

Laporan Praktikum Pemrograman Web Lanjut
Jobsheet 5: Blade View, Web Templating(AdminLTE), Datatables.



Oleh :
Danica Nasywa Putriniair (2341760122)
Kelas SIB 2B / 05

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS JURUSAN
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur

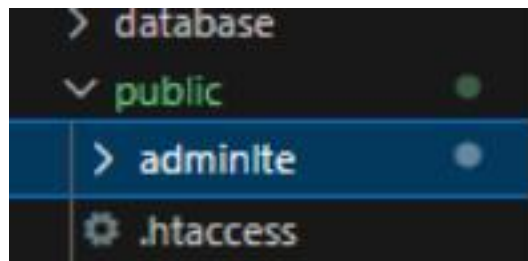
65141

Praktikum 1 - Layouting AdminLTE

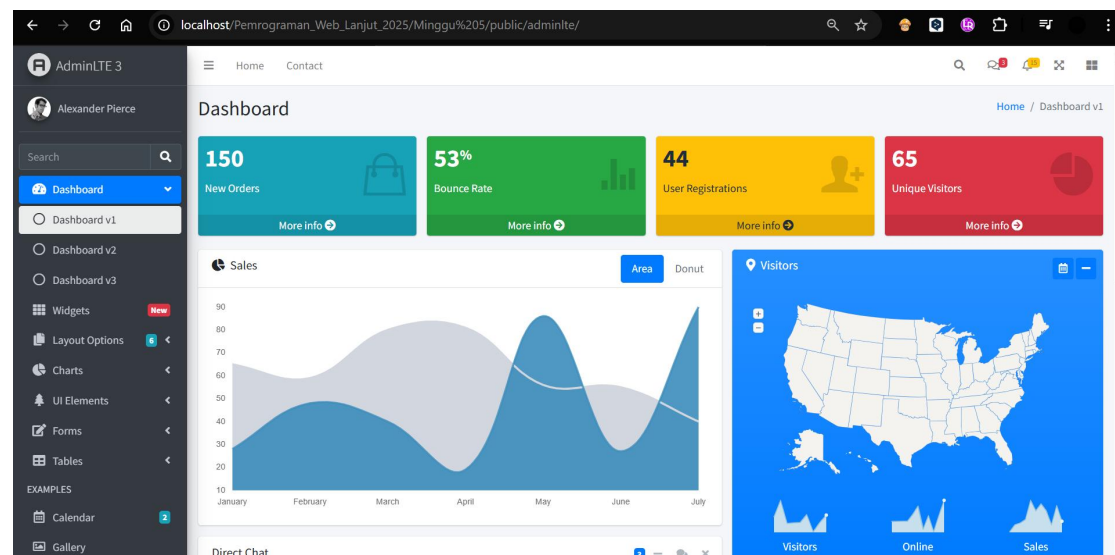
1. Kita Download AdminLTE v3.2.0 yang rilis pada 8 Feb 2022



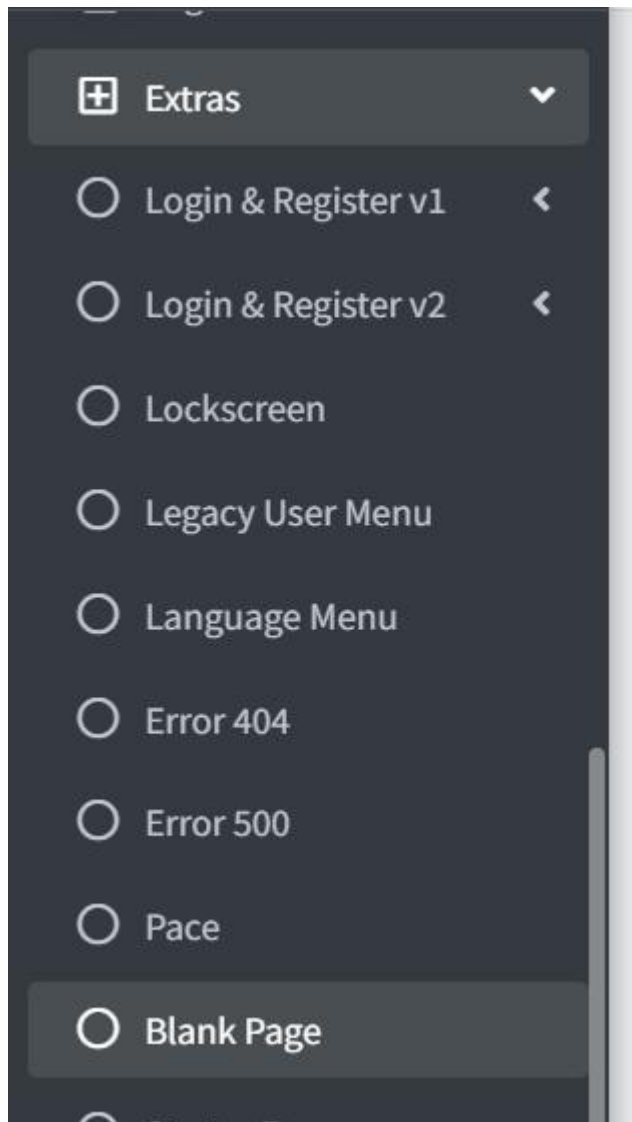
2. Setelah kita berhasil download, kita ekstrak file yang sudah di download ke folder project PWL_POS/public, kemudian kita rename folder cukup menjadi adminlte



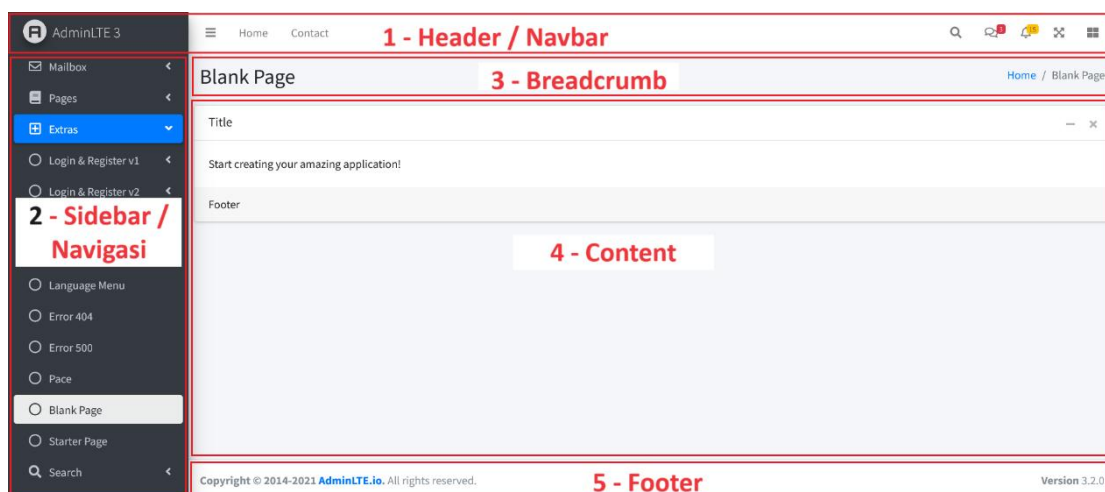
3. Selanjutnya kita buka di browser dengan alamat http://localhost/PWL_POS/public/adminlte maka akan muncul tampilan seperti berikut



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



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



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

View page source

Ctrl+U

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>AdminLTE 3 | Blank Page</title>

  <!-- Google Font: Source Sans Pro -->
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
  <!-- Font Awesome -->
  <link rel="stylesheet" href=".../plugins/fontawesome-free/css/all.min.css">
  <!-- Theme style -->
  <link rel="stylesheet" href=".../dist/css/adminlte.min.css">
</head>
<body class="hold-transition sidebar-mini">
<!-- Site wrapper -->
<div class="wrapper">
  <!-- Navbar -->
  <nav class="main-header navbar navbar-expand navbar-white navbar-light">
    <!-- Left navbar links -->
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>
      </li>
      <li class="nav-item d-none d-sm-inline-block">
        <a href=".../index3.html" class="nav-link">Home</a>
      </li>
      <li class="nav-item d-none d-sm-inline-block">
        <a href="#" class="nav-link">Contact</a>
      </li>
    </ul>

    <!-- Right navbar links -->
    <ul class="navbar-nav ml-auto">
      <!-- Navbar Search -->
      <li class="nav-item">
        <a class="nav-link" data-widget="navbar-search" href="#" role="button">
          <i class="fas fa-search"></i>
        </a>
        <div class="navbar-search-block">
          <form class="form-inline">
            <div class="input-group input-group-sm">
              <input class="form-control form-control-navbar" type="search" placeholder="Search" aria-label="Search">
              <div class="input-group-append">
                <button class="btn btn-navbar" type="submit"></button>
              </div>
            </div>
          </form>
        </div>
      </li>
    </ul>
  </nav>
</div>
```

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)

```

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=block">
10 <!-- Font Awesome -->
11 <link rel="stylesheet" href="../../plugins/fontawesome-free/css/all.min.css">
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">
18 <!-- Navbar -->
19 <nav class="main-header navbar navbar-expand navbar-white navbar-light">
20 <!-- Left navbar links -->
21 <ul class="navbar-nav">
22 <li class="nav-item">
23 <a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars">
24 </li>
25 <li class="nav-item d-none d-sm-inline-block">
26 <a href="../../index3.html" class="nav-link">Home</a>
27 </li>
28 <li class="nav-item d-none d-sm-inline-block">
29 <a href="#" class="nav-link">Contact</a>
30 </li>
31 </ul>
32
33 <!-- Right navbar links -->
34 <ul class="navbar-nav ml-auto">
35 <!-- Navbar Search -->
36 <li class="nav-item">
37 <a class="nav-link" data-widget="navbar-search" href="#" role="button">
38 <i class="fas fa-search"></i>
39 </a>
40 <div class="navbar-search-block">
41 <form class="form-inline">
42 <div class="input-group input-group-sm">
43 <input class="form-control form-control-navbar" type="search" placeholder="Search">
44 <div class="input-group-append">
45 <button class="btn btn-navbar" type="submit">
46 <i class="fas fa-search"></i>
47 </button>
48 <button class="btn btn-navbar" type="button" data-widget="navbar-search">
49 <i class="fas fa-times"></i>
50 </button>
51 </div>
52 </div>
53 </form>
54 </div>

```

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

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

```

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=block">
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


```

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
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>
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>
29
30         <!-- Sidebar -->
31         <div class="sidebar">
32             <!-- Sidebar user (optional) -->
33             <div class="user-panel mt-3 pb-3 mb-3 d-flex">
34                 <div class="image">
35                     
37                 <div class="info">
38                     <a href="#" class="d-block">Alexander Pierce</a>
39                 </div>
40             </div>
41
42             <!-- SidebarSearch Form -->
43             <div class="form-inline">
44                 <div class="input-group" data-widget="sidebar-search">
45                     <input class="form-control form-control-sidebar" type="search" placeholder="Sear
46                     <div class="input-group-append">
47                         <button class="btn btn-sidebar">
48                             <i class="fas fa-search fa-fw"></i>
49                         </button>
50                     </div>
51                 </div>
52             </div>
53
54             <!-- Sidebar Menu -->

```

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

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

```

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

```

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

```

</div>

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

<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>Blank Page</h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><a href="#">Home</a></li>
            <li class="breadcrumb-item active">Blank Page</li>
          </ol>
        </div>
      </div>
    </div>
  </section>
</div>

```

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

```

</div>

<!-- /.content-wrapper -->

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

<!-- ./wrapper -->

<!-- jQuery -->

```

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

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

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

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

```
</aside>

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

17. Layout terakhir adalah pada bagian konten. Layout untuk konten bisa kita buat dinamis, sesuai dengan apa yang ingin kita sajikan pada web yang kita bangun.

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

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


```

<div class="wrapper">
  <!-- Navbar -->
  @include('layouts.header')
  <!-- /.navbar -->

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

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

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

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

  @include('layouts.footer')
</div>
<!-- /.wrapper -->

<!-- jQuery -->

```

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

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

Praktikum 2 - Penerapan Layouting

1. Kita buat file controller dengan nama WelcomeController.php

```
pp > Http > Controllers > WelcomeController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  class WelcomeController extends Controller
6  {
7      public function index() {
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 }
18
```

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

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

```

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

```

resources > views > layouts > sidebar.blade.php > div.sidebar
1 <div class="sidebar">
2
3   <div class="form-inline mt-2">
4     <div class="input-group" data-widget="sidebar-search">
5       <input class="form-control form-control-sidebar" type="search" placeholder="Search" aria-label="Search">
6       <div class="input-group-append">
7         <button class="btn btn-sidebar">
8           <i class="fas fa-search fa-fw"></i>
9         </button>
10      </div>
11    </div>
12  </div>
13
14  <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      <li class="nav-item">
30        <a href="{{ url('/user') }}" class="nav-link {{ ($activeMenu == 'user') ? 'active' : '' }}">
31          <i class="nav-icon far fa-user"></i>
32          <p>Data User</p>
33        </a>
34      </li>
35      <li class="nav-header">Data Barang</li>
36      <li class="nav-item">
37        <a href="{{ url('/kategori') }}" class="nav-link {{ ($activeMenu == 'kategori') ? 'active' : '' }}">
38          <i class="nav-icon far fa-bookmark"></i>
39          <p>Kategori Barang</p>
40        </a>
41      </li>
42      <li class="nav-item">
43        <a href="{{ url('/barang') }}" class="nav-link {{ ($activeMenu == 'barang') ? 'active' : '' }}">
44          <i class="nav-icon far fa-list-alt"></i>
45          <p>Data Barang</p>
46        </a>
47      </li>
48      <li class="nav-header">Data Transaksi</li>
49      <li class="nav-item">
50        <a href="{{ url('/stok') }}" class="nav-link {{ ($activeMenu == 'stok') ? 'active' : '' }}">
51          <i class="nav-icon fas fa-cubes"></i>
52          <p>Stok Barang</p>
53        </a>
54      </li>
55    </ul>
56  </nav>

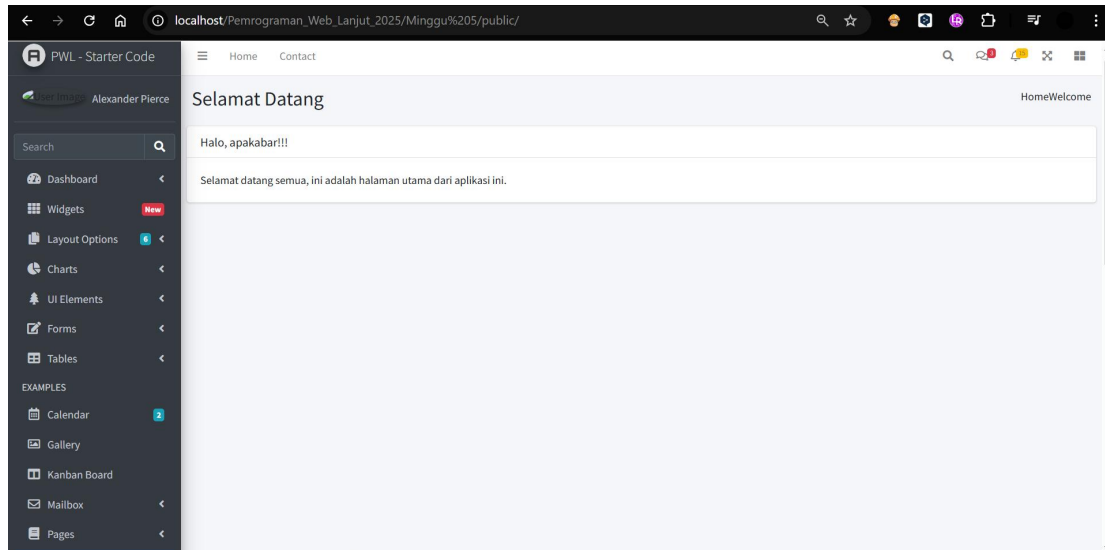
```

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/Pemrograman_Web_Lanjut_2025/Minggu%205/public/



Praktikum 3 - Implementasi jQuery Datatable di AdminLTE

1. Kita modifikasi proses CRUD pada tabel m_user pada praktikum ini
2. Kita gunakan library Yajra-datatable dengan mengetikkan perintah pada CMD

composer require yajra/laravel-datatables:^10.0 atau

composer require yajra/laravel-datatables-oracle

```
PS E:\Aplik\laragon\www\Pemrograman Web Lanjut 2025\Minggu 5> composer require yajra/laravel-datatables:^10.0
The "10.0" constraint for "yajra/laravel-datatables" appears too strict and will likely not match what you want. See https://getcomposer.org/constraints
./composer.json has been updated
Running composer update yajra/laravel-datatables
Loading composer repositories with package information
Updating dependencies
Lock file operations: 7 installs, 0 updates, 0 removals
  - Locking league/fractal (0.20.2)
  - Locking yajra/laravel-datatables (v10.0.0)
  - Locking yajra/laravel-datatables-buttons (v10.0.9)
  - Locking yajra/laravel-datatables-editor (v1.25.4)
  - Locking yajra/laravel-datatables-fractal (v10.0.0)
  - Locking yajra/laravel-datatables-html (v10.12.0)
  - Locking yajra/laravel-datatables-oracle (v10.11.4)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 7 installs, 0 updates, 0 removals
  - Installing yajra/laravel-datatables-oracle (v10.11.4): Extracting archive
  - Installing yajra/laravel-datatables-html (v10.12.0): Extracting archive
  - Installing league/fractal (0.20.2): Extracting archive
  - Installing yajra/laravel-datatables-fractal (v10.0.0): Extracting archive
  - Installing yajra/laravel-datatables-editor (v1.25.4): Extracting archive
  - Installing yajra/laravel-datatables-buttons (v10.0.9): Extracting archive
  - Installing yajra/laravel-datatables (v10.0.0): Extracting archive
1/7 [====>-----] 14%
```

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

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

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 data
    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
});
```

4. Kita buat atau modifikasi penuh untuk UserController.php. Kita buat fungsi index() untuk menampilkan halaman awal user
5. Lalu kita buat view pada PWL/resource/views/suer/index.blade.php


```

2  @endpush
3
4  @push('js')
5  <script>
6      $(document).ready(function () {
7          var dataUser = $('#table_user').DataTable({
8              serverSide: true,
9              ajax: {
10                  url: "{{ url('user/list') }}",
11                  "dataType": "json",
12                  "type": "POST"
13              },
14              columns: [
15                  {
16                      // nomor urut dari laravel datatable addIndexColumn()
17                      data: "DT_RowIndex",
18                      className: "text-center",
19                      orderable: false,
20                      searchable: false
21                  },
22                  {
23                      data: "username",
24                      className: "",
25                      orderable: true,
26                      searchable: true
27                  },
28                  {
29                      data: "nama",
30                      className: "",
31                      orderable: true,
32                      searchable: true
33                  },
34                  {
35                      // mengambil data level hasil dari ORM berelasi
36                      data: "level.level_nama",
37                      className: "",
38                      orderable: false,
39                      searchable: false
40                  },
41                  {
42                      data: "aksi",
43                      className: "",
44                      orderable: false,
45                      searchable: false
46                  }
47              ]
48          });
49      });
50  </script>
51  @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

```

8 <meta name="csrf-token" content="{{ csrf_token() }}"> <!-- Untuk mengirimkan token Laravel CSRF pada setiap request -->
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=flex" -->
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>

```

```

<script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script>
<!-- Bootstrap 4 -->
<script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<!-- DataTables & Plugins -->
<script src="{{ asset('adminlte/plugins/datatables/jquery.dataTables.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/datatables-responsive/js/dataTables.responsive.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/dataTables.buttons.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/jszip/jszip.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/pdfmake/pdfmake.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/pdfmake/vfs_fonts.js') }}"></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>
<script src="{{ asset('adminlte/plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>
<!-- AdminLTE App -->
<script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>
</script>
// Untuk mengirimkan token Laravel CSRF pada setiap request ajax
$.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}});
</script>
@stack('js')
</body>

```

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

```

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

    return DataTables::of($users)
        // Menambahkan kolom index / no urut (default nama kolom: DT_RowIndex)
        ->addIndexColumn()
        ->addColumn('aksi', function ($user) { // Menambahkan kolom aksi
            $btn = '<a href="'.url('/user/'. $user->user_id).'>Detail</a> ';
            $btn .= '<a href="'.url('/user/'. $user->user_id . '/edit') . '>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 y
            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.

Home
Contact

Daftar User

HomeUser

Daftar user yang terdaftar dalam sistem

Tambah

Show 10 entries

Search:

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

Showing 1 to 8 of 8 entries

Previous

1

Next

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'
    ];

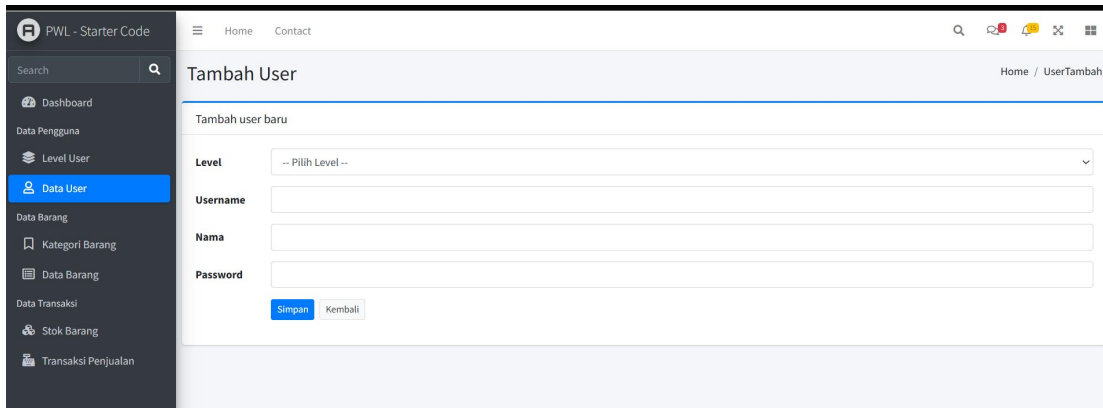
    $level = LevelModel::all(); // ambil data level untuk ditampilkan di form
    $activeMenu = 'user'; // set menu yang sedang aktif

    return view('user.create', ['breadcrumb' => $breadcrumb, 'page' => $page, 'level' =>

```

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

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

```
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
```

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]);
}
```

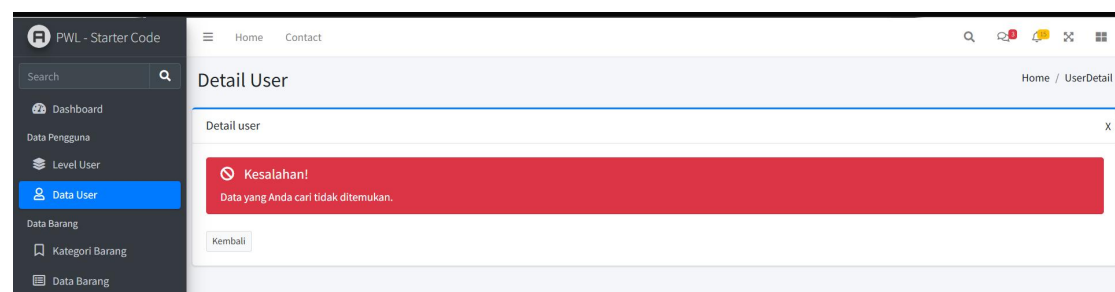
15. Kemudian kita buat view di PWL/resources/views/user/show.blade.php


```

sources / views / user / show.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">{{ $page->title }}</h3>
7         <div class="card-tools">X</div>
8     </div>
9     <div class="card-body">
10         @empty($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         @else
16             <table class="table table-bordered table-striped table-hover table-sm">
17                 <tr>
18                     <th>ID</th>
19                     <td>{{ $user->user_id }}</td>
20                 </tr>
21                 <tr>
22                     <th>Level</th>
23                     <td>{{ $user->level->level_nama }}</td>
24                 </tr>
25                 <tr>
26                     <th>Username</th>
27                     <td>{{ $user->username }}</td>
28                 </tr>
29                 <tr>
30                     <th>Nama</th>
31                     <td>{{ $user->nama }}</td>
32                 </tr>
33                 <tr>
34                     <th>Password</th>
35                     <td>*****</td>
36                 </tr>
37             </table>
38             @endempty
39             <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
40         </div>
41     </div>
42 @endsection
43
44 @push('css')
45 @endpush
46
47 @push('js')
48 @endpush

```

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



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

```

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

```

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

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

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

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

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

```

<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>
      <a href="{{ url('user') }}" class="btn btn-sm btn-default mt-2">Kembali</a>
    @else
      <form method="POST" action="{{ url('/user/'.$user->user_id) }}" class="form-horizontal">
        @csrf
        {!! method_field('PUT') !!} <div class="form-group row">
          <label class="col-1 control-label col-form-label">Level</label>
          <div class="col-11">
            <select class="form-control" id="level_id" name="level_id" required>
              <option value="">-- Pilih Level --</option>
              @foreach($level as $item)
                <option value="{{ $item->level_id }}" @if($item->level_id == $user->level_id) se
              @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('use
            @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', $use
            @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')

```

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

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Home Contact

Daftar User

HomeUser

Daftar user yang terdaftar dalam sistem

Tambah

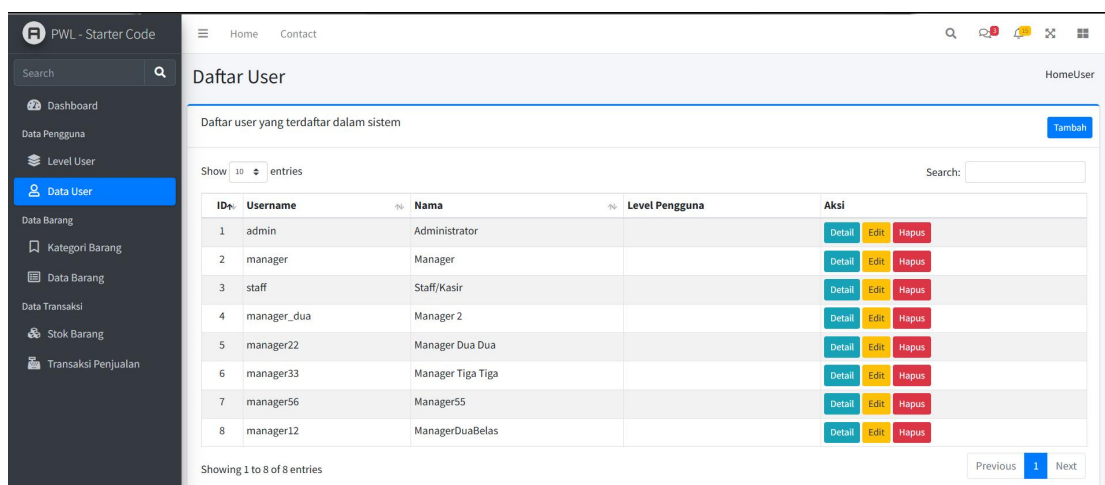
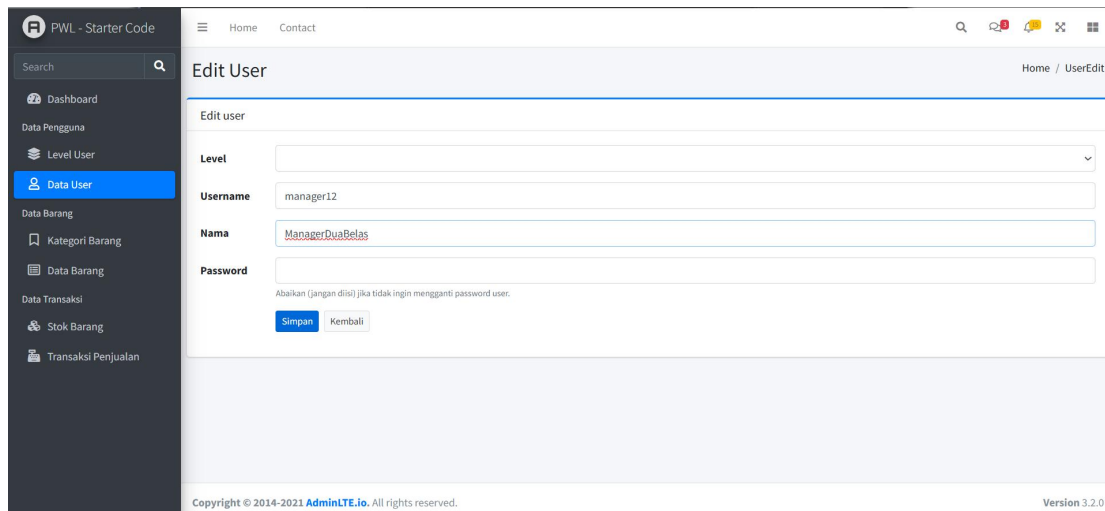
Show 10 entries

Search:

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

Showing 1 to 8 of 8 entries

Previous 1 Next



Data yang dirubah telah tersimpan

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']); // menghapus data user
```

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

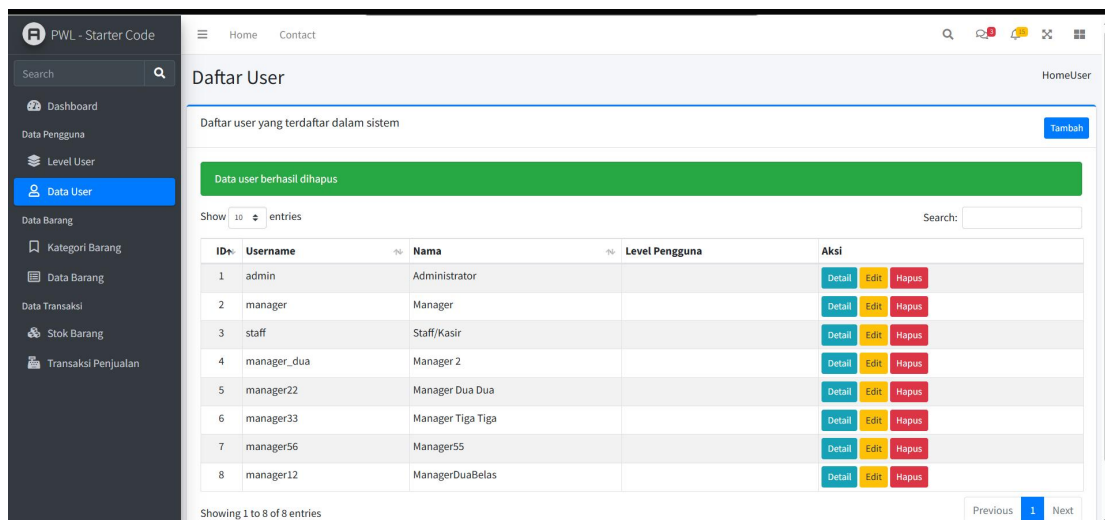
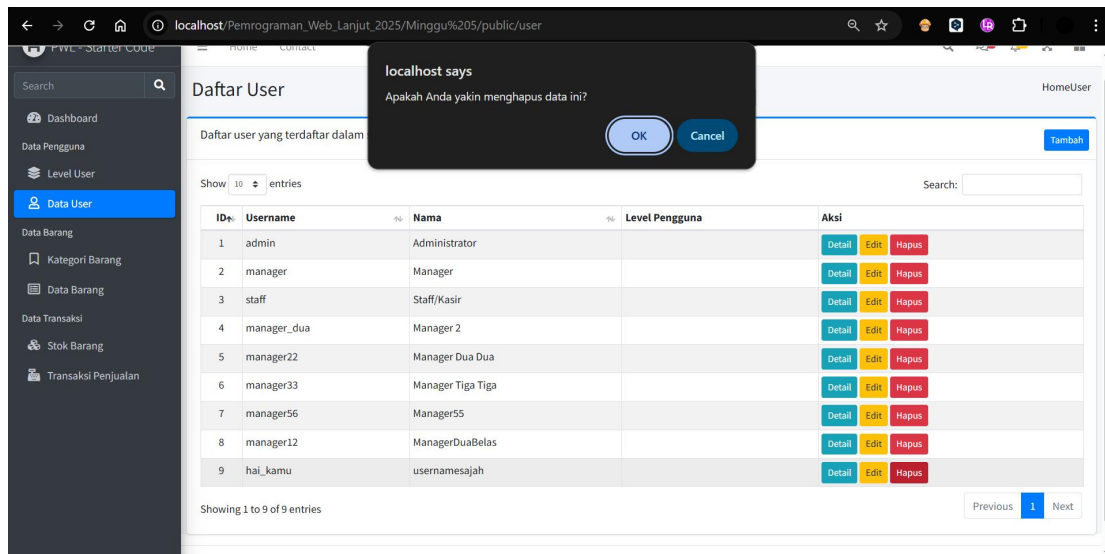
        return redirect('/user')->with('success', 'Data user berhasil dihapus');
    } catch (\Illuminate\Database\QueryException $e) {
        // Jika terjadi error ketika menghapus data, redirect kembali ke halaman dengan membawa pesan error
        return redirect('/user')->with('error', 'Data user gagal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

23. Selanjutnya kita modifikasi file PWL/resources/views/user/index.blade.php untuk

menambahkan tampilan jika ada pesan error

```
</div>
<div class="card-body">
  @if (session('success'))
    <div class="alert alert-success">{{ session('success') }}</div>
  @endif
  @if (session('error'))
    <div class="alert alert-danger">{{ session('error') }}</div>
  @endif
</div>
```

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



25. Selamat, kalian sudah membuat Laravel Starter Code untuk proses CRUD dengan

menggunakan template AdminLTE dan plugin jQuery Datatables.

26. Jangan lupa commit dan push ke github PWL_POS kalian

Praktikum 4 - Implementasi *Filtering* Datatables

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

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

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

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

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

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

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

```
@endiv
<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">
```

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

```

ajax: {
    url: "{{ url('user/list') }}",
    "dataType": "json",
    "type": "POST",
    "data": function (d) {
        d.level_id = $('#level_id').val();
    }
},

```

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

```

});

$('#level_id').on('change', function () {
    dataUser.ajax.reload();
});

});
</script>
@endpush

```

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

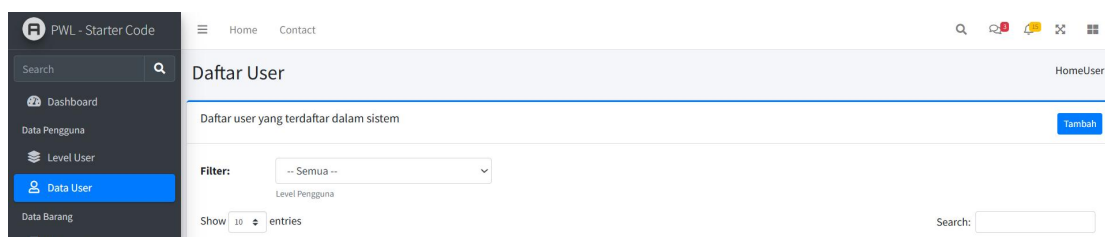
```

->with('level');

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

```

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



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

8. Jangan lupa commit dan push ke github PWL_POS kalian

Tugas

PWL - Starter Code

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Data Supplier

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

HomeContact

Daftar Kategori

HomeKategori

Daftar Kategori yang terdaftar dalam sistem

Tambah

Show 10 entries

Search:

ID	Kode Kategori	Nama Kategori	Aksi
1	BABY	BabyKid	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
2	FASH	Fashion	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
3	FOOD	Makanan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
4	BEAU	Kecantikan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
5	HOME	HomeCare	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
6	SMK	Snack/Makanan Ringan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
7	CML	Camilan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>
8	MNR	Minuman Ringan	<div>Detail</div> <div>Edit</div> <div>Hapus</div>

Showing 1 to 8 of 8 entries

Previous1Next