

## JOBSHEET 07

### PWL – LARAVEL STARTER CODE

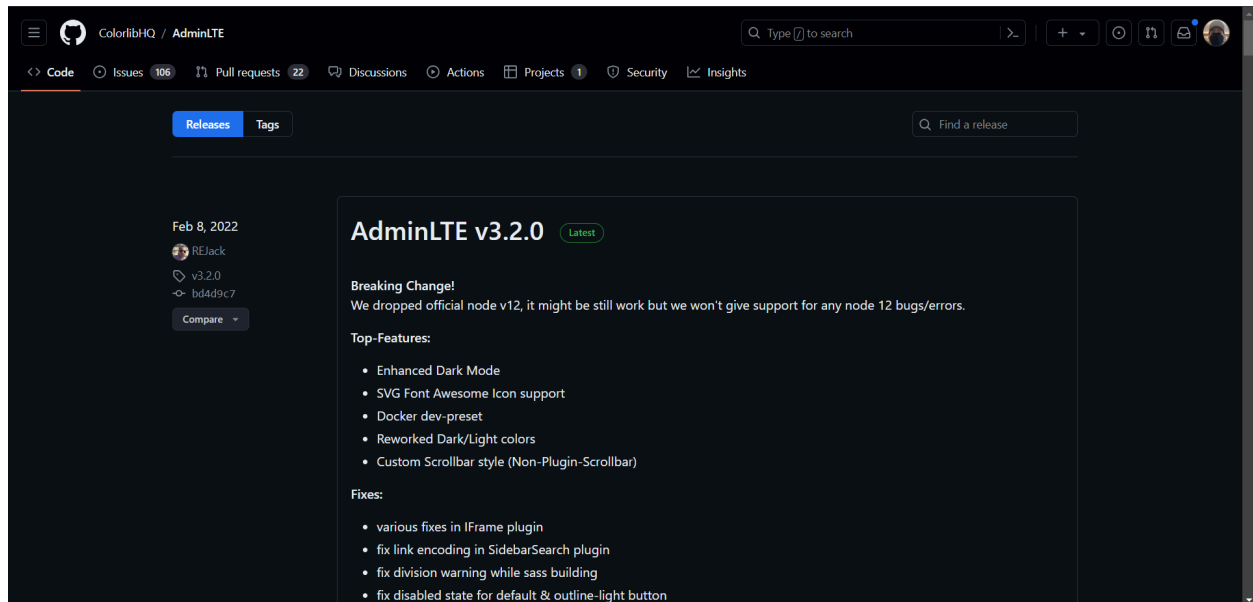
Nama : Raffy Jamil Octavialdy

Kelas : TI - 2H

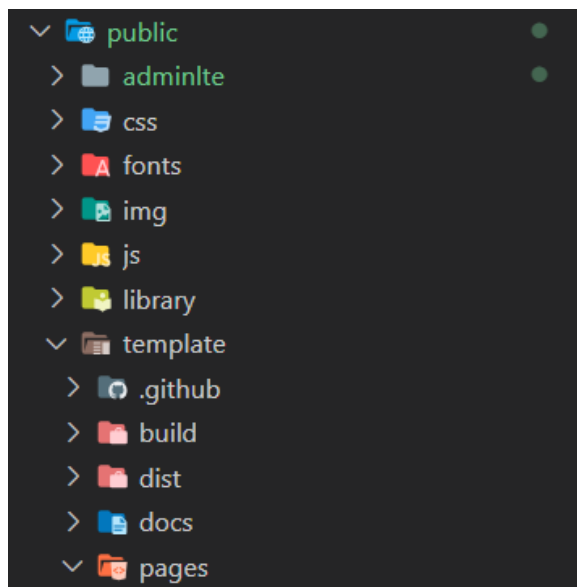
NIM : 2241720082

#### Praktikum 1 – Layouting AdminLTE

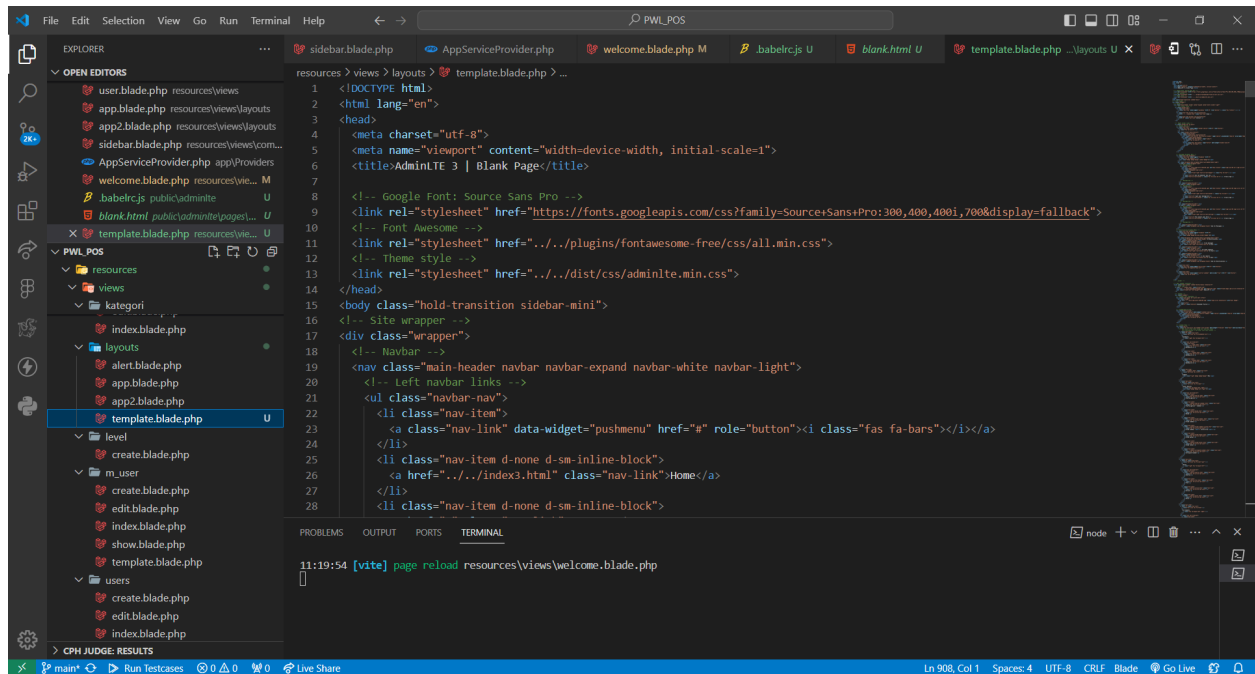
1. Kita download AdminLTE v3.2.0 yang rilis pada 8 Feb 2022



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 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)



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

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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title>

  <!-- Google Font: Source Sans Pro -->
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
  <!-- Font Awesome -->
  <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
  <!-- Theme style -->
  <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
</head>
```

6, 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

```
<body class="hold-transition sidebar-mini">
<!-- Site wrapper -->
<div class="wrapper">
  <!-- Navbar -->
  @include('layouts.header')
  <!-- /.navbar -->

  <!-- Main Sidebar Container -->
  <aside class="main-sidebar sidebar-dark-primary elevation-4">
    <!-- Brand Logo -->
    <a href="../../../index3.html" class="brand-link">
      
      <span class="brand-text font-weight-light">AdminLTE 3</span>
    </a>
```

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

```
<!-- 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 Code</span>
  </a>
```

8. 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

```
<!-- 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 Code</span>
  </a>

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

9. 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

```
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

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

10. 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/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<!-- AdminLTE App -->
<script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>
<!-- AdminLTE for demo purposes -->
<script src="{{ asset('adminlte/dist/js/demo.js') }}"></script>
</body>
</html>
```

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

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

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

  <!-- Main content -->
  <section class="content">

    <!-- Default box -->
    <div class="card">
      <div class="card-header">
        <h3 class="card-title">Title</h3>

        <div class="card-tools">
          <button type="button" class="btn btn-tool" data-card-widget="collapse" title="Collapse">
            <i class="fas fa-minus"></i>

```

13. 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.

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

15. 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 Code</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>
```

## Praktikum 2 – Penerapan Layouting

1. Kita buat file controller dengan nama WelcomeController.php

```
class WelcomeController extends Controller
{
    public function index()
    {
        $breadcrumb = (object) [
            'title' => 'Selamat Datang',
            'list' => ['Home', 'Welcome']
        ];

        $activeMenu = 'dashboard';

        return view('welcome', ['breadcrumb' => $breadcrumb, 'activeMenu' => $activeMenu]);
    }
}
```

2. Kita buat file pada PWL\_POS/resources/views/welcome.blade.php

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

@section('content')

<div class="card">
    <div class="card-header">
        <h3 class="card-title">Halo, Apakabar!!!</h3>
        <div class="card-tools"></div>
    </div>
    <div class="card-body">
        Selamat datang semua, ini adalah halaman utama dari aplikasi ini.
    </div>
</div>
@endsection
```

3. Kita modifikasi file PWL\_POS/resources/views/layouts/breadcrumb.blade.php

```
<section class="content-header">
  <div class="container-fluid">
    <div class="row mb-2">
      <div class="col-sm-6">
        <h1>{{ $breadcrumb->title }}</h1>
      </div>
      <div class="col-sm-6">
        <ol class="breadcrumb float-sm-right">
          @foreach ($breadcrumb->list as $key => $value)
            @if ($key == count($breadcrumb->list) - 1)
              <li class="breadcrumb-item active">{{ $value }}</li>
            @else
              <li class="breadcrumb-item">{{ $value }}</li>
            @endif
          @endforeach
        </ol>
      </div>
    </div>
  </div><!-- /.container-fluid -->
</section>
```

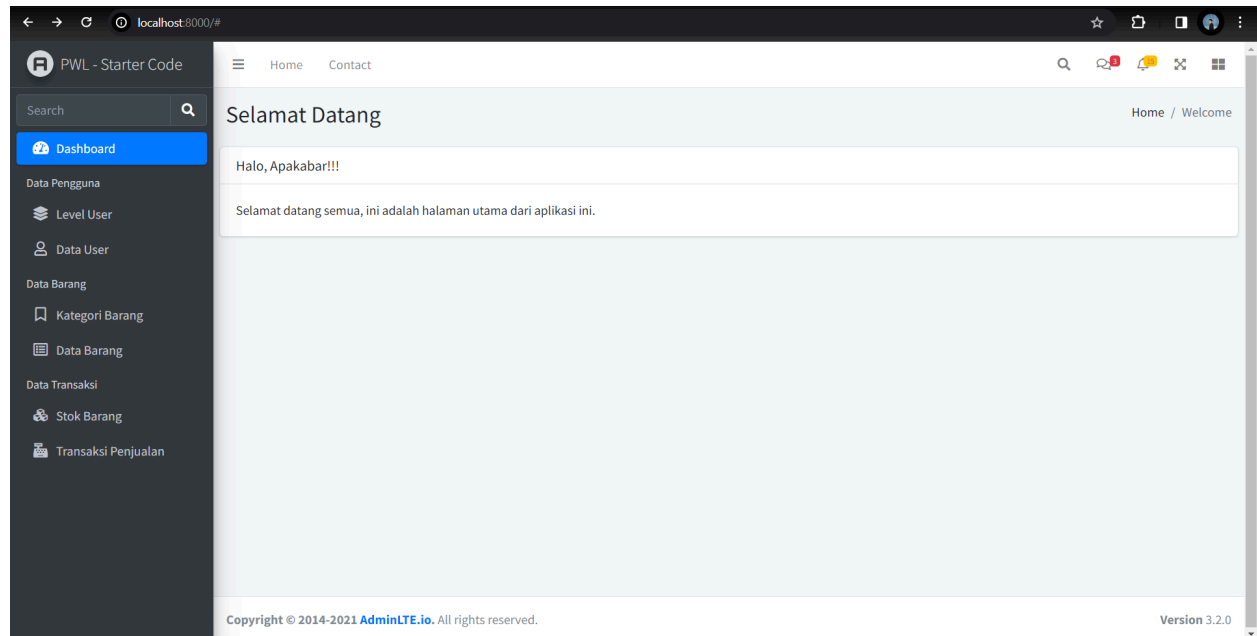
4. Kita modifikasi file PWL\_POS/resources/views/layouts/sidebar.blade.php

```
<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>
    </ul>
  </nav>
```

5. Kita tambahkan kode berikut router web.php

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

6. Sekarang kita coba jalankan di browser





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

```
Route::group(['prefix' => 'user'], function () {
    Route::get('/', [UserController::class, 'index']);
    Route::post('/list', [UserController::class, 'list']);
    Route::get('/create', [UserController::class, 'create']);
    Route::post('/', [UserController::class, 'store']);
    Route::get('/{id}', [UserController::class, 'show']);
    Route::get('/{id}/edit', [UserController::class, 'edit']);
    Route::get('/user/tambah', [UserController::class, 'tambah']);
    Route::put('/{id}', [UserController::class, 'update']);
    Route::delete('/{id}', [UserController::class, 'destroy']);
});
```

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

```
// 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

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

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>
            </table>
        </div>
    </div>
@endsection
```

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

```
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>{{ config('app.name', 'PWL Laravel Starter Code') }}</title>

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

    <!-- Google Font: Source Sans Pro -->
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
    <!-- DataTables -->
    <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-bs4/css/dataTables.bootstrap4.min.css') }}">
    <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}">
    <link rel="stylesheet" href="{{ asset('adminlte/plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}">
    <!-- Theme style -->
    <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">

    @stack('css') <!-- Digunakan untuk memanggil custom css dari perintah push('css') pada masing-masing view -->
</head>
<body class="hold-transition sidebar-mini">
<!-- Site wrapper -->
<div class="wrapper">
    <!-- Navbar -->
    @include('layouts.header')
    <!-- /.navbar -->
```

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')
        ->get();

    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()
                . '<input type="button" value="Hapus" class="btn btn-danger btn-sm"></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.

Daftar User

Daftar user yang terdaftar dalam sistem

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 Pertama	Pelanggan	Detail Edit Hapus
5	Manager_dua	Manager 2	Manager	Detail Edit Hapus
6	Manager_tiga	Manager 3	Manager	Detail Edit Hapus
7	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus

Showing 1 to 8 of 8 entries Previous 1 Next

Copyright © 2014-2021 AdminLTE Inc. All rights reserved. Version 3.2.0

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.index', ['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>
                </div>
            </div>
        </form>
    </div>
</div>
```

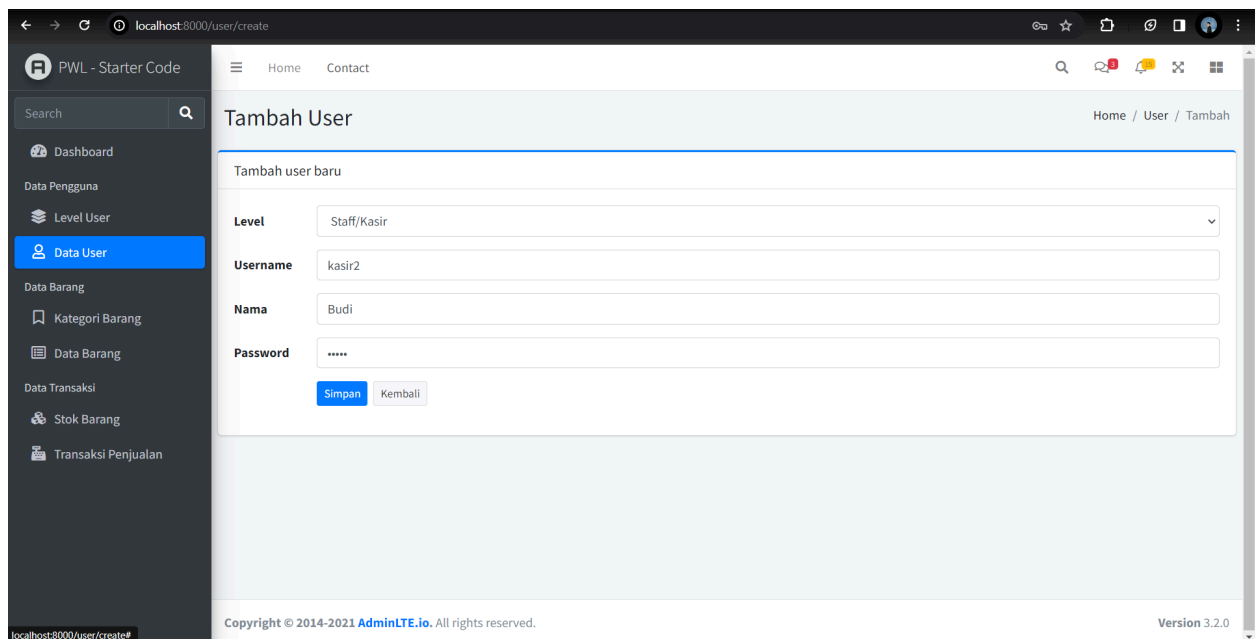
11. Kemudian untuk bisa handle data yang akan disimpan ke database, kita buat fungsi store() di UserController.php

```
// Menyimpan data user baru
public function store(Request $request)
{
    $request->validate([
        'username' => 'required|string|min:3|unique:m_user,username',
        'nama' => 'required|string|max:100',
        'password' => 'required|min:5',
        'level_id' => 'required|integer',
    ]);

    UserModel::create([
        'username' => $request->username,
        'nama' => $request->nama,
        'password' => bcrypt($request->password),
        '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::get('/{id}', [UserController::class, 'show']);
```

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

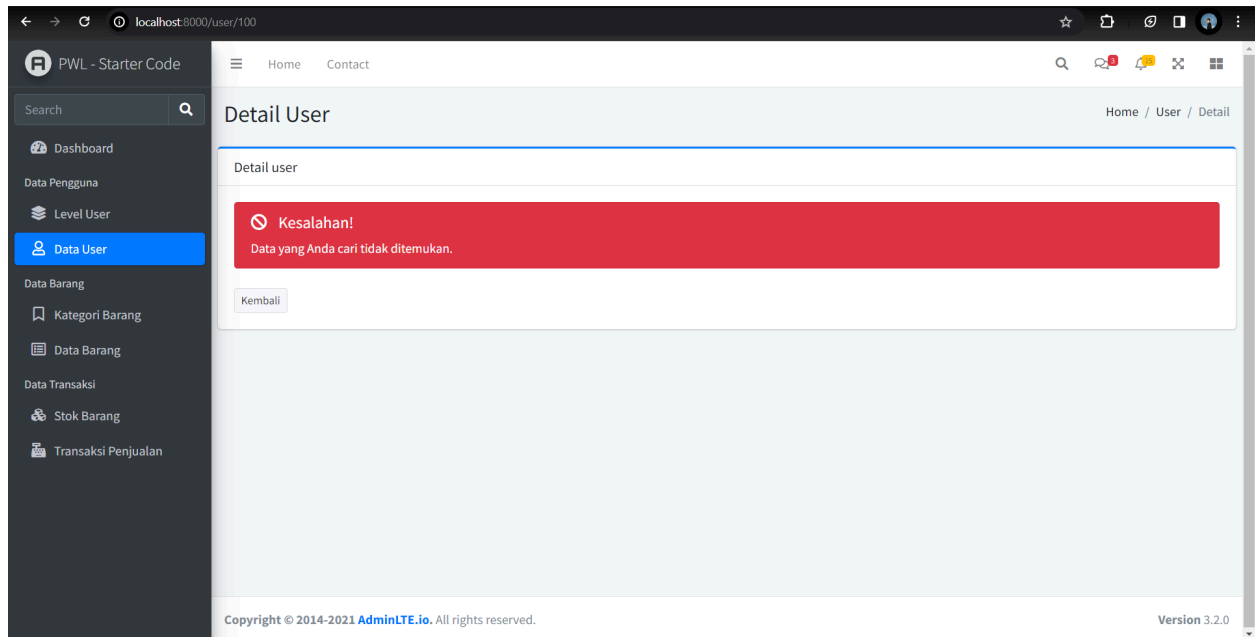
    return view('user.show', ['breadcrumb' => $breadcrumb, 'page' => $page, 'user' => $user, 'activeMenu' => $activeMenu]);
}
```

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 table-sm">
                <tr>
                    <th>ID</th>
                    <td>{{ $user->user_id }}</td>
                </tr>
                <tr>
                    <th>Level</th>
                    <td>{{ $user->level->level_nama }}</td>
                </tr>
                <tr>
                    <th>Username</th>
                    <td>{{ $user->username }}</td>
                </tr>
            </table>
        @endempty
    </div>
</div>
```

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` amati apa yang terjadi, dan laporkan!!!



Akan keluar pesan error karena user dengan id 100 tidak ada.

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

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

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',
    ];

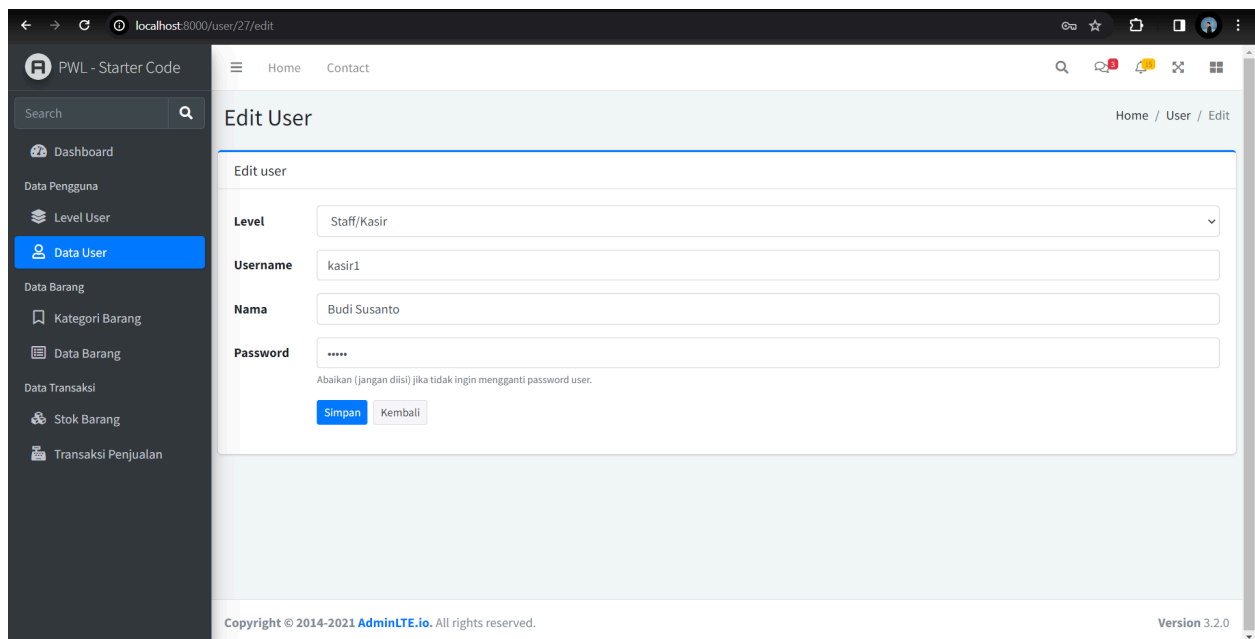
    $activeMenu = 'user'; // Set menu yang sedang aktif

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





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



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) {
        return redirect('/user')->with('error', 'Data user tidak ditemukan');
    }

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

        return redirect('/user')->with('success', 'Data user berhasil dihapus');
    } catch (\Throwable $th) {
        // Membawa pesan error jika terdapat error ketika menghapus data
        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

```
<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
    <table class="table table-bordered table-striped table-hover table-sm" id="table_user">
```

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

The screenshot shows the 'Daftar User' page in the PWL - Starter Code application. A confirmation dialog box is displayed in the center, asking 'Apakah Anda yakin menghapus data ini?' (Are you sure you want to delete this data?). The dialog has 'OK' and 'Cancel' buttons. In the background, the 'Daftar User' table is visible, showing a list of users with columns for ID, Username, Nama, Level Pengguna, and Aksi. The 'Aksi' column contains 'Detail', 'Edit', and 'Hapus' buttons for each user.

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 Pertama	Pelanggan	Detail Edit Hapus
5	Manager_dua	Manager 2	Manager	Detail Edit Hapus
6	Manager_tiga	Manager 3	Manager	Detail Edit Hapus
7	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus
9	kasir1	Budi Susanto	Staff/Kasir	Detail Edit Hapus

The screenshot shows the 'Daftar User' page in the PWL - Starter Code application after a user has been successfully deleted. A green success message banner is displayed at the top of the table area, stating 'Data user berhasil dihapus' (User data successfully deleted). The table now shows 8 entries instead of 9. The 'Aksi' column still contains 'Detail', 'Edit', and 'Hapus' buttons for each user.

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 Pertama	Pelanggan	Detail Edit Hapus
5	Manager_dua	Manager 2	Manager	Detail Edit Hapus
6	Manager_tiga	Manager 3	Manager	Detail Edit Hapus
7	manager22	Manager Dua Dua	Manager	Detail Edit Hapus
8	manager56	Manager55	Manager	Detail Edit Hapus

## 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\_POS/resources/views/user/index.blade.php

```
<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" name="level_id" id="level_id">
                    <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>
```

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

```
"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();  
})
```

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);  
    }  
}
```

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

Daftar User

Daftar user yang terdaftar dalam sistem

Filter: Manager

Level Pengguna

Show 10 entries

ID	Username	Nama	Level Pengguna	Aksi
1	manager	Manager	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	Manager_dua	Manager 2	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	Manager_tiga	Manager 3	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	manager22	Manager Dua Dua	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	manager56	Manager55	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 5 of 5 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

## **PERTANYAAN**

1. Apa perbedaan frontend template dengan backend template?

### **Jawab:**

#### Frontend Template

Frontend template digunakan untuk merancang tampilan depan (frontend) dari sebuah aplikasi web. Frontend template bertujuan untuk menyajikan informasi kepada pengguna akhir dan memberikan pengalaman pengguna yang baik. Contoh dari penggunaan frontend template adalah template untuk halaman beranda, halaman produk, halaman profil pengguna, dll.

#### Backend Template

Backend template digunakan untuk merancang tampilan dan logika pada bagian belakang (backend) dari sebuah aplikasi web. Backend template digunakan oleh pengelola atau administrator situs untuk mengelola aplikasi, melakukan manajemen data, autentikasi, otorisasi, dan tugas administratif lainnya. Contoh dari penggunaan backend template adalah template untuk dasbor administratif, halaman manajemen pengguna, halaman manajemen konten, dll.

2. Apakah layouting itu penting dalam membangun sebuah website?

### **Jawab:**

Penting, karena layout menentukan cara konten disusun dan disajikan kepada pengguna. Layout yang baik dapat meningkatkan pengalaman pengguna, membuat navigasi lebih mudah dipahami, dan memastikan bahwa informasi disajikan dengan cara yang jelas dan teratur.

3. Jelaskan fungsi dari komponen laravel blade berikut : `@include()`, `@extend()`, `@section()`, `@push()`, `@yield()`, dan `@stack()`

### **Jawab:**

- `@include()`: Digunakan untuk menyertakan file blade lain ke dalam file blade saat ini.
- `@extend()`: Digunakan untuk menunjukkan bahwa file blade saat ini "mewarisi" dari layout blade tertentu.
- `@section()`: Digunakan untuk menentukan bagian dari konten yang akan digantikan oleh konten dari file blade lain.
- `@push()`: Digunakan untuk menambahkan konten ke dalam satu atau beberapa stack. Konten ini kemudian bisa ditampilkan di dalam file blade lain.
- `@yield()`: Digunakan untuk menampilkan konten dari suatu section yang ditentukan di dalam file blade lain.
- `@stack()`: Digunakan untuk menampilkan konten yang telah ditambahkan ke dalam suatu stack.

4. Apa fungsi dan tujuan dari variable \$activeMenu?

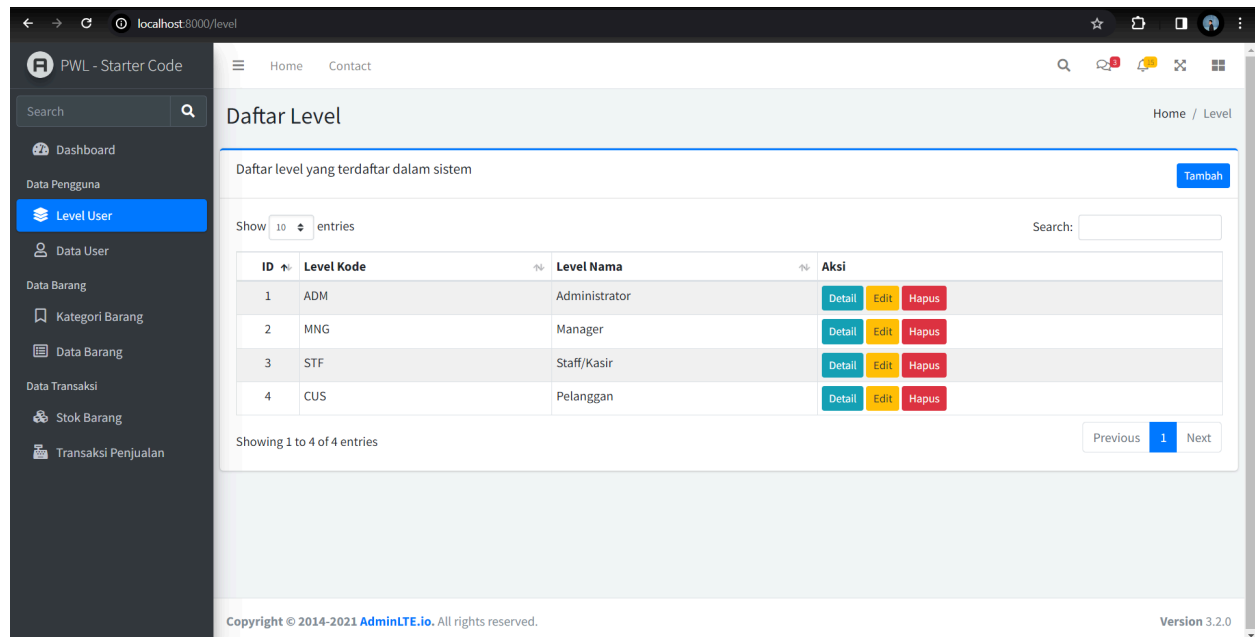
**Jawab:**

Variabel \$activeMenu digunakan untuk menunjukkan menu atau halaman yang sedang aktif atau terpilih di dalam aplikasi ini.

## 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.

### - Level User



The screenshot shows a web application interface for managing user levels. The sidebar on the left contains the following navigation items: Dashboard, Data Pengguna, Level User (highlighted), Data User, Data Barang, Kategori Barang, Data Barang, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area is titled 'Daftar Level' and includes a search bar, a 'Tambah' button, and a table of user levels. The table has columns for ID, Level Kode, Level Nama, and Aksi. The data rows are as follows:

ID	Level Kode	Level Nama	Aksi
1	ADM	Administrator	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	MNG	Manager	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	STF	Staff/Kasir	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	CUS	Pelanggan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Below the table, it says 'Showing 1 to 4 of 4 entries' and includes 'Previous', '1', and 'Next' pagination links. The footer of the page contains the copyright notice 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and the version number 'Version 3.2.0'.

## - Kategori Barang

Daftar kategori yang terdaftar dalam sistem

Show 10 entries Search:

ID	Kategori Kode	Kategori Nama	Aksi
1	KTG01	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	KTG02	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	KTG03	Alat Masak	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	KTG04	Alat Mandi	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	KTG05	Elektronik	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 5 of 5 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

## - Data Barang

Daftar barang yang terdaftar dalam sistem

Filter: - Semua - Kategori Barang

Show 10 entries Search:

ID	Kode Barang	Nama Barang	Kategori Barang	Harga Beli	Harga Jual	Aksi
1	BRG01	Tahu Telor	Makanan	10000	12000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	BRG02	Pecel Ayam	Makanan	12000	15000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	BRG03	Es Teh Jumbo	Minuman	3000	4000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	BRG04	Es Milo	Minuman	4000	5000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	BRG05	Panci	Alat Masak	15000	20000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	BRG06	Pisau	Alat Masak	10000	13000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	BRG07	Sampo	Alat Mandi	8000	10000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	BRG08	Pasta Gigi	Alat Mandi	12000	14000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	BRG09	Televisi	Elektronik	1000000	1200000	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

- Stok Barang

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 Stok

Home / Stok

Daftar stok yang terdaftar dalam sistem

Tambah

Filter: - Semua -

Nama Barang

Show 10 entries

Search:

ID	Nama Barang	Penyetok	Tanggal Stok	Jumlah	Aksi
1	Tahu Telor	Manager	2024-04-02 00:00:00	125	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	Pecel Ayam	Administrator	2024-03-03 14:09:45	100	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	Es Teh Jumbo	Administrator	2024-03-03 14:09:45	100	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	Es Milo	Administrator	2024-03-03 14:09:45	100	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	Panci	Administrator	2024-03-03 14:09:45	100	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	Pisau	Administrator	2024-03-03 14:09:45	100	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	Sampo	Administrator	2024-03-03 14:09:45	100	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	Pasta Gigi	Administrator	2024-03-03 14:09:45	100	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	Televisi	Administrator	2024-03-03 14:09:45	100	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

- Transaksi Penjualan

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 Penjualan

Home / Penjualan

Daftar Penjualan yang terdaftar dalam sistem

Tambah

Filter: - Semua -

Nama Petugas

Show 10 entries

Search:

ID	Kode Penjualan	Pembeli	Petugas	Total Harga	Tanggal	Aksi
1	PJL001	Krisna	Staff/Kasir	33000	2024-03-03 14:10:01	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	PJL002	Krisna	Staff/Kasir	42000	2024-03-03 14:10:01	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	PJL003	Febrio	Staff/Kasir	213000	2024-03-03 14:10:01	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	PJL004	Febrio	Staff/Kasir	195000	2024-03-03 14:10:01	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	PJL005	Denny	Staff/Kasir	55000	2024-03-03 14:10:01	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	PJL006	Denny	Staff/Kasir	63000	2024-03-03 14:10:01	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	PJL007	Al Byan	Staff/Kasir	64000	2024-03-03 14:10:01	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	PJL008	Al Byan	Staff/Kasir	61000	2024-03-03 14:10:01	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	PJL009	Ageng	Staff/Kasir	62000	2024-03-03 14:10:01	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>