

LAPORAN PRAKTIKUM JOBSHEET 5

**MATA KULIAH PEMROGRAMAN WEB LANJUT
BLADE VIEW, WEB TEMPLATING(AdminLTE), DATABASE**

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Di Susun Oleh :

Louise Nazarossa (2341760117)

SIB-2A/18

PROGRAM STUDI D4 SISTEM INFORMASI BISNIS

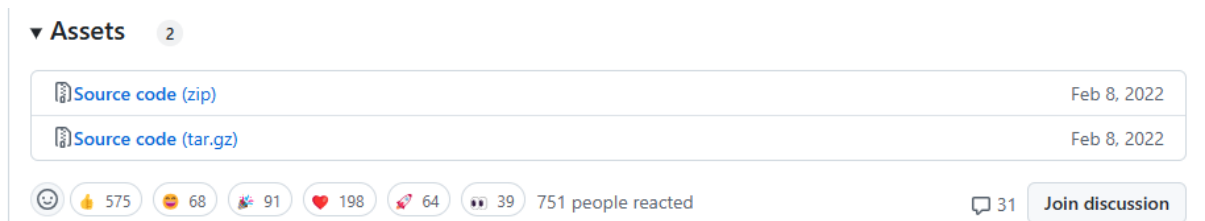
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

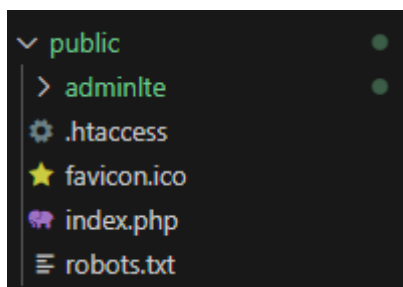
2025

PRAKTIKUM 1: Layouting AdminLTE

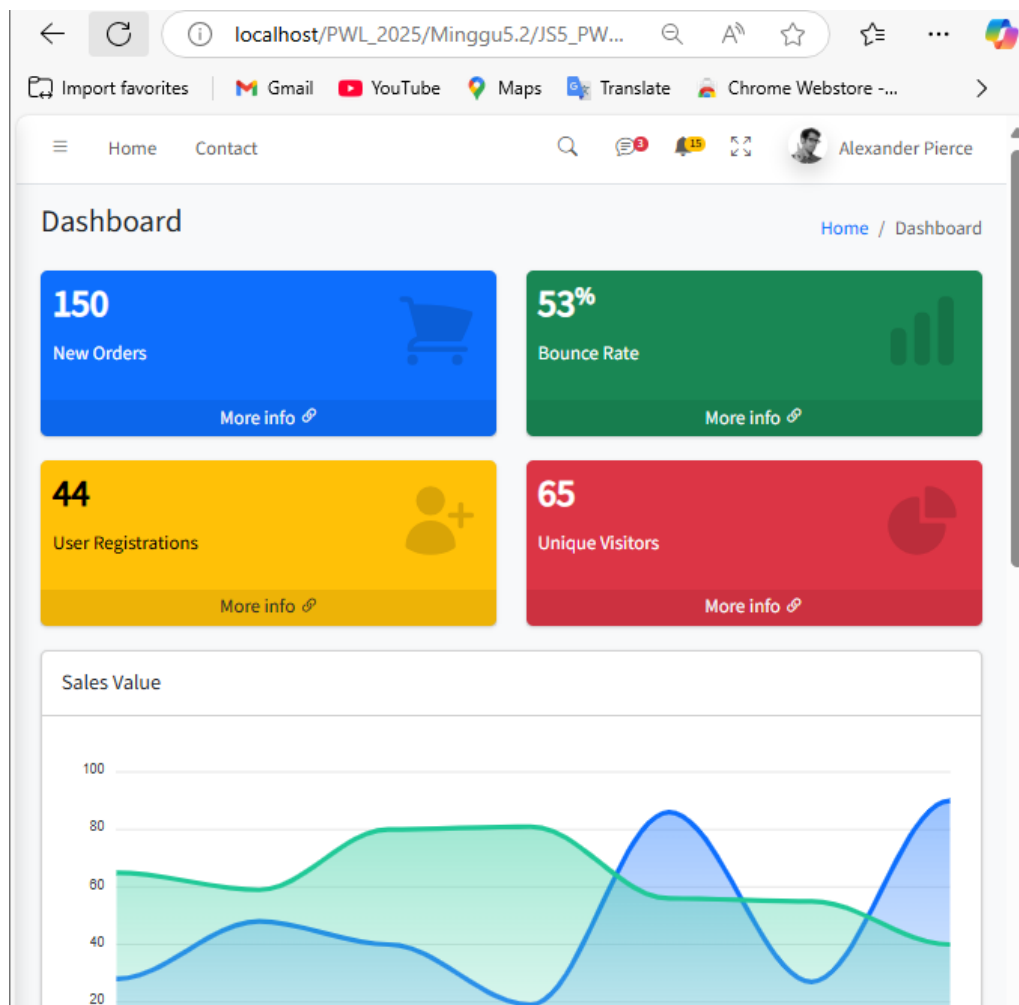
a.) Download AdminLTE v3.2.0



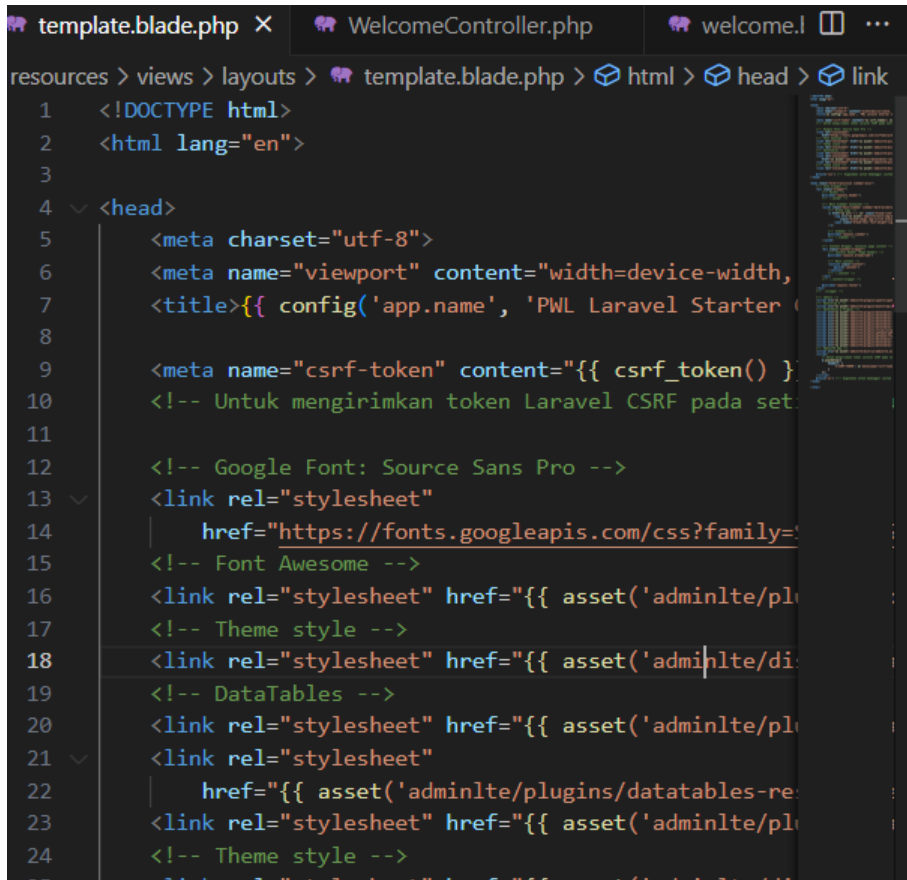
b.) Setelah kita berhasil download, kita ekstrak file yang sudah di download ke folder project PWL_POS/public, kemudian kita rename folder cukup menjadi adminlte



c.) Selanjutnya kita buka di browser dengan alamat maka akan muncul tampilan seperti berikut



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



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width,
7     <title>{{ config('app.name', 'PWL Laravel Starter
8
9     <meta name="csrf-token" content="{{ csrf_token() }}"
10    <!-- Untuk mengirimkan token Laravel CSRF pada set
11
12    <!-- Google Font: Source Sans Pro -->
13    <link rel="stylesheet"
14        href="https://fonts.googleapis.com/css?family=
15    <!-- Font Awesome -->
16    <link rel="stylesheet" href="{{ asset('adminlte/pl
17    <!-- Theme style -->
18    <link rel="stylesheet" href="{{ asset('adminlte/di
19    <!-- DataTables -->
20    <link rel="stylesheet" href="{{ asset('adminlte/pl
21    <link rel="stylesheet"
22        href="{{ asset('adminlte/plugins/datatables-re
23    <link rel="stylesheet" href="{{ asset('adminlte/pl
24    <!-- Theme style -->
```

- e.) File layouts/template.blade.php adalah file utama untuk templating website
- f.) Pada baris 1-14 file template.blade.php, kita modifikasi



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

- g.) 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

```

1 <!-- Navbar -->
2 <nav class="main-header navbar navbar-expand navbar-white navbar-light">
3 <!-- Left navbar links -->
4 <ul class="navbar-nav">
5 <li class="nav-item">
6 <a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>
7 </li>
8 <li class="nav-item d-none d-sm-inline-block">
9 <a href="../../index3.html" class="nav-link">Home</a>
10 </li>
11 <li class="nav-item d-none d-sm-inline-block">
12 <a href="#" class="nav-link">Contact</a>
13 </li>
14 </ul>

```

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

i.) Kita modifikasi baris 25 dan 26 pada template.blade.php

```

37 <!-- Main Sidebar Container -->
38 <aside class="main-sidebar sidebar-dark-primary elevation-4">
39 <!-- Brand Logo -->
40 <a href="{{ url('/') }}" class="brand-link">
41 
43 <span class="brand-text font-weight-light">PWL - Starter Code</span>
44 </a>
45

```

j.) 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

```

resources > views > layouts > sidebar.blade.php > div.sidebar
1 <div class="sidebar">
2 <!-- SidebarSearch Form -->
3 <div class="form-inline mt-2">
4 <div class="input-group" data-widget="sidebar-search">
5 <input class="form-control form-control-sidebar" type="search" placeholder="Search" ar
6 <div class="input-group-append">
7 <button class="btn btn-sidebar">
8 <i class="fas fa-search fa-fw"></i>
9 </button>
10 </div>
11 </div>
12 </div>
13 <!-- Sidebar Menu -->
14 <nav class="mt-2">
15 <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-
16 <li class="nav-item">
17 <a href="{{ url('/') }}" class="nav-link {{ $activeMenu == 'dashboard' ? 'active'
18 <i class="nav-icon fas fa-tachometer-alt"></i>
19 <p>Dashboard</p>
20 </a>
21 </li>

```

k.) 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

```

template.blade.php x footer.blade.php x header.blade.php WelcomeController.php welcome.blade.php u
resources > views > layouts > footer.blade.php > ...
1 <footer class="main-footer">
2   <div class="float-right d-none d-sm-block">
3     <b>Version</b> 3.2.0
4   </div>
5   <strong>Copyright &copy; 2014-2021 <a href="https://adminlte.io">AdminLTE.io</a>. All rights reserved.
6 </footer>
7
8 <!-- Control Sidebar -->
9 <aside class="control-sidebar control-sidebar-dark">
10   <!-- Control sidebar content goes here -->
11 </aside>
12

```

l.) Kemudian kita modifikasi file template.blade.php baris 91-100

```

68 <!-- jQuery -->
69 <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script>
70 <!-- Bootstrap 4 -->
71 <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
72 <!-- DataTables & Plugins -->
73 <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}"></script>

```

m.) Sekarang masuk pada bagian konten. Konten kita bagi menjadi 2, yaitu elemen untuk breadcrumb dan elemen untuk content

n.) Perhatikan file template.blade.php pada baris 38-52 kita jadikan sebagai elemen 4 breadcrumb. Kita blok baris 38-52 lalu kita cut, dan PVL_POS/resource/view/layouts/breadcrumb.blade.php breadcrumb.blade.php jika paste-kan di file (buat file belum ada). Sehingga tampilan dari file template.blade.php menjadi seperti berikut

```

43
44
45
46 <!-- Sidebar -->
47 @include('layouts.sidebar')
48 <!-- /.sidebar -->
49 </aside>
50
51 <!-- Content Wrapper. Contains page content -->
52 <div class="content-wrapper">
53   <!-- Content Header (Page header) -->
54   @include('layouts.breadcrumb')
55
56   <!-- Main content -->
57   <section class="content">
58     @yield('content')
59   </section>
60   <!-- /.content -->
61 </div>
62 <!-- /.content-wrapper -->

```

o.) 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.

p.) Untuk content, kita akan menghapus baris 42-66 pada file template.blade.php. dan kita ganti dengan kode seperti ini @yield('content')

q.) Hasil akhir pada file utama layouts/template.blade.php adalah seperti berikut

PRAKTIKUM 2: Penerapan Layouting

a.) Kita buat file controller dengan nama WelcomeController.php

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

b.) Kita buat file pada PWL_POS/resources/views/welcome.blade.php

```

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

```

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

```

resources > views > layouts > breadcrumb.blade.php > section.content-header > div.container-fluid > div
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>

```

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

```

1  <div class="sidebar">
2      <!-- SidebarSearch Form -->
3      <div class="form-inline mt-2">
4          <div class="input-group" data-widget="sidebar-search">
5              <input class="form-control form-control-sidebar" type="search" placeholder="Search" aria-label="Search">
6              <div class="input-group-append">
7                  <button class="btn btn-sidebar">
8                      <i class="fas fa-search fa-fw"></i>
9                  </button>
10             </div>
11         </div>
12     </div>

```

```

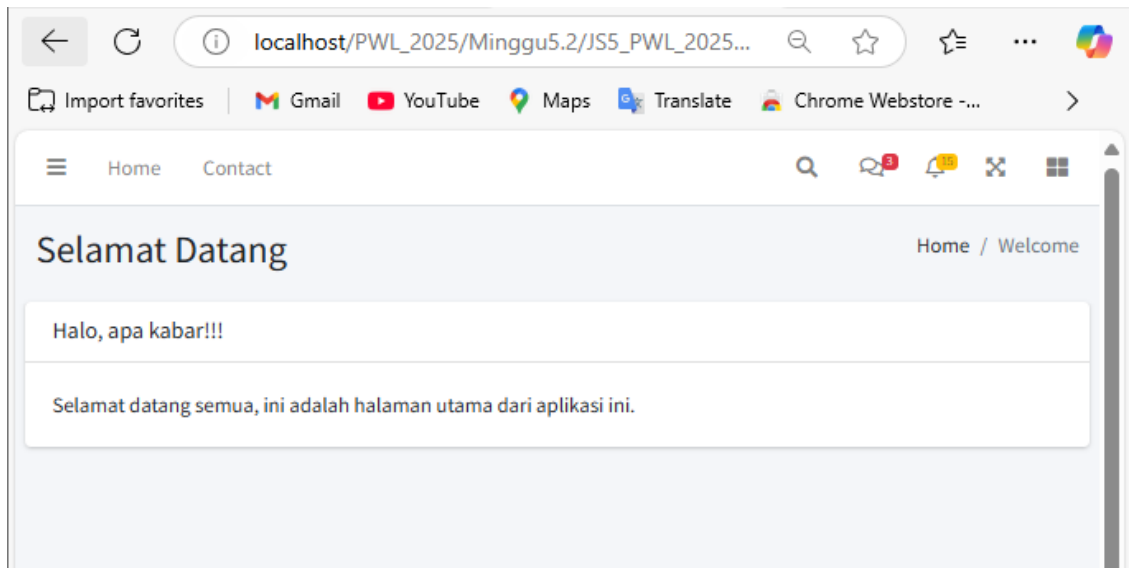
13 <!-- Sidebar Menu -->
14 <nav class="mt-2">
15     <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">
16         <li class="nav-item">
17             <a href="{{ url('/') }}" class="nav-link {{ $activeMenu == 'dashboard' ? 'active' : '' }}">
18                 <i class="nav-icon fas fa-tachometer-alt"></i>
19                 <p>Dashboard</p>
20             </a>
21         </li>
22         <li class="nav-header">Data Pengguna</li>
23         <li class="nav-item">
24             <a href="{{ url('/level') }}" class="nav-link {{ $activeMenu == 'level' ? 'active' : '' }}">
25                 <i class="nav-icon fas fa-layer-group"></i>
26                 <p>Level User</p>
27             </a>
28         </li>
29
30         <li class="nav-item">
31             <a href="{{ url('/user') }}" class="nav-link {{ $activeMenu == 'user' ? 'active' : '' }}">
32                 <i class="nav-icon far fa-user"></i>
33                 <p>Data User</p>
34             </a>
35         </li>
36         <li class="nav-header">Data Barang</li>
37         <li class="nav-item">
38             <a href="{{ url('/kategori') }}" class="nav-link {{ $activeMenu == 'kategori' ? 'active' : '' }}">
39                 <i class="nav-icon far fa-bookmark"></i>
40                 <p>Kategori Barang</p>
41             </a>
42         </li>
43         <li class="nav-item">
44             <a href="{{ url('/barang') }}" class="nav-link {{ $activeMenu == 'barang' ? 'active' : '' }}">
45                 <i class="nav-icon far fa-list-alt"></i>
46                 <p>Data Barang</p>
47             </a>
48         </li>
49         <li class="nav-item">
50             <a href="{{ url('/supplier') }}" class="nav-link {{ $activeMenu == 'supplier' ? 'active' : '' }}">
51                 <i class="nav-icon fas fa-truck"></i>
52                 <p>Data Supplier</p>
53             </a>
54         </li>
55         <li class="nav-header">Data Transaksi</li>
56         <li class="nav-item">
57             <a href="{{ url('/stok') }}" class="nav-link {{ $activeMenu == 'stok' ? 'active' : '' }}">
58                 <i class="nav-icon fas fa-cubes"></i>
59                 <p>Stok Barang</p>
60             </a>
61         </li>
62         <li class="nav-item">
63             <a href="{{ url('/barang') }}" class="nav-link {{ $activeMenu == 'penjualan' ? 'active' : '' }}">
64                 <i class="nav-icon fas fa-cash-register"></i>
65                 <p>Transaksi Penjualan</p>
66             </a>
67         </li>
68     </ul>
69 </nav>

```

e.) Kita tambahkan kode berikut router web.php

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

f.) Sekarang kita coba jalankan di browser dengan mengetikkan url http://localhost/PWL_POS/public



PRAKTIKUM 3: Implementasi jQuery Database di AdminLTE

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

```
13 Route::group(['prefix' => 'user'], function () {
14     Route::get('/', [UserController::class, 'index']);
15     Route::post('/list', [UserController::class, 'list']);
16     Route::get('/create', [UserController::class, 'create']);
17     Route::post('/', [UserController::class, 'store']);
18     Route::get('/{id}', [UserController::class, 'show']);
19     Route::get('/{id}/edit', [UserController::class, 'edit']);
20     Route::put('/{id}', [UserController::class, 'update']);
21     Route::delete('/{id}', [UserController::class, 'destroy']);
22 });
```

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

```

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

```

e.) Lalu kita buat view pada PWL/resources/views/user/index.blade.php

```

1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title">{{ $page->title }}</h3>
7              <div class="card-tools">
8                  <a class="btn btn-sm btn-primary mt-1" href="{{ url('user/create') }}">Tambah</a>
9              </div>
10         </div>
11         <div class="card-body">
12             @if (session('success'))
13                 <div class="alert alert-success">{{ session('success') }}</div>
14             @endif
15
16             @if (session('error'))
17                 <div class="alert alert-danger">{{ session('error') }}</div>
18             @endif

```

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

```

72 <!-- DataTables & Plugins -->
73 <script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}"></script>
74 <script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}"></script>
75 <script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}"></script>
76 <script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}"></script>
77 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}"></script>
78 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}"></script>
79 <script src="{{ asset('adminlte/plugins/jszip/jszip.min.js') }}"></script>
80 <script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}"></script>
81 <script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}"></script>
82 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>
83 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>
84 <script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>

```

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

```

33 public function list(Request $request)
34 {
35     $users = UserModel::select('id_user', 'username', 'nama', 'level_id')->with('level');
36
37     // Filter data user berdasarkan level_id
38     if ($request->level_id) {
39         $users->where('level_id', $request->level_id);
40     }
41
42     return DataTables::of($users)
43         // Menambahkan kolom index / nomor urut (default nama kolom: DT_RowIndex)
44         ->addIndexColumn()
45         ->addColumn('aksi', function ($user) { // Menambahkan kolom aksi
46             $btn = '<a href="' . url('/user/' . $user->user_id) . '" class="btn btn-info btn-sm">Detail</a> ';
47             $btn .= '<a href="' . url('/user/' . $user->user_id . '/edit') . '" class="btn btn-warning btn-sm">Edit';
48             $btn .= '<form class="d-inline-block" method="POST" action="' . url('/user/' . $user->user_id) . '">';
49                 . csrf_field() . method_field('DELETE') .
50                 '<button type="submit" class="btn btn-danger btn-sm" onclick="return confirm(\''Apakah Anda yakin me
51             return $btn;
52         })
53         ->rawColumns(['aksi']) // Memberitahu bahwa kolom aksi berisi HTML
54         ->make(true);
55 }

```

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

i.) Selanjutnya kita modifikasi UserController.php untuk form tambah data user

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

```

1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title">{{ $page->title }}</h3>
7              <div class="card-tools"></div>
8          </div>
9          <div class="card-body">
10             <form method="POST" action="{{ url('user') }}" class="form-horizontal">
11                 @csrf
12                 <div class="form-group row">
13                     <label class="col-1 control-label col-form-label">Level</label>
14                     <div class="col-11">
15                         <select class="form-control" id="level_id" name="level_id" required>
16                             <option value="">- Pilih Level -</option>
17                             @foreach ($level as $item)
18                                 <option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
19                             @endforeach
20                         </select>
21                         @error('level_id')
22                             <small class="form-text text-danger">{{ $message }}</small>
23                         @enderror
24                     </div>
25                 </div>

```

- k.) Kemudian untuk bisa menng-handle data yang akan disimpan ke database, kita buat fungsi store() di UserController.php
- l.) Sekarang coba kalian buka form tambah data user dengan klik tombol tambah. Amati dan pelajari..!!!
- m.) Selanjutnya, kita masuk pada bagian menampilkan detail data user (klik tombol) pada halaman user. Route yang bertugas untuk menangkap request detail adalah
- n.) Jadi kita buat/modifikasi fungsi show() pada UserController.php seperti berikut

```

81  public function store(Request $request)
82  {
83      // Validasi input
84      $request->validate([
85          'username' => 'required|string|min:3|unique:m_user,username', // Usern
86          'nama' => 'required|string|max:100', // Nama harus diisi, berupa s
87          'password' => 'required|min:5', // Password minimal 5 karakter
88          'level_id' => 'required|integer' // Level harus berupa angka dan wajib
89      ]);
90
91      // Simpan data ke database
92      UserModel::create([
93          'username' => $request->username,
94          'nama' => $request->nama,
95          'password' => bcrypt($request->password), // Enkripsi password sebelum
96          'level_id' => $request->level_id
97      ]);
98
99      // Redirect ke halaman user dengan pesan sukses
100     return redirect('/user')->with('success', 'Data user berhasil disimpan');
101 }

```

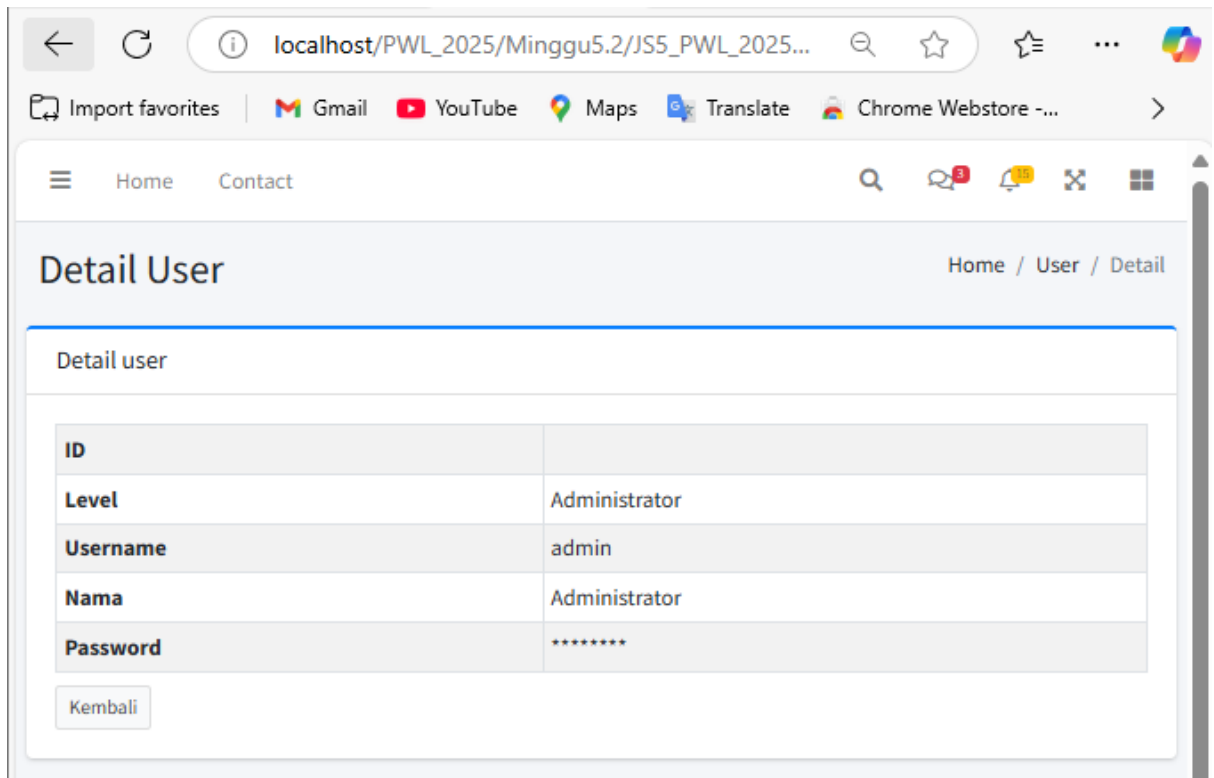
- o.) Kemudian kita buat view di PWL/resources/views/user/show.blade.php

```

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

```

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



- q.) Selanjutnya, kita masuk pada bagian untuk memodifikasi data user. Route yang bertugas untuk menangkap request edit adalah
- r.) Jadi kita buat fungsi `edit()` dan `update()` pada `UserController.php`

```
103 public function show(string $id)
104 {
105     // Ambil data user berdasarkan ID dengan relasi level
106     $user = UserModel::with('level')->find($id);
107
108     // Jika user tidak ditemukan, tampilkan halaman 404
109     if (!$user) {
110         abort(404, 'User tidak ditemukan');
111     }
112
113     // Konfigurasi breadcrumb untuk navigasi
114     $breadcrumb = (object) [
115         'title' => 'Detail User',
116         'list' => ['Home', 'User', 'Detail']
117     ];
118
119     // Konfigurasi judul halaman
120     $page = (object) [
121         'title' => 'Detail user'
122     ];
123
124     // Menentukan menu yang sedang aktif
125     $activeMenu = 'user';
126
127     // Mengembalikan tampilan dengan data yang sudah dikonfigurasi
128     return view('user.show', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
129 }
```

- s.) Selanjutnya, kita buat view untuk melakukan proses edit data user di `PWL/resources/views/user/edit.blade.php`

```

resources > views > user > edit.blade.php > ...
1  @extends('layouts.template')
2
3  @section('content')
4      <div class="card card-outline card-primary">
5          <div class="card-header">
6              <h3 class="card-title">{{ $page->title }}</h3>
7              <div class="card-tools"></div>
8          </div>
9          <div class="card-body">
10             @empty($user)
11                 <div class="alert alert-danger alert-dismissible">
12                     <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
13                     Data yang Anda cari tidak ditemukan.
14                 </div>
15                 <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
16             @else
17                 <form method="POST" action="{{ url('/user/' . $user->user_id) }}" class="form-horizontal">
18                     @csrf
19                     {!! method_field('PUT') !!} <!-- tambahkan baris ini untuk proses edit yang butuh method PUT -->
20                     <div class="form-group row">
21                         <label class="col-1 control-label col-form-label">Level</label>
22                         <div class="col-11">
23                             <select class="form-control" id="level_id" name="level_id" required>
24                                 <option value="">- Pilih Level -</option>

```

- t.) Sekarang kalian coba untuk mengedit data user di browser, amati, pahami, dan laporkan!
- u.) 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']);
```

- v.) Jadi kita buat fungsi destroy() pada UserController.php

```

182 // Menghapus data user
183 public function destroy(string $id)
184 {
185     // Mengecek apakah data user dengan ID yang dimaksud ada atau tidak
186     $check = UserModel::find($id);
187     if (!$check) {
188         return redirect('/user')->with('error', 'Data user tidak ditemukan');
189     }
190
191     try {
192         // Menghapus data user berdasarkan ID
193         UserModel::destroy($id);
194
195         return redirect('/user')->with('success', 'Data user berhasil dihapus');
196     } catch (\Illuminate\Database\QueryException $e) {
197         // Jika terjadi error ketika menghapus data,
198         // redirect kembali ke halaman dengan pesan error
199         return redirect('/user')->with('error', 'Data user gagal dihapus karena masih terdapat tabel lain yang terk
200     }
201 }
202 }

```

- w.) 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
16     @if (session('error'))
17         <div class="alert alert-danger">{{ session('error') }}</div>
18     @endif

```

PRAKTIKUM 4: Implementasi Filtering Datatables

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

```
27     $level = LevelModel::all(); // Ambil data level untuk filter level
28     return view('user.index', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' => $level, 'activeMenu' => $ac
```

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

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

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

```
$(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",
            "data": function(d) {
                d.level_id = $('#level_id').val();
            }
        },
    },
```

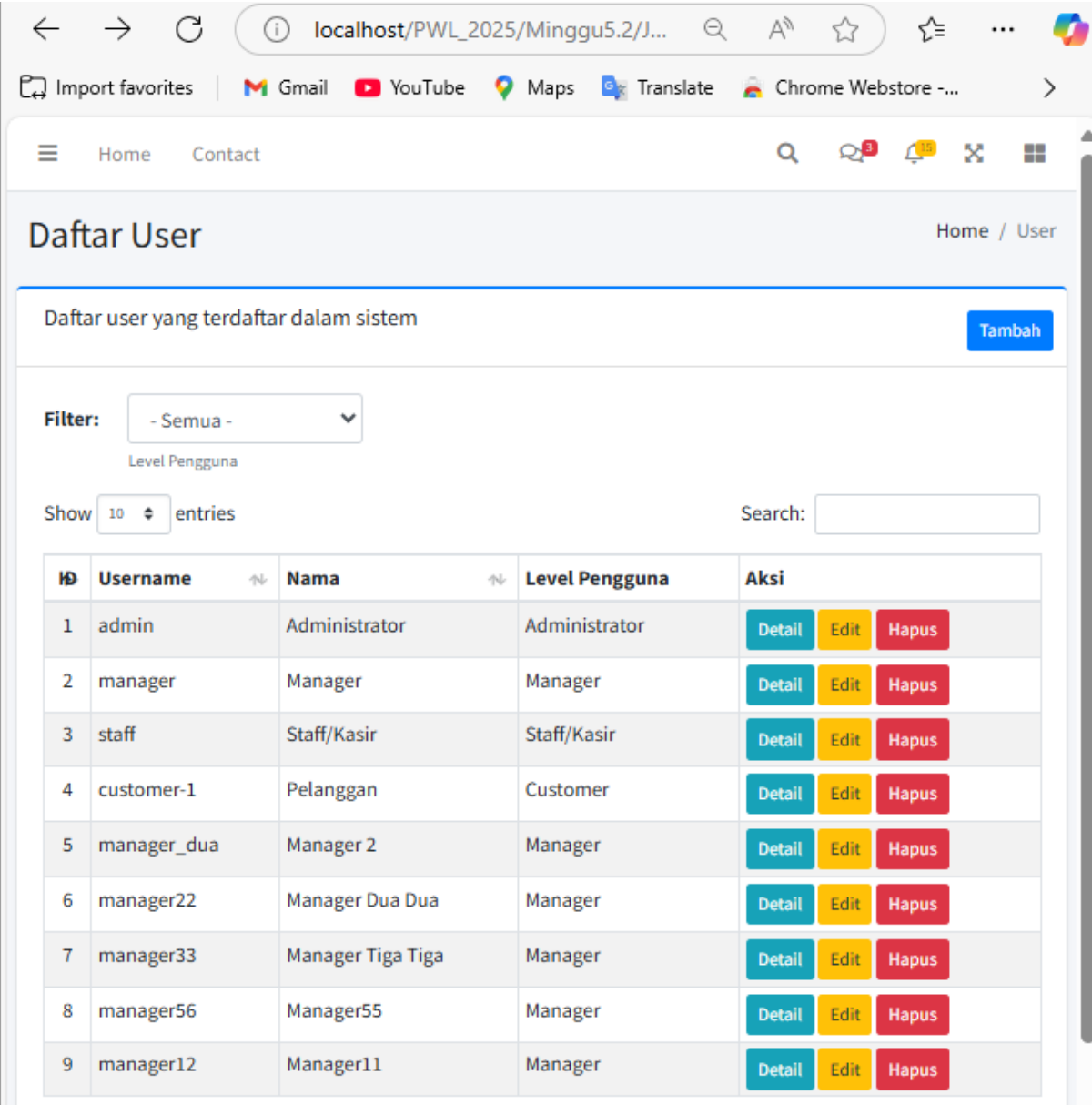
- d.) Kemudian kita edit pada bagian akhir script `@push('js')` untuk menambahkan listener jika data filtering dipilih

```
100         $('#level_id').change(function() {
101             dataUser.ajax.reload();
102         });
103     });
104 </script>
105 @endpush
```

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

```
37 // Filter data user berdasarkan level_id
38 if ($request->level_id) {
39     $users->where('level_id', $request->level_id);
40 }
```

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



Daftar user yang terdaftar dalam sistem

Filter: - Semua -
Level Pengguna

Show 10 entries Search:

ID	Username	Nama	Level Pengguna	Aksi
1	admin	Administrator	Administrator	Detail Edit Hapus
2	manager	Manager	Manager	Detail Edit Hapus
3	staff	Staff/Kasir	Staff/Kasir	Detail Edit Hapus
4	customer-1	Pelanggan	Customer	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

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

PWL - Starter Code

- Dashboard
- Data Pengguna
- Level User
- Data User
- Data Barang
- Kategori Barang
- Data Barang
- Data Supplier
- Data Transaksi
- Stok Barang
- Transaksi Penjualan

[Home](#) / [User](#)

m sistem

Tambah

	Level Pengguna	Aksi
Administrator	Administrator	Detail Edit Hapus
Manager	Manager	Detail Edit Hapus
Kasir	Staff/Kasir	Detail Edit Hapus
Ganggan	Customer	Detail Edit Hapus
ger 2	Manager	Detail Edit Hapus
ger Dua Dua	Manager	Detail Edit Hapus

Home

Contact

3

35

Daftar Level User

Home / Level

Daftar level user yang terdaftar dalam sistem

Tambah

Show 10 entries

Search:

ID	Kode	Nama	Aksi
1	ADM	Administrator	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
2	MNG	Manager	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
3	STF	Staff/Kasir	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
4	CUS	Customer	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

Showing 1 to 4 of 4 entries

Previous

1

Next

HomeContact

Daftar Kategori

Home / Kategori

Daftar kategori yang terdaftar dalam sistem

Tambah

Show 10 entriesSearch:

ID	Kode	Nama	Aksi
1	ELK	Elektronik	<div>DetailEditHapus</div>
2	PKN	Pakaian	<div>DetailEditHapus</div>
3	MKN	Makanan	<div>DetailEditHapus</div>
4	MNM	Minuman	<div>DetailEditHapus</div>
5	PRT	Peralatan Rumah Tangga	<div>DetailEditHapus</div>
6	MNB	Minuman Bersoda	<div>DetailEditHapus</div>
7	DST	Dessert	<div>DetailEditHapus</div>

Showing 1 to 7 of 7 entries

Previous1Next

localhost/PWL_2025/Minggu5.2/JS5_PW...GmailYouTubeMapsTranslateChrome Webstore -...

HomeContact

Daftar Barang

Home / Barang

Daftar barang yang terdaftar dalam sistem

Tambah

Filter: - Semua -

Kategori

Show 10 entriesSearch:

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	ELK001	Laptop	8000000	10000000	Elektronik	<div>DetailEditHapus</div>
2	ELK002	Smartphone	4000000	5000000	Elektronik	<div>DetailEditHapus</div>
3	PKN001	Kaos	75000	100000	Pakaian	<div>DetailEditHapus</div>
4	PKN002	Celana Jeans	150000	200000	Pakaian	<div>DetailEditHapus</div>
5	MKN001	Nasi Goreng	20000	25000	Makanan	<div>DetailEditHapus</div>

