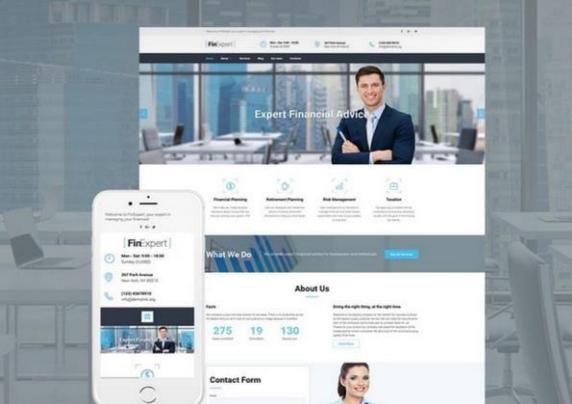
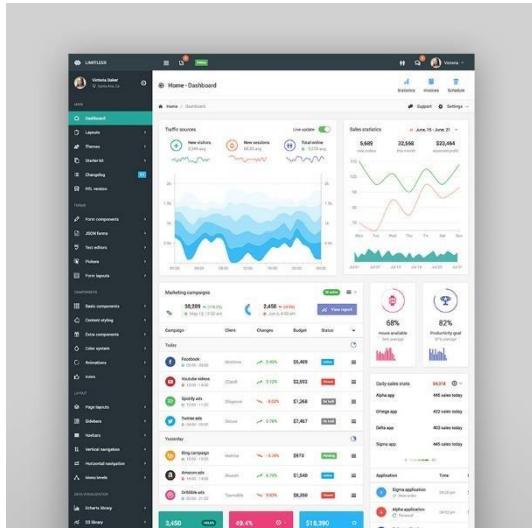
menggunakan template yang sesuai dengan kebutuhan, kita bisa mengurangi waktu pembuatan suatu website secara drastis.

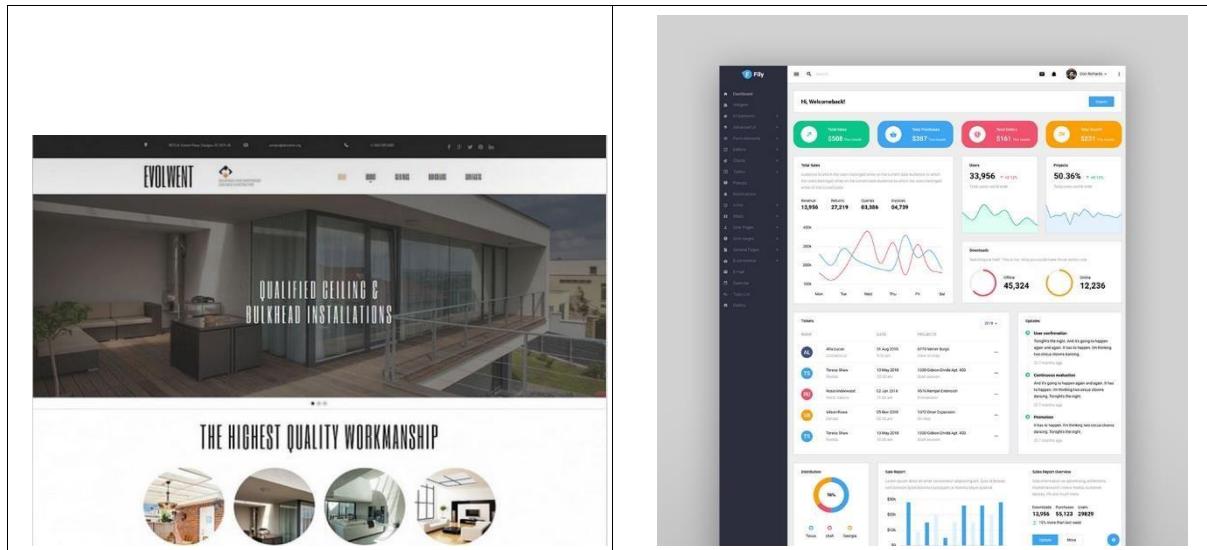
Secara umum kita bisa menggunakan template website yang sudah ada di internet untuk mempermudah kita dalam membuat website. Ada banyak sekali template yang ada di internet dan kita ambil satu contoh template untuk halaman administrator, seperti **AdminLTE** template.

**AdminLTE** adalah salah satu template yang sering digunakan oleh web developer sebagai template backend/admin pada proyek yang sering dikerjakan. Template ini dibuat menggunakan framework bootstrap yang merupakan framework CSS yang paling banyak digunakan di kalangan web desainer.

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

#### Contoh Web template

Frontend Template	Backend Template
	



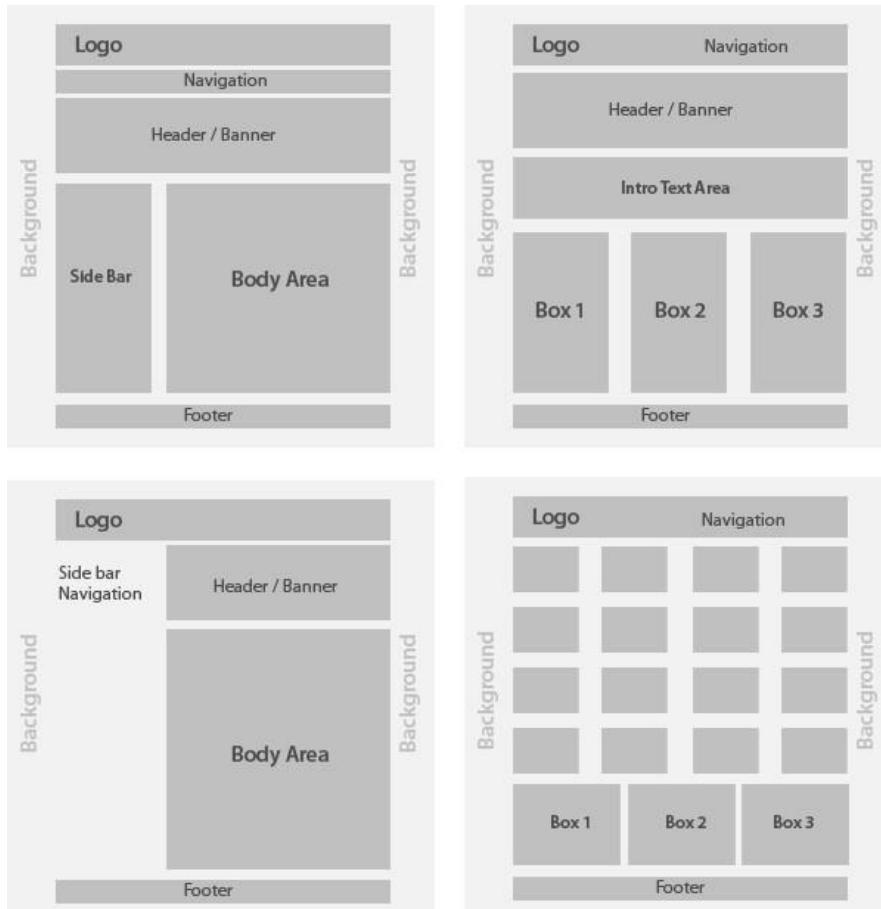
## B. WEB LAYOUT

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

- **Header** ⑦ Pada bagian header, layout designer dapat mengisi bagian ini dengan logo website, navigasi situs, ikon media sosial, dan menu pencarian.
- **Navigasi** ⑦ Navigasi dapat dipahami sebagai panduan. Di website, navigasi dapat berkisar dari menu yang muncul di bagian atas halaman web hingga menu pendukung lainnya yang biasa ditemukan di bawah situs.
- **Body/Konten** ⑦ Selain itu, terdapat elemen body/konten yang biasanya diisi dengan informasi produk, fitur produk, dan deskripsi produk yang dijual.
- **Sidebar** ⑦ elemen ini merupakan elemen yang berada di samping kanan/kiri sebuah website. Element ini kadang sudah tidak dipakai, terutama pada Web Frontend. Namun, sidebar masih digunakan di banyak halaman artikel, yang kemudian akan diisi dengan informasi produk, produk terpopuler, dan navigasi tambahan.



- **Footer** ⑦ Elemen terakhir adalah footer. Footer merupakan elemen yang berada dibawah sendiri sebagai menu pendukung atau sebagai identitas dari sebuah website.



## C. LAYOUTING ADMINLTE

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

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

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



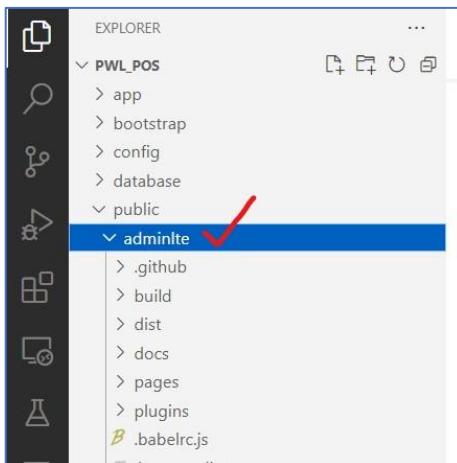
- [@yield\(\)](#)
- [@stack\(\)](#)

## Praktikum 1 – Layouting AdminLTE:

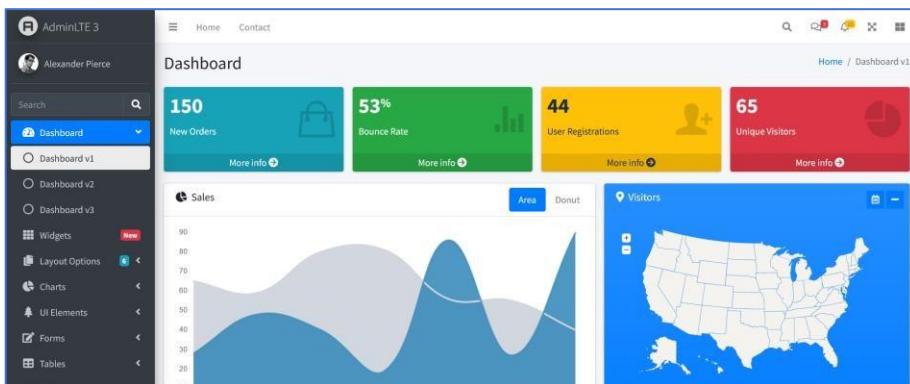
The screenshot shows the GitHub release page for AdminLTE v3.2.0. At the top, there are tabs for 'Releases' (which is selected) and 'Tags'. A search bar is on the right. Below the tabs, it shows the release date: Feb 8, 2022, and the author: REJack. It lists two commits: v3.2.0 and bd4d9c7. There is a 'Compare' button. The main content area features the title 'AdminLTE v3.2.0' with a 'Latest' badge. It includes a 'Breaking Change!' note: 'We dropped official node v12, it might still work but we won't give support for any node 12 bugs/errors.' Below that is a 'Top-Features:' section with a bulleted list: Enhanced Dark Mode, SVG Font Awesome Icon support, Docker dev-preset, Reworked Dark/Light colors, and Custom Scrollbar style (Non-Plugin-Scrollbar). The 'Contributors' section shows 17 contributors with their profile icons. The 'Assets' section has two items: 'Source code (zip)' with a red checkmark and 'Source code (tar.gz)'. Below the assets are reaction counts: 493, 56, 79, 166, 51, and 23, followed by 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](#)

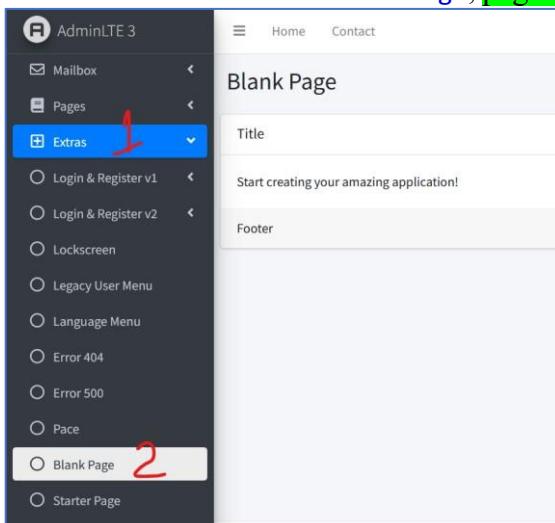
<sup>1</sup> . Kita download **AdminLTE v3.2.0** yang rilis pada 8 Feb 2022



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



4. Kita klik menu [Extras > Blank Page](#), page inilah yang akan menjadi dasar web template





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

The screenshot shows a web application interface with the following layout structure:

- 1 - Header / Navbar:** Located at the top of the main content area.
- 2 - Sidebar / Navigasi:** A vertical sidebar on the left side of the screen.
- 3 - Breadcrumb:** Located above the main content area.
- 4 - Content:** The central area where content is displayed.
- 5 - Footer:** Located at the bottom of the page.

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

The screenshot shows a right-click context menu on a web page. The 'View Page Source' option is circled in red.

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



EXPLORER

...

template.blade.php X

resources > views > layouts > template.blade.php > ...

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>AdminLTE 3 | Blank Page</title>
7
8      <!-- Google Font: Source Sans Pro -->
9      <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
10     <!-- Font Awesome -->
11     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css" integrity="sha384-5sLW6Bf41HcQ1Z8P4E6WwqOZ8l9RrYUk6F4P3" crossorigin="anonymous">
12     <!-- Theme style -->
13     <link rel="stylesheet" href="dist/css/adminlte.min.css">
14 </head>
15 <body class="hold-transition sidebar-mini">
16     <!-- Site wrapper -->
17     <div class="wrapper">
```

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

template.blade.php X

resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > nav.main-header.navbar.navbar-expand.navbar

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>AdminLTE 3 | Blank Page</title>
7
8      <!-- Google Font: Source Sans Pro -->
9      <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
10     <!-- Font Awesome -->
11     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css" integrity="sha384-5sLW6Bf41HcQ1Z8P4E6WwqOZ8l9RrYUk6F4P3" crossorigin="anonymous">
12     <!-- Theme style -->
13     <link rel="stylesheet" href="dist/css/adminlte.min.css">
14 </head>
```

Menjadi

template.blade.php X

resources > views > layouts > template.blade.php > html > body.hold-transition.sidebar-mini > div.wrapper > nav.main-header.navbar.navbar-expand.navbar

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title>
7
8      <!-- Google Font: Source Sans Pro -->
9      <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
10     <!-- Font Awesome -->
11     <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
12     <!-- Theme style -->
13     <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
14 </head>
```

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



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

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

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

```
17   <div class="wrapper">
18     <!-- Navbar -->
19     @include('layouts.header')
20     <!-- /.navbar -->
21
22     <!-- Main Sidebar Container -->
23     <aside class="main-sidebar sidebar-dark-primary elevation-4">
24       <!-- Brand Logo -->
25       <a href="../../index3.html" class="brand-link">
26         
27         <span class="brand-text font-weight-light">AdminLTE 3</span>
28       </a>
29
30     <!-- Sidebar -->
31     <div class="sidebar">
```

Menjadi

```
17   <div class="wrapper">
18     <!-- Navbar -->
19     @include('layouts.header')
20     <!-- /.navbar -->
21
22     <!-- Main Sidebar Container -->
23     <aside class="main-sidebar sidebar-dark-primary elevation-4">
24       <!-- Brand Logo -->
25       <a href="{{ url('/') }}" class="brand-link">
26         
27         <span class="brand-text font-weight-light">PWL - Starter Code</span>
28       </a>
29
30     <!-- Sidebar -->
31     <div class="sidebar">
```

#### 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>
```

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

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

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

Menjadi



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

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

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



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>
```

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

## Praktikum 2 – Penerapan Layouting:

---

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

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



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

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

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

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

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

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



```
<div class="sidebar">
    <!-- SidebarSearch Form -->
    <div class="form-inline mt-2">
        <div class="input-group" data-widget="sidebar-search">
            <input class="form-control form-control-sidebar" type="search"
placeholder="Search" aria-label="Search">      <div class="input-
group-append">
                <button class="btn btn-sidebar">
                    <i class="fas fa-search fa-fw"></i>
                </button>
            </div>
        </div>
    </div>
    <!-- Sidebar Menu -->
    <nav class="mt-2">
        <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview"
role="menu" data-accordion="false">
            <li class="nav-item">
```



```
<a href="{{ url('/') }}" class="nav-link {{ ($activeMenu == 'dashboard')? 'active' : '' }} ">
    <i class="nav-icon fas fa-tachometer-alt"></i>
    <p>Dashboard</p>
</a>
</li>
<li class="nav-header">Data Pengguna</li>
<li class="nav-item">
    <a href="{{ url('/level') }}" class="nav-link {{ ($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](http://localhost/PWL_POS/public)

## D. LARAVEL DATATABLE

Untuk menampilkan banyak data, kita bisa menampilkan data tersebut dalam format tabel. DataTable ini merupakan plugin **jQuery** yang dibuat untuk mengelola data informasi dalam bentuk grid / table.



## INFO

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

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

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

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

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

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

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

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\\_POS/resources/views/user/index.blade.php](#)

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

@section('content')
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h3 class="card-title">{{ $page->title }}</h3>
            <div class="card-tools">
                <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}>Tambah</a>
            </div>
        </div>
        <div class="card-body">
            <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
                <thead>
                    <tr><th>ID</th><th>Username</th><th>Nama</th><th>Level Pengguna</th><th>Aksi</th></tr>
                </thead>
                <tbody>
                </tbody>
            </table>
        </div>
    @endsection

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



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141  
Telp. (0341) 404424 – 404425, Fax (0341) 404420  
<http://www.polinema.ac.id>

---



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



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

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

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

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



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

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

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



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

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\\_POS/resources/views/user/create.blade.php](#)



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

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

```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141  
Telp. (0341) 404424 – 404425, Fax (0341) 404420  
<http://www.polinema.ac.id>

---

</div>



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

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..!!!

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\_POS/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
tablesmt">
                    <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
```

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

The screenshot shows a web browser window with three tabs open. The active tab is titled 'localhost/PWL\_POS/public/user/100'. The page content is a red error box with the text 'Kesalahan! Data yang Anda cari tidak ditemukan.' (Error! The data you searched for was not found.). Below the error box is a 'Kembali' (Back) button. The left sidebar has a blue-highlighted 'Data User' menu item under 'PWL - Starter Code'.

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_POS/resources/views/user/edit.blade.php`

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

@section('content')


<div class="card-header">
        <h3 class="card-title">{{ $page->title }}</h3>
        <div class="card-tools"></div>
    </div>
    <div class="card-body">
        @empty($user)
            <div class="alert alert-danger alert-dismissible">
                <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
                Data yang Anda cari tidak ditemukan.
            </div>
            <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
        @else
            <form method="POST" action="{{ url('/user/'.$user->user_id) }}"
class="form-horizontal">


```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141  
Telp. (0341) 404424 – 404425, Fax (0341) 404420  
<http://www.polinema.ac.id>

---



```
@csrf
{!! method_field('PUT') !!}<!-- tambahkan baris ini untuk proses edit
yang butuh method PUT -->
<div class="form-group row">
    <label class="col-1 control-label col-form-label">Level</label>
    <div class="col-11">
        <select class="form-control" id="level_id" name="level_id" required>
            <option value="">- Pilih Level -</option>
            @foreach($level as $item)
                <option value="{{ $item->level_id }}" @if($item->level_id ==
$user->level_id) selected @endif>{{ $item->level_nama }}</option>
            @endforeach
        </select>
        @error('level_id')
            <small class="form-text text-danger">{{ $message }}</small>
        @enderror
    </div>
</div>
<div class="form-group row">
    <label class="col-1 control-label col-form-label">Username</label>
    <div class="col-11">
        <input type="text" class="form-control" id="username"
name="username" value="{{ old('username', $user->username) }}" required>
        @error('username')
            <small class="form-text text-danger">{{ $message }}</small>
        @enderror
    </div>
</div>
<div class="form-group row">
    <label class="col-1 control-label col-form-label">Nama</label>
    <div class="col-11">
        <input type="text" class="form-control" id="nama" name="nama"
value="{{ old('nama', $user->nama) }}" required>
        @error('nama')
            <small class="form-text text-danger">{{ $message }}</small>
        @enderror
    </div>
</div>
<div class="form-group row">
    <label class="col-1 control-label col-form-label">Password</label>
    <div class="col-11">
        <input type="password" class="form-control" id="password"
name="password">
        @error('password')
            <small class="form-text text-danger">{{ $message }}</small>
        @enderror
    </div>
</div>
```



```
@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
```

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



The screenshot shows two screenshots of a Laravel application interface:

- Edit User:** A form for editing a user. It has fields for Level (set to Manager), Username (lithia), Name (Indra), and Password (\*\*\*\*\*). Below the form is a note: "Abalkan (jangan diisi) jika tidak ingin mengganti password user." At the bottom are 'Simpan' and 'Kembali' buttons.
- Daftar User:** A table listing users. The columns are ID, Username, Name, Level Pengguna, and Aksi (Actions). The 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	staff	Staff	Staff/Kasir	[Detail] [Edit] [Hapus]
4	customer-1	Pelanggan Pertama	Pelanggan	[Detail] [Edit] [Hapus]
5	manager_dua	Manager 2	Manager	[Detail] [Edit] [Hapus]
6	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	lithia	Indra	Manager	[Detail] [Edit] [Hapus]

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

Route::delete('/{id}', [UserController::class, 'destroy']); 22.

Jadi kita buat fungsi **destroy()** pada UserController.php

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

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

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



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

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

Menjadi

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

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

The screenshot shows a Laravel application interface. On the left is a sidebar with navigation links: Data Pengguna, Level User, and Data User (which is currently selected). The main content area has a title "Daftar User" and a table titled "Daftar User yang terdaftar dalam sistem". The table has columns: ID, Username, Nama, Level Pengguna, and Aksi. The data in the table is as follows:

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



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

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

## E. DATA SEARCHING DAN FILTERING

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

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

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

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



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

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

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

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

#### Praktikum 4 – Implementasi **Filtering** Datatables:

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



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

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

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

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

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

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

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

```
@if (session('error'))
    <div class="alert alert-danger">{{ session('error') }}</div>
@endif
<div class="row">
    <div class="col-md-12">
        <div class="form-group row">
            <label class="col-1 control-label col-form-label">Filter:</label>
            <div class="col-3">
                <select class="form-control" id="level_id" name="level_id" required>
                    <option value="">- Semua -</option>
                    @foreach($level as $item)
                        <option value="{{ $item->level_id }}>{{ $item->level_nama }}</option>
                    @endforeach
                </select>
                <small class="form-text text-muted">Level Pengguna</small>
            </div>
        </div>
    </div>
</div>
<table class="table table-bordered table-striped table-hover table-sm" id="table_user">
    <thead>
        <tr><th>ID</th><th>Username</th><th>Nama</th><th>Level Pengguna</th><th>Aksi</th></tr>
    </thead>
</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 datatable addIndexColumn()
```

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

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

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



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

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

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

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

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.

## F. PERTANYAAN

Jawablah pertanyaan berikut sesuai pemahaman materi di atas

1. Apa perbedaan *frontend template* dengan *backend template*?

Frontend template digunakan untuk memberikan tampilan dan interaksi pengguna di sisi pengguna, sedangkan backend template digunakan untuk mengatur logika dan fungsi di sisi server.

2. Apakah *layouting* itu penting dalam membangun sebuah website?

Layouting sangat penting dalam membangun sebuah website karena layout adalah kerangka dasar dari tampilan halaman web.

3. Jelaskan fungsi dari komponen laravel blade berikut : `@include()`, `@extend()`,

`@section()`, `@push()`, `@yield()`, dan `@stack()`

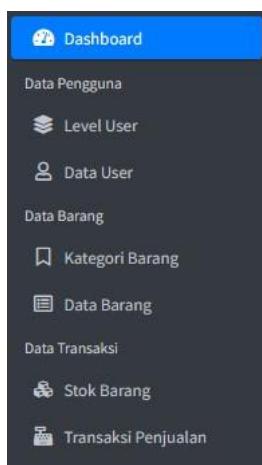
- `@include()`: Digunakan untuk menyisipkan konten dari file view lain ke dalam view saat ini.
- `@extend()`: Digunakan untuk mengambil layout utama yang akan digunakan oleh view saat ini.
- `@section()`: Digunakan untuk menentukan bagian konten dari sebuah layout.
- `@push()`: Digunakan untuk menambahkan script baru ke dalam sebuah stack, biasanya digunakan untuk menambahkan script pada bagian bawah halaman.
- `@yield()`: Digunakan untuk menampilkan konten dari sebuah section yang telah didefinisikan di layout.



- `@stack()`: Digunakan untuk menampilkan stack tertentu yang telah didefinisikan di layout.
4. Apa fungsi dan tujuan dari variable `$activeMenu` ?

Variabel `$activeMenu` berfungsi untuk menentukan style yang akan digunakan di bagian sidebar.

## G. TUGAS



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

- Data Barang

ID	Kategori Nama	Barang Kode	Barang Nama	Harga Beli	Harga Jual	Aksi
1	Elektronik	BRG001	Smartphone	1000000	1500000	[Detail, Edit, Hapus]
2	Elektronik	BRG002	Laptop	3000000	4100000	[Detail, Edit, Hapus]
3	Pakaian	BRG003	Kemeja	500000	700000	[Detail, Edit, Hapus]
4	Pakaian	BRG004	Celana Panjang	400000	600000	[Detail, Edit, Hapus]
5	Alat Rumah Tangga	BRG005	Panci	100000	150000	[Detail, Edit, Hapus]
6	Alat Rumah Tangga	BRG006	Sendok Garpu Set	50000	75000	[Detail, Edit, Hapus]
7	Otomotif	BRG007	Oli Mesin	50000	75000	[Detail, Edit, Hapus]
8	Otomotif	BRG008	Wiper Blade	30000	50000	[Detail, Edit, Hapus]
9	Olahraga	BRG009	Bola Sepak	50000	75000	[Detail, Edit, Hapus]
10	Olahraga	BRG010	Sepeda Lipat	1000000	1500000	[Detail, Edit, Hapus]



The screenshot displays two consecutive pages from a web application:

**Tambah Barang (Add Item Page):**

- Page URL: localhost/PWL\_POS/public/barang/create
- Page Title: PWL - Starter Code
- Content: A form titled "Tambah Barang Baru" with fields:
  - Kategori: dropdown menu labeled "- Pilih Kategori -"
  - Barang Kode: input field labeled "Masukkan Barang Kode"
  - Barang Nama: input field labeled "Masukkan Barang Nama"
  - Harga Beli: input field labeled "Masukkan Harga Beli"
  - Harga Jual: input field labeled "Masukkan Harga Jual"
- Buttons: "Simpan" and "Kembali"

**Detail Barang (Item Detail Page):**

- Page URL: localhost/PWL\_POS/public/barang/1
- Page Title: PWL - Starter Code
- Content: A table titled "Detail Barang" showing item details:

ID	1
Kategori	Elektronik
Kode Barang	BRG001
Nama Barang	Smartphone
Harga Beli	1000000
Harga Jual	1500000
- Buttons: "Kembali"

Both pages include a footer with copyright information: "Copyright © 2014-2021 AdminLTE.io. All rights reserved." and "Version 3.2.0".



Search

Home Contact

### Edit Barang

Kategori: Elektronik

Barang Kode: BRG001

Barang Name: Smartphone

Harga Beli: 1000000

Harga Jual: 1500000

Simpan Kembali

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

Version 3.2.0

## • Stok Barang

Search

Home Contact

### Daftar Stok Barang

Daftar Stok Barang yang Terdaftar Dalam Sistem

Tambah

Filter: - Semua -

ID	Barang	User	Tanggal Stok	Jumlah Stok	Aksi
1	Smartphone	Manager	2024-03-07 00:00:00	10	[Detail] [Edit] [Hapus]
2	Laptop	Administrator	2024-03-07 00:00:00	6	[Detail] [Edit] [Hapus]
3	Kemeja	Staff	2024-03-07 00:00:00	20	[Detail] [Edit] [Hapus]
4	Celana Panjang	Administrator	2024-03-07 00:00:00	25	[Detail] [Edit] [Hapus]
5	Panci	Manager	2024-03-07 00:00:00	10	[Detail] [Edit] [Hapus]
6	Sendok Garpu Set	Staff	2024-03-07 00:00:00	30	[Detail] [Edit] [Hapus]
7	Oli Mesin	Administrator	2024-03-07 00:00:00	10	[Detail] [Edit] [Hapus]
8	Wiper Blade	Manager	2024-03-07 00:00:00	40	[Detail] [Edit] [Hapus]
9	Bola Sepak	Administrator	2024-03-07 00:00:00	20	[Detail] [Edit] [Hapus]
10	Sepeda Lipat	Administrator	2024-03-07 00:00:00	20	[Detail] [Edit] [Hapus]

Showing 1 to 10 of 11 entries

Previous 1 2 Next

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

Version 3.2.0



localhost/PWL\_POS/public/stok/create

Tambah Stok

Tambah Stok Barang Baru

Barang: - Pilih Barang -

User: - Pilih User -

Tanggal Stok: dd/mm/yyyy

Jumlah Stok: Masukkan Jumlah Stok

Simpai | Kembali

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

Version 3.2.0

localhost/PWL\_POS/public/stok/1

Detail Stok

Detail Stok Barang

ID	1
Barang	Smartphone
User	Manager
Tanggal Stok	2024-03-07 00:00:00
Jumlah Stok	10

Kembali

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

Version 3.2.0



Search  Home Contact

**Edit Stok**

**Barang**: Smartphone

**User**:

**Tanggal Stok**:

**Jumlah Stok**:  10

**Actions**: Simpan | Kembali

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

Version 3.2.0

### ● Transaksi Penjualan

Search  Home Contact

**Daftar Penjualan**

Daftar Penjualan yang Terdapat Dalam Sistem

**Filter**: - Semua -

Show 10 entries

ID	User	Kode Penjualan	Pembeli	Tanggal Penjualan	Aksi
1	Staff	PEN001	Budi	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>
2	Staff	PEN002	Bagas	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>
3	Staff	PEN003	Luis	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>
4	Staff	PEN004	Dika	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>
5	Staff	PEN005	Huda	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>
6	Staff	PEN006	Dimas	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>
7	Staff	PEN007	Rusdi	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>
8	Staff	PEN008	Tono	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>
9	Staff	PEN009	Hisyam	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>
10	Staff	PEN010	Heri	2024-03-07 00:00:00	<a href="#">Detail</a> <a href="#">Hapus</a>

Showing 1 to 10 of 10 entries

**Actions**: Tambah Transaksi | Previous | Next

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

Version 3.2.0



Screenshot of a POS system interface showing two windows:

**Tambah Transaksi (Add Transaction)**

This window allows adding a new transaction. It includes fields for User (selected as 'PEN001'), Kode Penjualan (PEN011), Pembeli (Customer Name), and Tanggal (Date). Below these, there's a section for 'Detail Transaksi Barang' (Transaction Item Details) with fields for Barang (selected as 'Smartphone'), Harga (Price), and Jumlah (Quantity). A green 'Tambah Barang' button is available to add more items. At the bottom are 'Simpan' (Save), 'Cek Harga' (Check Price), and 'Kembali' (Back) buttons.

**Detail Penjualan (Sales Detail)**

This window displays the details of a specific sale. It shows the ID (1), User (PEN001), Kode Penjualan (PEN001), Pembeli (Budi), and Tanggal Penjualan (2024-03-07 00:00:00). Below this, a table lists the items sold:

No	Barang	Jumlah	Harga	Sub-Total
1	Smartphone	1	1000000	1000000
2	Celana Panjang	1	400000	400000
3	Kemeja	1	500000	500000

The total amount is displayed as Total : RP. 1,900,000.00. A 'Kembali' button is at the bottom.

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